

Samuel Ady Sanjaya

Generating Combination of Biblical Baby Names using Recurrent Neural Network (RNN) and Optimization Comparison

 Quick Submit Quick Submit Universitas Multimedia Nusantara

Document Details

Submission ID

trn:oid:::1:3308588900

Submission Date

Aug 5, 2025, 4:20 PM GMT+7

Download Date

Aug 5, 2025, 4:29 PM GMT+7

File Name

erating_Combination_of_Biblical_Baby_Names_Document_Combined.pdf

File Size

9.1 MB

264 Pages

177,123 Words

942,616 Characters

19% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text
- Cited Text
- Small Matches (less than 8 words)

Exclusions

- 19 Excluded Sources

Match Groups

- 935** Not Cited or Quoted 19%
Matches with neither in-text citation nor quotation marks
- 0** Missing Quotations 0%
Matches that are still very similar to source material
- 0** Missing Citation 0%
Matches that have quotation marks, but no in-text citation
- 0** Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

- 4% Internet sources
- 16% Publications
- 5% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- 935** Not Cited or Quoted 19%
Matches with neither in-text citation nor quotation marks
- 0** Missing Quotations 0%
Matches that are still very similar to source material
- 0** Missing Citation 0%
Matches that have quotation marks, but no in-text citation
- 0** Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

- 4% Internet sources
- 16% Publications
- 5% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Publication	
Md. Sabbir Hossain, Dipayan Ghose, All Masror Partho, Minhaz Ahmed et al. "Perf...		2%
2	Publication	
Maherun Nessa Isty, Nishat Vasker, Fahmida Anjum, Warda Ruhin Parsub, Amit R...		2%
3	Publication	
Sarmin Sultana Snigdha, Reyad Bin Satter, Waliul Islam, Shafiur Rahman Chowdh...		2%
4	Student papers	
Laguna State Polytechnic University		1%
5	Publication	
Oda I. B. Hariyanto, Doni Purnama Alamsyah. "Student Individual Performance in...		1%
6	Publication	
Sakorn Mekruksavanich, Ponnipa Jantawong, Anuchit Jitpattanakul. "Improving E...		1%
7	Student papers	
George Bush High School		1%
8	Publication	
Habibur Rahman, Md. Munam Kazi, Md. Samiul Islam, Jannatul Maoya et al. "An A...		1%
9	Publication	
Akibul Haque, Mahamudul Hasan, Sumaya Akter Suma, Nishat Vasker et al. "Dete...		1%
10	Internet	
www.proceedings.com		1%

11	Publication	Mahir Tagwar, Md Asfakul Islam, Md Rokibul Hasan, Md Faisal Azhar et al. "Data ...	1%
12	Internet	scholars.ncu.edu.tw	<1%
13	Publication	Kanij Fatema Arpa, Tanni Mittra, Tasnim Ferdous, Nusrat Jahan et al. "A Machine ...	<1%
14	Publication	Thangaprakash Sengodan, Sanjay Misra, M Murugappan. "Advances in Electrical ...	<1%
15	Publication	H.L. Gururaj, Francesco Flammini, S. Srividhya, M.L. Chayadevi, Sheba Selvam. "Co...	<1%
16	Publication	Arvind Dagur, Karan Singh, Pawan Singh Mehra, Dharendra Kumar Shukla. "Artific...	<1%
17	Internet	shdl.mmu.edu.my	<1%
18	Publication	Poonam Nandal, Mamta Dahiya, Meeta Singh, Arvind Dagur, Brijesh Kumar. "Pro...	<1%
19	Publication	"Proceedings of the 5th International Conference on Data Science, Machine Lear...	<1%
20	Internet	scholarworks.bwise.kr	<1%
21	Publication	V. Sharmila, S. Kannadhasan, A. Rajiv Kannan, P. Sivakumar, V. Vennila. "Challeng...	<1%
22	Publication	Pawan Singh Mehra, Dharendra Kumar Shukla. "Artificial Intelligence, Blockchain,...	<1%
23	Publication	Deb, Dipok. "Application and Analysis of Machine Learning and Deep Learning AI...	<1%
24	Publication	Sumaya Akter Suma, Akibul Haque, Nishat Vasker, Mahamudul Hasan, Jesan Aha...	<1%

25	Publication	H L Gururaj, Francesco Flammini, V Ravi Kumar, N S Prema. "Recent Trends in He...	<1%
26	Publication	H L Gururaj, M R Pooja, Francesco Flammini. "Recent Trends in Computational Sci...	<1%
27	Publication	Salem Al-Naemi, Rachid Benlamri, Rehan Sadiq, Aitazaz Farooque, Michael Phillip...	<1%
28	Internet	fse.ewubd.edu	<1%
29	Internet	bilselkongreleri.com	<1%
30	Internet	assets-eu.researchsquare.com	<1%
31	Publication	Salwa Belaqziz, Salma El Hajjami, Hicham Amellal, Redouan Lahmyed, Lahcen Kou...	<1%
32	Publication	Amit Kumar Tyagi. "Data Science and Data Analytics - Opportunities and Challeng...	<1%
33	Publication	Ahmed A. Abd El-Latif, Mohammed A ElAffendi, Mohamed Ali AlShara, Yassine Ma...	<1%
34	Internet	www.mdpi.com	<1%
35	Publication	Yassine Maleh, Justin Zhang, Abderrahim Hansali. "Advances in Emerging Financi...	<1%
36	Publication	Samarjeet Borah, Ratna Raja Kumar Jambi, Sharifah Sakinah Syed Ahmad, Mahen...	<1%
37	Publication	Oluwatobi Adeleke, Sina Karimzadeh, Tien-Chien Jen. "Machine Learning-Based M...	<1%
38	Publication	Adesokan, Ademola Abdulganiyu. "Leveraging Deep Learning Models and Social ...	<1%

39	Publication	Inam Ullah Khan, Salma El Hajjami, Mariya Ouaisa, Salwa Belaqziz, Tarandeep Ka...	<1%
40	Publication	Rashmi Agrawal, Marcin Paprzycki, Neha Gupta. "Big Data, IoT, and Machine Lear...	<1%
41	Student papers	National College of Ireland	<1%
42	Internet	library.cbit.ac.in	<1%
43	Publication	Ankur Gupta. "Next Generation Computing and Information Systems", CRC Press,...	<1%
44	Publication	Harahsheh, Khawlah. "Enhancing IoT Security Using Lightweight Machine Learni...	<1%
45	Publication	Achin Jain, Arun Kumar Dubey, Meenakshi Gupta, Sarita Yadav, Arvind Panwar, R...	<1%
46	Publication	Tiago Augusto Orcajo Demay Cordeiro. "Detecção de anomalias na comunicação ...	<1%
47	Publication	Yijun Zhang, Zimeng Chen, Zhuxuan Wan, Qianhui Jiang, Xiaoling Lu, Bin Yan, Jing...	<1%
48	Publication	Haithami, Mahmood. "Towards Real-World Clinical Colonoscopy Deep Learning M...	<1%
49	Publication	Sujata Dash, Subhendu Kumar Pani, Joel J. P. C. Rodrigues, Babita Majhi. "Deep Le...	<1%
50	Publication	Canedo, Daniel Duarte. "Data-Centric Artificial Intelligence in the Context of Com...	<1%
51	Publication	Prashant Pranav, Archana Patel, Sarika Jain. "Machine Learning in Healthcare an...	<1%
52	Publication	Sibiya, Malusi. "Classification and Severity Prediction of Maize Leaf Diseases Usin...	<1%

53	Publication	Thales de Lorena Peixoto Junior. "Análise de permeabilidade de materiais granul...	<1%
54	Publication	Cuomo, Michael Roman. "Movie Genre Classification Using Script Texts.", Florida ...	<1%
55	Publication	D. Jeya Mala, Anto Cordelia Tanislaus Antony Dhanapal, Saurav Sthapit, Anita Kha...	<1%
56	Publication	Sajjan Singh, Sarabpreet Kaur. "Latest Trends in Engineering and Technology - AI...	<1%
57	Publication	Shruti Sharma, Ashutosh Sharma, Trinh Van Chien. "The Intersection of 6G, AI/Ma...	<1%
58	Publication	Javid, Irfan. "Modified Gated Recurrent Unit Based on Update-Prime Gate for Hea...	<1%
59	Publication	Nishu Gupta, Sandeep S. Joshi, Milind Khanapurkar, Asha Gedam, Nikhil Bhawe. "...	<1%
60	Publication	Om Prakash Mahela, Baseem Khan, Puneet Kumar Jain. "Emerging Electrical and ...	<1%
61	Publication	Grilo, Tiago Manuel Martins. "Predicting Demand in Fashion Retailing: A Data Ana...	<1%
62	Publication	Sakorn Mekruksavanich, Ponnipa Jantawong, Anuchit Jitpattanakul. "A Hybrid De...	<1%
63	Publication	da Costa, Tiago André Queiroz Soares. "Enhanced Multiview Experiences Through...	<1%
64	Publication	zhonglin zhao, Baohong Lu, Shuo Zhang, Daoli Wang, Jiaquan Wan, Ranyu Liu, Hu...	<1%
65	Publication	Qi, Jing. "Pathology Data Prioritisation in the Absence of a Ground Truth.", The Un...	<1%
66	Publication	Krishneel Sundar, Pritika Reddy, Kaylash C. Chaudhary, Shaireen Khan. "A System...	<1%

67	Internet	journal.uob.edu.bh	<1%
68	Publication	Celorico, Maria Inês Monteiro. "Classic Car Spare Parts for Restoration: Internatio...	<1%
69	Publication	Jaidev Sanjay Khalane, Nilesh D. Gawande, Shanmuganathan Raman, Subramani...	<1%
70	Publication	Muhindo, Ricky Birambovote. "Evaluate the Performance of Support Vector Mach...	<1%
71	Publication	Shrikaant Kulkarni, Jaiprakash Narain Dwivedi, Dinda Pramanta, Yuichiro Tanaka....	<1%
72	Student papers	Universitas Diponegoro	<1%
73	Publication	Vandana Mohindru Sood, Yashwant Singh, Bharat Bhargava, Sushil Kumar Naran...	<1%
74	Publication	Inam Ullah, Inam Ullah Khan, Mariya Ouaisa, Mariyam Ouaisa, Salma El Hajjami...	<1%
75	Publication	Fischer, Wayne A.. "Comparative Quantitative Analysis of Machine Learning Algor...	<1%
76	Publication	Gomes, Nuno Teixeira. "Previsão de Séries Temporais com R ² Negativo: Uma Aplic...	<1%
77	Publication	de SeifãRasteiro, Rui Manuel Mendes. "O Canto e as Suas Características como Fe...	<1%
78	Internet	sajhrm.co.za	<1%
79	Publication	Md. Huzaifa, Fahiha Bushra Fabin, Nazia Tabassum Natasha, Imran Fakir et al. "D...	<1%
80	Student papers	Southern New Hampshire University - Continuing Education	<1%

81	Publication	Kazi Mostaq Hridoy, Saiful Islam Akash, Fahmida Afrose Dipti, Mahamudul Hasan,...	<1%
82	Student papers	Korea Advanced Institute of Science and Technology	<1%
83	Publication	Muhammad Khalil Afzal, Muhammad Ateeq, Sung Won Kim. "Data-Driven Intellig...	<1%
84	Student papers	University of Sheffield	<1%
85	Publication	Kouremenou, Eleftheria Georgia. "Data Mining, Cleaning, Feature Extraction, and...	<1%
86	Student papers	Liverpool John Moores University	<1%
87	Publication	Bahjat Fakieh, Farrukh Saleem. "COVID-19 From Symptoms to Prediction: A Statis...	<1%
88	Publication	Mohammad Abdullah Bari, Aliva Bakshi, Jahid Chowdhury Al Choton, Swaraj Pra...	<1%
89	Publication	Luiz Afonso Batalha Marão. "Deep reinforcement learning multi-sensor based na...	<1%
90	Internet	data.crossref.org	<1%
91	Publication	Shrikaant Kulkarni, P. William, Vijaya Prakash, Jaiprakash Narain Dwivedi. "Lever...	<1%
92	Publication	Yılmaz, Ümit. "Yatırım Fonu Kapanış Fiyatının (Net Aktif Değerinin) ve Performans...	<1%
93	Publication	Ashish Bagwari, Geetam Singh Tomar, Jyotshana Bagwari, Jorge Luis Victória Bar...	<1%
94	Publication	Deema Mohammed AlSekait, Mohammed Zakariah, Syed Umar Amin, Zafar Iqbal ...	<1%

95

Publication

Manisha Guduri, V Uma Maheswari, Rajanikanth Aluvalu, Amit Krishna Dwivedi, ...

<1%

Generating Combination of Biblical Baby Names using Recurrent Neural Network (RNN) and Optimization Comparison

1st Samuel Ady Sanjaya

Faculty of Technology and Informatics
Universitas Multimedia Nusantara
Tangerang, Indonesia
samuel.ady@umn.ac.id

2nd Suyoto

Faculty of Technology and Industry
Universitas Atma Jaya Yogyakarta
Yogyakarta, Indonesia
suyoto@uajy.ac.id

Abstract— Giving names to newborns in every country has its own uniqueness based on ethnicity, religion, race, culture, and names from past history. All parents in the world should give their baby a name with a good meaning because there is a saying that a name is a prayer for the child. However, these names have been used repeatedly based on trends every year, so it is very possible that humans on Earth have the same name. In this research, an approach will be made to generate names based on names from the Bible to make combinations of characters into new names. This Bible-based name is used because it is the most common source of names for Christians around the world. Names will be generated using a Recurrent Neural Network (RNN) with a comparison of several Optimizers such as Adadelta, Adagard, SGD, Adam, Adamax, and RMSProp. This comparison found that the 3 optimizers with the best loss function were Adam at 1,755, RMSProp at 1,814, and Adamax at 2,064. The optimizer with the best loss function can produce data with a good combination of consonants and vowels, but the optimizer with a bad loss function will produce a large number of combinations of consonants. This research resulted in a new approach for generating names from the RNN model generation with the best optimizer that can be used as inspiration for baby names.

Keywords—Biblical names, loss function, RNN, optimizer algorithm

I. INTRODUCTION

The name given by the parents is very meaningful, in some areas including Indonesia, it is stated that the name is a prayer. Parents will try to give the best name to their children. The giving of names is influenced by various ethnic, cultural, and religious factors and even references from singers or film actors. Most modern people in some areas have thought of a creative child's name to differentiate as other children's names across cultures or countries. In Indonesia, now many people adopt names that come from outside their area, even abroad to get a name that is unique and not the same as their surroundings. However, it cannot be denied that giving names in an area is also usually carried out from generation to generation, especially for last names or family names. In Catholic beliefs, it is also customary to use a baptismal name for a person's first name. Moreover, if the population of Christians in the world increases, it is recorded that there is a growth in Christian beliefs from 1900 as much as 9.1% to 59.1% in 2022 [1]. So if someone usually has three to five words for their full name, it's getting narrower to create their name. Moreover, coupled with the names in the Bible, the number is limited. Apart from naming because of belief, naming can also be based on culture as in the example of

research to see naming phenomena in Javanese culture. [2]. Cultures in other places, such as Japanese culture and Mexican culture, also influence the evolution of baby names in the future. [3][4]. Merging names is also a solution to obtain a unique name, for example, combining parents' names or combining two popular names. Based on this inspiration, it was found that the purpose of this research was to generate names using one of the deep learning methods. The use of deep learning can also be used to identify Arabic name recognition with a combination of several methods such as RBF, CNN, BiLSTM, or the use of B-GRU [5], [6]. There is also the use of deep learning to process text by considering a network embedding scheme with a text merging [7]. As for the use of another deep learning method that is good for text processing, namely Recurrent Neural Network (RNN). RNN is one of the best RNN models that can be used in various study cases, especially text processing [8]. Some examples of the use of RNN in text modeling are sentiment analysis, miss-information detection, text summarization, character-level language models, and text classification [9]–[12]. This research will also use the text processing approach to generate new names based on existing patterns, especially names in the Bible. Determining the best model is still not enough, so this research will also compare the best optimizer to be combined with RNN.

II. METHODS

A. Data Collection and Preprocessing

The dataset is taken from a collection of Biblical names consisting of the names of prophets, people, kings, and angels totaling 3668 names [13]. This dataset also has several attributes which explain the proper name to use, gender, date, and place of birth and death:

- 1) Name
- 2) IsProperName
- 3) Gender
- 4) BirthYear
- 5) DeathYear
- 6) BirthPlace
- 7) DeathPlace

From this dataset, information on each name regarding being born and died can be obtained, although there are still some names information is not complete. Not all names in the Bible are good for use as name references because the names are irrelevant or too long, so filtering will be done first according to the IsProperName attribute. The name attribute will also be cleaned beforehand by deleting special characters

and spaces, including names that have more than 2 words, the goal is that no special characters will be combined into a new name [14][15]. Furthermore, after the data is cleaned, it is followed by tokenization. Tokenization is the process of converting words or letters into numbers. After that, the tokenized name pool is then converted into a matrix so that all name pools can be processed by the model.

B. Modeling

Name generation will be carried out using a semi-supervised learning approach because even though there is no labeled data or class classification, the model still requires data training to be able to learn patterns from existing names. This model will focus on studying the patterns and structure of names in the Bible, then it will try to do generation based on existing names using one of the deep learning models, namely RNN. The data to be modeled by the RNN must first exist in the form of a name tokenization result matrix to be processed. This RNN model will be run in several epochs to get the optimal value [9]. The preparation of several hidden layers is also considered to get the appropriate evaluation value. An overview of the deep learning process using RNN is shown in Figure 1.

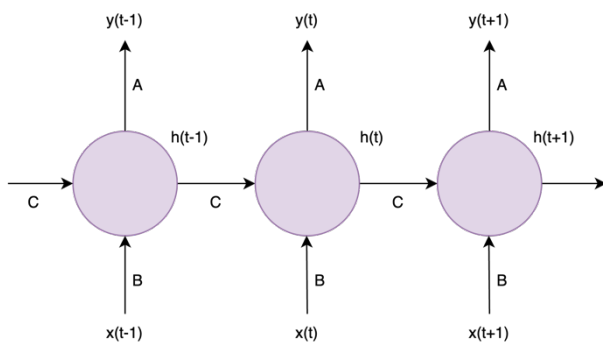


Fig. 1. RNN Schema

with equations,

$$h(t) = f_c(h(t-1), x(t))$$

where,

- $h(t)$: new state
- f_c : function with parameter c
- $h(t-1)$: old state
- $x(t)$: input vector at time step t

So x is the input layer, h is the hidden layer and y is the hidden layer. A , B , and C are the network parameters used to improve the output of the model. At any given time t , the current input is a combination of inputs at $x(t)$ and $x(t-1)$. The output at any given time is fetched back to the network to improve the output.

C. Evaluation of Different Optimizers

The RNN model will be evaluated using a loss function to get good evaluation results for a model. Loss functions are also commonly used for several other neural network-based deep learning models to be a good benchmark. The loss function is used to compare the target and predicted value with the aim of minimizing the loss between the predicted

and target output. So the value of the loss function will be better if the value is the minimum, lower better. Several evaluation metric comparisons for loss function also ensure that loss function is an evaluation metric that can be used for text processing or computer vision [16], [17]. Several related studies also use the loss function as the main metric for evaluating the model [18], [19]. After the evaluation metric is determined, the next step is to determine several optimizers that will be paired with the RNN with several comparisons according to Table 1 [20].

Table 1. Optimization Algorithm for Deep Learning

No	Optimizer		
	Algorithm	Advantages	Disadvantages
1	Adam	Fast and converges rapidly, high learning rate	Has probability to suffer a weight decay problem
2	Adadelata	Learning rate is not decay and the training is not stop	High computation cost
3	RMSProp	Has pseudo-curvature information and robust optimizer	Learning rate still manual
4	Stochastic Gradient Descent (SGD)	Run in less memory and less time to converges	High variance in model parameters

Based on Table 1, there are advantages and disadvantages of each optimizer which will be tested in this study with the RNN model. Besides the four optimizer algorithms, two other optimizer algorithms will be added, namely: Adamax and Adagrad. So that later there will be a total of six optimizer algorithms used. From this comparison of optimizer algorithms, it will be tested which one has the best loss function and will also see how the performance of the name that has been generated from a combination of RNN and several optimizer algorithms will be seen.

III. RESULT AND DISCUSSIONS

A. Data Collection and Preprocessing

Based on the data that has been collected as many as 3668 lists of names derived from the Bible to be used. After selecting the data based on the eligibility of the name, the length of the name and also the name that does not contain special characters, 1789 names are obtained which will be processed. The name to be used has entered the transformation process to lower case, removing special characters and also filtering based on proper names. The following is an example of the dataset in Table 2 which consists of several attributes such as name, gender and place of birth.

Table 2. Dataset example

No	Attribute			
	Name	IsProperName	Gender	Birth Place
1.	Aaron	1	M	Egypt
2.	Abel	1	M	Eden
3.	Eve	1	F	Eden
4.	Nehusta	1	F	null
5.	Samuel	1	M	null

Even though there are several attributes, in this study only use name variables to make name combinations. Later the model will learn to do naming based on the names in the Bible. The distribution of names in this dataset has various name length variations, in Figure 2 it can be seen that most names have a character length between 4 to 8. Names with a total length of more than 12 characters are rarely used. One of the requirements that the name is proper is not too long, so names that are too long will be eliminated.

Name Length

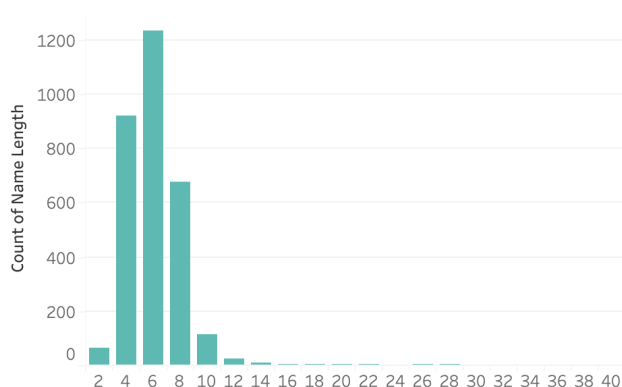


Fig. 2. Name Length Distribution

The list of names that will be used will then be tokenized to change the letters in each name into a token. The name that has been changed to lowercase, then tokenized, then the number of tokens is 26 letters, plus 3 characters. One letter will be represented by one number, then one name will be made into a number matrix which is a transformation from the previous tokenization.

```
aaron
arvadite
jaakobah
dinah
hamuel
kadmial
nahash
sallu
theudas
[[ 0  3  3 20 17 16  1  1  1]
 [ 0  3 20 24  3  6 11 22  7]
 [ 0 12  3  3 13 17  4  3 10]
 [ 0  6 11 16  3 10  1  1  1]
 [ 0 10  3 15 23  7 14  1  1]
 [ 0 13  3  6 15 11  7 14  1]
 [ 0 16  3 10  3 21 10  1  1]
 [ 0 21  3 14 14 23  1  1  1]
 [ 0 22 10  7 23  6  3 21  1]]
```

Fig. 3. Tokenization and Matrix Transformation

The results of the tokenization and matrix transformation are depicted in Figure 3. The name that has been changed to this matrix will be processed into the RNN model for training. An example of tokenization in Figure 3 is the name "aaron" will be changed to token "3 3 20 17 16". The number 1 in the matrix is the result of conversion to fill in the blank space so that the length is the same as the other names.

B. Modeling

Modeling with RNN is done using the Tensorflow library developed by Google. In Tensorflow, there is an RNN model that can be compiled with a combination of layers to get good results. In this study, there are 5 layers arranged to form an RNN. Starting from the first input layer with an embedding constraint of 16 and with an embedding initializer of 11. Then the RNN layer is formed into 512 units with relu activation. In the next three layers added dense with various kinds of activation with relu. The summary model used can be seen in Table 3 which consists of several layers.

Table 3. RNN model build layers

Layer	Layers		
	Layer (type)	Output Shape	Param#
1	Embedding	11, 16	464
2	RNN	11, 512	270848
3	Dense I	11, 128	65664
4	Dense II	11, 64	8256
5	Dense III	11, 29	1885

So the model obtained a total of 347,117 parameters and trainable parameters. This model is run with a batch size and epoch of 50. In the next stage, this model will be run several times using a different optimizer to get the best model based on loss function validation.

C. Evaluation of Different Optimizers

The evaluation carried out for this model is by softmax loss, or in other words, it is a combination of softmax activation combined with categorical cross-entropy with the following equation:

Softmax equation,

$$f(s)_i = \frac{e^{s_i}}{\sum_j^c e^{s_j}}$$

Cross Entropy equation,

$$CE = - \sum_i^c t_i \log(f(s)_i)$$

Categorical cross-entropy is used because the data uses multiple classes or the output can be in various letter combinations. The input will be converted first with encoding as in Fig 3.

Based on Table 4, there are three optimizers that have a loss function value that tends to be constant at around 3.36 compared to the epoch value. During the 50th epoch, the loss function results were still the same, so there was no significant loss function improvement. The Adadelta, Adagrad and SGD optimizer algorithms do not have a good learning rate because they fail to learn the pattern of constructing names.

Table 4. Loss functions with bad score

Epoch	Optimizer Algorithm		
	<i>Adadelta</i>	<i>Adagard</i>	<i>SGD</i>
10	3.36601591	3.36432672	3.3652246
20	3.36567068	3.36284447	3.36269403
30	3.36538386	3.36158252	3.36027384
40	3.37230325	3.36048651	3.35790443
50	3.3721447	3.35946655	3.35554481

Meanwhile, based on Table 5, there are three other optimizers that get good loss function values at the end of 50 epochs, namely Adam, Adamax and RMSProp. Adamax will look less good if the iteration is up to 30, but in the next epoch up to 50 it has a good result at number 2. Meanwhile the best performance is obtained by Adam optimizer with a number of 1.75 at the 50th epoch. For RMSProp it is also good at 1.81 but RMSProp has a good loss function value if only a few epochs are executed. This is evident from the first 10 epochs, which has the best loss function is RMSProp. So RMSProp can be an alternative to Adam if there aren't many epochs being run.

Table 5. Loss functions with good score

Epoch	Optimizer Algorithm		
	<i>Adam</i>	<i>Adamax</i>	<i>RMSProp</i>
10	2.04107976	3.37183929	1.94330382
20	1.91443384	3.3716104	2.07952929
30	1.85466015	3.37136817	1.94642556
40	1.80001903	2.12565208	1.86542249
50	1.75560939	2.06411362	1.81419039

The graph of the overall loss function comparison to the RNN model and various types of optimizers can be seen in Figure 4. From the figure, there are three constant models according to Table 4. In general, Adam is a good optimizer because it reduces loss function faster compared to RMSProp and also Adamax.

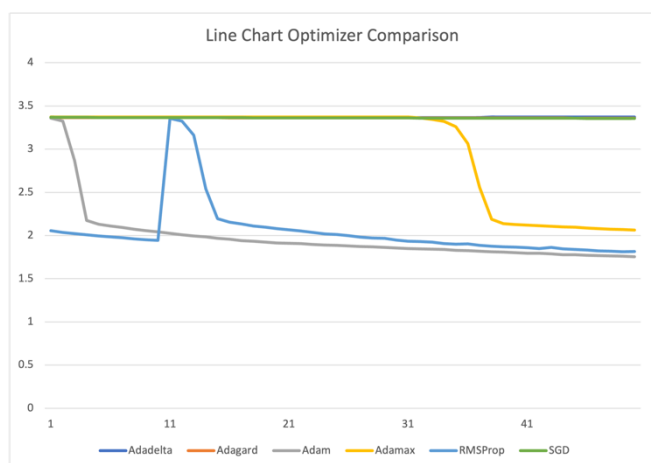


Fig. 4. Optimizer Comparison by Epochs

D. Result Example

The loss function table of the six comparisons has obtained the best based on Table 4 and Table 5. Next is the test of the

name generation results which have been grouped based on the loss function values in Table 6 and Table 7.

Table 6. Result example with bad loss function score

No	Optimizer Algorithm		
	<i>Adadelta</i>	<i>Adagard</i>	<i>SGD</i>
1.	xvairwokmq	qpaaevvi	cjbvpvskn
2.	hdampgptop	ecdoitmh	oyuaayyqtp
3.	pcroic	owpzuijlry	mgnlouylhc
4.	vaoswlfyee	ageuiyrjc	ybxxfpjqnqq
5.	opihsnd	reekeg	grubunedgb

Table 6 is a sample selection from the results of the running model and it is found that the names that are generated are not good because they are difficult to read because there are too many compositions of consonants compared to vowels. From the example in Table 6, it means that the value of the loss function is in line with the results of the sample name generation. The higher the value of the loss function, the more difficult it is for the model to be able to learn the name patterns in the dataset.

Table 7. Result example with good loss function score

No	Optimizer Algorithm		
	<i>Adam</i>	<i>Adamax</i>	<i>RMSProp</i>
1.	ehaci	balra	zahas
2.	qikeia	zaeone	xumataa
3.	mleinahe	zhuala	eumaa
4.	lanonoa	zojoar	zaiare
5.	ucasaha	iaana	magra

Different results are shown in Table 7, where the generated names are easier to read because the composition of the consonants and vowels matches. However, for some people this name still feels foreign because the names in the Bible come from Greece, Israel, Egypt or other Middle Eastern countries. With the results in Table 7, the aim of this research has been achieved to generate new baby names using the right RNN and optimizer algorithm combination.

IV. CONCLUSIONS

Based on RNN modeling with a combination of several optimizers, it is found that the optimizer is very influential on the generation of new names. Optimizers with low loss function scores will be able to form names that are more suitable for the composition of vowels and consonants, such as the optimizer Adam, Adamax, RMSProp. But with a large loss function, it will also have an effect that names become unreadable because there are too many consonant letter compositions that are close together, such as in the Adadelta, Adagard, SGD optimizer. So that there is indeed a strong relationship between the loss function and the resulting name generation results. The smaller the loss function, the better the name generation results, and vice versa. The conclusion of this study, the best composition of the comparison of six optimizers is the application of RNN using the Adam optimizer with a loss function score of 1.755 in the 50th

epoch, followed by the RMSProp optimizer with a value of 1.814 and Adamax 2.064.

ACKNOWLEDGMENT

This research was supported by the Institution of Research and Community Services of Universitas Multimedia Nusantara. We thank our colleagues from Big Data Laboratory of Universitas Multimedia Nusantara part of the Information Systems Department who provided insight and expertise that greatly assisted the research.

REFERENCES

- [1] G. A. Zurlo, T. M. Johnson, and P. F. Crossing, "World Christianity and Religions 2022: A Complicated Relationship," *Int. Bull. Mission Res.*, vol. 46, no. 1, pp. 71–80, Jan. 2022, doi: 10.1177/23969393211046993.
- [2] E. K. Aribowo and N. Herawati, "Trends in Naming System on Javanese Society: A Shift From Javanese to Arabic," *Ling. Cult.*, vol. 10, no. 2, p. 117, Nov. 2016, doi: 10.21512/lc.v10i2.1730.
- [3] G. Unser-Schutz, "For Whom and by Whom Children Are Named: Family Involvement in Contemporary Japanese Naming Practices," *Genealogy*, vol. 3, no. 2, p. 29, May 2019, doi: 10.3390/genealogy3020029.
- [4] J. Thurtell and E. K. Merchant, "Gender-Differentiated Tarascan Surnames in Michoacán," *J. Interdiscip. Hist.*, vol. 48, no. 4, pp. 465–483, Feb. 2018, doi: 10.1162/JINH_a_01195.
- [5] N. Alsaaran and M. Alrabiah, "Arabic Named Entity Recognition: A BERT-BGRU Approach," *Comput. Mater. Contin.*, vol. 68, no. 1, pp. 471–485, 2021, doi: 10.32604/cmc.2021.016054.
- [6] A. Mousa, I. Shahin, A. B. Nassif, and A. Elnagar, "Cascaded RBF-CBiLSTM for Arabic Named Entity Recognition," in *2020 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI)*, Nov. 2020, pp. 1–5. doi: 10.1109/CCCI49893.2020.9256638.
- [7] H. Jiang, S. Fan, N. Zhang, and B. Zhu, "Deep learning for predicting patent application outcome: The fusion of text and network embeddings," *J. Informetr.*, vol. 17, no. 2, p. 101402, May 2023, doi: 10.1016/j.joi.2023.101402.
- [8] F. Nakhaei and M. Irannajad, "Application and comparison of RNN, RBFNN and MNLN approaches on prediction of flotation column performance," *Int. J. Min. Sci. Technol.*, vol. 25, no. 6, pp. 983–990, Nov. 2015, doi: 10.1016/j.ijmst.2015.09.016.
- [9] X. Dong and L. Qian, "Semi-supervised bidirectional RNN for misinformation detection," *Mach. Learn. with Appl.*, vol. 10, p. 100428, Dec. 2022, doi: 10.1016/j.mlwa.2022.100428.
- [10] I. Banerjee *et al.*, "Comparative effectiveness of convolutional neural network (CNN) and recurrent neural network (RNN) architectures for radiology text report classification," *Artif. Intell. Med.*, vol. 97, pp. 79–88, Jun. 2019, doi: 10.1016/j.artmed.2018.11.004.
- [11] M. Bani-Almarjeh and M.-B. Kurdy, "Arabic abstractive text summarization using RNN-based and transformer-based architectures," *Inf. Process. Manag.*, vol. 60, no. 2, p. 103227, Mar. 2023, doi: 10.1016/j.ipm.2022.103227.
- [12] S. A. Ali Shah, W. M. S. Yafooz, A.-H. M. Emara, and H. Sajid, "Generating Muslim Name using Character-Level Language Model in Deep Learning," in *2023 International Conference on Smart Computing and Application (ICSCA)*, Feb. 2023, pp. 1–6. doi: 10.1109/ICSCA57840.2023.10087673.
- [13] MetaV, "KJV Bible database with metadata MetaV," 2016. <https://github.com/arthurmilliken/KJV-bible-database-with-metadata-MetaV-> (accessed May 23, 2023).
- [14] D. A. Kristiyanti, A. H. Umam, M. Wahyudi, R. Amin, and L. Marlinda, "Comparison of SVM & Naive Bayes Algorithm for Sentiment Analysis Toward West Java Governor Candidate Period 2018-2023 Based on Public Opinion on Twitter," in *2018 6th International Conference on Cyber and IT Service Management (CITSM)*, Aug. 2018, pp. 1–6. doi: 10.1109/CITSM.2018.8674352.
- [15] D. A. Kristiyanti and M. Wahyudi, "Feature selection based on Genetic algorithm, particle swarm optimization and principal component analysis for opinion mining cosmetic product review," in *2017 5th International Conference on Cyber and IT Service Management (CITSM)*, Aug. 2017, pp. 1–6. doi: 10.1109/CITSM.2017.8089278.
- [16] Y. Tian, D. Su, S. Lauria, and X. Liu, "Recent advances on loss functions in deep learning for computer vision," *Neurocomputing*, vol. 497, pp. 129–158, Aug. 2022, doi: 10.1016/j.neucom.2022.04.127.
- [17] S. H. Rudy and T. P. Sapsis, "Output-weighted and relative entropy loss functions for deep learning precursors of extreme events," *Phys. D Nonlinear Phenom.*, vol. 443, p. 133570, Jan. 2023, doi: 10.1016/j.physd.2022.133570.
- [18] Y. Li, Y. Liu, and D. S. Williamson, "On Loss Functions for Deep-Learning Based T60 Estimation," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Jun. 2021, pp. 486–490. doi: 10.1109/ICASSP39728.2021.9414442.
- [19] Q. Gao, Y. Jiang, and S. Wang, "Improvement to Naive Loss Functions with Outlier Identifier," in *2020 IEEE Conference on Telecommunications, Optics and Computer Science (TOCS)*, Dec. 2020, pp. 203–205. doi: 10.1109/TOCS50858.2020.9339696.
- [20] N. Kumari, N. Sharma, and P. Singh, "Performance of Optimizers in Text Summarization for News Articles," *Procedia Comput. Sci.*, vol. 218, pp. 2430–2437, 2023, doi: 10.1016/j.procs.2023.01.218.



Certificate of Recognition

This certificate is proudly presented to

Samuel Ady Sanjaya

for presenting a paper entitled

*Generating Combination of Biblical Baby Names Using Recurrent Neural Network (RNN)
and Optimization Comparison*

The 4th International Conference on Big Data Analytics and Practices (IBDAP 2023)

held on August 25 - 27, 2023

Organized by Big Data Institute (Public Organization)

Assoc. Prof. Tiranee Achalakul, PhD

President of Big Data Institute (Public Organization)

IBDAP 2023

The 4th International Conference
on Big Data Analytics
and Practices

**Bangkok, Thailand
25-27 August 2023**

Big Data Analytics and Mining
Algorithms and systems for big data search and analytics
Machine learning for big data
Predictive analytics and simulation
Big data visualization and interactive data exploration
Big data mining applications
Knowledge extraction, discovery, analysis, and presentation
Big Data Platforms and Technologies
Big data processing frameworks and technologies
Big data services and application development methods and tools
Big data quality evaluation and assurance technologies
Big data system reliability, dependability, and availability
Open-source development and technology for big data
Big Data as a Service (BDaaS) platform and technologies
Big Data and Machine Learning Applications and Experiences
Innovative big data applications and services
Big data analytics in the public sector
Large-scale recommendation systems
Link and graph mining, social network mining
Mobility and big data
Stream data mining
Real-world and large-scale practices of big data

Organizing Committee
Big Data Institute (Public Organization)

28

Proceedings of
The 4th International Conference on
Big Data Analytics and Practices (IBDAP 2023)

August 25-27, 2023

Bangkok, Thailand

Big Data Institute (Public Organization)
Ministry of Digital Economy and Society
ISBN (e-Book): 978-616-8333-02-0

IBDAP 2023, August 25-27, Bangkok, Thailand, 2023

Organized by

Big Data Institute (Public Organization)



Sponsored

Digital Economy Promotion Agency



Ministry of Digital Economy and Society



Supported by

IEEE



IEEE Thailand Section



IEEE Computer Society



The Association of Council of Information

Technology Deans of Thailand



Google Cloud

denodo

alteryx



IBDAP 2023, August 25-27, Bangkok, Thailand, 2023

The 4th International Conference on Big Data Analytics and Practices (IBDAP 2023)

Big Data analytics and smart computing are emerging fields that have recently drawn much attention from research communities in computer science, information technology, social science, and many other disciplines.

The International Conference on Big Data Analytics and Practices (IBDAP 2023), initiated by Big Data Institute (Public Organization), provides an international forum for exchanging ideas and information on current progresses, challenges, research results, system developments, and practical experiences in these emerging fields.

IBDAP calls for submission of work from a wide range of domains, including but not limited to smart systems (e.g. in cities, automobiles, farms, etc.), cyber-physical systems, Internet of Things applications, healthcare, social networks, and media. Authors are encouraged to submit papers covering big data topics, such as service-oriented technologies, machine learning, predictive analytics, data modeling, system architectures, data mining, and simulation.

Published by

Big Data Institute (Public Organization)

Ministry of Digital Economy and Society

ISBN (e-Book): 978-616-8333-02-0

Website IBDAP: <https://www.ibdap.org/>

Contents

	Page
Conference Organizers	1
Keynote Speaker	7
Program Schedule	13
Oral Presentation Session 1	
KNN and Naïve Bayes Algorithms for Improving Prediction of Indonesian Film Ratings Using Feature Selection Technique <i>Frentzen, Jansen Wiratama, Raymond Sunardi Oetama</i>	19
Breast Cancer Prediction of Benign and Malignant Tumors by Classification Algorithms <i>Santad Promtan, Phungern Khongthong, Chidchanok Choksuchat</i>	25
An Integrated Approach for Sentiment Analysis and Topic Modeling of a Digital Bank in Indonesia Using Naïve Bayes and Latent Dirichlet Allocation Algorithms on Social Media Data <i>Johan Setiawan, Anastasia Milenia, Ahmad Faza</i>	31
The Impact of COVID-19 on Volatility Spillover in Stock Exchange of Thailand <i>Nuttaphat Sukchitt, Manad Khamkong</i>	38
Oral Presentation Session 2	
Deep Learning-Based Slip-Trip Falls and Near-Falls Prediction Model Using a Single Inertial Measurement Unit Sensor for Construction Workplace <i>Oleksandr Yuhai, Hyunggun Kim, Ahnryul Choi, Joung Hwan Mun</i>	42
Deep Reinforcement Learning Using a Multi-Scale Agent with a Normalized Reward Strategy for Automatic Cephalometric Landmark Detection <i>Hong Woojae, Seong-Min Kim, Joongyeon Choi, Jun-Young Paeng, Joung Hwan Mun, Hyunggun Kim</i>	48
Implementation of the CNN-SVM Hybrid Model for Leukocoria Detection <i>Siti Ummi Masruroh, Muhammad Hugo Athallah Hardy, Khodijah Hulliyah, Rosa Adelina, Rizka Amalia Putri</i>	54
Reinforcement Learning-Based Contention Window Adjustment for Wireless Body Area Networks <i>Jung-Hyok Kwon, Dongwan Kim, Eui-Jik Kim</i>	60

IBDAP 2023, August 25-27, Bangkok, Thailand, 2023

Page

Generating Combination of Biblical Baby Names Using Recurrent Neural Network (RNN) and Optimization Comparison

64

Samuel Ady Sanjaya, Suyoto

The Online Communication as Mobility in City of Learning: Management of City Learning for Senior Citizen

69

Pastraporn Thipayasothorn, Amornchai Chaichana

Smart Technology Artificial Intelligence Robotics Algorithms (STARA)

73

Awareness Companies in East Java Indonesia?

Imam Mukhlis, Hariyono, Dediek Tri Kurniawan, Rizky Firmansyah, Adelia Shabrina Prameka

Oral Presentation Session 3

Data-To-Question Generation Using Deep Learning

78

Nicole Rachel Koshy, Anshuman Dixit, Siddhi Shrikant Jadhav, Arun V Penmatsa, Sagar V Samanthapudi, Mothi Gowtham Ashok Kumar, Sydney Oghenetega Anuyah, Gourav Vemula, Patricia Snell Herzog, Davide Bolchini

Can Herd Mentality Be Negative? The Data Analysis

84

Shiu-Wan Hung, Jyun-Hao Jian

Classification and Prediction of Pineapple Quality Using Deep Learning

88

Zhehui Li, Yung-Wey Chong, Mohd Nadhir Ab Wahab, Gin-Keat Lim, Rahmad Dawood

Masked Face Recognition Using Deep Learning Techniques

94

Yiyang Ma, Naruemon Wattanapongsakorn

Oral Presentation Session 4

Analysis, Visualization and Interpretation of Determinants of Degree Program

98

Selection of the Senior High School Graduates Utilizing Classification and Clustering Algorithms

Loyd Soriano Echalar, Jonalyn Joy B Labayne, Merardo Camba

Forecasting Air Passengers in Phuket, Thailand: An Application of Machine Learning Algorithms

104

Aphinan Photun, Thanakorn Thaminkaew

IBDAP 2023, August 25-27, Bangkok, Thailand, 2023

Page

10	Machine Learning Models for Micro-Bubble Image Detection in Mosquito Sprayer Quality Control: Addressing Class and Scale Imbalance <i>Suppasit Srisaeng, Pongsakorn Sadakorn, Kritchavat Ploddi, Darin Areechokechai, Worasait Suwannik</i>	110
10	Real-Time Prediction of Free Lime in Cement Clinker Using Support Vector Machine Algorithm <i>Rathimala Kannan, Ahmad Hanif, Ghanim, Kannan Ramakrishnan, Satesh More</i>	116
10	Performance Analysis of TSP Training Instance Generation Methods in Policy-Based Hyper-Heuristic Optimization Framework <i>Orachun Udomkasemsub, Booncharoen Sirinaovakul, Tiranee Achalakul</i>	120
Oral Presentation Session 5		
10	Student Individual Performance in Online Learning: Technology Adoption <i>Oda I. B. Hariyanto, Doni Purnama Alamsyah</i>	126
10	Real-Time Self-Harm Detection Ensuring Safety in Every Moment <i>Nishat Vasker, Mahamudul Hasan</i>	131
10	Comparison of Feasibility Between Machine Learning Algorithms in Terms of Predicting Energy Consumption of Smart Grid <i>Md Tanvir Chowdhury, Noshin Samiha Prova, Md Akram Uddin, Diba Akter Supti, Md. Sabbir Hossain, Mahamudul Hasan, Md Sawkat Ali, Taskeed Jabid, Maheen Islam</i>	137
	Data Security in Cloud Computing Using a Hybrid Algorithm Approach <i>Mahir Tagwar, Md. Asfakul Islam, Md Rokibul Hasan, Md Faisal Azhar, A M Feroz Ehsas, Habibur Rahman, Mahamudul Hasan, Taskeed Jabid, Md Sawkat Ali, Maheen Islam</i>	143
	An Analysis of Bangla Tweets on Social Media Platform for Polarity Detection Using Machine Learning Algorithms <i>Habibur Rahman, Md. Munam Kazi, Mohammad Samiul Islam, Jannatul Mawa, Md. Monjurul Arif, Mahir Tagwar, Mahamudul Hasan, Md Sawkat Ali, Taskeed Jabid, Maheen Islam</i>	149
10	Heart Disease Prediction Using Machine Learning Algorithms <i>Kazi Mostaq Hridoy, Saiful Islam Akash, Fahmida Afroze Dipti, Mahamudul Hasan, Jesan Ahammed Ovi, Md Al-Imran, Sarwar Jahan</i>	155

IBDAP 2023, August 25-27, Bangkok, Thailand, 2023

Page

DeepTestDroid: A Platform for Automated Application Testing Using Deep Learning 161

Md. Huzaifa, Fabiha Bushra Fabin, Nazia Tabassum Natasha, Imran Fakir, Sabiha Afsana Falguni, Mahamudul Hasan, Mohammad Rifat Ahmmad Rashid

Oral Presentation Session 6

A Machine Learning and Deep Learning Integrated Model to Detect Criminal Activities 167

Kanij Fatema Arpa, Tanni Mittra, Tasnim Ferdous, Nusrat Jahan, Riffat Ara Khan Tayna, Mahamudul Hasan, Mohammad Rifat Ahmmad Rashid, Md Sawkat Ali

DeepCONVSVM: A Combining Model for the Detection of a Diseased Mango Leaf 173

Md. Shafayat Hossain, Sarder Iftekhar Ahmed, Md. Nadim, Md. Mizanur Rahman, Maria Mehjabin Shenjuti, Mahamudul Hasan, Taskeed Jabid, Maheen Islam, Md Sawkat Ali

Unmasking Ovary Tumors: Real-Time Detection with YOLOv5 179

Nishat Vasker, Ab. Rahim Ahmed Sowrov, Mahamudul Hasan, Md Sawkat Ali, Mohammad Rifat Ahmmad Rashid, Mohammad Manzurul Islam

Performance Evaluation of Intrusion Detection System Using Machine Learning and Deep Learning Algorithms 185

Md. Sabbir Hossain, Dipayan Ghose, All Masror Partho, Minhaz Ahmed, Md. Tanvir Chowdhury, Mahamudul Hasan, Md Sawkat Ali, Taskeed Jabid, Maheen Islam

Improving EEG-Based Epileptic Seizures Detection Using Self-Normalizing Neural Network 191

Sakorn Mekruksavanich, Ponnipa Jantawong, Anuchit Jitpattanakul

Hybrid Firefly Algorithm with Rao Algorithm for Optimization Problems 196

Sakkayaphop Pravesjit, Krittika Kantawong, Duangjai Jitkongchuen

A Hybrid Deep Learning Neural Network for Recognizing Exercise Activity Using Inertial Sensor and Motion Capture System 200

Sakorn Mekruksavanich, Ponnipa Jantawong, Anuchit Jitpattanakul

Oral Presentation Session 7

Beans Disease Detection Using Convolutional Neural Network 205

Sumaya Akter Suma, Akibul Haque, Nishat Vasker, Mahamudul Hasan, Jesan Ahammed Ovi, Maheen Islam

IBDAP 2023, August 25-27, Bangkok, Thailand, 2023

Page

10	Image Recognition of Different Tea Quality Based on Convolutional Neural Network	210
	<i>Sarmin Sultana Snigdha, Reyad Bin Satter, Waliul Islam, Shafiur Rahman Chowdhury Tonay, Mahamudul Hasan, Maheen Islam, Md Sawkat Ali, Taskeed Jabid</i>	
	Deep Learning-Assisted Fracture Diagnosis: Real-Time Femur Fracture Diagnosis and Categorization	216
	<i>Nishat Vasker, Syed Nawazish Haider, Mahamudul Hasan, Mohammad Salah Uddin</i>	
	Deep Learning for Real-Time Leaf Disease Detection: Revolutionizing Apple Orchard Health	222
	<i>Maheerun Nessa Isty, Nishat Vasker, Fahmida Anjum, Warda Ruhin Parsub, Amit Roy, Akibul Haque, Mahamudul Hasan, Maheen Islam</i>	
	Detecting Pneumonia from X-Ray Images of Chest Using Deep Convolutional Neural Network	228
	<i>Akibul Haque, Mahamudul Hasan, Sumaya Akter Suma, Nishat Vasker, Md. Mahir Ashhab, Md.Sawkat Ali, Mohammad Rifat Ahmmad Rashid, M Saddam Hossain Khan</i>	
10	Using Deep Learning Systems for Imaging Methods to Recognising Brain Tumors	234
	<i>Monjurul Islam Sumon, Md Tanvir Chowdhury, Syed Salman Hayat, Mahir Tagwar, Habibur Rahman, Md. Sabbir Hossain, Mahamudul Hasan, Taskeed Jabid, Maheen Islam</i>	
10	Enhancing IoT Forensics Through Deep Learning: Investigating Cyber-Attacks and Analyzing Big Data for Improved Security Measures	240
	<i>Karam Muhammed Mahdi Salih, Najla Badi Ibrahim</i>	

IBDAP 2023, August 25-27, Bangkok, Thailand, 2023

International Advisory Committee

Kaoru Sezaki (University of Tokyo, Japan)
Lerwen Liu (National University of Singapore, Singapore)
Pierre Vande Vyvre (CERN, Switzerland)
Simon See (Shanghai Jiao Tong University, Singapore)

General Chair

Tiranee Achalakul (BDI, Thailand)

TPC Chair

Duangjai Jitkongcheun (BDI, Thailand) (IEEE Member)

TPC Co-chair

Krisana Chinnasarn (BUU, Thailand) (IEEE Thailand Section)

Publication Chair

Santitham Prom-on (KMUTT, Thailand) (IEEE Member)

Technical Program Chairs

Datchakorn Tancharoen (PIM, Thailand) (IEEE Member)
Parisut Jitpakdee (GBDi, Thailand)
Vorapitchaya Rabiablock (GBDi, Thailand)

Organizing Committees

Akkradach Watcharapupong (KMUTL, Thailand)
Arnond Sakworawich (NIDA, Thailand)
Chaiyaporn Khemapatapan (DPU, Thailand)
Chatchai Powthongchin (SU, Thailand)
Chetneti Srisaan (RSU, Thailand)
Jirapon Sunkpho (TU, Thailand)
Khajonpong Akkarajitsakul (KMUTT, Thailand)
Natawut Nupairoj (Chula, Thailand)
Nattagit Jiteurtragool, (TNI, Thailand)

IBDAP 2023, August 25-27, Bangkok, Thailand, 2023

Organizing Committees

Nawaporn Wisitpongphan (KMUTNB, Thailand)
 Pagaporn Pengsart (MU, Thailand)
 Pakorn Ubolkosold (BU, Thailand)
 Pichaya Tandayya (PSU, Thailand)
 Soonthorn Pibulcharoensit (ABAC, Thailand)
 Sukumal Kitisin (KU, Thailand)
 Surachai Thongkaew (SPU, Thailand)

Technical Program Committees

Adisak Intana (Prince of Songkla University), Thailand
 Ajchara Phu-ang (Thammasat University), Thailand
 Akadej Udomchaiporn (King Mongkut's Institute of Technology Ladkrabang), Thailand
 Akkradach Watcharapupong (King Mongkut's Institute of Technology Ladkrabang), Thailand
 Amarita Ritthipakdee (Phranakorn Rajabhat University), Thailand
 Anantaporn Hanskunatai (King Mongkut's Institute of Technology Ladkrabang), Thailand
 Angkhana Prommarat (Big Data Institute (Public Organization)), Thailand
 Arnond Sakworawich (National Institute of Development Administration), Thailand
 Artit Sagoolmuang (Big Data Institute (Public Organization)), Thailand
 Assadarat Khurat (Mahidol University), Thailand
 Aziz Nanthaamornphong (Prince of Songkla University), Thailand
 Boonsit Yimwadsana (Mahidol University), Thailand
 Chadaporn Keatmanee (Thai-Nichi Institute of Technology), Thailand
 Chakree Teekapakvisit (King Mongkut's Institute of Technology Ladkrabang), Thailand
 Chaloemphon Sirikayon (Dhurakij Pundit University), Thailand
 Chatchai Powthongchin (Silpakorn University), Thailand
 Chetneti Srisaan (Rangsit University), Thailand
 Chidchanok Choksuchat (Prince of Songkla University), Thailand
 Cholrit Luangjinda (Thai-Nichi Institute of Technology), Thailand
 Chuleekorn Nuansomsri (Rangsit University), Thailand
 Datchakorn Tancharoen (Panyapiwat Institute of Management (PIM)), Thailand
 Dechanuchit Katanyutaveetip (Siam University), Thailand
 Eakasit Pacharawongsakda (Dhurakij Pundit University), Thailand
 Ekarat Rattagan (NIDA), Thailand
 Hutchatai Chanlekha (Kasetsart University), Thailand

IBDAP 2023, August 25-27, Bangkok, Thailand, 2023

Technical Program Committees

Jian Qu (Panyapiwat Institute of Management (PIM)), Thailand
 Jirapon Sunkpho (Thammasat University), Thailand
 Kanjana Laosen (Prince of Songkla University), Thailand
 Kanticha Kittipeerachon (Thai-Nichi Institute of Technology), Thailand
 Karin Sumongkayothin (Mahidol University), Japan
 Karn Yongsiriwit (Rangsit University), Thailand
 Kasem Thiptarajan (Thai-Nichi Institute of Technology), Thailand
 Kasikrit Damkliang (Prince of Songkhla University), Thailand
 Kawiwat Amnatchotiphan (Thai-Nichi Institute of Technology), Thailand
 Khajonpong Akkarajitsakul (King Mongkut's University of Technology Thonburi), Thailand
 Khwansiri Sirimangkhalala (Big Data Institute (Public Organization)), Thailand
 Kitsiri Chochiang (Prince of Songkla University), Thailand
 Kitsuchart Pasupa (King Mongkut's Institute of Technology Ladkrabang), Thailand
 Kriangkrai Limthong (Bangkok University), Thailand
 Krittika Kantawong (University of Phayao), Thailand
 Kulwadee Somboonviwat (Kasetsart University Sriracha), Thailand
 Kwankamon Dittakan (Prince of Songkla University), Thailand
 Maleerat Sodanil (King Mongkut's University of Technology North Bangkok), Thailand
 Nantika Prinyapol (Dhurakij Pundit University), Thailand
 Narit Hnoohom (Mahidol University), Thailand
 Narongdech Keeratipranon (Dhurakij Pundit University), Thailand
 Narungsun Wilaisakoolyong (Thai-Nichi Institute of Technology), Thailand
 Natawut Nupairoj (Chulalongkorn University), Thailand
 Nattagit Jiteurtragool (King Mongkut's University of Technology North Bangkok), Thailand
 Nattagit Jiteurtragool (Thai-Nichi Institute of Technology), Thailand
 Nattapol Aunsri (Mae Fah Luang University), Thailand
 Nattapong Sanchan (Bangkok University), Thailand
 Nawaporn Wisitpongphan (King Mongkut's University of Technology North Bangkok), Thailand
 Nilubon Kurubanjerdjit (Mae Fah Luang University), Thailand
 ์Nittaya Kerdyam (Siam University), Thailand
 Nopporn Chotikakamthorn (King Mongkut's Institute of Technology Ladkrabang, Bangkok), Thailand
 Pagaporn Pengsart (Mahidol University), Thailand
 Pakachart Puttipakorn (Thai-Nichi Institute of Technology), Thailand
 Pakapan Limtrairut (Bangkok University), Thailand
 Pakorn Ubolkosold (Bangkok University), Thailand

IBDAP 2023, August 25-27, Bangkok, Thailand, 2023

Technical Program Committees

Panita Thusaranon (Dhurakij Pundit University), Thailand
 Paralee Maneerat (Chulalongkorn University), Thailand
 Peeradon Samasiri (King Mongkut's University of Technology Thonburi), Thailand
 Phaisarn Sudwilai (Advanced Magnetic and Motor Drive Research Laboratory (AMDRL)), Thailand
 Phayung Meesad (King Montkut's University of Technology North Bangkok), Thailand
 Pichaya Tandayya (Prince of Songkla University), Thailand
 Pichit Sukchareonpong (Thai-Nichi Institute of Technology), Thailand
 Pirommas Techitnutsarut (Big Data Institute (Public Organization)), Thailand
 Pornavalai Chotipat (King Mongkut's Insitute of Technology Ladkrabang), Thailand
 Pornthep Rojanavasv (University of Phayao), Thailand
 Prajak Chertchom (Thai-Nichi Institute of Technology), Thailand
 Prajaks Jitngernmadan (Burapha University), Thailand
 Pramuk Boonsieng (Thai-Nichi Institute of Technology), Thailand
 Pranisa Israsena (Thai Nichi Institute of Technology), Thailand
 Preecha Tangworakitthaworn (Mahidol University), Thailand
 Ratchakoon Pruengkarn (Dhurakij Pundit University), Thailand
 Rattana Wetprasit (Prince of Songkla University), Thailand
 Saiyan Saiyod (Khon Kaen University), Thailand
 Sakorn Mekruksavanich (University of Phayao), Thailand
 Salil Boonbrahm (Walailak University), Thailand
 Sangsuree Vasupongayya (Prince of Songkla University), Thailand
 Saprangsit Mruetusatorn (Thai-Nichi Institute of Technology), Thailand
 Saranthorn Phusingha (Big Data Institute (Public Organization)), Thailand
 Sarawut Ramjan (Thammasat University), Thailand
 Sarayut Nonsiri (Thai-Nichi Institute of Technology), Thailand
 Saromporn Charoenpit (Thai-Nichi Institute of Technology), Thailand
 Silada Intarasothonchun (Khon Kaen University), Thailand
 Sinchai Kamolphiwong (Prince of Songkla University), Thailand
 Sirion Vittayakorn (King Mongkut's Institute of Technology Ladkrabang), Thailand
 Songsri Tangsripairoj (Mahidol University), Thailand
 Soontarin Nupap (Mae Fah Luang University), Thailand
 Soonthorn Pibulcharoensit (Assumption University), Thailand
 Suchart Khummanee (Mahasarakham University), Thailand
 Sukumal Kitisin (Kasetsart University), Thailand
 Supaporn Chairungsee (Walailak University), Thailand

IBDAP 2023, August 25-27, Bangkok, Thailand, 2023

Technical Program Committees

Suppakarn Chansareewittaya (Mae Fah Luang University), Thailand
 Suppat Rungraungsilp (Walailak University), Thailand
 Surachai Thongkaew (Sripatum University), Thailand
 Surangkana Rawungyot (University of Phayao), Thailand
 Surapong Uttama (Mae Fah Luang University), Thailand
 Suttisak Jantavongso (Rangsit University), Thailand
 Tanapon Jensuttiwetchakul (King Mongkut's University of Technology North Bangkok), Thailand
 Taravichet Titijaronroj (King Mongkut's Institute of Technology Ladkrabang), Thailand
 Thanu Udomsripaiboon (University of Phayao), Thailand
 Thanapat Kangkachit (Dhurakij Pundit University), Thailand
 Thanapon Noraset (Mahidol University), Thailand
 Thanathip Limna (Prince of Songkla University), Thailand
 Theekapun Charoenpong (Srinakharinwirot University), Thailand
 Thitinan Tantidham (Mahidol University), Thailand
 Thitiporn Lertrudachakul (Thai-Nichi Institute of Technology), Thailand
 Thongchai Kaewkiriya (Panyapiwat Institute of Management), Thailand
 Toshiaki Kondo (Sirindhorn International Institute of Technology), Thailand
 Ungsumalee Suttapakti (Burapha University), Thailand
 Vanvisa Chutchavong (King Mongkut's Institute of Technology Ladkrabang), Thailand
 Vasin Chooprayoon (Rangsit University), Thailand
 Virach Sornlertlamvanich (Musashino University), Japan
 Vithida Chongsuphajaisiddhi (King Mongkut's University of Technology Thonburi), Thailand
 Vorapitchaya Rabiablock (Big Data Institute (Public Organization)), Thailand
 Warakorn Srichavengsup (Thai-Nichi Institute of Technology), Thailand
 Warangkhan Kimpan (King Mongkut's Institute of Technology Ladkrabang), Thailand
 Waraporn Jirapanthong (Dhurakij Pundit University), Thailand
 Wasimon Panichpattanakul (Prince of Songkla University), Thailand
 Watchareewan Jitsakul (King Mongkut's University of Technology North Bangkok), Thailand
 Weerapat Satitkanitkul (Big Data Institute (Public Organization)), Thailand
 Weerawut Thanhikam (Panyapiwat Institute of Management), Thailand
 Wimol San-Um (Thai-Nichi Institute of Technology), Thailand
 Wiphada Wettayaprasit (Prince of Songkla University), Thailand
 Wisarn Patchoo (Bangkok University), Thailand
 Worapan Kusakunniran (Mahidol University), Thailand
 Worapat Paireekreng (Dhurakij Pundit University), Thailand

IBDAP 2023, August 25-27, Bangkok, Thailand, 2023

Technical Program Committees

Worasak Rueangsirarak (Mae Fah Luang University), Thailand

Woratat Makasiranondh (Rangsit University), Thailand

Wudhichart Sawangphol (Mahidol University), Thailand

IBDAP 2023, August 25-27, Bangkok, Thailand, 2023

Keynote Speaker



Opening Ceremony

Assoc. Prof. Tiranee Achalakul, Ph.D.

President of Big Data Institute (Public Organization)



IBDAP 2023, August 25-27, Bangkok, Thailand, 2023

Keynote Speaker

Keynote Session 1



Asst. Prof. Phond Phunchongharn, Ph.D.

Acting Director of Industrial Promotion and Partnership Division

Big Data Institute (Public Organization)

Title: Exploring Thailand's Data Business Ecosystem



IBDAP 2023, August 25-27, Bangkok, Thailand, 2023

Keynote Speaker

Keynote Session 2



Dr. Nitipong boonlong

Regional Director for ASEAN Region,
Indochina Denodo



Dr. Santisook Limpeeticharoenchot

Chief Executive Officer of STelligence
Company

Mastering Big Data Using a Logical Approach for Faster Analytics and Unified Insights

Big Data and data warehouses offer organizations centralized data delivery platforms. In this session, we will share how organizations can leverage a logical data management approach to achieve a holistic view of their data across Big Data, data warehouses and data sources, and help deliver faster analytics and unified insights in a secured way. Learn how using this logical approach, combined with Data Virtualization, organizations can accelerate data integration, simplify access and data governance, and foster self-service capabilities.



IBDAP 2023, August 25-27, Bangkok, Thailand, 2023

Keynote Speaker

Keynote Session 3



Mr. Niti Charoenkwanchai

Account Executive, Large Enterprise – Thailand and Vietnam at Alteryx

Title: The Democratization of Analytics - Scale the Impact of Analytics Across the
Organization

alteryx

IBDAP 2023, August 25-27, Bangkok, Thailand, 2023

Keynote Speaker

Keynote Session 4



Prachya Boonkwan, Ph.D.

Senior Researcher, NECTEC, Thailand

Topic: ChatGPT and Business Opportunities



IBDAP 2023, August 25-27, Bangkok, Thailand, 2023

Keynote Speaker

Special Session 5



Mr. Wittawin Waiyapat

Customer Engineering Consultant, Google Cloud

Title: [Generative AI Updates by Google](#)



IBDAP 2023, August 25-27, Bangkok, Thailand, 2023

Program Schedule

Program (Day 1) Friday, 25 August 2023	
08.00 - 15.00	Registration
09.00 - 09.30	Opening Ceremony Assoc. Prof. Tiranee Achalakul, Ph.D. President of Big Data Institute (Public Organization)
09.30 - 10.15	Keynote Session 1 Keynote Speaker: Asst. Prof. Phond Phunchongharn, Ph.D. Acting Director of Industrial Promotion and Partnership Division, Big Data Institute (Public Organization) <i>Title: Exploring Thailand's Data Business Ecosystem</i>
10.15 - 10.30	Session Break
10.30 - 12.00	Keynote Session 2 Keynote Speaker: (1) Dr. Nitipong boonlong - Regional Director for ASEAN Region, Indochina Denodo (2) Dr. Santisook Limpeeticharoenchot - Chief Executive Officer of STelligence Company <i>Title: Mastering Big Data Using a Logical Approach for Faster Analytics and Unified Insights</i>
12.00 - 13.30	Lunch
13.30 - 14.45	Oral Session 1
13.30 - 13.45	KNN and Naïve Bayes Algorithms for Improving Prediction of Indonesian Film Ratings Using Feature Selection Technique
13.45 - 14.00	Breast Cancer Prediction of Benign and Malignant Tumors by Classification Algorithms
14.00 - 14.15	An Integrated Approach for Sentiment Analysis and Topic Modeling of a Digital Bank in Indonesia Using Naïve Bayes and Latent Dirichlet Allocation Algorithms on Social Media Data
14.15 - 14.30	The Impact of COVID-19 on Volatility Spillover in Stock Exchange of Thailand

IBDAP 2023, August 25-27, Bangkok, Thailand, 2023

Program (Day 1) Friday, 25 August 2023	
14.45 - 15.00	Session Break
15.00 - 17.00	Oral Session 2
15.00 - 15.15	Deep Learning-Based Slip-Trip Falls and Near-Falls Prediction Model Using a Single Inertial Measurement Unit Sensor for Construction Workplace
15.15 - 15.30	Deep Reinforcement Learning Using a Multi-Scale Agent with a Normalized Reward Strategy for Automatic Cephalometric Landmark Detection
15.30 - 15.45	Implementation of the CNN-SVM Hybrid Model for Leukocoria Detection
15.45 - 16.00	Reinforcement Learning-Based Contention Window Adjustment for Wireless Body Area Networks
16.00 - 16.15	Generating Combination of Biblical Baby Names Using Recurrent Neural Network (RNN) and Optimization Comparison
16.15 - 16.30	The Online Communication as Mobility in City of Learning: Management of City Learning for Senior Citizen
16.30 - 16.45	Smart Technology Artificial Intelligence Robotics Algorithms (STARA) Awareness Companies in East Java Indonesia?
18.00 - 21.00	Banquet

IBDAP 2023, August 25-27, Bangkok, Thailand, 2023

Program (Day 2) Saturday, 26 August 2023	
08.00 - 15.00	Registration
09.00 - 09.40	Keynote Session 3 Keynote Speaker: Mr. Niti Charoenkwanchai Account Executive, Large Enterprise - Thailand and Vietnam at Alteryx <i>Title: The Democratization of Analytics - Scale the Impact of Analytics Across the Organization</i>
09.40 - 10.20	Keynote Session 4 Keynote Speaker: Prachya Boonkwan, Ph.D. Senior Researcher, NECTEC, Thailand <i>Title: ChatGPT and Business Opportunities</i>
10.20 - 10.30	Session Break
10.30 - 12.00	Keynote Session 5 Keynote Speaker: Mr. Wittawin Waiyapat Customer Engineering Consultant, Google Cloud <i>Title: Generative AI Update by Google</i>
12.00 - 13.30	Lunch
13.30 - 14.45	Oral Session 3
13.30 - 13.45	Data-To-Question Generation Using Deep Learning
13.45 - 14.00	Can Herd Mentality Be Negative? The Data Analysis
14.00 - 14.15	Classification and Prediction of Pineapple Quality Using Deep Learning
14.15 - 14.30	Masked Face Recognition Using Deep Learning Techniques
14.45 - 15.00	Session Break
15.00 - 17.00	Oral Session 4
15.00 - 15.15	Analysis, Visualization and Interpretation of Determinants of Degree Program Selection of the Senior High School Graduates Utilizing Classification and Clustering Algorithms
15.15 - 15.30	Forecasting Air Passengers in Phuket, Thailand: An Application of Machine Learning Algorithms

IBDAP 2023, August 25-27, Bangkok, Thailand, 2023

Program (Day 2) Saturday, 26 August 2023	
15.30 – 15.45	Machine Learning Models for Micro-Bubble Image Detection in Mosquito Sprayer Quality Control: Addressing Class and Scale Imbalance
15.45 – 16.00	Real-Time Prediction of Free Lime in Cement Clinker Using Support Vector Machine Algorithm
16.00 – 16.15	Performance Analysis of TSP Training Instance Generation Methods in Policy- Based Hyper-Heuristic Optimization Framework

IBDAP 2023, August 25-27, Bangkok, Thailand, 2023

Program (Day 3) Sunday, 27 August 2023	
09.00 - 11.00	Oral Session 5
09.00 - 09.15	Student Individual Performance in Online Learning: Technology Adoption
09.15 - 09.30	Real-Time Self-Harm Detection Ensuring Safety in Every Moment
09.30 - 09.45	Comparison of Feasibility Between Machine Learning Algorithms in Terms of Predicting Energy Consumption of Smart Grid
09.45 - 10.00	Data Security in Cloud Computing Using a Hybrid Algorithm Approach
10.00 - 10.15	An Analysis of Bangla Tweets on Social Media Platform for Polarity Detection Using Machine Learning Algorithms
10.15 - 10.30	Heart Disease Prediction Using Machine Learning Algorithms
10.30 - 10.45	DeepTestDroid: A Platform for Automated Application Testing Using Deep Learning
09.00 - 11.00	Oral Session 6
09.00 - 09.15	A Machine Learning and Deep Learning Integrated Model to Detect Criminal Activities
09.15 - 09.30	DeepCONVSVM: A Combining Model for the Detection of a Diseased Mango Leaf
09.30 - 09.45	Unmasking Ovary Tumors: Real-Time Detection with YOLOv5
09.45 - 10.00	Performance Evaluation of Intrusion Detection System Using Machine Learning and Deep Learning Algorithms
10.00 - 10.15	Improving EEG-Based Epileptic Seizures Detection Using Self-Normalizing Neural Network
10.15 - 10.30	Hybrid Firefly Algorithm with Rao Algorithm for Optimization Problems
10.30 - 10.45	A Hybrid Deep Learning Neural Network for Recognizing Exercise Activity Using Inertial Sensor and Motion Capture System
09.00 - 11.00	Oral Session 7
09.00 - 09.15	Beans Disease Detection Using Convolutional Neural Network
09.15 - 09.30	Image Recognition of Different Tea Quality Based on Convolutional Neural Network

IBDAP 2023, August 25-27, Bangkok, Thailand, 2023

Program (Day 3) Sunday, 27 August 2023	
09.30 - 09.45	Deep Learning-Assisted Fracture Diagnosis: Real-Time Femur Fracture Diagnosis and Categorization
09.45 - 10.00	Deep Learning for Real-Time Leaf Disease Detection: Revolutionizing Apple Orchard Health
10.00 - 10.15	Detecting Pneumonia from X-Ray Images of Chest Using Deep Convolutional Neural Network
10.15 - 10.30	Using Deep Learning Systems for Imaging Methods to Recognising Brain Tumors
10.30 - 10.45	Enhancing IoT Forensics Through Deep Learning: Investigating Cyber-Attacks and Analyzing Big Data for Improved Security Measures

KNN And Naive Bayes Algorithms for Improving Prediction of Indonesian Film Ratings using Feature Selection Techniques

¹Frentzen

InformationSystems, Faculty of
Engineering and Informatics
Universitas Multimedia Nusantara
Tangerang, Indonesia
frentzen@student.umn.ac.id

²Jansen Wiratama*

InformationSystems, Faculty of
Engineering and Informatics
Universitas Multimedia Nusantara
Tangerang, Indonesia
jansen.wiratama@umn.ac.id

³Raymond Sunardi Oetama

InformationSystems, Faculty of
Engineering and Informatics
Universitas Multimedia Nusantara
Tangerang, Indonesia
raymond@umn.ac.id

Abstract—This electronic document is a “live” template and already defines the components of your paper [title, text, heads, etc.] in its style sheet. ***CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract.** (Abstract) A survey conducted by IDNTimes with 411 respondents shows that audiences in Indonesia choose which films to watch based on several factors. People chose films to watch in theatres based on the genre by 16.4%, actors or actors by 14.5%, and directors by 12.7%. There needs to be more interest from researchers regarding Indonesian film ratings. Most of them discussed foreign films more than Indonesian films. So research is required that discusses Indonesian film ratings. This research will use data mining tools, namely RapidMiner, with the CRISP-DM framework method. The study compared two algorithms, Naïve Bayes and KNN, to predict Indonesian film ratings using the feature selection technique. This research yielded that the KNN algorithm performs better than Naïve Bayes with or without feature selection techniques. Of the three models made for KNN, namely with Forward Selection, Backward Elimination, and without feature selection, all produce the same accuracy value, namely 86.65%. At the same time, Naïve Bayes gets an accuracy value of 85.13% with forward selection, 84.91% with backward elimination, and 71.32% without feature selection technique.

Keywords—CRISP-DM, Feature Selection, KNN, Naïve bayes, rating.

I. INTRODUCTION

According to the Big Indonesian Dictionary, the film means a play or a moving picture story. In English, a film is usually called a movie [1]. New films will usually be shown in theatres or cinemas. Cinemas or cinemas in Indonesia do not only show films originating from the film industry from abroad, for example, Hollywood (United States) or Bollywood (India) but also show films originating from Indonesia itself. The film industry in Indonesia is experiencing development. The development of the film industry in Indonesia can be seen from the increasing audience of Indonesian films. Based on Indonesia.go.id website, in 2016, the number of viewers reached 34.5 million; in 2017, it increased to 40.5 million, and in 2018 it touched more than 50 million viewers. Even Indonesian films produced in 2018 totalled 200 titles, an increase from the previous year, which only produced 143 titles [2]. A survey conducted by IDNTimes with 411 respondents shows that audiences in Indonesia choose which films to watch based on several factors. People choose movies to watch in theatres based on the genre by 16.4%, film actors or actors by 14.5%, and directors by 12.7% [3]. Regarding the player or actor factor, the influence of the actor's popularity is also quite influential on a film's rating. Many Indonesian film players or actors also

work as YouTubers and have YouTube channels. Of course, this can increase their popularity, regardless of whether it has increased due to viral positive or negative things [4]. To find out the ratings of Indonesian films from spectators, an evaluation of a film is required, usually in the form of a rating. Rating is a conclusion that describes the quality of the elements in a film [5]. Films with high ratings will have good quality so that they will be in demand by the audience. Conversely, films with low ratings will be less desirable. This film rating can also determine the success or popularity of a film.

There needs to be more interest from researchers regarding Indonesian film ratings. Most of them discussed films abroad more than Indonesian films [5] [6] [7] [8] [9]. Because there is still very little research on Indonesian film ratings, research that discusses Indonesian film ratings is needed. Research on Indonesian film ratings can be done using data mining techniques. Data mining is looking for patterns and obtaining helpful information using specific techniques or methods. Data mining has several fields: prediction, classification, description, association, estimation, and clustering [10]. Prediction is a technique of systematically estimating something that is most likely to happen in the future based on data from the past and currently owned data [8]. Prediction can be interpreted as forecasting or forecasting a value that has yet to be known for the future. With this prediction, it is hoped that it can produce valuable information for filmmakers to improve film quality and for future film development in Indonesia.

Previous studies to predict film ratings based on these ratings have used several data mining algorithms. The methods used in previous studies were: Naïve Bayes with an accuracy of 55.80% [5], Naïve Bayes and K-Nearest Neighbor of 53.99% and 51.26% [8]. Based on some of the previous studies, it can be concluded that the Naïve Bayes algorithm is the most popular algorithm to be used in predicting film rating classifications but with unfavourable results.

For this study, predictions will be made using film data from Indonesia from 2011 to 2020 to improve model performance based on previous research. The model will be created using the Naïve Bayes and KNN (K-Nearest Neighbor) algorithms with the Feature Selection technique to determine ratings for Indonesian films. The Feature Selection technique is used in this study because of its usefulness in selecting the most relevant features and improving the performance of the model created [11].

Based on previous studies relevant to this research, the results show that the KNN and Naïve Bayes algorithms are famous for predicting ratings and feature selection techniques

to improve model accuracy. CRISP-DM is a suitable craft method. However, the results obtained in these studies are still not good enough, so they are used as the basis for improving the algorithm's performance that will be carried out in this study. This research aims to produce the K-Nearest Neighbor (KNN) model and the Naïve Bayes model to predict Indonesian film ratings. After the predictions are obtained, this research contributes to improving the KNN and the Naïve Bayes models' accuracy based on previous research using feature selection techniques. The results obtained in this study contributed to an increase in ratings on Indonesian film ratings.

II. METHODOLOGY

A. CRISP-DM

Cross Industry Standard Process for Data mining, commonly known by the acronym CRISP-DM, is a popular method for increasing the success of data mining models. The CRISP-DM method is a cycle consisting of six stages that will help the implementation of data mining to be applied in natural environments, such as helping to make business decisions [12]. The six stages in the CRISP-DM method cycle, according to [13], are:

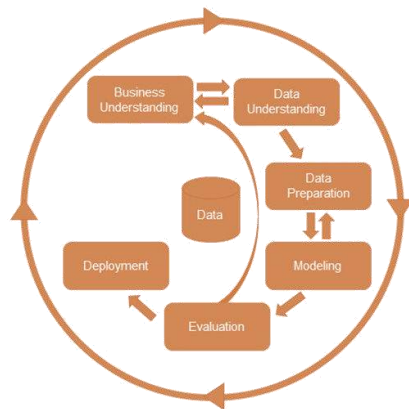


Fig 1. CRISP-DM six stages

1. The first stage carried out in the CRISP-DM method is Business Understanding. Business understanding is carried out to plan and understand the objectives to be achieved in research. is the first stage of the CRISP-DM method. At this stage, an understanding of the project's business will be carried out by determining the goals and needs of the project, as well as preparing an initial strategy to achieve the desired goals.
2. Data Understanding is the second stage of the CRISP-DM method. The data comes from a dataset of Indonesian film ratings from the Kaggle.com website collected by Dionisius Darryl Hermansyah from the Bandung Institute of Technology under the title IMDb Indonesian Movies. The data comes from the IMDb.com website (Internet Movie Database) using the IMDb-scraper tool, which is then converted and cleaned into CSV form. This dataset contains Indonesian film ratings from 1926 to 2020, with 1272 film titles. The Indonesian Movies IMDb dataset also has 11 attributes, eight attributes of nominal type and three attributes of numeric type. These attributes will later be used to build a model

for predicting Indonesian film ratings. Several values still need to be found in the IMDb Indonesian Movies dataset. This missing value needs to be corrected in the data preparation stage by preprocessing the data so that this dataset can be used for the modelling stage later.

3. Data Preparation is the third stage of the CRISP-DM method. The data preparation stage is to prepare the dataset for modelling. Data preprocessing is carried out at this stage to improve and produce good data ready for the following modelling stage. The data preprocessing carried out in this study is filtering and data cleansing. Data filtering is done to select the data to be used; the dataset starts from 1926, while only data from 2011 to 2020 will be used for this study. After the data is filtered, data cleansing will be done to correct the missing values in the attributes on the dataset. One way to overcome missing values is to delete rows from data with a missing value. After the data has been filtered and cleaned, this new dataset is ready for the next stage, namely modelling.
4. Data Modeling is the fourth stage of the CRISP-DM method. Data mining techniques will be applied to create a model in the modelling stage. The data mining technique chosen for this research is Naïve Bayes and KNN. The tools used for implementation are Rapid Miner. The Naïve Bayes technique is chosen because Naïve Bayes is a popular algorithm for predicting a film's rating. Based on previous research, Naïve Bayes was used twice with accuracy results of 53.99% [8] and 55.80% [6], and KNN was chosen as the comparison algorithm in this study. Naïve Bayes is an algorithm that calculates a set of probabilities by adding the sum of the frequencies and combinations of values from the given dataset [17]. In contrast to the Naïve Bayes algorithm, the KNN algorithm finds the shortest distance between data to be evaluated with k-neighbours in training data [17].
5. The Evaluation stage is the stage where an evaluation of the model made in the previous stage is carried out to determine whether it is following the business objectives in the Business Understanding stage. Data testing, which amounts to 30% of the total dataset, will be used to test and validate using the 10-fold cross-validation method.
6. The Deployment stage, according to previous research [13], can be done by producing a final report that has been made based on the deployment plan. In this study, the researcher did not carry out the Deployment stage because this research needed to be implemented in companies but only for research purposes.

B. Naïve Bayes Technique

Naïve Bayes is a statistical grouping technique that can predict the probability of class members [6]. Naïve Bayes also has extreme accuracy and speed when applied to databases with big data, according to [8]. The following is the formula from Naïve Bayes:

$$P(Y) = \frac{P(X|Y)P(Y)}{P(X)}$$

Equation 1. Naïve bayes formula

Description of Equation 1:

- 14
- X : Data with unknown class
 - Y : The data hypothesis is a specific class
 - $P(Y|X)$: Probability of the hypothesis based on condition X (posteriori probability)
 - $P(Y)$: The probability of the hypothesis Y (prior probability)
 - $P(X|Y)$: Probability of X under the condition of the hypothesis Y
 - $P(X)$: Probability of X

C. KNN Algorithm

The KNN or K-Nearest Neighbor algorithm is a method for classifying objects based on learning data with the closest distance to the thing. Knowing data will be projected into a multidimensional space where each dimension will represent the features of the data [8]. The following is the Equation of the KNN algorithm:

$$distance = \sqrt{\sum_{i=1}^n (x_{training}^i - x_{testing})^2}$$

Equation 2. KNN Algorithm formula

- $x_{training}^i$: Data training n-i
- $x_{testing}$: Data testing
- I : Row of- from the table
- n : Total data training

D. Feature Selection

Feature selection is a preprocessing technique in data mining to select relevant features and reduce data by removing unimportant and redundant attributes from datasets. Feature selection aims to get better prediction performance, speed up the prediction process, reduce costs, and know how to produce better data [11].

E. Forward Selection

Forward selection is one method of feature selection technique. Forward selection aims to add controlled variables one by one into the equation. Forward selection starts from the empty set, then adds variables individually and is evaluated until all variables have been tested [14].

F. Backward Elimination

Backward elimination is also a method of feature selection technique. Backward elimination works oppositely to the forward selection method. The backwards elimination process begins by including all existing variables and eliminating 1 per 1 variable that is considered insignificant based on the value of accuracy or error. The process continues until there are no more insignificant variables [14].

G. RapidMiner tools

The tools used in carrying out this research are the RapidMiner software. RapidMiner can import information from various database sources, which are then analyzed in an

application [6]. RapidMiner is a standard data mining tool developed by the RapidMiner company in Germany [15]. RapidMiner is focused on processes that may contain subprocesses. Procedures include operators in the form of visual components. Operators are implementations of DM algorithms and data sources. Data flows are constructed by drag-and-drop operators and by connecting the inputs and outputs of the appropriate operators [15] [18].

III. RESULTS AND DISCUSSION

The data comes from a dataset of Indonesian film ratings from the Kaggle.com website collected by Dionisius Darryl Hermansyah from the Bandung Institute of Technology under the title IMDb Indonesian Movies. The data comes from the IMDb.com website (Internet Movie Database) using the IMDb-scraper tool, which is then converted and cleaned into CSV form. This Indonesian film rating dataset contains variables in the form of numeric and nominal that affect the rating that will be obtained from a film. The Indonesian film rating dataset variables are title, year, description, genre, rating, votes, languages, directors, actors, and runtime, user rating. These variables are divided into two data types: Nominal and Numeric. Numeric is data in the form of numbers, while nominal is categorical data [16].

Table 1. Dataset and data type

No	Attribute	Description	Type
1	title	Film title	Nominal
2	year	Year of release	Numeric
3	description	Film description	Nominal
4	genre	Film genre	Nominal
5	MPAA rating	Age rating certifications	Nominal
6	votes	Total vote	Numeric
7	languages	Film language	Nominal
8	directors	Film director	Nominal
9	actors	Actor/Actress	Nominal
10	runtime	Film duration	Nominal
11	user rating	IMDB user rating	Numeric

A. Data Preparation

Data preparation is the stage where the dataset that has been collected will be prepared so that it can be used in the subsequent modelling process. At this stage, we will handle missing values and generalize several variables for analysis and classification [7]. The variable that will be generalized at this stage is users_rating. The users_rating variable is in the form of a number with a scale of 1 to 10. A value of 1 indicates that the film is bad and is one of the worst films ever watched, and a value of 10 indicates that the film is excellent or good. The users_rating variable, which will become a class in the prediction, is generalized into 2, namely Good and Poor and forms a new variable named rating_2class with the following division:

Table 2. Class Division

users_rating	class
1-5	Poor
>5	Good

Next, the missing values on the attributes will be checking used using the Rstudio tools. The following is the syntax used. After that, the variables used in the modelling process are clean of missing values and ready for use in this study:

```
missingvalue = dfSummary(Indonesian_Movies, valid.col=FALSE,
                          graph.col=FALSE, labels.col=FALSE,
                          stats.col=FALSE)
missingvalue$`Stats / Values` = NULL
missingvalue$`Freqs (% of Valid)` = NULL
view(missingvalue)
```

Fig 2. Syntax for the missing value

B. Modelling

Before starting the modelling process, a K search must be carried out to determine the value of K that will be used for the KNN algorithm later. This K search was carried out using Rstudio tools with the following results:

```
k-Nearest Neighbors
464 samples
8 predictor
2 classes: 'Good', 'Poor'

No pre-processing
Resampling: Cross-validated (10 fold)
Summary of sample sizes: 418, 418, 417, 417, 417, ...
Resampling results across tuning parameters:

k  Accuracy  Kappa
5  0.8730342  0.2812610
7  0.8706753  0.2036466
9  0.8707216  0.1944859

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 5.
```

Fig 3. K Finding results

Based on the results in Figure 4.4, there are 3 K values of 5, 7, and 9 with respective accuracy values of 0.873, 0.870, and 0.870. With this, it can be determined that the best K is five, which is considered suitable for this study. The work on the modelling stage is carried out based on the flowchart in Figure 4 as follows:

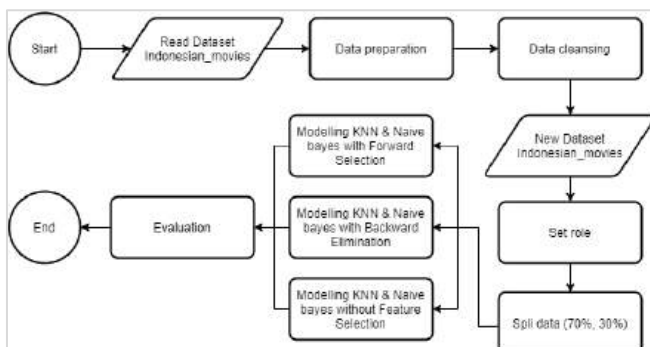


Fig 4. Flowchart modelling

C. Multiply

Because modelling is carried out using two algorithms, KNN and Naïve Bayes [17] have three comparisons: Forward selection, Backward elimination and only Cross-Validation. A Multiply operator is needed to make copies of RapidMiner [18] objects to be connected to other ports.

D. Modelling forward selection KNN

In this simulation, the Forward selection operator selects five attributes, genre, year, directors, MPAA_rating, and actors_1, as the attribute with the best performance. The accuracy obtained from the KNN model with Forward

selection is also quite good at 86.65%, which can be seen in the following figure:

PerformanceVector

```
PerformanceVector:
accuracy: 86.65% +/- 3.34% (micro average: 86.65%)
ConfusionMatrix:
True:   Good   Poor
Good:  371    43
Poor:   18    25
```

Fig 5. Forward selection of KNN performance and accuracy

E. Modelling forward selection Naïve bayes

In this simulation, the Forward selection operator only selects one attribute, namely the year, as the attribute with the best performance. The accuracy obtained from the Naïve Bayes model with Forward selection is also quite good at 85.13%, which can be seen in the following figure:

PerformanceVector

```
PerformanceVector:
accuracy: 85.13% +/- 1.25% (micro average: 85.12%)
ConfusionMatrix:
True:   Good   Poor
Good:  389    68
Poor:    0     0
```

Fig 6. Forward selection of Naive bayes performance and accuracy

F. Modelling backward elimination KNN

In this simulation, the Backward elimination operator selects six attributes, namely genre, year, directors, MPAA_rating, actors_1, and runtime_generalized, as the attributes that have the best performance. The accuracy obtained from the KNN model with Forward selection is also quite good at 86.65%, which can be seen in the following figure:

PerformanceVector

```
PerformanceVector:
accuracy: 86.65% +/- 3.34% (micro average: 86.65%)
ConfusionMatrix:
True:   Good   Poor
Good:  371    43
Poor:   18    25
```

Fig 7. Backward elimination of KNN performance and accuracy

G. Modelling backward elimination Naïve bayes

In this simulation, the Backward elimination operator selects four attributes, namely year, MPAA_rating, languages, and runtime_generalized, as the attributes that have the best performance. The accuracy obtained from the Naïve Bayes model with Forward selection is also quite good, which is 84.91%, which can be seen in the following figure:

PerformanceVector

```
PerformanceVector:
accuracy: 84.91% +/- 1.49% (micro average: 84.90%)
ConfusionMatrix:
True:   Good   Poor
Good:  388    68
Poor:    1     0
```

Fig 8. Backward elimination of Naive bayes performance and accuracy

Identify applicable funding agency here. If none, delete this text box.

H. Modelling KNN without feature selection

Furthermore, the accuracy obtained from the KNN model without the Feature Selection technique is 86.65%.

```

PerformanceVector

PerformanceVector:
accuracy: 86.65% +/- 3.34% (micro average: 86.65%)
ConfusionMatrix:
True:    Good    Poor
Good:    371     43
Poor:    18      25
  
```

Fig 9. KNN performance and accuracy without feature selection

I. Modelling Naïve bayes without feature selection

Then in the last simulation, the accuracy obtained from the Naïve Bayes model without the Feature selection technique was 71.32%.

```

PerformanceVector

PerformanceVector:
accuracy: 71.32% +/- 7.13% (micro average: 71.33%)
ConfusionMatrix:
True:    Good    Poor
Good:    284     26
Poor:    105     42
  
```

Fig 10. Naive bayes performance and accuracy without feature selection

J. Evaluation

The evaluation stage is carried out by comparing the model's performance using data testing of 30%. In Fig 11, we can see a comparison of the version of the model with and without feature selection techniques.

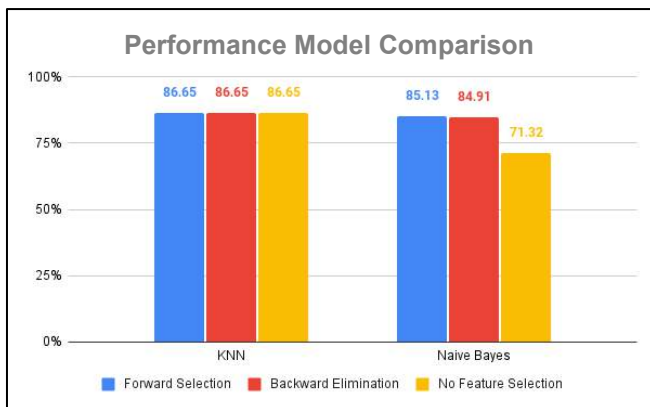


Fig 11. Performance model comparison

Table 3. Selected Variables

	actors	directors	genres	languages	MPAA_rating	runtime	year
KNN	✓	✓	✓		✓		✓
NB							✓
KNN	✓	✓	✓		✓	✓	✓
NB				✓	✓	✓	✓
KNN	✓	✓	✓	✓	✓	✓	✓
NB	✓	✓	✓	✓	✓	✓	✓

Based on Table 1 regarding the dataset used to compare the KNN and Naive Bayes algorithms, seven variables were chosen because of their relevance to the tests of the two algorithms: actors, directors, genres, languages, MPAA_rating, runtime, and year. In the forward selection technique, the variables used by the KNN algorithm are

actors, directors, genres, and MPAA_rating for testing because the KNN algorithm requires much more information than the Naive Bayes algorithm. Meanwhile, the variable used by the Naive Bayes algorithm is only the year variable because the year variable has relevance for predicting based on previous probability.

Furthermore, in the backward elimination technique, the KNN algorithm uses six variables, except languages, because it is less relevant to the backward elimination technique with the KNN algorithm. In comparison, the Naive Bayes algorithm uses four variables: languages, MPAA_rating, runtime, and year. Then in testing without features selection, the two algorithms use all available variables.

The K-Nearest Neighbor (KNN) algorithm consistently achieves an accuracy of 86.65% across all three models. In contrast, the Naïve Bayes algorithm demonstrates varying accuracy values: 85.13% with forward selection, 84.91% with backward elimination, and 71.32% without feature selection. Notably, the year variable is frequently included in both KNN and Naïve Bayes models. The impact of feature selection techniques differs between the algorithms. For KNN, feature selection has no discernible effect as the accuracy remains constant across the three models, possibly due to limited selectable variables. In contrast, feature selection significantly affects the Naïve Bayes model. Without feature selection, the model achieves only 71.32% accuracy. With backward elimination, accuracy increases to 84.91%, and forward selection yields the highest accuracy at 85.13%. The comparison of accuracy values presented in Figure 11 and Table 3 indicates that the KNN algorithm outperforms the Naïve Bayes algorithm, regardless of the use of feature selection techniques. The KNN algorithm consistently achieves better results, suggesting its superiority in this context.

IV. CONCLUSION

This research produced models for two different algorithms, KNN and Naïve Bayes, using forward selection, backward elimination, and without feature selection. Based on the research that has been done, the following conclusions are obtained:

- The K-Nearest Neighbor algorithm obtains an accuracy value of 86.65% with forward selection, 86.65% with backward elimination, and 86.65% without feature selection techniques.
- The Naïve Bayes algorithm obtains an accuracy value of 85.13% with forward selection, 84.91% with backward elimination, and 71.32% without feature selection techniques.
- The feature selection technique does not affect the KNN algorithm in this study, as evidenced by the results that the accuracy of the 3 KNN models made is the same, with an accuracy of 86.65%.
- The feature selection technique affects the Naïve Bayes algorithm in this study, as evidenced by the results of the model accuracy without the feature selection technique only getting 71.32% and increasing with the backward elimination method of 84.91%, forward selection 85.13%.

- e. The feature selection technique with the forward selection method produces the best model for the Naïve Bayes algorithm with an accuracy value of 85.13%.
- f. By optimizing the selection of variables which are fewer in number compared to previous research, where the variables selected in this study are relevant variables according to the criteria in the KNN and Naïve Bayes algorithms, this research shows an increase in the performance of models created using the Feature Selection technique and also improves the accuracy of the Naïve Bayes and KNN models in previous studies, as shown in Table 3.

Table 3. Comparison of Accuracy to the previous research

Research	Accuracy	
	Naïve Bayes	KNN
This research	85.13%	86.65%
A. Yosafat, Y. Kurnia [8]	53,99%	51,26%

ACKNOWLEDGMENT

We extend our heartfelt gratitude to Universitas Multimedia Nusantara for their invaluable support, which played a pivotal role in the successful completion of this research endeavor. Their substantial contribution was instrumental in achieving our objectives, and we are deeply grateful for their unwavering assistance.

REFERENCES

- [1] D. Mursid Alfathoni, Muhammad Ali; Manesah, *Pengantar Teori Film*, Avinda Yud. Yogyakarta: Deepublish Publisher, 2020. [Online]. Available: https://books.google.co.id/books?hl=en&lr=&id=_G4PEAAQBAJ&oi=fnd&pg=PP1&dq=Film+menurut+Kamus+Besar+Bahasa+Indonesia+memiliki+arti+yaitu+lakon+atau+cerita+gambar+hidup,+dalam+Bahasa+Inggris+film+biasanya+disebut+sebagai+movie+&ots=eHTSBOneot&sig=Z_QOjdokg3xc9xfu hGFie8ip2vE&redir_esc=y#v=onepage&q&f=false
- [2] Anonymous, "Tren Positif Film Indonesia," *News article*, 2019. <https://indonesia.go.id/ragam/seni/sosial/tren-positif-film-indonesia> (accessed Apr. 03, 2023).
- [3] S. Azasya, "Home Hype Entertainment [INFOGRAFIS] Minat Penonton Terhadap Film Indonesia, Banyak Gak Sih?26 Jul 19 | 16:00 [INFOGRAFIS] Minat Penonton Terhadap Film Indonesia, Banyak Gak Sih?," *News article*, 2019. <https://www.idntimes.com/hype/entertainment/stella/infografis-minat-penonton-terhadap-film-indonesia> (accessed Apr. 03, 2023).
- [4] J. Siahaan, W. Wella, and R. Desanti, "Apakah Youtuber Indonesia Kena Bully Netizen?," *Ultim. InfoSys J. Ilmu Sist. Inf.*, vol. 11, no. 2 SE-Articles, Dec. 2020, doi: <https://doi.org/10.31937/si.v11i2.1764>.
- [5] R. A. Abarja, "Movie Rating Prediction using Convolutional Neural Network based on Historical Values," *Int. J. Emerg. Trends Eng. Res.*, vol. 8, no. 5, pp. 2156–2164, 2020, doi: [10.30534/ijeter/2020/109852020](https://doi.org/10.30534/ijeter/2020/109852020).
- [6] R. Wijayatun Pratiwi and Y. Sulisty Nugroho, "Prediksi Rating Film Menggunakan Metode Naïve Bayes," *Kinabalu*, vol. 12, no. 1, pp. 50–57, 2017.
- [7] M. H. Latif and H. Afzal, "Prediction of Movies popularity Using Machine Learning Techniques," *IJCSNS Int. J. Comput. Sci. Netw. Secur.*, vol. 16, no. 8, p. 127, 2016.
- [8] A. R. Yosafat and Y. Kurnia, "Aplikasi Prediksi Rating Film dengan Perbandingan Metode Naïve Bayes dan KNN Berbasis Website Menggunakan Framework Codeigniter," *J. ALGOR*, vol. 1, no. 1, pp. 16–26, 2019.
- [9] D. Gaikar, R. Solanki, H. Shinde, P. Phapale, and I. Pandey, "Movie Success Prediction Using Popularity Factor from Social Media," *Int. Res. J. Eng. Technol.*, vol. 6, no. 4, pp. 5185–5190, 2019, [Online]. Available: www.irjet.net
- [10] Y. Mardi, "Data Mining : Klasifikasi Menggunakan Algoritma C4.5," *Edik Inform.*, vol. 2, no. 2, pp. 213–219, 2017, doi: [10.22202/ei.2016.v2i2.1465](https://doi.org/10.22202/ei.2016.v2i2.1465).
- [11] F. F. Firdaus, H. A. Nugroho, and I. Soesanti, "A Review of Feature Selection and Classification Approaches for Heart Disease Prediction," *IJITEE (International J. Inf. Technol. Electr. Eng.)*, vol. 4, no. 3, p. 75, 2021, doi: [10.22146/ijitee.59193](https://doi.org/10.22146/ijitee.59193).
- [12] S. Moro, P. Cortez, and R. M. S. Laureano, "A Data Mining Approach for Bank Telemarketing Using the rminer Package and R Tool," no. September, pp. 1–23, 2013.
- [13] D. Feblian and D. Daihani Umar, "Implementasi model (Dina F, dkk) IMPLEMENTASI MODEL CRISP-DM UNTUK MENENTUKAN SALES PIPELINE PADA PT X," *J. Tek. Ind.*, pp. 1–12, 2016.
- [14] M. R. Fanani, "Algoritma Naïve Bayes Berbasis Forward Selection Untuk Prediksi Bimbingan Konseling Siswa," *J. DISPROTEK*, vol. 11, no. 1, pp. 13–22, 2020, doi: [10.34001/jdpt.v11i1.952](https://doi.org/10.34001/jdpt.v11i1.952).
- [15] A. Jović, K. Brkić, and N. Bogunović, "An overview of free software tools for general data mining," *2014 37th Int. Conv. Inf. Commun. Technol. Electron. Microelectron. MIPRO 2014 - Proc.*, no. May, pp. 1112–1117, 2014, doi: [10.1109/MIPRO.2014.6859735](https://doi.org/10.1109/MIPRO.2014.6859735).
- [16] R. Nithin, V. S. Babu, P. M., and L. A., "Predicting Movie Success Based On Imdb Data," vol. 5, no. X, p. 504, 2017, doi: [http://doi.org/10.22214/ijraset.2017.10074](https://doi.org/10.22214/ijraset.2017.10074).
- [17] R. N. Devita, H. W. Herwanto, and A. P. Wibawa, "Perbandingan Kinerja Metode Naïve Bayes dan K-Nearest Neighbor untuk Klasifikasi Artikel Berbahasa Indonesia," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 4, p. 427, 2018, doi: [10.25126/jtiik.201854773](https://doi.org/10.25126/jtiik.201854773).
- [18] C. F. Mohd Foozy, R. Ahmad, M. A. Faizal Abdollah, and C. C. Wen, "A Comparative Study with RapidMiner and WEKA Tools over some Classification Techniques for SMS Spam," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 226, no. 1, 2017, doi: [10.1088/1757-899X/226/1/012100](https://doi.org/10.1088/1757-899X/226/1/012100).

Breast Cancer Prediction of Benign and Malignant Tumors by Classification Algorithms

Santad Promtan

College of Digital Science
Prince of Songkla University
Songkhla, Thailand
6510035007@psu.ac.th

Phungern Khongthong

Medical Oncologist
Hatyai Namom Cancer Centre
Hatyai Hospital
Songkhla, Thailand
p.khongthong6@gmail.com

Chidchanok Choksuchat

Division of Computational Science
Faculty of science,
Prince of Songkla University
Songkhla, Thailand
chidchanok.ch@psu.ac.th

Abstract— Breast cancer is a disease that affects both males and females, but it is more commonly found in females. It is considered the most common form of cancer among women worldwide. Unfortunately, breast cancer can be fatal and is the most invasive cancer in women globally. The absence of symptoms in the early stages of the disease often leads to a delayed diagnosis, rendering medical treatment ineffective. However, early screenings can reduce the mortality rate associated with breast cancer. In addition to traditional medical diagnosis methods, machine learning techniques can be utilized to predict breast cancer risk using health data. In this study, we attempted to compare various machine learning classification algorithms, such as the Decision Tree Classifier, Logistic Regression, KNeighbors Classifier, and Naïve Bayes Classifier. The results indicate that the Naïve Bayes Classifier outperformed the other algorithms in terms of performance. Experimental results show that the Naïve Bayes Classifier yields the highest accuracy (93%). Future research will explore fuzzy logic, rule-based methods for accurate breast cancer diagnosis, and expert validation of the model.

Keywords—Machine Learning, Classification Algorithms, Breast Cancer, Decision Tree Classifier, Logistics Regression, KNeighbors Classifier, Naïve Bayes Classifier

I. INTRODUCTION

Cancer is a disease caused by abnormalities in the cells of the body's organs, resulting in genetic alterations. These changes cause abnormal growth that is diffuse and difficult to control, leading to the potential for the disease to spread rapidly to other organs, including the lungs, liver, brain, bones, spinal cord, lymph nodes, heart, colon, breast, and others. The causes and risk factors of cancer vary depending on factors such as diet, lifestyle, and certain behaviors, as well as factors that are beyond people's control, such as family history, age range, hormone levels, and inherited traits.

Breast cancer is a type of cancer that can occur in both males and females, but it is the most common cancer diagnosed in women. It is considered the most common type of cancer in women who attend breast clinics. Abnormal division of the mammary glands can cause the cancer to spread to the lymphatic area in the armpit and other organs, potentially leading to death. Risk factors for breast cancer include a family history of breast and ovarian cancer, as well as lifestyle and diet choices.

Nowadays, there are several methods to detect and diagnose breast cancer, including medical diagnosis, mammogram (an x-ray of the breast), breast ultrasound (a check-up using high-frequency sound waves), biopsy (a procedure to obtain a sample of breast cells for testing), and MRI (breast magnetic resonance imaging). The diagnosis of breast cancer involves determining the disease or condition

that explains a person's symptoms and signs. There are two types of tumors that can be classified as malignant or benign. Benign tumors are noncancerous and do not spread, while malignant tumors are cancerous and can spread to other cells in the body.

Breast cancer is a serious health concern, as statistics from 2020 released by the World Health Organization (WHO) show that 2.3 million women were diagnosed with breast cancer, and 685,000 died from the disease worldwide. Although early screenings can help decrease the mortality rate, breast cancer is still a dreadful and worrisome disease due to the absence of symptoms in the early stages, resulting in delayed diagnosis and making medical treatment less effective. Therefore, it is crucial to raise awareness and take cancer seriously.

In this study, our aim is to explore the Breast Cancer Wisconsin Data Set, which is a dataset relevant to breast cancer screening using Fine Needle Aspiration (FNA) cytology of breast masses to describe the results of a biopsy performed by a medical laboratory. We focus on applying various classification algorithms in machine learning to predict the presence of benign or malignant tumors in breast cancer.

This paper includes sections such as Related Work, Classification Algorithms, Methodology and Results, Discussion, and Conclusion following the Introduction.

II. RELATED WORK

In previous research, various techniques have been proposed for predicting breast cancer, which poses a big challenge in using big data to analyze it. Classification algorithms in machine learning are the most important techniques in data analysis. They are the most popular and widely used algorithms among researchers. Machine learning models applied to medical data analysis have been surveyed by researchers.

In a study conducted by Hiba Asri, Hajar Mousannif, Hassan Al Moatassime, and Thomas Noel [2], the performance of machine learning models, such as Support Vector Machine (SVM), Decision Tree (DT), Naive Bayes (NB), and k Nearest Neighbors (k-NN), was compared on the Wisconsin Breast Cancer dataset to assess the correctness in classifying data with respect to the efficiency and effectiveness of each algorithm in terms of accuracy, precision, sensitivity, and specificity. The results showed that Support Vector Machine (SVM) gave the best accuracy score (97.13%) with the lowest error score.

In another study by Vikas Chaurasia and Saurabh Pal [3], RepTree, RBF Network, and Simple Logistic were used for the diagnosis system to detect breast cancer. The results showed that Simple Logistic gave the best accuracy score (74.47%) and a total modeling time of 0.62 seconds.

In previous studies, various techniques have been proposed for predicting breast cancer, which presents a significant challenge in big data analysis. Among these techniques, classification algorithms in machine learning are crucial for data analysis and are widely used and popular among researchers.

In [4], Mengfan Li proposes a computer-based feature fusion Convolutional Neural Network breast cancer image classification and detection method. The researcher pre-trains two Convolutional Neural Networks with different structures and then uses them to automatically extract features. The features extracted from the two structures are fused, and finally, a classifier is used to classify the fused features. The results showed that Convolutional Neural Network (CNN) achieved an accuracy score of 89%.

In [5], Abdoulaye Bah and Muhammed Davud used various machine-learning techniques for breast cancer classification, including Random Forest, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Convolutional Neural Networks (CNN). The results showed that CNN obtained a higher accuracy score than the other techniques.

In [6], Muhammet Fatih Ak used machine learning techniques for data visualization and to predict breast cancer tumors. The modeling techniques used were Logistic Regression, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Naïve Bayes (NB), Decision Tree (DT), Random Forest (RF), and Rotation Forest (RF). The results showed that Logistic Regression had the best accuracy score of 98.1%.

In [7], Shweta S. Kaddi and Malini M. Patil constructed regression models using the NKI breast cancer dataset. The regression methods used were linear, multiple, and polynomial. The results of the prediction model helped analyze the behavior of the features of breast cancer datasets and provided useful insights in diagnosing patients. The results showed that multiple regression performed better with an accuracy score of 83%.

In [8], Heena Nankani, Shruti Gupta, Shubham Singh, and S. S. Subashka Ramesh compared the performance of machine learning models to detect and analyze each type of cancer. They used machine learning techniques such as K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Logistic Regression (LR), and Linear Regression (LR) to create models for data analysis. The results showed that Logistic Regression (LR) was the best model with a Test log loss of 0.9257.

The above-mentioned research compares various classification algorithms in machine learning for predicting breast cancer. Different machine learning and data analysis methods, including Support Vector Machine, Decision Tree, Naive Bayes, k Nearest Neighbors, RepTree, RBF Network, Regression, Convolutional Neural Network, Random Forest, and Rotation Forest, are employed. The accuracy of these classification algorithms is compared by Muhammet Fatih Ak [6], with Logistic Regression achieving the highest accuracy score of 98.1%.

In this study, we attempted to compare various classification algorithms of machine learning, such as the Decision Tree Classifier, Logistic Regression, KNeighbors Classifier, and Naïve Bayes Classifier. The goal was to achieve the highest accuracy and the lowest error rate.

III. CLASSIFICATION ALGORITHMS

In the field of machine learning, numerous classification techniques are available for efficient data classification in medical applications, including Naïve Bayes, Logistic Regression, K-Nearest Neighbors, Decision Tree, Random Forest, Support Vector Machine, and others. In this paper, we focus on presenting four specific classification techniques, which are as follows:

A. Decision Tree Classifier

Decision trees are one of the most popular supervised learning algorithms in the field of machine learning. This algorithm uses a tree structure that resembles a flowchart to illustrate the different decisions and rules used for classifying data [9]. It is a model capable of learning attributes of data and creating a decision tree that can be applied to classification to indicate which data belongs to which group. These groups may include two or more categories, such as high risk, moderate risk, low risk, or normal/abnormal.

In a decision tree, internal nodes represent attributes or features, while the branches represent the decision rules, and each leaf node represents the final decision outcome. The construction of decision tree rules depends on the selection of important features that have a significant influence on the decisions related to the data in the analysis [10].

B. Logistics Regression

Logistic regression analysis is a multivariate analysis technique aimed at estimating or predicting events that will or will not occur under a given factor. The logistic model comprises two variables: independent variables, which are causal variables expected to influence dependent variables, and dependent variables, which are variables whose values are opposite to those of independent variables, and their values depend on the independent variable [11-12]. Logistic regression analysis consists of two types:

- Binary Logistic Regression: This is a regression model used in biological logistics analysis where the target variable or the dependent variable has only two binary values: occurrence ($y = 1$) and non-occurrence ($y = 0$). This model is popular among data analysts.

- Multinomial Logistic Regression: This is a multinomial logistics analysis where the dependent variable has more than two values, for example, good ($y = 3$), medium ($y = 2$), and low ($y = 1$).

C. KNeighbors Classifier

The KNeighbors Classifier is a versatile algorithm that can work on both classification and regression data. It operates by finding new data points and classifying them by looking at neighboring points. When information is found near any point, the data results are grouped together, and the value of k is used to indicate the desired number of groups or points [13-14].

The KNeighbors Classifier is optimized for classification data and relies on the similarity between new and old information to classify the new data into the category that is most similar to the existing category. However, its

effectiveness in analysis depends on the quality and nature of the data used in the analysis.

D. Naïve Bayes Classifier

The Naïve Bayes Classifier is a supervised learning algorithm used for classification models. This model is employed to classify or predict upcoming events based on probability by applying Bayes (Bayesian) theory to analyze and forecast incoming data [15].

The model's objective is similar to the Logistic Regression model, as it predicts the occurrence or non-occurrence of events. It can also increase the likelihood of an event occurring. This model is often used to analyze dependent event data, such as the risk of disease, which requires statistical calculations to predict such events.

IV. METHODOLOGY

In this work, we propose a process for building a classification model using the Breast Cancer Wisconsin (Diagnostic) Data Set, as depicted in Figure 1.

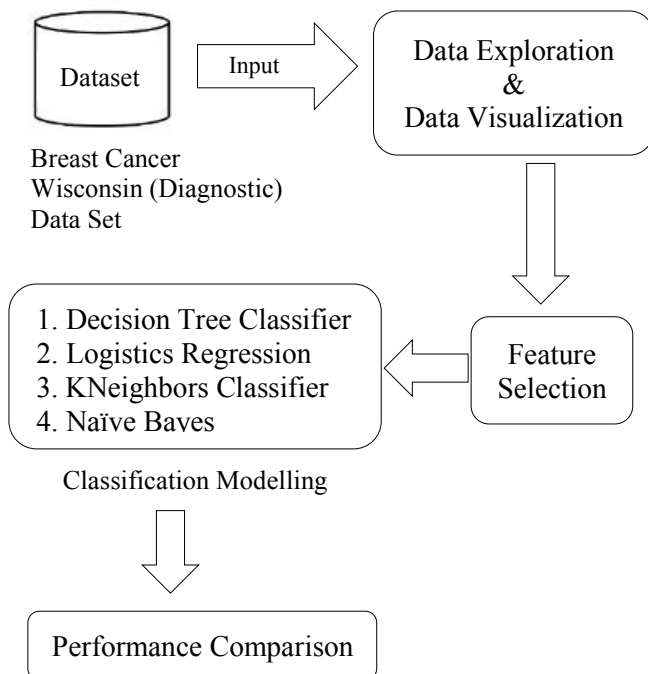


Fig. 1. Process of building the classification model

A. Dataset

Data can be collected from two sources: primary and secondary sources. In this work, the data collection method from the secondary source was used, specifically from the Kaggle website, utilizing the Breast Cancer Wisconsin (Diagnostic) Data Set. This dataset is readily available and reusable, and it was used for data analysis and modeling to predict the risk of breast cancer. The attributes were calculated from digital images of Fine Needle Puncture (FNA) breast masses, which characterize the cell nuclei present in the image. Some examples of the attributes in the dataset are as follows:

- radius_mean
- texture_mean
- perimeter_mean

- area_mean
- smoothness_mean
- concave_points
- compactness
- concavity_mean
- diagnosis

B. Data Exploration

The Breast Cancer Wisconsin (Diagnostic) Data Set consists of 29 continuous variables, 1 categorical variable (M = malignant, B = benign), and 569 rows, as depicted in Table I.

TABLE I. EXAMPLE DATA OF BREAST CANCER WISCONSIN (DIAGNOSTIC) DATA SET

radius_mean	texture_mean	perimeter_mean	area_mean	diagnosis
17.99	1038	122.80	1001.0	M
20.57	17.77	132.90	1326.0	M
19.69	21.25	130.00	1203.0	M
13.08	15.71	85.63	520	B
13.54	14.36	87.46	566.3	B

C. Data Visualization

The aim of data visualization is to provide a summary of information that is easy to understand and to show patterns or structures in the dataset, allowing us to view the data clearly and with ease. Examples of data visualization are depicted in Figures 2 and 3.

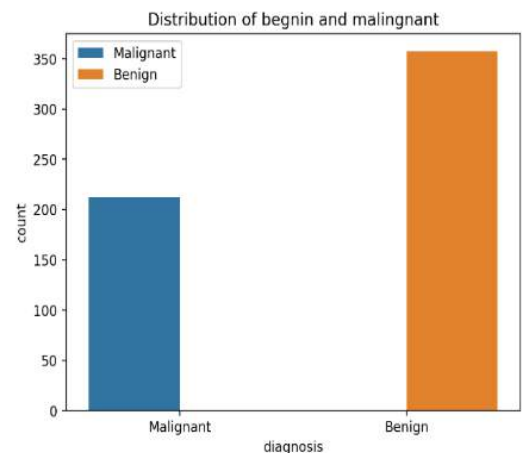


Fig. 2. Plot bar chart distribution of benign and malignant

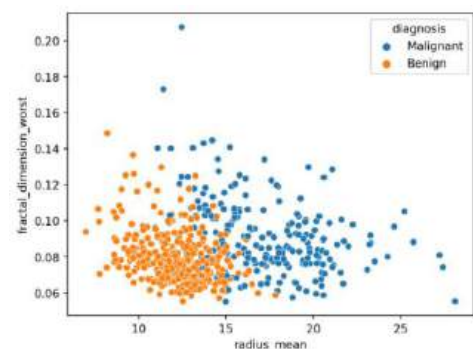


Fig. 3. Scatter plots between diagnosis and fractal_dimension_worst

D. Identifying Variables

You have explained the concepts of independent and dependent variables correctly.

In this dataset, the independent variables are radius_mean, texture_mean, perimeter_mean, area_mean, and so on. These variables are being studied and their values are independent of other variables.

The dependent variable in this dataset is the diagnosis. It is the variable that is affected by the independent variables, and its value depends on the values of the independent variables.

E. Data Preprocessing

Data preprocessing is the process of preparing raw data to be transformed into data that is ready for analysis and for creating a model for data analysis.

Missing values: Missing values refer to some data that is absent in a dataset. If not managed properly, this can have serious consequences for data analysis. In this dataset, no missing values were found.

Duplicate values: De-duplication of data is an important step in data preprocessing. In this process, duplicates should be eliminated from the dataset to ensure accurate data analysis. In this dataset, no duplicate values were found.

Outliers: An outlier is a value that lies at an abnormal distance from other values in a dataset. In the graph below, we can clearly see that values above 25 in the radius_worst column and above 150 in the perimeter_worst column is acting as outliers.

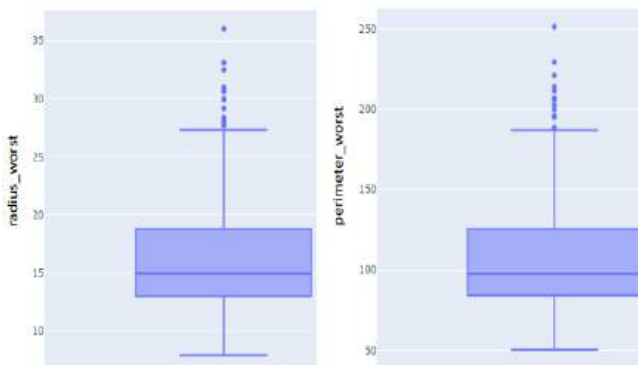


Fig. 4. Outliers present in the radius_worst and perimeter_worst column

F. Feature Selection

To train a machine learning model, it is necessary to collect a large quantity of data. Usually, datasets have a large number of features. However, not all of these features may be significant in building the model. Some features may not contribute much to the training process and may even lead to erroneous data analysis. Therefore, it is crucial to select important features in the dataset that affect data analysis and model building. The process of reducing the number of input variables is essential in this regard. However, selecting an appropriate statistical technique for feature selection can be challenging for researchers. To address this issue, various feature selection techniques have been proposed in the literature,

including Support Vector Machine [16], Random Forest [17], Lasso [18], and Decision Tree [19].

In this work, we used the Random Forest Classifier for feature selection by finding the importance scores of all features. The feature importance scores are depicted in Figure 5.

In Figure 5, the feature selection process using the Random Forest Classifier model is demonstrated to select only the important features, build the model using these features, and measure the accuracy of a classification model. The features were selected by considering scores greater than 0.1 ($>.1$).

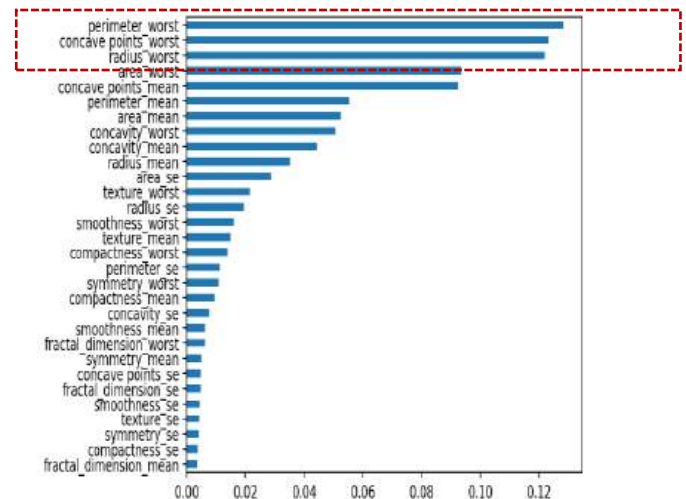


Fig. 5. Visualize feature scores of dataset

Three features were selected, namely: radius_worst (important score = 0.122073), concave_points_worst (important score = 0.123387), and perimeter_worst (important score = 0.128214).

G. Split the data

The intelligence of machine learning depends on the size of the dataset used for training. However, even with large datasets, the learning efficiency of machine learning algorithms may not be optimal.

To address this issue, we use a process called data splitting, which involves separating the data into two sets: the training set and the test set.

The training set is used to train the machine learning model, while the test set is used to evaluate the performance of the model after training. The proportion of data allocated to each set may vary depending on the dataset and its characteristics.

In this work, we split the dataset into a training set containing approximately 75% of the rows and a test set containing the remaining 25%.

H. Classification Modelling

To develop a model, the Breast Cancer Wisconsin (Diagnostic) Data Set was split into a 75:25 ratio to be used as training and test sets. The classification algorithms used to create the model were the Decision Tree Classifier, Logistic Regression, KNeighbors Classifier, and Naïve Bayes Classifier. In this dataset, the independent variables (features) are radius_worst, concave_points_worst, and

perimeter_worst. The dependent variable (label) is the diagnosis column.

I. Experiment Results

In this section, after developing a classification algorithm model, we evaluate the performance of the algorithms using statistical methods. The methods used in the analysis include the Confusion Matrix, Accuracy, Precision, Recall, and F1 Score, which are explained below.

Confusion matrix:

When performing predictions on a classification model, the results are classified into four categories: True positives (TP): These are instances where the model correctly predicted a positive answer. True negatives (TN): These are instances where the model correctly predicted a negative answer. False positives (FP): These are instances where the model predicted a positive answer, but it was incorrect. False negatives (FN): These are instances where the model predicted a negative answer, but it was incorrect.

The example of a confusion matrix for the Naïve Bayes Classifier model is depicted in Figure 6.

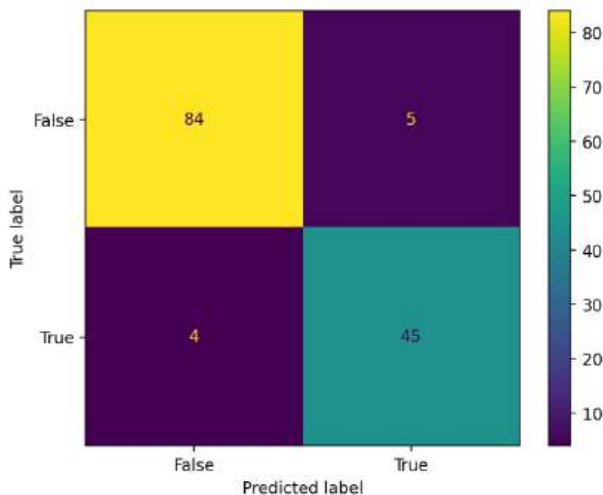


Fig. 6. The confusion matrix for Naïve Bayes Classifier

In the above figure, there are a total of 552 samples after data preprocessing, and the dataset was split into a test set containing approximately 25% (138 rows) of the total data. The diagonal elements represent the correctly predicted samples. Out of the total 138 samples in the test set, 129 samples were correctly predicted, resulting in an overall accuracy of 75.92%. On the contrary, 9 samples were wrongly predicted out of the total 138 samples.

Accuracy: The percentage of correct predictions for model. Accuracy is define by as:

$$\text{Accuracy scores} = \frac{\text{Correct Predictions}}{\text{All Predictions}}$$

Precision: Compare true positives and false positives. Precision is define by as:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

Recall: Compare true positives and false negatives. Recall is define by as:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

F1 Score: Average Precision and Recall. F1 Score is define by as:

$$\text{F1 score} = \frac{2 * TP}{2 * TP + FP + FN} = 2 * \frac{\text{Precision} * \text{recall}}{\text{Precision} + \text{recall}}$$

The model evaluation results for the classification algorithms are presented in Table II.

TABLE II. MODEL EVALUATION RESULT

Model	Accuracy	Precision	Recall	F1 Score
Decision Tree Classifier	0.92	0.93	0.83	0.93
Logistics Regression	0.92	0.93	0.85	0.93
KNeighbors Classifier	0.91	0.89	0.85	0.89
Naïve Bayes Classifier	0.93	0.90	0.91	0.90

Table II presents the results of the model evaluation for classification algorithms. It shows the best and worst performance on the evaluation data based on Accuracy, Precision, Recall, and F1 Score. The comparison of model performance across different evaluation metrics leads to the conclusion that the Naïve Bayes Classifier is the best accuracy model, with an accuracy score of 0.93 (93%), a precision score of 0.90 (90%), a recall score of 0.91 (91%), and an F1 score of 0.90 (90%). On the contrary, the KNeighbors Classifier is the worst accuracy model, with an accuracy score of 0.91 (91%), a precision score of 0.89 (89%), a recall score of 0.85 (85%), and an F1 score of 0.89 (89%). The table presents the results for all the models evaluated.

V. DISCUSSION

Notice from Table II that:

(1) the Decision Tree Classifier has obtained an accuracy score of 0.92, a precision score of 0.93, a recall score of 0.83, and an F1 score of 0.93;

(2) the Logistic Regression has obtained an accuracy score of 0.92, a precision score of 0.93, a recall score of 0.85, and an F1 score of 0.93;

(3) the KNeighbors Classifier has obtained an accuracy score of 0.91, a precision score of 0.89, a recall score of 0.85, and an F1 score of 0.89;

(4) the Naïve Bayes Classifier has obtained an accuracy score of 0.93, a precision score of 0.90, a recall score of 0.91, and an F1 score of 0.90.

VI. CONCLUSION

Breast cancer is a deadly disease that has claimed countless lives. Currently, machine learning has a considerable impact on medicine, and it can be used for breast cancer diagnosis using various techniques of machine learning and data analysis methods. In this work, models have been developed to accurately predict breast cancer. All models were evaluated for performance using statistical methods, including Confusion Matrix, Accuracy, Precision, Recall, and F1 Score.

The best model for predicting breast cancer is the Naïve Bayes Classifier, which achieved an accuracy score of 0.93 (93%). In this study, we used the Naïve Bayes Classifier model, which can be applied at the Medical Oncologist at Hatyai-Namom Cancer Centre, Songkhla, by combining the model with the development of an automated program to analyze real-time diagnosis data. This automated program can help provide initial diagnosis data support to medical oncologists and breast care nurses, facilitating their operations.

In the future, interesting machine learning techniques such as fuzzy logic and rule-based methods will be further applied to make the diagnosis of breast cancer more accurate and effective, providing recommendations for breast cancer treatment. Additionally, the model will be validated by experts in breast cancer treatment.

REFERENCES

- [1] World Health Organization (WHO). (2021, March 26). Breast cancer. [Online]. Available: <https://www.who.int/news-room/fact-sheets/detail/breast-cancer>.
- [2] H. Asri and T. Noel, "Using Machine Learning Algorithms for Breast Cancer Risk Prediction and Diagnosis," *The 6th International Symposium on Frontiers in Ambient and Mobile Systems*, vol. 83, pp. 1064-1069, May 2016, doi: 10.1016/j.procs.2016.04.224.
- [3] V. Chaurasia and S. Pal, "Data Mining Techniques: To Predict and Resolve Breast Cancer Survivability," *International Journal of Computer Science and Mobile Computing*, vol. 3, no. 1, pp. 10-22, January 2014.
- [4] M. Li, "Research on the Detection Method of Breast Cancer Deep Convolutional Neural Network Based on Computer Aid," in *IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC)*, 2021, pp. 536-540.
- [5] A. Bah and M. Davud, "Analysis of Breast Cancer Classification with Machine Learning based Algorithms," in *International Conference on Computing and Machine Intelligence (ICMI)*, 2023.
- [6] Muhammet Fatih Ak, "A Comparative Analysis of Breast Cancer Detection and Diagnosis Using Data Visualization and Machine Learning Applications," *Healthcare*, vol. 8, no. 16, pp. 1-23, March 2020, doi: 10.3390/healthcare8020111.
- [7] S. S. Kaddi and M. M. Patil, "Forecasting the survival rate of breast cancer patients using a supervised learning method," in *Global Transitions Proceedings*, 2022, pp. 25-30.
- [8] H. Nankani, S. Gupta, S. Singh, S. S. Subashka Ramesh, "Detection Analysis of Various Types of Cancer by Logistic Regression using Machine Learning," *International Journal of Engineering and Advanced Technology*, vol. 9, no. 1, pp. 99-104, October 2019, doi: 10.35940/ijeat.A1055.109119.
- [9] S. Mitrofanov and E. Semenkin, "An Approach to Training Decision Trees with the Relearning of Nodes," in *Proceedings of the 2021 International Conference on Information Technologies (InfoTech-2021), IEEE Conference*, 2021.
- [10] W. Yanxia, "Student Information Management Decision System Based on Decision Tree Classification Algorithm," in *International Conference on Information Systems and Computer Aided Education (ICISCAE)*, 2021, pp. 827-831.
- [11] V. N. Vasu, S. R. S. M.S and M. N., "Prediction of Defective Products Using Logistic Regression Algorithm against Linear Regression Algorithm for Better Accuracy," in *International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, 2022, pp. 161-166.
- [12] A. Yamini and K. Sashi Rekha, "Improved Accuracy for Identifying At-Risk Students at Different Percentage of Course Length using Logistic Regression Compared with Random Forest Predictive Model," in *International Conference on Contemporary Computing and Informatics (IC3I)*, 2022, pp. 1869-1873.
- [13] A. Bansal and A. Singhrova, "Performance Analysis of Supervised Machine Learning Algorithms for Diabetes and Breast Cancer Dataset," in *Proceedings of the International Conference on Artificial Intelligence and Smart Systems (ICAIS-2021)*, 2021, pp. 137-143.
- [14] M. Ati, M. Ikram Alib and M. Usman Ghani Khan, "An Ensemble Learning to Predict Glucose Level of Patient by Urine Sample," in *International Conference on Electrical and Computing Technologies and Applications (ICECTA)*, 2022, pp. 366-369.
- [15] D. P. He, Z. L. He* and C. Liu, "Recommendation Algorithm Combining Tag Data and Naive Bayes Classification," in *International Conference on Electron Device and Mechanical Engineering (ICEDME)*, 2020, pp. 662-666.
- [16] Y. Bouchlaghem, Y. Akhiat and S. Amjad, "Feature Selection: A Review and Comparative Study," in *E3S Web of Conferences*, 2022, pp. 1-6.
- [17] R. Chen, C. Dewi, S. Huang and R. E. Caraka, "Selecting critical features for data classification based on machine learning methods," *Journal of Big Data*, vol. 52, no. 7, pp. 1-26, July 2020, doi: 10.1186/s40537-020-00327-4.
- [18] N. Pudjihartono, T. Fadason, A. W. Kempa-Liehr and J. M. O'Sullivan, "A Review of Feature Selection Methods for Machine Learning-Based Disease Risk Prediction," *Frontiers in Bioinformatics*, vol. 2, pp. 1-17, June 2022, doi: 10.3389/fbinf.2022.927312.
- [19] A. A. Altae, A. E. Rad, R. Tati, "Comparative Study on Effective Feature Selection Methods," *International Journal for Innovative Engineering & Management Research*, Forthcoming, vol. 12, no. 2, pp. 290-293, Feb 2023.

An Integrated Approach for Sentiment Analysis and Topic Modeling of a Digital Bank in Indonesia using Naïve Bayes and Latent Dirichlet Allocation Algorithms on Social Media Data

Johan Setiawan

Information Systems Study
Universitas Multimedia Nusantara
Tangerang, Indonesia
johan@umn.ac.id

Anastasia Milenia

Information Systems Study
Universitas Multimedia Nusantara
Tangerang, Indonesia
anastasia.milenia@gmail.com

Ahmad Faza

Information Systems Study
Universitas Multimedia Nusantara
Tangerang, Indonesia
ahmad.faza@umn.ac.id

Abstract— Social media provides a public platform for expressing complaints and opinions. Researchers can use text mining techniques such as sentiment analysis and topic modeling on social media data to compare features and gauge public opinion on competing digital banks in Indonesia. The aim of this study is to classify sentiments and identify topics in social media data related to a specific digital bank. To accomplish this, the Naïve Bayes algorithm is used for sentiment analysis, while Latent Dirichlet Allocation is used for topic modeling. The social media data is sourced from Twitter and Instagram for Line Bank digital bank. The study finds that the Naïve Bayes algorithm performs well in classifying sentiments, achieving a maximum F1 score of 0.863. Positive sentiments are more prevalent in Twitter data, while negative sentiments are more prevalent on Instagram. Topic modeling using Latent Dirichlet Allocation algorithm identifies four optimal topics for positive sentiment and five for negative sentiment. The coherence value obtained is 0.426279 for positive sentiment and 0.397232 for negative sentiment.

Keywords— Digital Bank, Latent Dirichlet Allocation (LDA), Naïve Bayes, Sentiment Analysis, Topic Modelling

I. INTRODUCTION

Over the past few years, there has been a noticeable shift from traditional banking transactions, such as those conducted through ATMs and branch offices, towards the utilization of mobile and internet banking services. Data from the Institute for Development Economy and Finance (INDEF) indicates a consistent decline in the frequency of ATM or debit transactions, from 62% in 2011 to 37% in 2018. Similarly, transactions at branch offices decreased from 17% in 2011 to 4% in 2018, while the frequency of mobile banking as an internet-based service increased from 6% in 2011 to 41% in 2018. Additionally, other internet-based services, such as internet banking, saw an increase from 11% in 2011 to 17% in 2018[1].

One of the internet-based banking services in Indonesia is digital banking. According to a survey conducted by Inventure Indonesia and Alvira Research Center, 43.6% of respondents used internet-based banking services more frequently after the COVID-19 pandemic compared to before[2]. Conversely, only 25.6% of respondents reported an increase in the use of e-wallet electronic payment instruments since the pandemic[2].

The growing number of Indonesians using internet-based banking services has led to an increase in the number of digital banks. Momentum Works reports that over the past five years, at least one digital bank product per year has started operating in Indonesia[3]. One such example is Line Bank, an application owned by KEB Hana Bank Indonesia and Line, a messaging application from Japan[4]. However, the rise in the number of digital banks has also been accompanied by an increase in consumer complaints, with financial services being the most frequently reported category of complaints over the past five years, according to data from the Consumers Association from Indonesia (YLKI)[5].

One of the complaints about financial services received by YLKI complaints related to financial services, is banking [6]. According to the Financial Services Authority (OJK) social media is a source of information that can be obtained quickly to find responses from the public, such as service complaints[7]. Based on this, sentiment analysis and topic modeling of social media data, particularly Twitter and Instagram, are needed, to better understand user comments and their implications. Indonesia's most widely used social media platforms, according to the 2020 report from We Are Social and HootSuite, respectively, are Youtube, WhatsApp, Facebook, Instagram, and Twitter[8]. As of July 2020, Indonesia is ranked sixth in terms of the number of Twitter users[9] and fourth in terms of Instagram users[10], according to data from Statista.

II. LITERATURE REVIEW

A. Sentiment Analysis

Sentiment analysis or opinion mining is a more profound analysis stage from text mining, which helps find opinions or tendencies of things in the text unit, such as positive, negative, or neutral emotions [11]. Sentiment analysis is one of the implementations of a popular classification method because it can find out a person's views, opinions, reactions, and emotions only based on the tone of speech from the written text [12]. The Naïve Bayes algorithm was chosen to perform the automatic sentiment classification from social media because it has proven to positively affect sentiment classification [13]–[16]. Some of these advantages include attributes that are not limited to numeric, speed in modeling until the deployment stage [17], modeling with a small amount

of training data, and better performance with fewer input dimensions [18], assuming all attributes are independent [17].

B. Topic Modelling

Topic modeling can summarize large text documents into a group of topics. The topic in question is a collection of words that often appear together in a document. Each word has its weight in a topic [19]. The process of getting topics from several documents is called topic modeling, based on the unsupervised concept [12]. The topic modeling task uses the Latent Dirichlet Allocation algorithm because it is proven to have advantages [20]–[22], such as the ability to process short, long, and mixed-length documents as well as being able to consider the form of relationships between documents [21].

C. Related Works

Several previous studies conducted sentiment analysis [23] and topic modeling [24] on digital banks and E-Commerce. Cheng and Sharmayne [23] predict sentiment classification and topic modeling based on review ratings of digital banks in the Philippines. Sentiment classification prediction is made without a special algorithm, while topic modeling uses LDA and association methods. This study resulted in seven main topics from application review data, keywords associated with each positive and negative label, and conclusions about the advantages and disadvantages of banks [23]. Yang [24] combines sentiment analysis and modeling by using LDA for topic modeling and a Python library called vader for sentiment analysis. The study compared five brands' product review data from e-commerce about healthy snacks ("new snacks"). The results of topic modeling almost all produce positive words for each store, then used for strategy recommendations for each store owner.

III. METHODOLOGIES

A. Data Collection

The source of research data is public opinion about one of the digital banks in Indonesia, which is published through social media. The social media data used are tweet data and comment data on official Instagram account content from Line Bank, a digital bank. The social media owned by Line Bank, which operates in Indonesia, are @linebankid for Instagram, Twitter, and Tiktok. While "Line Bank by Hana Bank" is the official account on Facebook and Youtube. The data is collected by scraping from social media using Python in Jupyter. The Python library used for scraping tweet data is Tweepy to access the Twitter API [25]. The Python Instagram-Comments- Scraper library collected comment data from Instagram [26]. Twitter and Instagram social media data took over three months, from November 6, 2021, to March 19, 2022. The period is determined based on the release date of the digital bank Line. Based on the provisions of the Twitter API, the period for scraping is also given a limit of seven days before the scraping date so that Twitter data will be retrieved periodically every week. The data collected within one week will then be examined by three students who have proficiency in Bahasa, whose results will determine the labelling of sentiment based on the majority vote results.

B. Research Design

The step of sentiment analysis research will follow a general procedure with some additional steps for modelling the topic of the digital bank. The step of sentiment analysis refers to the research conducted by Pierre [27] with

modifications made. Modifications were carried out at the preprocessing stage, where previous studies used only one social media, while in this study, two social media were used. The following change is that the research step does not stop at the sentiment classification stage but continues with the topic modelling task. The stages of the topic modelling task use the reference of the research framework of Asmussen and Møller [28] with modification in the clean document step.

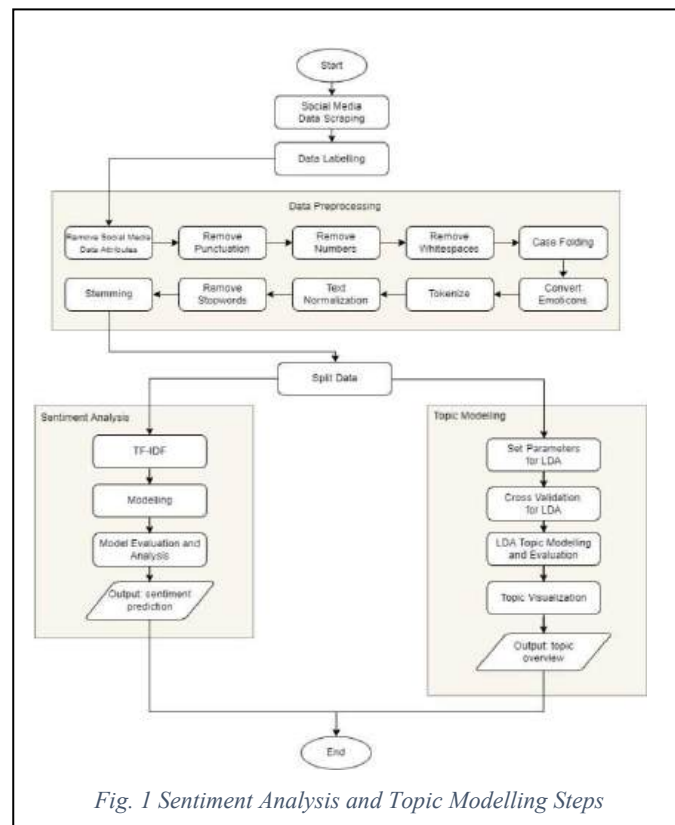


Fig. 1 Sentiment Analysis and Topic Modelling Steps

These stages are combined with the preprocessing step when conducting sentiment analysis for efficiency. Fig. 1. shows the results of improvements for this study.

IV. ANALYSIS AND RESULTS

A. Data Scrapping

Twitter data retrieval is done using the Tweepy library. Tweet data retrieval is carried out every week according to the limits of the Twitter API, with the keyword 'line bank'. The results of scraping tweet data are then stored in a spreadsheet file. The second social media data, Instagram comments data, were obtained using the Instagram-Comments-Scraper library. The results of scraping the comment data obtained are in the form of a spreadsheet file for each Instagram post. The data is combined into a single file consisting of username data and comment content. Based on the scraping process of the two social media, there were 4257 Twitter tweets and 1930 Instagram comments.

B. Data Labelling

Results of scraping Twitter data and subsequent Instagram comments manually labelled sentiment by three students who are digital bank users and have proficiency in Bahasa. Labelling is divided into data with positive sentiment (expressed by 'P') or negative (expressed by 'N'). Each data will then be determined as the final label based on

the most labels obtained. Tweets with positive labels were collected as many as 3894 and negative labels as many as 363. In contrast, positive comments from Instagram were obtained, as many as 276 comments and 1654 negative comments.

C. Data Attributes Punctuation Numbers and Whitespaces Removal

The first stage of data processing is to delete data attributes often found in social media data, such as links, hashtags, usernames, mentions, and the description of 'RT', which represents retweets. The deletion is done with the `re.sub()` library, which replaces word patterns based on regex with empty strings. Existing punctuation marks other than those contained in the unique attributes of social media data will then be deleted based on the list of punctuation marks in the `string.punctuation`. The script works the same way as the previous process, which replaces punctuation marks with empty strings. The process of deleting numbers is done by using the script `"\d+"` to find one or more digits which are then deleted.

D. Case Folding

Case folding is changing letters to non-capital to prevent calculating the same word but having a different capitalization [29]. The data resulting from the next step is then capitalized to be non-capital using the `str.lower()` function.

E. Convert Emoji

Changing the emoji is done by using an `emot` library that will convert the emoji into keywords that describe the emoji. Research only divides emoji pool changes into 'sad' and 'happy' words [29].

F. Tokenize

The tokenization stage is a step to break the existing sentences into smaller units. In the case of social media data, it will be broken down into word-for-word [30] using the `string` library's `split()` function.

G. Text Normalization

The normalization process works by mapping some words, such as informal words or abbreviations (slang), to become standardized formal word forms [31]. This study uses a dictionary according to the KBBI standard, which is then mapped according to the dictionary.

H. Remove Stop Words

Stop words are a collection of words that are very often found in a sentence and can make other more relevant words uncountable [29]. Some words included in stop words will be deleted using the `nlTK` library in Bahasa to get more relevant data.

I. Stemming

Each existing word will be changed to its basic word based on conformity with the Indonesian language rules [29]. The word then changes using the `StemmerFactory` function from the `litrery` library.

J. Split Data

Before starting the modelling process, all data that has gone through the preprocessing stage is separated into several parts. In the first step, data from each social media

will be separated by 80%. Furthermore, the two data from social media will be combined into training and validation data. The second data split process will take 20% of each social media and then use it as testing data to test the model on the new data. The training and validation data collected was 3406 on Twitter and 1544 on Instagram. The first data split process also generates 20% of data for model testing on each social media, which are 851 on Twitter and 386 on Instagram. The second data split process divides the combined results of the social media data from the previous process by dividing 80% for training data and 20% for validation data. The split process uses the `train_test_split` function. The data training collected was 3960, and 990 data were collected for validation. The data separation is divided based on negative and positive sentiments for the topic modelling task. The combined data comes from the combined Twitter and Instagram data. There are 4170 data with positive sentiment and 2017 data with negative sentiment.

K. Create TF-IDF (Term Frequency-Inverse Document Frequency)

The Term Frequency-Inverse Document Frequency (TF-IDF) is a word weighting method from the dataset. TF-IDF process is required to convert text data into a word matrix and its weight so that the algorithm can process it [32]. Words that appear less frequently will get a higher score and are considered more relevant, and words that occur too often will be reduced because they are considered less relevant [33]. TF-IDF is divided into two processes: the calculation of the frequency of words in a document divided by the number of words in the document (TF). The IDF stage is when words are given weights based on the logarithmic calculation of the number of documents divided by the number of documents containing the word [34]. After the TF-IDF process, the classification model is also made using the pipeline. Parameters tested for modelling using `MultinomialNB()` are alpha of 0.01, 0.01, and 0.001.

L. Model Selection

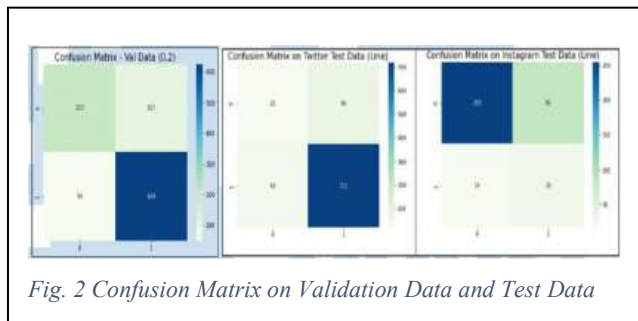
The classification model is also made using the pipeline after the TF-IDF process. Parameters tested for modelling using `MultinomialNB()` are alpha of 0.01, 0.01, and 0.001. Determination of the best parameters for TF-IDF and Naïve Bayes modelling is done using the `GridSearchCV` function, with a total of five cross-validations, on the training data. The results of the parameter determination process show that the `max_df` (TF-IDF) parameter has a value of 0.75, `ngram_range` (TF-IDF) has a value of (1,2), and alpha (`MultinomialNB`) has a value of 0.1.

M. Model Evaluation and Analysis

In determining the performance of the classification model, the F1 score will be used to measure performance for an unbalanced amount of data per label [35]. The accuracy, precision, and recall values will be used to analyze its performance based on the confusion matrix. The sentiment classification process stops at this step after the entire process is repeated for each social media data. Table 1 shows the model's performance value on the validation data from the dataset. Fig. 2. show the results of the model's performance value.

Table 1 – Model Performance

Dataset	Precision	Recall	Accuracy	F1-Score
Two social media	0.839	0.841	0.841	0.837
Data testing (Twitter)	0.856	0.872	0.872	0.863
Data testing (Instagram)	0.852	0.733	0.733	0.775



In data validation, there are six hundred seventy-six data with positive sentiment and three hundred fourteen data with negative sentiment. Seven hundred sixty-four data with positive and eighty-seven with negative sentiments were produced from testing data with Twitter sources. In testing data with Instagram sources, forty-six data with positive and three hundred forty with negative sentiments. Those show that Twitter produces data with the most positive sentiment. In contrast, Instagram has data with the most negative sentiments.

N. Set Parameters for LDA

The LDA algorithm requires parameters before starting the modelling stage, namely the optimal number of topics to be generated. A more significant number of issues usually indicates a more complex topic, while a small number of topics indicates a more general topic [28]. In addition, the parameters prepared are data in the form of a dictionary, corpus, and bag of words that consider n-grams. The first step in topic modelling is to prepare data for the LDA algorithm, creating a dictionary, corpus, and bag of words. Function phrases from the gensim library are used to detect the presence of phrases that often appear but consist of more than one word; then converted into a single unit. Making the bag of words format is done with the doc2bow function and the Dictionary function to convert bigram data into a dictionary.

O. Cross Validation for LDA

The cross-validation stage was performed to ensure that the modelling performed with the LDA algorithm resulted in the optimal number of topics [28]. The process of validating the optimal number of topics will use a coherence value, where the level of similarity between words incorporated in one topic is described in this value [36]. The greater the coherence value, the better the resulting topic represents the document [37]. The highest coherence value is used from the number of topics to get the optimal number of topics for topic modelling. The coherence value is calculated using the CoherenceModel function with the coherence value parameter value 'c_v'. A cross-validation process was performed to get the coherence value from each number of topics. The process is repeated on the test on 75%

of the data and 100% or all of the data. The topic range is determined from 2 topics to 10 topics [24]. The parameters and coherence values obtained will then be stored in the data frame for visualization purposes.

P. LDA Topic Modeling and Evaluation

Making a topic modelling model with LDA is done using the best parameters obtained from the previous stage, then implemented on social media data that has gone through the preprocessing stage. The topic will be determined based on the data per sentiment label so that the main topics on the data with positive and negative sentiments can be identified. This stage resulted in four topics with positive sentiments with a coherence value of 0.426279. However, on topics with negative sentiments, there are five topics with a coherence value of 0.397232.

Q. Topic Visualization

The results of the topic modelling process will be visualized by displaying a list of topics along with graphs using the Python pyLDAVis library. The main display on the graph is an inter-topic distance map that can be used interactively or not statically [38]. If the cluster map visualization is focused on one of the topics, the bar chart will change according to the frequency of words in the selected topic. Visualization using pyLDAVis can add to understanding the resulting topic results [38]. The results of the topic modelling show a total of nine topics. There are four topics with positive sentiment and five topics with negative sentiment.

Four topics with positive sentiments are:

- Topic 1: 36.7% of Line Bank's social media data discusses administration fees and free transfer fees.
- Topic 2: Integration with digital banking platforms or other payment methods covered 36.6% of social media data about Line Bank
- Topic 3: 14.7% of social media data discusses Line Bank physical cards, which have different designs from the usual bank cards in a positive way. Based on the data obtained, the design of the Line Bank's card is considered funny.
- Topic 4: 12.1% of the data discusses Line Bank features which are easy to understand and fast application performance.

Five topics with negative sentiments are:

- Topic 1: 30% of social media data discusses obstacles when opening a Line Bank account.
- Topic 2: Old physical card creation and problems related to card status change is discussed in as much as 24.9% of the data.
- Topic 3: Promotional activities carried out by Line Bank often fail and are discussed in 21.3% of its social media data.
- Topic 4: as much as 14.2% of social media data with negative sentiments discusses obstacles and errors that occur when transactions.
- Topic 5: 9.7% of social media data discusses customer service performance that is considered flawed.

Table 2. Comparison of model evaluation results

Authors	Algorithms			
	Naïve Bayes		SVM	
	Accuracies	F1-Scores	Accuracies	F1-Scores
Kristiyanti et al.	0.94	0.94	0.755	0.89
Arora et al	0.7944	0.794	0.7933	0.7907
Milenia et al. (author)	0.872	0.863	-	-

Table 3. Comparison of data amount and preprocessing method

Authors	Accuracies	F1-Scores	Data Training		Data Pre-processing Method
			Positive	Negative	
Kristiyanti et al.	0.94	0.94	100	100	Tokenization, use 2-grams, remove punctuation and numbers, 10-fold cross-validation.
Arora et al	0.7944	0.794	500	500	Tokenization, remove HTML markups, convert number digits to 'NUMBER', and punctuations to 'SYMBOL', and remove less frequent words.
Milenia et al. (author)	0.872	0.863	764	87	Remove punctuations, numbers, whitespaces, case folding, stop words, convert emoji, tokenization, normalization, stemming, 5-fold cross-validation, use 1-3-grams.

Table 4. Comparison of coherence values with previous research

Authors	Coherence Values	Topics	Data Sources	Pre-processing Method
Qomariyah et al	0.1376	4	Twitter	Remove punctuations, stop words, case folding, stemming, tokenization
Cheng & Sharmayne	0.4308	7	Review from Playstore	Remove punctuations, stop words, case folding, and lemmatization.
Milenia et al. (author)	0.426279	4	Twitter and Instagram	Remove punctuations, numbers, whitespaces, case folding, stop words, convert emoji, tokenization, normalization, stemming

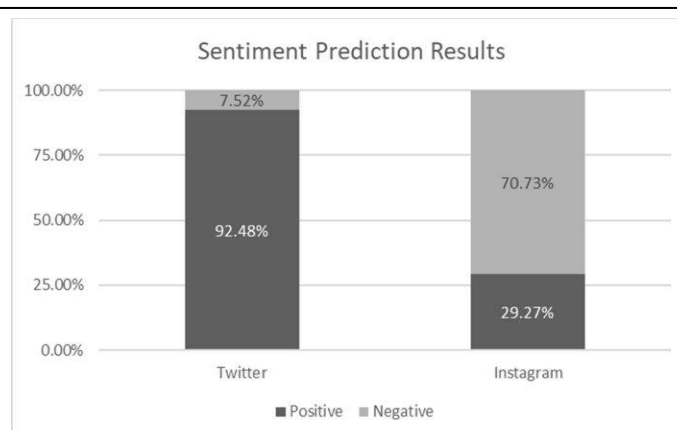


Fig. 3 Amount of Predicted Data based on Sentiment

V. DISCUSSION

Based on the evaluation results in Table 2, the value of the metrics selected for the model will be used to compare with the results of previous research on sentiment analysis [14][39]. Table 2 provides information on comparing the evaluation results of the model for the same task, namely sentiment classification of social media data, by comparing the Naïve Bayes algorithm and the Support Vector Machine (SVM). Both studies concluded that the Naïve Bayes algorithm is superior based on the accuracy value and F1 score. The findings align with this research result, where the Naïve Bayes algorithm also has a high accuracy value and F1 score.

This study resulted in a higher value than Arora et al. [39], with an accuracy value of 0.872 and the F1 score of 0.863. Both studies use a balanced amount of data for each sentiment label. However, this study uses unbalanced data on each sentiment. Then, the research also goes through different preprocessing stages so that it can affect the results of the model's performance. Table 3 compares the training data and preprocessing methods between the three studies [14], [39]. Fig. 3. shows the amount of data based on sentiment prediction. Data from Twitter has more positive sentiment tweets, with a percentage of 92.48%.

On the contrary, the data on Instagram has the highest number of negative sentiments, which is 7.52%. The positive things the digital bank has carried out can be seen from the results of the topic modelling task. The optimal number of topics has been evaluated, and a coherence value of 0.426279 is obtained on topics with positive sentiment. Table 4 compares coherence values with the research of Qomariyah et al. [20] and Cheng & Sharmayne [23]. This study has a higher coherence value than Qomariyah et al. [20]. However, this study has a lower coherence value than Cheng & Sharmayne study [23].

On the one hand, the results of topic modelling on the positive sentiment dataset show a discussion of the features provided by Line Bank, such as integration with external platforms and the application of free administration and transfer fees. Another topic also discusses cute physical card designs so that Line Bank's image becomes positive on social media. The ease of users' understanding and the speed of response from the features provided also get a positive response. On the other hand, topics with negative sentiments indicate the need for improvement in customer service in handling account openings, transaction settlement performance, and the availability of information regarding certain promotional activities.

VI. CONCLUSION

Sentiment analysis and topic modelling using social media data from a digital bank in Indonesia using the Naïve Bayes algorithm and Latent Dirichlet Allocation has been successfully carried out. On Twitter data prediction results, a lot of positive sentiment data is gathered that contains the general opinion of Twitter users regarding the digital bank. The results of the data prediction from Instagram are more of a negative sentiment because it is a place for digital bank customers who experience service problems or dissatisfaction. The data sentiment classification results show that the Naïve Bayes algorithm obtains a relatively good F1 score of 0.863. The topic

modelling process using the Latent Dirichlet Allocation (LDA) algorithm receives the optimal number of topics of four topics in the dataset with positive sentiments and five topics with negative sentiments. The resulting coherence value is 0.426279 with positive sentiment and 0.397232 with negative sentiment.

Limitation

The present study is circumscribed by a limited scope, as its findings are solely derived from an investigation into one specific digital bank operating in the Indonesian context. Consequently, the generalizability of the results is inherently constrained, and they may not be applicable to other digital banking institutions or alternative geographic locations.

Moreover, this study employs unbalanced data on each sentiment, meaning that the sample sizes for positive, negative, and neutral sentiments are not of equal size. Consequently, caution must be exercised in interpreting the findings, as the skewed data may not accurately reflect the sentiments of the broader population.

Future Research

The present study offers an analysis of digital banks using a specific set of algorithms. However, it is important for future scholars to expand the scope of investigation by considering additional digital banks and utilizing alternative algorithms. By doing so, future research will contribute to a more comprehensive understanding of the digital banking industry.

ACKNOWLEDGMENT

We gratefully acknowledge the support from Universitas Multimedia Nusantara for this research.

REFERENCES

- [1] H. Widowati, "Transaksi Digital Menggeser Peran Kantor Cabang dan ATM Bank," 2019. [Online]. Available: <https://databoks.katadata.co.id/datapublish/2019/08/08/transaksi-digitalmenggeser-peran-kantor-cabang-dan-atm-bank> [Accessed: 01-Oct-2021].
- [2] D. . Bayu, "Layanan Perbankan Digital Makin Sering Digunakan saat Pandemi," 2020. [Online]. Available: <https://databoks.katadata.co.id/datapublish/2020/11/18/layanan-perbankandigital-makin-sering-digunakan-saat-pandemi#> [Accessed: 01-Oct-2021].
- [3] Momentum Works, "Rise of Digital Banks in Indonesia," 2021. [Online]. Available: https://thelowdown.momentum.asia/country_sector/rise-of-digital-banksin-indonesia/?option=2021-indonesia&code=11502. [Accessed: 01-Oct-2021].
- [4] Line Corporation, "LINE and PT Bank KEB Hana Indonesia Launch LINE Bank in Indonesia," 2021.
- [5] R. Pahlevi, "YLKI Catat 535 Aduan Konsumen Sepanjang 2021 | Databoks," 2022. [Online]. Available: <https://databoks.katadata.co.id/datapublish/2022/01/10/ylki-catat-535-%0Aaduan-konsumen-sepanjang-2021> [Accessed: 05-Jul-2022].
- [6] N. Sahara, "YLKI Terima Banyak Pengaduan Soal Lembaga Keuangan," 2020.
- [7] Otoritas Jasa Keuangan, "OJK Government PR Forum 2016," 2016. .
- [8] Hootsuite and We Are Social, "Digital 2020: Indonesia — DataReportal — Global Digital Insights," 2020.
- [9] Statista, "Twitter: most users by country," 2021.
- [10] Statista, "Instagram: users by country," 2021.
- [11] D. Sarkar, R. Bali, and T. Sharma, Practical Machine Learning with Python. Apress, 2018.
- [12] B. Benjamin, Applied Text Analysis with Python, vol. 53, no. 9. O'Reilly, 2018.

- [13] V. A. Fitri, R. Andreswari, and M. A. Hasibuan, "Sentiment Analysis of Social Media Twitter with Case of Anti-LGBT Campaign in Indonesia using Naïve Bayes, Decision Tree, and Random Forest Algorithm," *Procedia Comput. Sci.*, vol. 161, pp. 765–772, 2019.
- [14] D. A. Kristiyanti, A. H. Umam, M. Wahyudi, R. Amin, and L. Marlinda, "Comparison of SVM & Naive Bayes Algorithm for Sentiment Analysis Toward West Java Governor Candidate Period 2018-2023 Based of Public Opinion on Twitter," 6th Int. Conf. Cyber IT Serv. Manag. (CITSM 2018), 2018.
- [15] A. Nayak and S. Natarajan, "Comparative study of Naïve Bayes, Support Vector Machine and Random Forest Classifiers in Sentiment Analysis of Twitter feeds," *Int. J. Adv. Stud. Comput. Sci. Eng.*, vol. 5, no. 1, 2016.
- [16] A. E. Khedr, S. E. Salama, and N. Yaseen, "Predicting Stock Market Behavior using Data Mining Technique and News Sentiment Analysis," *MECSJ. Intell. Syst. Appl.*, vol. 7, pp. 22–30, 2017.
- [17] V. Kotu and B. Deshpande, *Data Science Concept and Practice*. Morgan Kaufmann Publishers, 2019.
- [18] M. D. Devika, C. Sunitha, and A. Ganesh, "Sentiment Analysis: A Comparative Study on Different Approaches," *Procedia Comput. Sci.*, vol. 87, pp. 44–49, 2016.
- [19] EMC Education Services, *Data Science & Big Data Analytics*. Wiley, 2015.
- [20] S. Qomariyah, N. Iriawan, and K. Fithriasari, "Topic modeling Twitter data using Latent Dirichlet Allocation and Latent Semantic Analysis," in *AIP Conference Proceedings*, 2019.
- [21] R. Albalawi, T. H. Yeap, and M. Benyoucef, "Using Topic Modeling Methods for Short-Text Data: A Comparative Analysis," *Front. Artif. Intell.*, vol. 3, p. 42, 2020.
- [22] E. S. Negara, D. Triadi, and R. Andryani, "Topic Modelling Twitter Data with Latent Dirichlet Allocation Method."
- [23] L. C. Cheng and L. R. Sharmayne, "Analyzing Digital Banking Reviews Using Text Mining," *IEEE/ACM Int. Conf. Adv. Soc. Networks Anal. Min.*, 2020.
- [24] Q. Yang, "Data Mining of New Snack E-commerce Reviews Based on Text Sentiment Analysis and Latent Dirichlet Allocation Topic Model," 2020.
- [25] J. Roesslein, "API — tweepy 4.10.0 documentation," 2022. [Online]. Available: <https://docs.tweepy.org/en/stable/api.html>. [Accessed: 20-May-2022].
- [26] A. Maulana, "Instagram-Comments-Scraper at feature-login," 2021. [Online]. Available: <https://github.com/AgiMaulana/Instagram-Comments-Scraper/tree/featurelogin>. [Accessed: 20-May-2022].
- [27] J. Pierre, "Philippine Twitter Sentiments during Covid-19 Pandemic using Multinomial Naïve-Bayes," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 1, pp. 408–412, 2020.
- [28] C. B. Asmussen and C. Møller, "Smart literature review: a practical topic modelling approach to exploratory literature review," *J. Big Data*, vol. 6, no. 93, pp. 1–18, 2019.
- [29] S. Sfenrianto, E. R. Kaburuan, A. Bayhaqy, and K. Nainggolan, "Sentiment Analysis about E-Commerce from Tweets Using Decision Tree, K-Nearest Neighbor, and Naïve Bayes," *Int. Conf. Orange Technol.*, 2018.
- [30] A. A. Lutfi, A. E. Permanasari, and S. Fauziati, "Sentiment Analysis in the Sales Review of Indonesian Marketplace by Utilizing Support Vector Machine," *J. Inf. Syst. Eng. Bus. Intell.*, vol. 4, no. 1, 2018.
- [31] M. Bollmann, "A Large-Scale Comparison of Historical Text Normalization Systems," 2019. [Online]. Available: <https://github.com/clarinsi/csmtiser>. [Accessed: 01-Jun-2022].
- [32] A. K. Singh and M. Shashi, "Vectorization of Text Documents for Identifying Unifiable News Articles," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 7, 2019.
- [33] R. Ali and S. Qaiser, "Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents," *Artic. Int. J. Comput. Appl.*, vol. 181, no. 1, pp. 975–8887, 2018.
- [34] R. Lourdasamy and S. Abraham, "A Survey on Text Pre-processing Techniques and Tools," *Int. J. Comput. Sci. Eng.*, vol. 6, no. 3, pp. 148–157, 2018.
- [35] A. Priya, S. Garg, and N. P. Tigga, "Predicting Anxiety, Depression and Stress in Modern Life using Machine Learning Algorithms," *Procedia Comput. Sci.*, vol. 167, pp. 1258–1267, 2020.
- [36] N. A. Tresnasari, T. B. Adji, and A. E. Permanasari, "Social-Child-Case Document Clustering based on Topic Modeling using Latent Dirichlet Allocation," *Indones. J. Comput. Cybern. Syst.*, vol. 14, no. 2, p. 179, 2020.
- [37] S. George and S. Vasudevan, "Comparison of LDA and NMF Topic Modeling Techniques for Restaurant Reviews," *Indian J. Nat. Sci.*, vol. 10, no. 62, pp. 28210–28216, 2020.
- [38] A. F. Hidayatullah and M. R. Ma'Arif, "Road traffic topic modeling on Twitter using latent dirichlet allocation," in *International Conference on Sustainable Information Engineering and Technology*, 2017, vol. 2018.
- [39] A. Arora, P. Patel, S. Shaikh, and A. Hatekar, "Support Vector Machine versus Naive Bayes Classifier: A Juxtaposition of Two Machine Learning Algorithms for Sentiment Analysis," *Int. Res. J. Eng. Technol.*, 2020.

The impact of COVID-19 on volatility spillover in Stock Exchange of Thailand

Nuttaphat Sukchitt

Department of Statistics, Faculty of Science
Chiang Mai University
Chiang Mai, Thailand
nuttaphat.sukch@cmu.ac.th

Manad Khamkong

Department of Statistics, Faculty of Science
Chiang Mai University
Chiang Mai, Thailand
manad.k@cmu.ac.th

Abstract—This research focuses on the volatility spillover between the returns of the Stock Exchange of Thailand (SET) index and the return of the SET index in each industry which has grouped the companies into 8 industries. The search for COVID-19 words in Thailand was retrieved from the Google Trends database as an exogenous variable. Our focus has been on four periods: the first started on 2 January 2020 and ended on 1 June 2020, the second started on 2 November 2020 and ended on 1 March 2021, the third started on 1 March 2021 and ended on 15 June 2021, and the last started on 16 June 2021 and ended on 1 November 2021. In this study, a multivariate generalized autoregressive conditional heteroscedasticity Diagonal BEKK model (DBEKK) with an exogenous variable is applied to measure spillovers in covolatility. According to this study, the autoregressive conditional heteroscedasticity (ARCH) effect was observed in the first and third periods. In the first period, the ARCH effect was observed on the return of the SET index, as well as the return of four industries, including consumer products, financials, property construction, and technology, while in the third period we noticed the ARCH effect on the following return of industries: the resources, services, and technology. Note that neither the second nor the fourth period showed an ARCH effect. According to Google Trends for COVID-19 searches in Thailand, the financials index had a higher level of shocks and sensitivity in the first period. Additionally, the SET index demonstrated high levels of volatility transmission and was highly correlated with other industries, particularly construction, financials, and technology. The technology industry was more sensitive to shocks in the third period. However, the level of volatility transmission was higher in the services industry. This indicated that the volatility of the services industry was highly correlated with that of other industries.

Keywords— COVID-19, The Stock Exchange of Thailand, BEKK, GARCH,

I. INTRODUCTION

The emergence of the COVID-19 pandemic in 2019 marked an unprecedented global crisis that quickly gripped nations around the world. The rapid transmission of the virus, which originated in Wuhan, China, shattered international borders and plunged countries into a state of unexpected turmoil. As 2020 began, the alarming rise in infection rates and deaths from the virus left an indelible mark on human history, prompting the World Health Organization (WHO) to declare it a full-fledged global pandemic on March 11, 2020. This momentous statement underscored We have recognized the seriousness of the situation and have unleashed a series of unprecedented responses from governments and institutions around the world.

Among the countries struggling with the virus attack, Thailand faced a formidable challenge in 2020. The contagion reached the Thai coast through international travelers, triggering alarms from the government. To curb the dangerous spread of the virus within its borders, the Thai government has taken a crucial step by imposing a provincial lockdown. This restrictive measure aimed to contain the transmission of the virus by confining people to their respective regions, a move that had immediate and far-reaching consequences for both the economy and society.[1] Thailand's economic landscape experienced a dramatic change as the pandemic spread. The International Monetary Fund (IMF) reported a staggering 6.1 percent drop in domestic production for 2020. This steep decline dealt a severe blow to the Thai economy and had far-reaching consequences for companies, industries and workers. As economic activity ground to a halt due to the lockdown measures, unemployment figures skyrocketed, compounding the social and economic challenges the country was facing[2].

The turmoil in the financial markets was another aspect of the impact of the pandemic. Investors found themselves in dangerous waters as market volatility increased due to the unforeseen and rapidly evolving situation. The uncertainty caused by the pandemic caused investors to panic, prompting them to divest from risky assets[3]. This retreat from riskier assets was emblematic of the prevailing apprehension and unpredictability gripping global financial markets. The aftermath of this uneasiness was felt even in established economies like the United States, where there was a noticeable fall in stock markets[4].

With the unprecedented uncertainty that marked this period, academics and researchers conducted numerous studies to understand the volatile financial landscape. Approaches such as the constant conditional correlation model and the dynamic conditional correlation model have been employed to analyze market volatility[5], [6]. On the other hand, the previous studies applied the conditional variance, covariance matrix to study the spillover effect [7], [8], [9].

However, amidst this research panorama there remained a noticeable gap, no concerted efforts were made to analyze the covolatility spillovers and the complex relationships between the return of the SET Index, the SET in different industries and the growth rate of COVID-19-related online searches.

We chose to analyze the covolatility spillovers and the transmission between the rate of returns of the SET index and the SET in each industry as well as the growth rate of the

27

COVID-19 search using DBEKK model since there were no researchers studying these factors in the stock market. The paper is divided into four sections: Introduction, Methodology, Results, Discussion, and Conclusion

II. METHODOLOGY

Multivariate generalized autoregressive conditional heteroscedasticity Diagonal BEKK model (DBEKK) is a multivariate restrictive volatility model and a dynamic conditional correlation model[10]. The study examined the volatility spillover between each industry and the SET index, as well as between the industries themselves, by using DBEKK model

The following equation has shown the conditional variance and covariance matrix: where Q_t is the time t associated $n \times n$ variance-covariance matrix. Since Q_t must be positive because of the quadratic form of Equation (1), ε_{t-1} is the disturbance vector with $n \times 1$, C is just the lower triangle matrix of intercept coefficients, and A and B are the diagonal matrix and E is an exogenous variable.

$$Q_t = C'C + A'\varepsilon_{t-1}\varepsilon'_{t-1}A + B'Q_{t-1}B + E'E \quad (1)$$

as the matrix form as follows:

$$Q_t = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}' \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} + \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}' \begin{bmatrix} \varepsilon_{t-1}^2 & \varepsilon_{1,t-1}\varepsilon_{2,t-1} \\ \varepsilon_{2,t-1}\varepsilon_{1,t-1} & \varepsilon_{2,t-1}^2 \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} + \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}' \begin{bmatrix} \sigma_{11,t-1} & \sigma_{12,t-1} \\ \sigma_{21,t-1} & \sigma_{22,t-1} \end{bmatrix} \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} + \begin{bmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \end{bmatrix}' \begin{bmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \end{bmatrix} \quad (2)$$

where, Q_t is an $n \times n$ conditional variance-covariance matrix, C is an upper triangular matrix of parameters, ε_{t-1} is an $n \times 1$ disturbance vector, E is an upper triangular matrix of exogenous variable, A and B are $n \times n$ diagonal parameter matrices.

The parameters of the Multivariate generalized autoregressive conditional heteroscedasticity diagonal BEKK model can be estimated by maximizing the loglikelihood function in accordance with conditional normality[10]. We study the return of the SET index and the return of eight industries. The first step is to determine how much the return of the SET index impacts the volatility spillover on the return of each industry, as well as between the return of industries themselves. The conditional variances of the return of the SET index and the return of eight industries are represented by the matrix Q_{tIS} . The parameters $C_{11,t}$ and $C_{22,t} \cdot C_{12,t}$ and $C_{21,t}$ represent the conditional covariance of the return on the SET index between industries, as well as between industries themselves.

The impact of prior shocks and volatility on the return of the set index and the return of eight industries conditional variance is measured by A_{11} and $B_{11} \cdot A_{22}$ and B_{22} . On the other hand, both capture the impact of past shocks and volatility in other return of industries conditional variance. The purpose of the analysis is to examine the effects of volatility variations. We are interested in parameters A and B . The coefficients A_{12} and A_{21} show how much cross-industry

spillover effects occur where the spillover effects of the volatility are represented by the coefficients B_{12} and B_{21} . The off-diagonal characteristics are investigated to see how volatility spillover affects a financial system's performance.

$$H_0: A_{12} = A_{21} = B_{12} = B_{21} = 0, \\ H_1: A_{12}, A_{21}, B_{12}, B_{21} \neq 0.$$

If H_0 is rejected and A_{12}, A_{21} significant, there are shock spillover effects between return of the SET index and return of each industry, as well as between the industries themselves. If H_0 is rejected and B_{12}, B_{21} significant, there are volatility spillover effects between return of the SET index and return of each industry, as well as between the return of industries themselves. Specifically, if all parameters are significant, the shock spillover and volatility spillover effect between return of the SET index and return of each industry, as well as between the return of industries themselves are indicated.

III. RESULT AND DISCUSSION

In this research, we consider the secondary data including the SET index and 8 industries as shown in Table 1.

We used data from 8 industries including the Agro and Food Industry (ARGO), Consumer Products (CONSUMP), Financials (FINCIAL), Industrials (INDUS), Property and Construction (PROPCON), Resources (RESOURC), Services (SERVICE) and Technology (TECH).

TABLE I. LIST OF INDUSTRIES UNDER STUDY[11]

Industry	Abbreviation	Sector
Agro and Food Industry	ARGO	Agribusiness, Food and Beverage
Consumer Products	CONSUMP	Fashion, Home and Office Products, Personal Products and Pharmaceuticals
Financials	FINCIAL	Banking, Finance and Securities, Insurance Automotive
Industrials	INDUS	Industrial Materials and Machinery, Paper and Printing Materials, Petrochemicals and Chemicals, Packaging, Steel
Property and Construction	PROPCON	Construction Materials, Construction Services, Property Fund and REITs, Property Development
Resources	RESOURC	Energy and Utilities, Mining
Services	SERVICE	Commerce, Health Care Services, Media and Publishing, Professional Services, Tourism and Leisure, Transportation and Logistics
Technology	TECH	Electronic Components, Information and Communication Technology

We also examined Google Trends Index as part of our study. It should be noted that the idea of this section is oriented by Napons' work. [12] The collected data was divided into four periods : The first period is from 2 January 2020 to 1 June 2020, the second period is from 2 November 2020 to 1 March 2021, the third period started from 1 March 2021 to 15 June 2021 and the fourth period is from 16 June 2021 to 1 November 2021. The stock index of each category and Google Trends index will be converted to the growth rate and return by using the conventional formula,

$$R_t = \ln\left(\frac{p_t}{p_{t-1}}\right), \quad (3)$$

where R_t = Return of index,
 p_t = Index on today,
 p_{t-1} = Index from yesterday.

In this study, there are the ARCH effects only in the first and the third periods. In the first period, the ARCH effects were observed in the SET index as well as in four industries, including consumer products, finance, property construction, and technology. Furthermore, in the third period, the ARCH effects were also observed in three industries which are resources, services, and technology. However, this study did not find the ARCH effect in the second and fourth periods.

The average shocks with exogenous variable of the SET index and four industries in the first period have been represented in table 2.

TABLE II. RANK OF AVERAGE SPILLOVERS SHOCKS OF EXOGENOUS VARIABLE ON THE SET INDEX AND FOUR INDUSTRY OF SET INDEX AT FIRST PERIOD

No	Abbreviation	Average shocks
1	FINCIAL	0.000725
2	PROPCON	0.000535
3	SET	0.000519
4	TECH	0.000363
5	CONSUMP	0.000125

The DBEKK can explain the effect of Google Trends with the SET and four industries. The results show that the Google Trends has a negative value related to other variables. Additionally, each industry's volatility affects others. In other words, there is a spillover effect from covolatility. Moreover, there are the SET index and four industries, mentioned in table 2, which has a high average spillover shock with exogenous variable. The highest is financials industries. And the lowest is consumer products.

TABLE III. RANK OF AVERAGE COVOLATILITY SPILLOVERS WITH EXOGENOUS VARIABLE FOR SET INDEX AND FOUR INDUSTRY OF SET INDEX AT FIRST PERIOD

No.	Abbreviation i	Abbreviation j	Average Covolatility Spillovers
1	SET	PROPCON	0.9292
2	SET	FINCIAL	0.8853
3	FINCIAL	PROPCON	0.8246
4	SET	TECH	0.7492
5	CONSUMP	FINCIAL	0.7213
6	CONSUMP	PROPCON	0.7182
7	SET	CONSUMP	0.6964
8	PROPCON	TECH	0.6929
9	CONSUMP	TECH	0.5386
10	FINCIAL	TECH	0.4952

The risk volatility spillover effects between the SET index's return and four industries in the first period are shown in terms of the average covolatility spillovers. Table 3 shows the sensitivity of the shock impact.

Highest average covolatility spillovers with exogenous variable for the SET index and four industries were observed in the first period as represented in table 3.

In the first period, considering the growth rate of Google Trend as an exogenous variable, the highest value of average covolatility spillovers is 0.9292, which indicates a relationship between the return of the SET Index and the return of the property and construction industry. If the returns

of the SET Index increase by 1 percent during this period, the return of the property and construction industry are expected to increase by 0.9292 percent.

TABLE IV. RANK OF AVERAGE SPILLOVER SHOCKS WITH EXOGENOUS VARIABLE FOR THREE INDUSTRY OF SET INDEX AT THE THIRD PERIOD

No	Abbreviation	Average shocks
1	TECH	0.0003
2	SERVICE	0.000126
3	RESOURC	0.000107

Since the spillovers could change over time, using the average covolatility to cover the shocks in the sample period is a reasonable approach to retrieve the value of spillover. The results of the average shocks with exogenous variable of three industries of SET index were observed in the third period.

DBEKK can explain the effect of Google Trends and three industries. It shows that the Google Trends has a negative correlation with other value variables. Additionally, the industry which has volatility to each other can be represented as an average covolatility spillovers effect. Besides, table 4 illustrates industries that have a high average spillover shock with exogenous variable.

TABLE V. RANK OF AVERAGE COVOLATILITY SPILLOVERS WITH EXOGENOUS VARIABLE FOR THREE INDUSTRY OF SET INDEX AT THE THIRD PERIOD

No.	Abbreviation i	Abbreviation j	Average Covolatility Spillovers
1	RESOURC	SERVICE	0.6274
2	SERVICE	TECH	0.3822
3	RESOURC	TECH	0.2636

The risk volatility spillover effects between returns in three industries in the third period are shown in terms of the average covolatility spillovers. The results from table 5 present the sensitivity of the impact of the shock and there is the highest average covolatility spillovers with exogenous variable for three industries in the third period.

In the third period, considering the growth rate of Google Trend as an exogenous variable, the highest value of Average Covolatility is 0.6274 indicates a relationship between the return of the resources industry and the return of the services industry. If the returns of the resources industry increase by 1 percent during this period, the return of the services industry are expected to increase by 0.6274 percent.

IV. CONCLUSION

The Google Trends has an impact on both The Stock Exchange of Thailand (SET) index by each industry volatility and the volatility transmission. The Google Trends is not only affecting the SET index in each industry, but also the volatility of the SET index in each industry. The investigation of the volatility of the SET index on each industry in the first period and the third period found the effect of the Google Trends.

Looking at the dynamics over time, it becomes clear that the Google Trends data has a discernible impact on the volatility of the SET index and the return of the SET index in each industry over different time periods.

Financials, property construction, and the SET index have high volatility in the first period. Besides, volatility in one industry can affect the volatility in other industries.

Moreover, we found that the SET index has the highest level of volatility transmission, which means the volatility of the SET index is highly correlated to the volatility from other Industries, especially the property and construction, financials, and technology.

In the third period, the technology, services, and resources industry have high volatility. It can be seen that the volatility from one industry of the SET index can affect the volatility in other industries. The level of volatility transmission is highest in the services, this volatility of the services is highly correlated to the volatility from other industries, especially the resources, and technology.

REFERENCES

- [1] S. Kaendera, and L. Leigh., "Five Things to Know About Thailand's Economy and COVID-19", [Online]. Available: <https://www.imf.org/en/News/Articles/2021/06/21/na062121-5-things-to-know-aboutthailands-economy-and-covid-19> [Accessed: 30-Jun.-2020]
- [2] Y. Thaicharoen., "Thailand after Covid-19 Prepare to get through the economist's perspective. Part 1: The impact of the economy and the Thai labor market", [Online]. Available: <https://www.scb.co.th/en/personal-banking/stories/business-maker/thailandafter-covid-ep1.html>. [Accessed 7- May-2020]
- [3] P. Domm., "How the pandemic drove massive stock market gains, and what happens next", [Online]. Available: <https://www.cnbc.com/2020/12/30/how-the-pandemicdrove-massive-stock-market-gains-and-what-happens-next.html> [Accessed 30-Dec-2020]
- [4] K. Amadeo., "How Does the 2020 Stock Market Crash Compare with Others?", <https://www.thebalancemoney.com/fundamentals-of-the-2020-market-crash-4799950> [Accessed 4-May-2022]
- [5] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.
- [6] S. Xiaojun, "Multivariate GARCH models for the Greater China stock markets," Dissertation, Singapore Management Univ., 2009.
- [7] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.
- [8] Y. H. Lee, Y. L. Huang and C.Y. Wu, "Dynamic Correlations and Volatility Spillovers between Crude Oil and Stock Index Returns: The Implications for Optimal Portfolio Construction," *International Journal of Energy Economics and Policy*, vol.4 no.3, pp. 327-336, 2014.
- [9] J. B. Su and J. C. Hung, "The Value-At-Risk Estimate of Stock and Currency-Stock Portfolios' Returns," *Risks*, vol. 6, no. 4, pp. 133, 2018.
- [10] H. Mohammadi and Y. Tan, "Return and Volatility Spillovers across Equity Markets in Mainland China, Hong Kong and the United States," *Econometrics*, vol. 3, no. 2, pp. 215-232, 2015.
- [11] P. Joshi, "Return and Volatility Spillovers Among Asian Stock Markets," *Sage Open*, vol. 1, no.1, 2011.
- [12] Y. Baba, R.F. Engle, D. Kraft, and K.F. Kroner, "Multivariate simultaneous generalized ARCH," Dept. Econ, University of California, San Diego, CA, USA, 1985.
- [13] "SET Industry Group Index and Sector", [Online]. Available: <https://classic.set.or.th/en/products/index/setindexp2.html.printable=true> [Accessed 15-Dec-2020]
- [14] N. Hongsakulvasu, C. Khiewngamdee and A. Liamukda, "Does COVID-19 crisis affects the spillover of oil Market's return and risk on Thailand's sectoral stock return?: Evidence from bivariate DCC GARCH-in-mean model," *International Energy Journal*, vol. 20, no. 4, pp. 647-662, 2020.

Deep learning-based slip-trip falls and near-falls prediction model using a single inertial measurement unit sensor for construction workplace

Oleksandr Yuhai¹, Hyunggun Kim¹, Ahnryul Choi^{1,2} and Joung Hwan Mun¹

¹Department of Bio-Mechatronic Engineering, Sungkyunkwan University, Suwon, Republic of Korea

²Department of Biomedical Engineering, Catholic Kwandong University, Gangneung, Republic of Korea

e-mail: oleksandr@g.skku.edu, hkim.bme@skku.edu, achoi@cku.ac.kr, jmun@skku.edu

Abstract—Slip-trip falls are one of the leading causes of injury in dynamic constructive workplace environments. To solve the problem of slips-trip in the construction industry, it is important not only to predict falls in order to be able to apply various fall prevention systems, but also to detect and monitor slip-trip near-falls in order to eliminate various dangerous anomalies in a timely manner. In this study, we propose a deep learning-based method for predicting and classifying activities of daily living (ADLs), trip falls, slip falls, trip near-falls and slip near-falls using a waist-attached wearable device with an embedded single inertial measurement unit (IMU) sensor containing a 3-axis accelerometer and a gyroscope. A total of 34 young and healthy participants took part in experiment to collect accelerometer and gyroscope data while performing 8 types of ADLs, slip-trip falls, and slip-trip near-falls. The data was processed and then 30 features were extracted from them. The resulting feature data was then transformed using bicubic interpolation to fit the input data size of the modified interpretation of LeNet-5 Convolutional Neural Network (CNN) deep learning algorithm. After completing 5-folds cross validation with 5 evaluation criteria, the proposed prediction and classification method showed high performance with an accuracy of about 0.9039, a specificity of about 0.9753 and F1-score of about 0.9049. Therefore, it is believed that the method proposed in this study can be used not only to increase accuracy by reducing false alarms of systems for early detection and prevention of slip-trip falls, but also for various systems for detecting and timely elimination of dangerous anomalies in the construction workplace.

Keywords—Slip and trip prediction, near-fall detection, pre-impact fall detection, deep learning, inertial measurement unit

I. INTRODUCTION

Falls that occur on the same level are one of the leading causes of non-fatal injuries in the construction workplace [1]. Usually, such incidents are associated with slips and trips leading to loss of balance and eventually a fall [2]. In the construction working environment, there are many pitfalls such as various construction aids, by-products, slippery and uneven surfaces that lead to frequent falls due to trips and slips [3]. It was found that 18.1% of all construction injuries and more than 85% of fall injuries at the same level were due to trips and slips (STs) falls [2, 4]. In addition, even those STs that do not cause a fall can lead to various secondary injuries due to active worker movements to restore balance in a dynamic construction work environment [4].

To prevent the occurrence of falls and other various incidents associated with STs in the construction work environment, it is important to identify and eliminate the cause

of their appearance in a timely manner [5]. In addition, the use of various fall prevention systems, such as a wearable airbag, is required to protect workers in the event of unavoidable falls. However, for the correct operation of such systems, it is required to use an algorithm capable of detecting a fall before the impact on the fall surface occurs [6]. One recent strategy for preventing falls is to monitor the detection of near-fall or near-miss fall cases [7]. Near-falls is a loss of balance phenomenon in which taking appropriate measures to restore a stable posture in a short time allow to avoid falling [8]. Monitoring and understanding near-falls can identify potential hazards that can be addressed early enough to reduce the likelihood of accidents [9]. Although falls due to STs have been recognized as a major cause of occupational injuries, which not only increases mortality in the work environment, but also deeply affects the economy of a country [10], the problem of STs has been underestimated [11]. Thus, in order to solve the various problems associated with STs in the construction work environment, it is important not only to be able to predict falls due to STs, but also to detect and monitor near-falls associated with STs events.

Many studies have been done to detect falls or near-falls using different types of sensors such as vision sensors, environmental sensors and various other types of wearable sensors [12]. However, systems based on vision sensors and environmental sensors have their own technical limitations, such as limited detection volume, sensitivity to obstacles in the detection volume, privacy and false alarms [13]. Recent research on fall detection on construction workplace is increasingly using an inexpensive and compact wearable inertial measurement unit (IMU) sensor that is more reliable than vision or environmental sensors-based methods in complex and dynamic construction site environment [14]. Most of the previous research on falls [15, 16] and near-falls [17, 18] detection based on the wearable IMU sensor used a threshold-based method, where threshold value was derived by calculating the sum of the total acceleration of the sensor readings along the X, Y, and Z axes. Despite the simplicity of calculating the threshold value, this method is highly susceptible to noise, sensor position, and person-to-person variability [19]. Therefore, in order to overcome the main limitations of the threshold-based method and improve recognition accuracy, many studies have been conducted on the use of machine learning algorithms such as k-nearest neighbor (k-NN), support vector machine (SVM), decision trees (DT), Naïve Bayes (NB) classifier, Hidden Markov model (HMM) and combined algorithms [20-24]. Machine learning methods have shown better performance than

threshold-based methods in classifying falls or near-falls, and more recent research on deep learning has contributed to even better performance [25].

Various studies have been conducted to detect STs falls or STs near-falls in a laboratory or construction environment using a wearable sensor such as an IMU and an AI algorithm [11, 25-27]. However, there have not yet been studies on the classification of activities of daily living (ADLs), STs falls and STs near-falls. Near-falls, such as STs near-falls can occur in a variety of conditions that intersect with real STs falls or ADLs, which makes it difficult to classify STs near-falls along with real STs falls and ADLs [7]. Therefore, the main goal of this study is to develop a deep learning algorithm for classifying ADLs, STs falls, and STs near-falls using a single IMU module. The pre-impact STs fall detection method developed in this study may be useful in the further development of construction workplace fall prevention systems, such as a wearable airbag. In addition, proposed STs near-falls along with STs falls and ADLs classification method, can be used as a robust and accurate system for monitoring and timely addressing fall risk factors in a construction environment.

The rest of this paper is organized as follows. Section II presents the materials and methods that were used to collect data, develop a detection algorithm, and evaluate performance. Then, Section III demonstrate the performance results of the proposed deep learning algorithm. Finally, in Section IV, we analyze and interpret results, indicate the main contributions and limitations of this study, and discuss a possible future direction for the study.

II. MATERIALS AND METHODS

A. Data collection

1. *Subjects.* This data collection experiment involved 34 young participants, among them 21 men and 13 women, aged from 21 to 34 years, with a body weight of 45 to 81 kg, and a height of 1.57m to 1.85m. All participants were completely healthy and had no musculoskeletal disorders. The entire process of the experiment was carried out in accordance with the relevant recommendations of the Sungkyunkwan University Ethics Committee.

2. *Apparatus.* A custom designed device with an embedded IMU module (MPU-6050, JK Electronics, Seoul, South Korea) containing a 3-axis accelerometer and a 3-axis gyroscope was used to collect ADLs, STs falls and STs near-falls data. The output data range was set to $\pm 2 - 16$ G for the accelerometer and $\pm 250 - 2,000$ d/s for the gyroscope. The process of collecting data from the sensory device by the workstation was carried out wirelessly using the Bluetooth V4.0 BLE (HM-11, JN Huamao Technology Co., Jinan, China) module at a sampling rate of 40 Hz. A windows form data acquisition program written in C # was used to collect real-time data from sensory device.

3. *Experimental protocols.* Before the start of the experiment, each subject watched a video clip showing an example of performing each activity. The sensory device was attached to the region of subject's left anterior iliac crest. Each of the subjects was instructed to conduct ADLs, STs falls and STs near-falls as realistically as possible. For the STs falls experiments, the subject had to perform a trip or slip and then fall onto a pre-prepared safety mat, while during STs near-falls experiment, the subject had to perform a trip or slip and

then regain balance in every possible way, as shown in the demo video. In the STs fall and STs near-fall experiments, dense gymnastic mats were placed in areas where the subject might fall to minimize impact. Subjects were asked to walk about 3m from the point of initiation of the action before performing a STs fall and STs near-fall. For trip fall and trip near-fall experiment, a plastic container of size 32 (length) $\times$ 23.5 (width) $\times$ 9 (height) cm was firmly taped to the floor at a distance of 15 cm from the safety mat. To conduct the slip fall and slip near-fall experiment, a polyvinyl chloride desk pad of size 81 (length) $\times$ 50 (width) $\times$ 0.2 (height) cm was strongly fixed in front of the safety mat. The surface of this desk pad was sprayed with water before each slip fall and slip near-fall experiment to reduce the coefficient of friction and create conditions similar to real slip environment. After performing STs falls or STs near-falls, the subject had to walk about 15 cm from the mat to the indicated position, and stand still for approximately 3 seconds before returning to the starting position. Based on previous studies [28], an ADLs experiment was modeled to conduct 8 types of activities, such as backward lying, standing, stand to sit, sit to stand, walking, upstairs, downstairs and jumping. As a result of this experiment, we collected a total of 177, 122, 124, 130, and 123 datasets of ADLs, trip falls, slip falls, trip near-falls, and slip near-falls, respectively.

B. An overview of the method for predicting ADLs, STs falls and STs near-falls

The method proposed in this paper for predicting and classifying ADLs, STs falls and STs near-falls consists of 7 main steps shown in Fig. 1. First, experimental data is collected from a 3-axis accelerometer and a gyroscope, then data processing occurs, where the data is cleaned and filtered. After that, the data is labeled, the main features are extracted, and the data is transformed to fit the format of the input data of the proposed deep learning algorithm. Finally, the deep learning model is trained and optimized, followed by its performance evaluation.

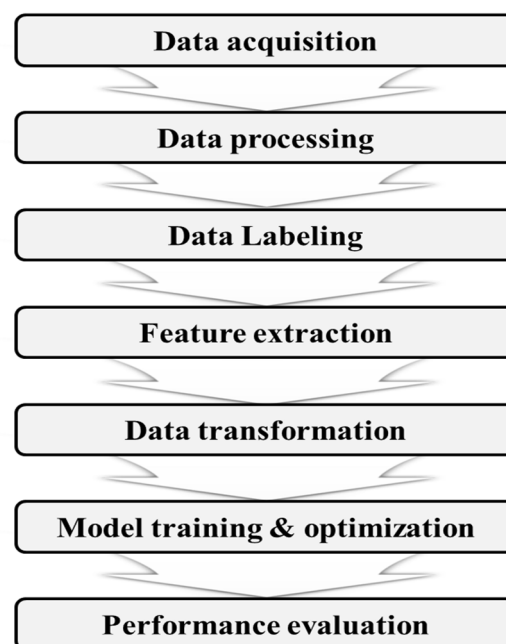


Fig. 1. Flowchart of the proposed method for predicting ADLs, STs falls and STs near-falls.

C. Data processing and labeling

The raw data received from the IMU sensor was thoroughly analyzed and cleaned to eliminate anomalous, erroneous and duplicate data that can negatively affect the performance of the deep learning algorithm. To improve data quality and eliminate various noises, a moving median filter was used.

As demonstrated in previous studies [7, 26], one of the most important data processing operations for developing supervised machine learning algorithms for prediction and classification is data labeling. Figure 2 shows a comparison of the main phases of STs falls (trip fall) and STs near-falls (trip near-fall) of the data extracted and used during the manual labeling step.

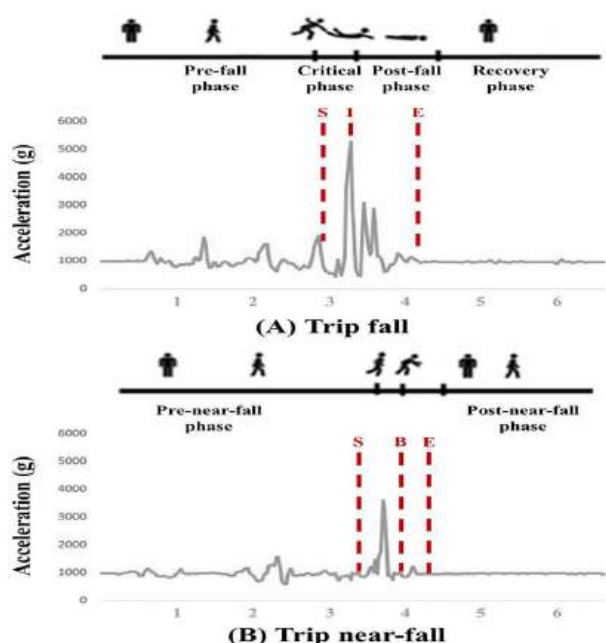


Fig.2. Comparison of falls ((A) trip fall) and near-falls ((B) trip near-fall) using the SMV signal of the 3-axis accelerometer signal. Here, S: start of the fall or near-fall; I: impact of fall; E: end of fall or near-fall; B: balance recovery phase.

During the data labeling process, the pre-fall/near-fall phase, post-fall/near-fall phase, recovery phase were assigned to the ADLs class. In the case of data obtained during the STs falls experiment, the critical phase starting from the start point of the fall and ending with the impact of the fall was assigned to the pre-impact STs falls (trip falls or slip falls) class, while in the case of data obtained during the STs near-falls experiment, the period starting from the start of the near-fall and ending at the balance recovery phase was assigned to STs near-falls (trip near-falls or slip near-falls) class.

D. Feature extraction and data transformation

When developing a machine learning algorithm for human activity recognition, features extraction is one of the essential steps. A total of 30 features, shown in Table I, were extracted based on previous studies [7]. Thus, we calculated 26 accelerometer features and 4 features corresponding to the orientations extracted using the Kalman filter (Quaternions) with accelerometer and gyroscope data.

The bicubic interpolation method was used to transform

TABLE I
EXTRACTED FROM IMU DEVICE HANDCRAFTED FEATURES

Extracted features		
	Accelerometer	Combination
1	Raw acceleration (x)	Quaternion (x)
2	Raw acceleration (y)	Quaternion (y)
3	Raw acceleration (z)	Quaternion (z)
4	Min. (x, y, z)	Quaternion (w)
5	Max. (x, y, z)	
6	Difference of Max. and Min.	
7	Mean (x, y, z)	
8	Variance (x, y, z)	
9	STD (x, y, z)	
10	Interquartile range	
11	Sum magnitude vector	
12	Sum magnitude area	
13	Sum vector in horizontal plane	
14	Tilt angle (x, y, z)	
15	Sagittal angle (x, y, z)	
16	Jerk (x)	
17	Jerk (y)	
18	Jerk (z)	
19	Jerk (x, y, z)	
20	Jerk (Max. (x, y, z))	
21	Jerk (Mean (x, y, z))	
22	Jerk (Variance (x, y, z))	
23	FFT (x)	
24	FFT (y)	
25	FFT (z)	
26	Sum of power of FFT (x, y, z)	

the size of the data matrix containing the handcrafted features to the size of the input data, i.e. 32 (height) x 32 (width) x 1 (number of channels), of the deep learning algorithm used in this study.

E. Deep learning network

In this study, to predict and classify ADLs, STs falls, and STs near-falls we used the modified interpretation of the LeNet-5 deep learning algorithm [29], which has a relatively high performance when dealing with variability of 2D data shapes, while having a simple architecture. The modified interpretation of the LeNet-5 convolutional neural network architecture used in this study consists of a total of 8 layers, including 3 convolutional layers, 2 subsampling layers, and 2 fully connected layers. The first layer is a convolution layer using 6 filters with a kernel size of 5x5, followed by a subsampling layer that performs average pooling with a pool size of 2x2 and a stride of 1. The output of this layer is then fed into another convolution layer that use 16 filters with a 5x5 kernel, followed by another subsampling layer. After that, the output of the previous layer is fed to another convolutional layer which has 120 filters with a kernel of size 5x5, followed by the last subsampling layer. Finally, the output of the last subsampling layer is flattened and fed to two fully connected layers, with the first layer having 84 neurons and the last layer having 5 neurons corresponding to 5 activity classes (ADLs, trip falls, slip falls, trip near-falls and slip near-falls). The modified LeNet-5 architecture uses a hyperbolic tangent activation function in the first three convolution layers and a SoftMax activation function in the output layer for classification. A more detailed visualization of the modified interpretation of the LeNet-5 architecture is shown in Fig. 3.

F. Performance measures

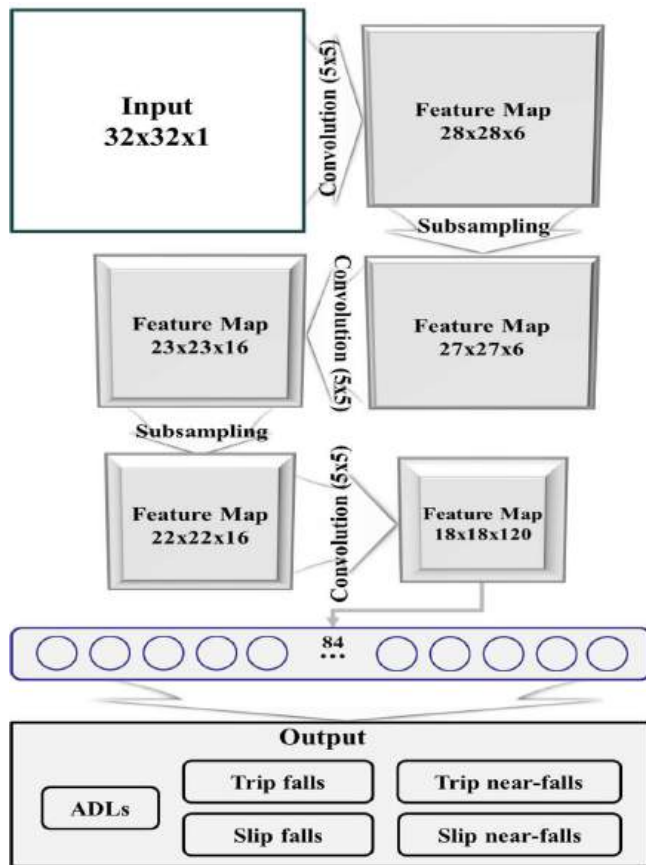


Fig. 3. Architecture of modified interpretation of LeNet-5 deep learning algorithm.

In this study, we used a 5-folds cross validation technique to validate the performance of the modified interpretation of the LeNet-5 deep learning algorithm. In total, 5 evaluation criteria such as accuracy, specificity, sensitivity, precision and F1-score were used to evaluate the performance of a deep learning model. A Windows based computer with AMD Ryzen 7 3700X 8-Core Processor @ 3.60 GHz, NVIDIA GeForce RTX 2080Ti 12GB and MATLAB (2022a, Mathworks, USA) were used for data processing, deep learning algorithm construction, training and evaluation.

III. RESULTS

After conducting 5-folds cross validation with 5 evaluation criteria, we obtained the performance results (Fig.

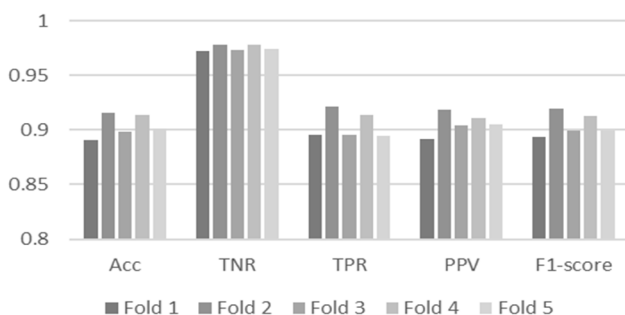


Fig. 4. Evaluation results of deep learning-based ADLs, STs falls and STs near-falls classification model. 5-folds cross-validation results: accuracy (Acc), specificity (TNR), sensitivity (TPR), precision (PPV) and F1-score.

4) of the modified interpretation of the LeNet-5 deep learning algorithm in predicting and classifying ADLs, trip falls, slip falls, trip near-falls, and slip near-falls.

The proposed deep learning-based ADLs, STs falls and STs near-falls prediction and classification method showed high performance results with average accuracy of 0.9039 ± 0.01 , specificity of 0.9753 ± 0.01 , sensitivity of 0.9043 ± 0.01 , precision of 0.9061 ± 0.01 and F1-score of 0.9049 ± 0.01 , respectively. It can also be seen from the confusion matrix shown in Figure 5 that the proposed method using a single IMU sensor is able to accurately classify ADLs, trip falls and slip falls, while it has experienced difficulty classifying trip near-falls and slip near-falls.

	ADLs	Trip Falls	Slip Falls	Trip Near-falls	Slip Near-falls
ADLs	92.2%	3.0%	1.7%	2.7%	6.5%
Trip Falls	0.8%	95.5%	1.7%	1.4%	1.3%
Slip Falls	1.6%	0%	96.6%	0%	0%
Trip Near-falls	1.6%	0%	0%	86.5%	9.1%
Slip Near-falls	3.9%	1.5%	0%	9.5%	83.1%

Fig. 5. Confusion matrix of a modified interpretation of the LeNet-5 deep learning algorithm obtained as a result of classification of ADLs, STs trips and STs near-falls using data from a single IMU sensor.

IV. DISCUSSION AND CONCLUSION

A. Discussion

A construction workplace environment has countless factors contributing to the frequent occurrence of trips and slips resulting in falls, which in turn are one of the leading causes of injury in a construction workplace [2, 4]. The highly accurate pre-impact STs falls detection algorithm can be effectively used in conjunction with wearable airbag systems, that can be utilized to prevent injury and various other negative consequences of STs falls. In addition, STs near-falls detection methods play an important role in providing a safe construction workplace environment. The high-performance STs near-falls detection algorithm can be used to monitor and timely eliminate various threats that cause falls and other harmful situations in a dynamic construction workplace. Hence, in order to eliminate the various problems associated with STs in the construction workplace, it is important to develop a method capable of not only detecting STs pre-impact falls but also tracking STs near-falls. Many studies have been done using robust and compact wearable sensors such as the IMU to detect pre-impact falls or near-falls [12]. Although studies using a large number of wearable sensors have shown a tendency to improve detection accuracy with more sensors used, however, utilizing of more sensors can be uncomfortable for the subject and may interfere with physical activity, especially in a dynamic construction workplace [14]. There have also been studies using sensors, such as an accelerometer and a gyroscope, built into mobile phones, to

detect falls or near-falls [11, 30]. However, the disadvantages of such systems are that not all mobile phones have built-in required sensors, the user must always remember to charge and carry the mobile phone correctly, and in case of a fall of the mobile phone, false positives may occur [30]. Therefore, in more recent studies, specially designed compact sensory devices, such as wearable IMU-based devices that can be integrated into work clothes or a wearable fall prevention airbag, are increasingly used to detect pre-impact falls or near-falls [11, 26]. In previous studies, in order to more accurately detect pre-impact falls or near-falls based on data obtained from a single wearable IMU device, various classification algorithms based on machine learning and deep learning techniques have often been used, which have higher performance than other classical methods, such as threshold-based method. However, most of the previous research has been heavily focused on detecting either only pre-impact STs falls or only STs near-falls in the construction workplace environment, and as far as we know there has not been a study to develop a method for detecting ADLs that include various basic types of activity, the pre-impact STs falls and STs near-falls in construction workplace. Therefore, the main goal of this study is to develop a high-performance method for detecting pre-impact STs falls and STs near-falls in a construction workplace environment using a single wearable sensory IMU device and a deep learning algorithm. The proposed method for predicting and classifying ADLs, including 8 types of normal activity, trip falls, slip falls, trip near-falls, and slip near-falls, demonstrated high performance with an accuracy of about 0.9039, a sensitivity of about 0.9043, and an F1-score of about 0.9049. The prediction and classification system developed in this study is highly resistant to false positives with a specificity of about 0.9753. From the confusion matrix in Fig. 5, we can conclude that the proposed modified interpretation of LeNet-5 deep learning algorithm, using handcraft features extracted from the single IMU sensor data, is able to accurately classify ADLs, trip falls, and slip falls, but has difficulty in identifying trip near-falls and slip near-falls. Due to the heterogeneous nature of near-falls and the various motion anomalies associated with a fairly large shift in the center of mass during the balance recovery phase, the prediction and classification of trip and slip near-falls of varying intensity, pre-impact trip and slip falls and ADLs that include various passive and active activities, as 5 separate classes is a particularly challenging task [7]. In addition, in this study due to the relatively small amount of STs falls and STs near-falls data artificially created in the simulation experiment, variation in the intensity of the STs near-falls simulation experiment from subject to subject, and the manual process of data labeling and feature extraction it was quite difficult to distinguish between trip near-falls and slip near-falls using single IMU sensor and lightweight deep learning algorithm.

One of the main limitations of this study was that the entire process of simulating STs falls and STs near-falls occurred in laboratory conditions with a pre-arranged scenario, which may differ from STs falls and STs near-falls occurring in real conditions of the construction work environment. In addition, the ADLs data used in this study included various passive and active activities encountered in daily life, and did not include various work activities occurring in the construction work environment. Various activities that occur in the construction work environment, especially types of work in the presence of intense vibration, such as drilling, can cause false alarms of

the proposed algorithm [32]. In addition, relatively little data was collected on ADLs, STs falls, and STs near-falls in this experiment. Since the performance of deep learning algorithms is highly dependent on the amount of data used, in order to achieve better classification results in future studies, it is recommended to collect more data with more participants. Hence, in future studies, it is recommended that an experiment be conducted to collect data from standard work frequently encountered in the construction work environment, including work in the presence of high vibration, and STs falls and STs near-falls data under real construction workplace with real workers as experimental subjects. Another limitation of this study is the use of a lightweight modified interpretation of LeNet-5 Convolutional Neural Network (CNN) deep learning algorithm. Although many different deep learning algorithms have been created, it has been shown that the performance of a deep learning algorithm is highly dependent on the quality and characteristics of the input training data [33]. However, in this study, we proposed a comprehensive methodology for the simultaneous prediction and classification of ADLs, trip falls, slip falls, trip near-falls, and slip near-falls. Therefore, future research needs to evaluate and compare performance using different deep learning models within this proposed process.

B. Conclusion

In conclusion, in this study, we have proposed a high-performance method for prediction and classification ADLs, STs falls and STs near-falls based on a deep learning algorithm and using a single wearable device with a built-in IMU sensor. Based on the processed signal of the 3-axis accelerometer and gyroscope, we extracted 30 features were, which were then utilized to train the deep learning algorithm. Modified interpretation of LeNet-5 CNN deep learning algorithm was used to predict and classify ADLs, trip falls, slip falls, trip near-falls and slip near-falls. The performance of the proposed method was evaluated using 5-folds cross validation with 5 evaluation criteria. This prediction and classification method showed high performance with an accuracy of about 0.9039, a specificity of about 0.9753, a sensitivity of about 0.9043, a precision of about 0.9061, and an F1-score of about 0.9049. Thus, the results of this study can be used to improve accuracy by reducing the number of false alarms of various systems for early detection and prevention of STs falls. In addition, it is believed that the proposed STs near-falls detection method can be used in the further development of the system of detection and the timely elimination of various dangerous anomalies in the construction workplace.

ACKNOWLEDGMENT

This work was supported by the Technology Innovation Program (Project No: 20016285) funded by the Ministry of Trade, Industry and Energy (MOTIE, Korea).

REFERENCES

- [1] A. Ardeshtir, M. Mohajeri, and M. Amiri, "Evaluation of safety risks in construction using fuzzy failure mode and effect analysis (FFMEA)," *Sci. Iran.*, vol. 23, no. 6, pp. 2546–2556, 2016.
- [2] H. J. Lipscomb, J. E. Glazner, J. Bondy, K. Guarini, and D. Lezotte, "Injuries from slips and trips in construction," *Appl. Ergon.*, vol. 37, no. 3, pp. 267–274, 2006.
- [3] T. A. Bentley and R. A. Haslam, "Identification of risk factors and countermeasures for slip, trip and fall accidents during the delivery of mail," *Appl. Ergon.*, vol. 32, no. 2, pp. 127–134, 2001.
- [4] M. F. Antwi-Afari, H. Li, J. Seo, S. Lee, D. J. Edwards, and A. Y. Lok Wong, "Wearable insole pressure sensors for automated detection and

classification of slip-trip-loss of balance events in construction workers,” In Construction Research Congress 2018, 2018.

- [5] Y. Shi, J. Du, C. R. Ahn, and E. Ragan, “Impact assessment of reinforced learning methods on construction workers’ fall risk behavior using virtual reality,” *Autom. Constr.*, vol. 104, pp. 197–214, 2019.
- [6] Hu, X., and Qu, X., “Pre-impact fall detection,” *Biomed. Eng. Online*, vol. 15, pp. 1–16.
- [7] A. Choi et al., “Deep learning-based near-fall detection algorithm for fall risk monitoring system using a single inertial measurement unit,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 30, pp. 2385–2394, 2022.
- [8] A. Weiss, I. Shimkin, N. Giladi, and J. M. Hausdorff, “Automated detection of near falls: algorithm development and preliminary results,” *BMC Res. Notes*, vol. 3, no. 1, p. 62, 2010.
- [9] M. Zhang, T. Cao, and X. Zhao, “Using smartphones to detect and identify construction workers’ near-miss falls based on ANN,” *J. Constr. Eng. Manag.*, vol. 145, no. 1, p. 04018120, 2019.
- [10] S. Sarkar, R. Raj, S. Vinay, J. Maiti, and D. K. Pratihari, “An optimization-based decision tree approach for predicting slip-trip-fall accidents at work,” *Saf. Sci.*, vol. 118, pp. 57–69, 2019.
- [11] T. K. Lim, S. M. Park, H. C. Lee, and D. E. Lee, “Artificial neural network-based slip-trip classifier using smart sensor for construction workplace,” *J. Constr. Eng. Manag.*, vol. 142, no. 2, p. 04015065, 2016.
- [12] S. C. Mukhopadhyay, “Wearable sensors for human activity monitoring: A review,” *IEEE Sens. J.*, vol. 15, no. 3, pp. 1321–1330, 2015.
- [13] H. Chander et al., “Wearable stretch sensors for human movement monitoring and fall detection in ergonomics,” *Int. J. Environ. Res. Public Health*, vol. 17, no. 10, p. 3554, 2020.
- [14] L. Joshua and K. Varghese, “Accelerometer-based activity recognition in construction,” *J. Comput. Civ. Eng.*, vol. 25, no. 5, pp. 370–379, 2011.
- [15] F. Wu, H. Zhao, Y. Zhao, and H. Zhong, “Development of a wearable-sensor-based fall detection system,” *Int. J. Telemed. Appl.*, vol. 2015, p. 576364, 2015.
- [16] I. T. Iliev, S. D. Tabakov, and A. I. Dotswinsky, “Automatic fall detection of elderly living alone at home environment,” *Global Journal of Medical research*, vol. 11, no. 4, pp. 49–54, 2011.
- [17] S. Chernbumroong, A. S. Atkins, and H. Yu, “Activity classification using a single wrist-worn accelerometer,” in 2011 5th International Conference on Software, Knowledge Information, Industrial Management and Applications (SKIMA) Proceedings, 2011.
- [18] S. D. Bersch, D. Azzi, R. Khusainov, I. E. Achumba, and J. Ries, “Sensor data acquisition and processing parameters for human activity classification,” *Sensors (Basel)*, vol. 14, no. 3, pp. 4239–4270, 2014.
- [19] W. C. Cheng and D. M. Jhan, “Triaxial accelerometer-based fall detection method using a self-constructing cascade-AdaBoost-SVM classifier,” *IEEE J. Biomed. Health Inform.*, vol. 17, no. 2, pp. 411–419, 2013.
- [20] A. T. Özdemir and B. Barshan, “Detecting falls with wearable sensors using machine learning techniques,” *Sensors (Basel)*, vol. 14, no. 6, pp. 10691–10708, 2014.
- [21] M. V. Albert, K. Kording, M. Herrmann, and A. Jayaraman, “Fall classification by machine learning using mobile phones,” *PLoS One*, vol. 7, no. 5, p. e36556, 2012.
- [22] D. Yacchirema, J. S. de Puga, C. Palau, and M. Esteve, “Fall detection system for elderly people using IoT and Big Data,” *Procedia Comput. Sci.*, vol. 130, pp. 603–610, 2018.
- [23] Y. Choi, A. S. Ralhan, and S. Ko, “A study on machine learning algorithms for fall detection and movement classification,” in 2011 International Conference on Information Science and Applications, 2011.
- [24] S. Yu, H. Chen, and R. A. Brown, “Hidden Markov model-based fall detection with motion sensor orientation calibration: A case for real-life home monitoring,” *IEEE J. Biomed. Health Inform.*, vol. 22, no. 6, pp. 1847–1853, 2018.
- [25] T. H. Kim, A. Choi, H. M. Heo, K. Kim, K. Lee, and J. H. Mun, “Machine learning-based pre-impact fall detection model to discriminate various types of fall,” *J. Biomech. Eng.*, vol. 141, no. 8, p. 081010, 2019.
- [26] M. Zhang, T. Cao, and X. Zhao, “Using smartphones to detect and identify construction workers’ near-miss falls based on ANN,” *J. Constr. Eng. Manag.*, vol. 145, no. 1, p. 04018120, 2019.
- [27] H. Lee, G. Lee, S. Lee, and C. R. Ahn, “Assessing exposure to slip, trip, and fall hazards based on abnormal gait patterns predicted from confidence interval estimation,” *Autom. Constr.*, vol. 139, no. 104253, p. 104253, 2022.
- [28] J. K. Lee, S. N. Robinovitch, and E. J. Park, “Inertial sensing-based pre-impact detection of falls involving near-fall scenarios,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 23, no. 2, pp. 258–266, 2015.
- [29] A. Ebrahimi, S. Luo, and Alzheimer’s Disease Neuroimaging Initiative, “Convolutional neural networks for Alzheimer’s disease detection on MRI images,” *J. Med. Imaging (Bellingham)*, vol. 8, no. 2, p. 024503, 2021.
- [30] A. Ramachandran and A. Karupiah, “A survey on recent advances in wearable fall detection systems,” *Biomed Res. Int.*, vol. 2020, p. 2167160, 2020.
- [31] K. Yang, C. R. Ahn, M. C. Vuran, and S. S. Aria, “Semi-supervised near-miss fall detection for ironworkers with a wearable inertial measurement unit,” *Autom. Constr.*, vol. 68, pp. 194–202, 2016.
- [32] S. Gelmini, S. Strada, M. Tanelli, S. Savaresi, and A. Guzzon, “Automatic detection of human’s falls from heights for airbag deployment via inertial measurements,” *Autom. Constr.*, vol. 120, no. 103358, p. 103358, 2020.
- [33] J. S. Sánchez, R. A. Mollineda, and J. M. Sotoca, “An analysis of how training data complexity affects the nearest neighbor classifiers,” *Pattern Anal. Appl.*, vol. 10, no. 3, pp. 189–201, 2007.

Deep reinforcement learning using a multi-scale agent with a normalized reward strategy for automatic cephalometric landmark detection

Woojae Hong

Department of Biomechatronic
Engineering

Sungkyunkwan University
Suwon, Republic of Korea
woojaehong94@gmail.com

Seong-Min Kim

Department of Biomechatronic
Engineering

Sungkyunkwan University
Suwon, Republic of Korea
seongminkim97@gmail.com

Joongyeon Choi

Department of Biomechatronic
Engineering

Sungkyunkwan University
Suwon, Republic of Korea
joongyeonchoi96@gmail.com

Jun-Young Paeng

Department of Oral and Maxillofacial
Surgery

Samsung Medical Center
Seoul, Republic of Korea
jypaeng@gmail.com

Joung Hwan Mun

Department of Biomechatronic
Engineering

Sungkyunkwan University
Suwon, Republic of Korea
jmun@skku.edu

Hyunggun Kim

Department of Biomechatronic
Engineering

Sungkyunkwan University
Suwon, Republic of Korea
hkim.bme@skku.edu

Abstract—Accurate detection of cephalometric landmarks is necessary for precise analysis, diagnosis, and surgical planning. Many studies on automated landmark detection have been conducted including machine learning, convolutional neural network, and reinforcement learning. In this study, deep Q network (DQN) with a normalized reward strategy in multi-scale agent was employed for automated cephalometric landmark detection. The performance of the standard DQN and DQN with a normalized reward strategy were evaluated using the IEEE International Symposium of Biomedical Imaging (ISBI) 2015 Challenge dataset and compared with previously proposed methods. Our data demonstrated that the DQN with a normalized reward strategy using a multi-scale agent achieved the top rank among recently published methods, with an average mean radius error of 1.2 mm and a success detection ratio of 82.34%. These findings indicate that the normalized reward strategy improved the performance of cephalometric landmark detection.

Keywords—cephalometric landmark detection, deep Q network, normalized reward

I. INTRODUCTION

Cephalometric analysis using 3D images is being actively studied due to the development of medical imaging equipment, however 2D cephalometric analysis, which is conducted from 2D lateral cephalogram, is still predominantly used in clinics [1, 2]. The 2D cephalometric analysis is based on the identification of specific landmarks to quantify the oral structure and jaw joint, with potential misclassification resulting from even slight deviations in landmark location due to the diagnostic concentration within millimeter and degree ranges [3, 4]. The degree of variability in this regard is contingent on the specific location of the identified landmarks. Therefore, accurate landmark detection is crucial for achieving accurate analysis, diagnosis, and surgical planning.

Accurate cephalometric landmark detection is a complicated and time-consuming task despite high-resolution images. The 2D X-ray cephalogram is taken by

projecting lateral view of skull, which is a highly sophisticated 3D object [5]. Each time the patient's head is not aligned correctly, it is taken differently. Although cephalometric landmark detection is generally conducted by trained orthodontists in clinical practice, the intra- and inter-observer variabilities occur due to the characteristics of the 2D lateral cephalogram [2, 6]. Under these circumstances, computer-aided, accurate, and fast landmark detection is highly desirable.

Many studies on automated landmark detection for cephalometric analysis have been conducted. Especially, many people participated in the IEEE International Symposium of Biomedical Imaging (ISBI) Challenge in 2014 and 2015, and various algorithms and frameworks were proposed [7, 8]. Ibragimov et al. tried to acquire the optimal landmark position by constructing an integrated framework that combined a tree-structured random forest classifier, statistical shape representation defined by the Gaussian kernel estimation, and game-theoretic optimization [9-11]. Lindner and Cootes used random forest regression-voting to roughly predict the location of each landmark, and applied a statistical shape model to the location of all landmarks to ensure consistency across all datasets [12, 13]. However, despite these efforts, the proposed methods had limitations, showing approximately 70% within the clinically acceptable threshold margin of 2 mm [8].

The performance of automated landmark detection for cephalometric analysis has been improved through the utilization of convolution neural network (CNN), which marks the inception of groundbreaking developments of machine learning in the fields of image analysis [14]. CNN enables the accurate localization of landmarks by efficiently extracting features of medical images [15]. Researchers have made several attempts to combine statistical shape models with CNN or adopt a U-shaped deep CNN structure [16, 17]. Furthermore, deep reinforcement learning (DRL) or deep Q network (DQN) based on reinforcement learning (RL) in conjunction with CNN has shown promise for 3D cephalometric landmark detection. Kang et al. proposed an

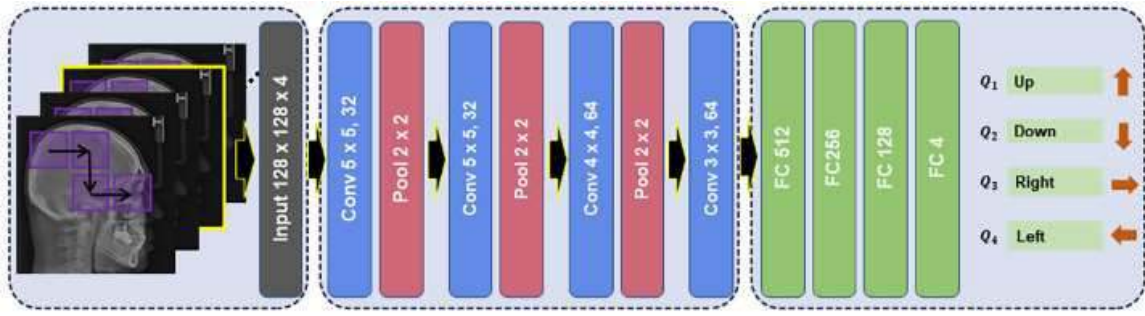


Fig. 1. A schematic of the proposed DQN-based network architecture for automated cephalometric landmark detection.

automatic 3D cephalometric annotation system based on multi-stage DRL and volume-rendered imaging [18]. Gillot et al. presented conventional DRL in 3D cone-beam computed tomography (CBCT) and trained 119 landmarks that commonly used in clinical studies for the quantification of changes in bone morphology and tooth position [19]. In DQN, the reward function is the most crucial factor and usually defined as the difference between the Euclidean distances from the previous location to the target landmark and from the current location to the target landmark. The standard reward clipping strategy for DQN in landmark detection involves constraining the calculated rewards between -1 and 1, and then importing the clipped rewards to DQN.

The present study investigated the application of a multi-scale agent employing a DQN with a normalized reward strategy for 2D cephalometric landmark detection, in comparison with the standard DQN and previously published methods. The standard DQN was trained with the default parameters suggested by Alansary et al. [20], whereas the DQN with the normalized reward strategy was trained with reward normalization instead of clipping. Normalization of rewards at each scale aimed to ensure consistent learning of actions at all scales equally, resulting in accurate landmark detection.

II. MATERIALS AND METHODS

The overview of the proposed DQN-based network for automated landmark detection is summarized in Fig. 1. For automated landmark detection, a full-size of cephalogram is set to environment, the state is derived by defining the $n \times n$ pixel data of the cephalogram, the agent is located at the center of the state, the action is defined to shift by pixels along the pixel axis, the reward defines the Euclidean distance to the target landmark, and the agent is designed to reach the target landmark. To effectively interpret a large size of state and select action, the multi-scale agent and dilated convolution strategies are applied to the primary DQN-based network.

A. Q-learning

RL is a method of learning in a way that the agent selects action to receive the maximum reward through the interaction between the agent and the environment. The agent selects an action by interpreting the state of the environment, and then the environment gives the agent a changed state and an immediate reward with respect to the agent's choice. The sum of rewards that can be received in the future is determined by the agent's choice of successive actions, and the agent learns to select the actions that can receive the largest accumulated rewards. The state-action value function $Q(s, a)$ is defined as the expected value of

the sum of discounted future rewards $E[r_{t+1} + \gamma r_{t+2} + \dots + \gamma^{n-1} r_{t+n} | s, a]$ where s , a , and r are the state, action, and reward, respectively, and $\gamma \in [0, 1]$ is a discount factor used to weight future rewards. The Q-function is defined as below, and can be solved recursively, so it can be solved iteratively.

$$Q_{i+1}(s, a) = E[r + \gamma \max_{a'} Q_i(s', a')] \quad (1)$$

where s' and a' are the next state and action, respectively.

B. DQN for Landmark Detection

The DQN-based network is configured to extract features from an image through CNN, define the image as a state, and select the next action determined through the Q-learning network $Q(s, a) \approx Q(s, a; \theta)$. The loss function of the algorithm is as follows:

$$L_{DQN}(\theta) = E_{s,a,r,s'} \left[\left(r + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta) \right)^2 \right] \quad (2)$$

where θ are the parameters of the current network $Q(\theta)$, θ^- are the parameters of the target network $Q(\theta^-)$, and s' denotes a new state exhibited by the changed environment when action a was performed in state s . The approximation of best parameters θ^* can be learned by stochastic gradient descent of the above loss function.

As DQN-based networks take consecutive frames as input, it is likely to involve learning bias due to the high correlation between the input data [21]. In order to prevent learning bias, the experience replay strategy is applied such that four consecutive frames representing the action and state of the agent $[s, a, r, s']$ are randomly selected and stored sequentially as much as the pre-allocated memory. Furthermore, the non-stationary target problem arises due to the repeatedly changed target $Q(\theta^-)$ network whenever the current $Q(\theta)$ network approximates the target value every step. Mnih et al. proposed a fixed Q-target strategy, whereby the $Q(\theta^-)$ value remains fixed during learning by separating the target $Q(\theta^-)$ network to reduce the instability of learning associated with this problem [21]. In addition, they employed the ϵ -greedy policy in which an agent randomly selects an action with probability ϵ during the learning process to avoid biased action selection. These techniques helped to minimize instability, facilitate diverse behavior selection during learning, and promote unbiased learning.

C. RL Agent Definition for Landmark Detection

The agent interacts with the environment to initiate movement by executing a set of four actions, $[\pm a_x, \pm a_y]$, where the direction of each action is defined along the positive and negative axes of the pixel grid. Each action step

involves shifting the scalar value of pixels along the corresponding axis.

The state is defined as the pixel data of size $n \times n$ in which the agent is located. The input data consists of four frames obtained from the actions continuously selected by the agent. The reward is determined by the Euclidean distance between the current state and the target landmark. However, this metric disregards the extent of progress made from the previous state to the present state. To address this limitation, a modified reward function that incorporates both the current and previous state distances is defined as follows:

$$R = D(P_{i-1}, P_t) - D(P_i, P_t), \quad (3)$$

where P_i is the agent's coordinates at step i , and P_t is the ground truth value of the target landmark.

The terminal state is reached when the agent has found the target landmark. The conditions for terminal state are divided into the learning and validation processes. In the learning process, the terminal state is defined when the agent reaches the target landmark below the pre-defined epsilon value. In the evaluation process, it is designed to be terminated when the agent oscillates.

The action step, limited to one pixel size, will inevitably take more time if the distance from the ground truth landmark is substantial. In order to avoid this problem, multi-scale agent and dilated convolution strategies are employed in a coarse-to-fine fashion [20, 22]. The agent moves rapidly converging to a target landmark through global structure information and moves in detail based on the features of a more specific area. The initial action step of the agent and dilated size of ROI are defined by the multi-scale agent parameter k . For instance, the environment is sampled through a fixed size image-grid with initial spacing (k_x, k_y) pixel around the current point of interest, and the agent searches for the target landmark using higher initial action steps m^k . If the agent oscillates, the environment sampling reduces in spacing to $((k-1)_x, (k-1)_y)$, and the agent uses smaller action steps m^{k-1} . This procedure is repeated until the target landmark is reached, which is marked by the multi-scale parameter k reaching one.

The reward values of a multi-scale agent can vary depending on the size of the action step. To maintain a consistent range, the standard method of reward clipping approach confines rewards within the bounds of -1 and 1. However, we employed a normalized reward strategy in this study to promote uniform learning across all scales. Specifically, our normalization method involves scaling rewards at action steps m^k to the range of $[-m^k, m^k]$, which are then normalized to $[-1, 1]$ using the normalization equation:

$$[a \ b] = \frac{(b-a)(x-x_{\min})}{x_{\max}-x_{\min}} + a \quad (4)$$

D. Implementation

Experiments were carried out utilizing an Intel Core i9-10940X CPU running at 3.30 GHz and 64 GB of memory, along with two NVIDIA TITAN RTX 24 GB GPUs connected through NVLink, providing a total capacity of 48 GB. Tensorflow environment was employed for training and validation. The input image size for the network was set at 128, with a multi-scale agent of 3, an action step of 16, a

batch size of 48, and an experience replay memory of size of $1e5$. PReLU was utilized as the activation function for convolutional layers, and leakyReLU for fully connected layers. For optimization, adam optimizer was employed with a learning rate of $\gamma = 0.9$, and $\varepsilon = 0.9-0.1$. The training process took 12 hours per landmark, and model performance was evaluated every 2 epochs, with a success ratio exceeding 0.95 considered to indicate over-fitting, leading to termination of training.

E. Dataset

The present investigation aimed to evaluate the effectiveness of the proposed DQN-based network in analyzing the cephalogram dataset provided for the ISBI 2015 Grand Challenges. This cephalogram dataset comprised images obtained from 400 patients using a Soredex CRANEX Excel Ceph unit (Tusula, Finland) and Soredex SorCom software (3.1.5, Version 2.0, Tusula, Finland). The images had a resolution of $1,935 \times 2,400$ pixels and an image spacing of 0.1 mm/pixel. For the detection challenge, a total of 19 landmarks in each of the 400 cephalogram images were annotated by two experienced clinicians. The ground truth for each landmark was defined as the average of the points identified by the two clinicians. The dataset was partitioned into three subsets: 150 images for training, 150 images for Test 1, and 100 images for Test 2.

F. Evaluation Metrics

Using the ISBI 2015 dataset, the performance of our DQN-based automated cephalometric landmark detection algorithm was evaluated employing the same evaluation methods as those utilized in the ISBI 2015 Challenge.

In the i^{th} landmark, the radical error (RE) in the j^{th} image was defined as the Euclidean distance between the predicted location and the ground truth location. The mean radial error (MRE) and the standard deviation (SD) of the i^{th} landmark were defined as follows in all the images:

$$MRE_i = \frac{\sum_{j=1}^M RE_i^j}{M}, \quad SD_i = \sqrt{\frac{\sum_{j=1}^M (RE_i^j - MRE_i)^2}{M}} \quad (5)$$

where M is the number of images.

Successful detection of a landmark is achieved when RE did not exceed δ mm. The success detection rate (SDR) for δ mm was defined as follows:

$$SDR_\delta = \frac{\#(\{\hat{y}_i: \|\hat{y}_i - y_i^{gt}\|_2 \leq \delta\})}{M} \quad (6)$$

where $\hat{y}_i$ is the detected landmark coordinate, y_i^{gt} is the ground truth landmark coordinate, M is the number of predicted images, and $\#(\cdot)$ refers to the counting function. SDR indicates the ratio between the number of detected images within a distance of δ mm and the total number of test images.

III. RESULTS

A. Landmark Detection with the Standard Reward Clipping Strategy

The performance outcomes of the DQN-based network for landmark detection with the standard reward clipping strategy are presented in Table I. The mean radius error

Table I. Performance of the DQN-based network for landmark detection with the standard reward clipping strategy in terms of MRE and SDR within 2.0, 2.5, 3.0, and 4.0 mm neighborhoods for the test dataset.

Test 1					
Number	MRE	2.0 mm	2.5 mm	3.0 mm	4.0 mm
1	1.04	92.00	94.00	95.33	96.67
2	1.63	75.33	84.67	88.67	94.00
3	1.41	80.00	88.00	90.67	96.67
4	2.36	51.33	64.00	69.33	84.67
5	2.25	58.00	66.00	75.33	86.67
6	1.50	77.33	84.00	91.33	96.00
7	1.01	89.33	98.00	98.67	99.33
8	0.93	94.00	97.33	98.67	98.67
9	1.12	87.33	90.67	96.00	99.33
10	2.20	52.00	66.67	83.33	92.67
11	1.21	88.00	90.67	94.67	98.00
12	1.22	85.33	90.00	92.67	94.00
13	1.15	92.00	96.00	97.33	97.33
14	0.98	95.33	96.67	97.33	97.33
15	1.47	82.00	86.00	90.00	92.67
16	2.02	60.67	73.33	85.33	94.00
17	1.13	86.00	94.00	96.67	98.00
18	1.75	67.33	82.67	88.67	94.00
19	2.00	62.00	76.67	80.67	89.33
Average	1.49	77.65	85.23	90.04	94.70
Test 2					
Number	MRE	2.0 mm	2.5 mm	3 mm	4 mm
1	1.09	94.00	95.00	97.00	98.00
2	1.37	77.00	87.00	90.00	94.00
3	1.79	71.00	82.00	85.00	95.00
4	1.88	68.00	81.00	83.00	90.00
5	1.79	61.00	81.00	88.00	96.00
6	2.52	40.00	53.00	69.00	85.00
7	1.02	91.00	98.00	99.00	100.0
8	0.76	98.00	100.0	100.0	100.0
9	0.87	93.00	99.00	99.00	99.00
10	2.07	52.00	69.00	79.00	90.00
11	1.05	87.00	90.00	94.00	97.00
12	0.99	84.00	89.00	91.00	95.00
13	1.84	64.00	77.00	84.00	92.00
14	1.45	56.00	75.00	87.00	94.00
15	1.52	71.00	82.00	91.00	94.00
16	3.70	14.00	27.00	48.00	69.00
17	1.53	75.00	84.00	88.00	96.00
18	1.55	69.00	85.00	95.00	100.0
19	1.62	74.00	82.00	87.00	93.00
Average	1.60	70.47	80.84	87.05	93.53

(MRE) for the 19 landmarks in Test 1 dataset was found to be 1.49 mm, which is superior to the performance of the Lindner and Cootes team (1.67 ± 1.65 mm) in the ISBI 2015 Challenge. Similarly, the average MRE for the 19 landmarks in Test 2 dataset was 1.60 mm, which is also lower than the performance of the Lindner and Cootes team (1.92 mm). The success detection ratio (SDR) within 2 mm was 77.65% in Test 1 dataset and 70.47% in Test 2 dataset, which outperforms the other methods published in 2015 and 2016 [8]. However, it is worth noting that the DQN-based approach with standard reward clipping strategy demonstrated lower accuracies compared to the recently published methods.

B. Landmark Detection with the Normalized Reward Strategy

The performance outcomes of the DQN-based network for landmark detection with the normalized reward strategy are presented in Table II. For the 19 landmarks tested, the average MRE was 1.12 mm and 1.28 mm in the Test 1 and Test 2 datasets, respectively, with the SDR within 2 mm of

Table II. Performance of the DQN-based network for landmark detection with the normalized reward strategy in terms of MRE and SDR within 2.0, 2.5, 3.0, and 4.0 mm neighborhoods for the test dataset

Test 1					
Number	MRE	2.0 mm	2.5 mm	3.0 mm	4.0 mm
1	0.58	98.00	98.00	98.00	98.67
2	1.22	85.33	90.67	92.67	94.67
3	1.21	86.00	90.67	96.00	100.0
4	1.75	57.33	64.00	72.00	85.33
5	1.63	69.33	82.00	88.00	97.33
6	1.23	83.33	91.33	92.67	95.33
7	0.92	91.33	96.67	97.33	98.67
8	0.84	94.00	98.67	98.67	100.0
9	0.85	93.33	96.67	98.00	100.0
10	1.88	61.33	72.67	83.33	92.67
11	0.83	90.00	93.33	94.67	98.00
12	0.53	94.00	97.33	98.67	98.67
13	1.18	93.33	98.67	100.0	100.0
14	0.76	99.33	100.0	100.0	100.0
15	0.81	92.67	96.00	98.00	100.0
16	1.37	85.33	91.33	93.33	97.33
17	0.81	95.33	98.00	98.00	99.33
18	1.23	84.67	89.33	93.33	98.00
19	1.67	66.00	77.33	84.67	92.67
Average	1.12	85.26	90.67	93.54	97.19
Test 2					
Number	MRE	2.0 mm	2.5 mm	3 mm	4 mm
1	0.52	99.00	99.00	100.0	100.0
2	0.85	88.00	94.00	97.00	98.00
3	1.92	60.00	78.00	82.00	98.00
4	1.58	78.00	86.00	87.00	90.00
5	1.44	78.00	86.00	94.00	100.0
6	2.06	56.00	64.00	76.00	94.00
7	0.60	99.00	100.0	100.0	100.0
8	0.58	99.00	99.00	100.0	100.0
9	0.59	100.0	100.0	100.0	100.0
10	1.70	67.00	75.00	85.00	93.00
11	0.80	92.00	94.00	96.00	98.00
12	0.42	96.00	97.00	97.00	100.0
13	1.98	56.00	64.00	76.00	98.00
14	1.71	76.00	87.00	94.00	99.00
15	0.87	94.00	97.00	100.0	100.0
16	3.57	9.00	17.00	40.00	64.00
17	1.01	90.00	96.00	99.00	99.00
18	1.01	91.00	96.00	99.00	100.0
19	1.15	81.00	92.00	97.00	99.00
Average	1.28	79.42	85.32	90.47	96.32

85.26% and 79.24%, respectively. These data demonstrate higher accuracy than that of the DQN network with the standard reward clipping strategy, as well as those reported in recent studies. Employing a multi-scale agent and normalizing rewards with action steps enhanced accuracy, with a decrease of 0.37 mm and 1.32 mm in the average MRE and an increase of 4.69% and 8.95% in the SDR within 2 mm in the Test 1 and Test 2 datasets, respectively, compared with the DQN utilizing the standard reward clipping strategy. Notably, three landmarks, namely upper lip (in Test 1, and 2), lower lip (in Test 2), and orbitale (in Test 2), exhibited a higher MRE. Performance of the proposed method was compared with other detection algorithms [16, 23-29] in Table III in terms of MRE and SDR in the ISBI 2015 Challenge dataset, including our DQN networks. Our findings revealed that the DQN with the normalized reward outperformed other methods in terms of SDR within 2 mm and achieved the lowest MRE in Test II. Overall, the proposed method exhibited superior accuracy compared to other methods that showed comparable performance, despite a slightly higher SDR.

Table III. Performance of the DQN-based networks for landmark detection in comparison with the previously proposed algorithms in the ISBI 2015 Challenge.

Model	Test 1					Test 2				
	MRE	2.0 mm	2.5 mm	3.0 mm	4.0 mm	MRE	2.0 mm	2.5 mm	3.0 mm	4.0 mm
Arik et al.	-	75.37	80.91	84.32	88.25	-	67.68	74.16	79.11	84.63
Qian et al.	1.28	82.51	86.25	89.30	90.62	1.54	72.40	76.15	79.65	85.90
Chen et al.	1.17	86.67	92.67	95.54	98.53	1.48	75.05	82.84	88.53	95.05
DRM	1.12	86.91	91.82	94.88	97.90	1.42	76.00	82.90	88.74	94.32
Cascaded	1.34	81.37	89.09	93.79	97.86	1.64	70.58	79.53	86.05	93.32
Zhou et al.	-	-	-	-	-	1.39	76.11	84.21	88.79	94.84
Swin-CE	1.13	87.31	92.93	95.87	98.74	1.44	76.23	83.36	89.15	95.24
GU2Net	1.12	87.21	92.55	95.6	97.7	1.37	74.11	85.05	89.39	94.04
SCN	1.34	81.47	89.36	93.15	97.01	1.65	69.94	78.84	85.74	93.89
DACFL	1.18	86.20	91.20	94.41	97.68	1.47	75.92	83.44	89.27	94.73
DQN / standard reward clipping	1.49	77.65	85.23	90.04	94.70	1.60	70.47	80.84	87.05	93.53
DQN / normalized reward	1.12	85.26	90.67	93.54	97.19	1.28	79.24	85.32	90.47	96.32

IV. DISCUSSION

This study introduces a novel approach to automated cephalometric landmark detection using DQN-based networks with a normalized reward strategy. The performance of two DQN-based networks for cephalometric landmark detection was evaluated using the ISBI 2015 Challenge dataset. The potential of the normalized reward strategy in a multi-scale agent was compared with previously published detection methods. The standard DQN model outperforms the other methods in the ISBI 2015 Challenge, while comparable outcomes were found with recently proposed methods. The DQN utilizing the standard reward clipping strategy demonstrated an average MRE of 1.49 mm and 1.60 mm for 19 landmarks in Test 1 and Test 2 dataset, respectively, which falls within the clinically accepted range. Furthermore, implementing a normalized reward strategy in a multi-scale agent improved the performance of the standard DQN model by yielding a lower average MRE and higher SDR. This approach outperforms the previously published methods and achieves a high rank.

The standard reward clipping strategy used in multi-scale agents exhibited suboptimal performance in landmark detection. This is attributed to the large displacement that occurs when an agent takes a high action step. The reward for each step is determined as the difference between the Euclidean distance from the previous agent location to the target landmark and from the current agent location to the target landmark. Therefore, early actions in an episode, which tend to have larger values, are of greater importance in multi-scale agents. In the process of learning, the agent aims to maximize the total amount of reward obtained. However, the reward obtained for actions performed in close proximity to the target landmark, characterized by low action steps, is lower than that for actions taken at larger action steps. Consequently, the training process fails to capture fine-grained information, indicating the insufficiency of the standard reward clipping strategy for achieving detailed and accurate performance in landmark detection. To address this issue and ensure uniform training across all scales, a normalized reward strategy was proposed. This strategy involves normalization of rewards across all action steps to the range $[-1, 1]$, enabling equivalent reward application at all steps and considerably enhancing the performance of cephalometric landmark detection.

The use of image data in RL can lead to partial observation, which poses a challenge for the DQN-based network that receives four fixed-size image grids as inputs

from cephalogram data. The conventional cephalogram has large resolution causing the partial observation problem by not seeing the entire image at once. To address this issue, the DQN approach developed by DeepMind employed a technique of stacking frames from the Atari simulator [30]. Rather than simply inputting a single frame into the network, the last four frames of the game were stacked and stored in the buffer memory before being passed to the network. Recent studies have employed the DQN framework of DeepMind in an attempt to improve landmark detection in medical image data. Specifically, these studies have focused on brain, heart, and fetal image data, utilizing a combination of multi-scale agent and dilated convolution strategies to increase learning efficiency and enhance landmark detection accuracy while minimizing the impact of partial observation [20]. These efforts are a crucial step towards improving the accuracy and reliability of landmark detection in medical imaging, with potential implications for diagnosis and treatment planning.

The framework of DQN involves extracting a fixed-size image grid and storing it in memory before inputting it to the CNN. Various hyperparameters are utilized within the DQN-based network, including the ROI size, area to be extracted, starting point of agent, and multi-scale agent parameter. These hyperparameters can have a significant impact on the performance of landmark detection, with larger ROI sizes requiring more computing resources and resulting in less efficient learning. During learning, images are randomly extracted from a pre-defined area and stored in buffer memory, highlighting the importance of this area for efficient learning and optimal performance. Given the dense distribution of cephalometric landmarks in the maxillofacial area, this area is of particular importance for efficient learning and accurate performance. The starting point of the agent, which is also randomly selected, has been found to impact accuracy. Additionally, the multi-scale agent strategy has an influence on landmark detection accuracy in large-sized medical image data in a coarse-to-fine fashion, with the degree of dilation and size of the action step affecting performance. Dilated approaches have been found to improve accuracy for landmarks such as maxilla 1 distal.

The efficacy of different DQN hyperparameters was evaluated in the context of automated cephalometric landmark detection, utilizing an open-source network with pre-defined image and action step sizes without further optimization. The DQN-based cephalometric landmark detection with the standard reward clipping strategy revealed suboptimal performance, as reported in recent

studies using the ISBI 2015 Challenge dataset. However, the proposed normalized reward strategy clearly improved landmark detection performance and exhibited superior accuracy compared to recently reported models.

V. CONCLUSIONS

In the present study, we utilized a DQN-based approach with a multi-scale agent and normalized reward strategy for automated cephalometric landmark detection. Two DQN models with different reward strategies were evaluated using the ISBI 2015 Challenge dataset and compared to recently published methods. Our findings indicate that the DQN model with the normalized reward strategy yielded superior performance in terms of MRE and SDR compared to both the standard DQN model and other methods. Despite these positive results, optimizing hyperparameters and episode construction remains an ongoing area of focus for continued improvement.

REFERENCES

- [1] A. Kaur and C. Singh, "Automatic cephalometric landmark detection using Zernike moments and template matching," *Signal Image Video P*, vol. 9, pp. 117-132, Jan 2015.
- [2] W. Yue, D. Yin, C. Li, G. Wang, and T. Xu, "Automated 2-D cephalometric analysis on X-ray images by a model-based approach," *IEEE Trans Biomed Eng*, vol. 53, pp. 1615-23, Aug 2006.
- [3] M. H. J. Doff, K. J. Finnema, A. Hoekema, P. J. Wijkstra, L. G. M. de Bont, and B. Stegenga, "Long-term oral appliance therapy in obstructive sleep apnea syndrome: a controlled study on dental side effects," *Clin Oral Investig*, vol. 17, pp. 475-482, 2013.
- [4] C. S. da Fontoura et al., "Candidate gene analyses of skeletal variation in malocclusion," *J Dent Res*, vol. 94, pp. 913-920, Jul 2015.
- [5] A. Kamoen, L. Dermaut, and R. Verbeeck, "The clinical significance of error measurement in the interpretation of treatment results," *Eur J Orthod*, vol. 23, pp. 569-78, Oct 2001.
- [6] A. P. R. Durão, A. Morosolli, P. Pittayapat, N. Bolstad, A. P. Ferreira, and R. Jacobs, "Cephalometric landmark variability among orthodontists and dentomaxillofacial radiologists: a comparative study," *Imaging Sci Dent*, vol. 45, pp. 213-220, 2015.
- [7] C. W. Wang et al., "Evaluation and comparison of anatomical landmark detection methods for cephalometric X-ray images: a grand challenge," *IEEE Trans Med Imaging*, vol. 34, pp. 1890-900, Sep 2015.
- [8] C. W. Wang et al., "A benchmark for comparison of dental radiography analysis algorithms," *Med Image Anal*, vol. 31, pp. 63-76, Jul 2016.
- [9] B. Ibragimov, B. Likar, F. Pernus, and T. Vrtovec, "A game-theoretic framework for landmark-based image segmentation," *IEEE Trans Med Imaging*, vol. 31, pp. 1761-1776, Sep 2012.
- [10] B. Ibragimov, B. Likar, F. Pernus, and T. Vrtovec, "Shape representation for efficient landmark-based segmentation in 3-d," *IEEE Trans Med Imaging*, vol. 33, pp. 861-874, Apr 2014.
- [11] B. Ibragimov et al., "Segmentation of tongue muscles from super-resolution magnetic resonance images," *Med Image Anal*, vol. 20, pp. 198-207, Feb 2015.
- [12] C. Lindner, P. A. Bromiley, M. C. Ionita, and T. F. Cootes, "Robust and accurate shape model matching using random forest regression-voting," *IEEE Trans Pattern Anal Mach Intell*, vol. 37, pp. 1862-1874, 2015.
- [13] C. Lindner, S. Thiagarajah, J. M. Wilkinson, G. A. Wallis, and T. F. Cootes, "Fully automatic segmentation of the proximal femur using random forest regression voting," *IEEE Trans Med Imaging*, vol. 32, pp. 1462-1472, Aug 2013.
- [14] G. Litjens et al., "A survey on deep learning in medical image analysis," *Med Image Anal*, vol. 42, pp. 60-88, Dec 2017.
- [15] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436-444, May 2015.
- [16] S. Arık, B. Ibragimov, and L. Xing, "Fully automated quantitative cephalometry using convolutional neural networks," *J Med Imaging*, vol. 4, p. 014501, Jan 2017.
- [17] E. N. D. Goutham, S. Vasamsetti, P. V. V. Kishore, and H. K. Sardana, "Automatic localization of landmarks in cephalometric images via modified U-Net," in *2019 International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1-6, Jul 2019.
- [18] S. H. Kang, K. Jeon, S. H. Kang, and S. H. Lee, "3D cephalometric landmark detection by multiple stage deep reinforcement learning," *Sci Rep*, vol. 11, p. 17509, Sep 2021.
- [19] M. Gillot et al., "Automatic landmark identification in cone-beam computed tomography," *Orthod Craniofac Res*, Feb 2023.
- [20] A. Alansary et al., "Evaluating reinforcement learning agents for anatomical landmark detection," *Med Image Anal*, vol. 53, pp. 156-164, 2019.
- [21] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529-533, 2015.
- [22] F. C. Ghesu et al., "Multi-scale deep reinforcement learning for real-time 3D-landmark detection in CT scans," *IEEE Trans Pattern Anal Machine Intell*, vol. 41, pp. 176-189, 2019.
- [23] R. N. Chen, Y. X. Ma, N. L. Chen, D. Lee, and W. P. Wang, "Cephalometric landmark detection by attentive feature pyramid fusion and regression-voting," *Lect Notes Comput Sc*, vol. 11766, pp. 873-881, 2019.
- [24] J. H. Qian, M. Cheng, Y. B. Tao, J. Lin, and H. Lin, "CephaNet: An improved Faster R-CNN for cephalometric landmark detection," *I S Biomed Imaging*, pp. 868-871, 2019.
- [25] C. Payer, D. Stern, H. Bischof, and M. Urschler, "Integrating spatial configuration into heatmap regression based CNNs for landmark localization," *Medical Image Analysis*, vol. 54, pp. 207-219, May 2019.
- [26] K. Oh, I. S. Oh, V. T. Le, and D. W. Lee, "Deep anatomical context feature learning for cephalometric landmark detection," *IEEE J Biomed Health*, vol. 25, pp. 806-817, Mar 2021.
- [27] M. M. Zeng, Z. L. Yan, S. Liu, Y. H. Zhou, and L. X. Qiu, "Cascaded convolutional networks for automatic cephalometric landmark detection," *Medical Image Analysis*, vol. 68, Feb 2021.
- [28] Z. S. Zhong, J. Li, Z. X. Zhang, Z. C. Jiao, and X. B. Gao, "An attention-guided deep regression model for landmark detection in cephalograms," *Medical Image Computing and Computer Assisted Intervention - MICCAI 2019, Pt Vi*, vol. 11769, pp. 540-548, 2019.
- [29] G. Q. Zhou et al., "Learn fine-grained adaptive loss for multiple anatomical landmark detection in medical images," *IEEE J Biomed Health*, vol. 25, pp. 3854-3864, Oct 2021.
- [30] V. Mnih et al., "Playing Atari with deep reinforcement learning," *arXiv*, vol. 1312.5602, 2013.

IMPLEMENTATION OF THE CNN-SVM HYBRID MODEL FOR LEUKOCORIA DETECTION

Siti Umami Masruroh

Department of Informatics
UIN Syarif Hidayatullah
Jakarta, Indonesia
ummi.masruroh@uinjkt.ac.id

Muhammad Hugo Athallah Hardy

Department of Informatics
UIN Syarif Hidayatullah
Jakarta, Indonesia
mhugoathalla@gmail.com

Khodijah Hullyyah

Department of Informatics
UIN Syarif Hidayatullah
Jakarta, Indonesia
khodijah.hullyyah@uinjkt.ac.id

Rosa Adelina

Department of Pharmaceutics
UIN Syarif Hidayatullah
Jakarta, Indonesia
rosa.adelina@uinjkt.ac.id

Rizka Amalia Putri

Department of Informatics
UIN Syarif Hidayatullah
Jakarta, Indonesia
rizka.amalia18@mhs.uinjkt.ac.id

Abstract—Retinoblastoma (Rb) is a tumor that is a primary intraocular malignancy of the retina that generally affects infants and children. The most common symptom of Rb is a white reflection emitted from the retina of the eye through the pupil called leukocoria. This disease must be treated earlier, otherwise, it can lead to dangerous diseases. Based on this, an early diagnosis will be very important, and this will be done by detecting leukocoria. With the development of technology, image classification has become easier. Convolutional Neural Network (CNN) is an advanced image processing algorithm, and Support Vector Machine (SVM) is an advanced algorithm for general classification. For this reason, the authors propose a CNN-SVM hybrid method to classify eyes with leukocoria. The model with the CNN-SVM hybrid method will extract features from the eye photographic image on the CNN layer and perform the final classification using the SVM algorithm. The model has good performance with accuracy, precision, recall, and F1-score values reaching 96.2%, 90.9%, 100.0%, and 95.2%.

Keywords—Convolutional Neural network, Image classification, Leukocoria, Hybrid Model, Retinoblastoma, Support Vector Machine

I. INTRODUCTION

Retinoblastoma is a rare type of cancer that affects the retina and is the most common primary intraocular cancer in children and infants. It occurs in about 1 in 18,000 to 30,000 live births worldwide [1]. One of the most common symptoms of retinoblastoma is a white reflection from the retina of the eye through the pupil. If not treated early, this condition can lead to serious diseases [2]. Making an early diagnosis is essential to avoid extraocular disease, which has a poor prognosis. However, with the advancement of technology in computer vision, early detection of retinoblastoma has become easier. One such technology is Convolutional Neural Networks (CNN). Deep learning methods called convolutional neural networks use filters or kernels with input images to extract information [3]. The method using a convolutional neural network yields the most notable outcomes for digital image recognition. This outcome is the consequence of CNN's implementation, which is based on the visual cortex's image recognition system [4]. CNN-based algorithms can directly extract "learned" features from the raw data of the relevant problem to improve classification accuracy [5].

In recent years, CNNs have been successful in several large-scale computer vision tasks [6]. Their convolutional filters scan the feature matrix and perform dimension reduction, making them precise for image classification [7]. Another algorithm commonly used for image classification is the Support Vector Machine (SVM), a supervised machine learning algorithm that can handle linear and non-linear data and can be used for classification or regression problems [8]. However, according to [9], SVM is known to not perform well with data that contains a lot of noise. To improve classification results, some studies have proposed hybrid models that combine multiple machine learning algorithms, such as the combination of the watershed, neural network segmentation, and self-organizing maps for brain tumor segmentation [10]. Based on the advantages and disadvantages of both algorithms, this research aims to develop a hybrid model that extracts features from eye images using CNN and then uses SVM as the final classifier to classify normal and leukocoria eye images using the learned features. SVM is used as a binary classifier and replaces the final layer in the CNN architecture.

II. LITERATURE REVIEW

A. A Convolutional Neural Network approach for classifying leukocoria.

In 2014, research conducted by Henning et al developed an approach for classifying recreational photography using a convolutional neural network algorithm. This approach was able to achieve good results with low error rates of smaller than 3% with a moderate amount of data and without the need for any preprocessing, the approach also uses a relatively small network [11].

B. Accurate leukocoria predictor based on deep VGGnet CNN technique

This research proposes a deep Visual Geometry Group-net CNN classifier for the automated detection of leukocoria. The proposed classifier includes pre-processing, feature extraction, and classification steps. It consists of convolution, pooling, and fully connected layers with weights developed for each image. The classifier was tested on some eye images including ordinary and leukocoria images from Flickr and showed superior results with an accuracy of 98.5% and an error rate below 2% [12].

C. Detection of leukocoria using a soft fusion of expert classifiers under non-clinical

This research presents a new classification method that utilizes fuzzy logic theory to combine multiple expert classifiers for the multichannel detection of leukocoria in recreational photography. The proposed method extracts features using the discrete cosine transform and Karhunen-Loeve transformation and achieves a 92% accuracy rate with an 89% true positive rate and an 11% false positive rate. The significance of this methodology is $p = 1.12 \times 10^{-5}$ [13].

D. A Hybrid CNN and RBF-Based SVM Approach for Breast Cancer Classification

This research investigates the use of transfer learning, where a Convolutional Neural Network (CNN) is utilized as a feature extractor to classify benign and malignant breast cancer images. The Support Vector Machine (SVM) classifier, based on the Radial Basis Function (RBF), is chosen for its ability to adjust to the data dimension space through the tuning of the kernel width. The combination of the CNN and RBF-Based SVM produces robust results for the dataset and task at hand [8].

E. Hybrid CNN-SVM Classifier for Handwritten Digit Recognition.

This paper presents a hybrid model combining a Convolutional Neural Network (CNN) and a Support Vector Machine (SVM) for handwritten digit recognition using the MNIST dataset. The proposed model combines the strengths of both classifiers, with the CNN serving as an automatic feature extractor and the SVM as a binary classifier. The MNIST dataset, consisting of diverse and highly distorted handwritten digits, is used for training and testing. The CNN's receptive field helps extract the most distinct features from these digits. The proposed model demonstrates high effectiveness with a recognition accuracy of 99.28% on the MNIST dataset [9].

F. A hybrid CNN-SVM classifier for weed recognition in winter rape field

This study presents a deep convolutional neural network (CNN) with a support vector machine (SVM) classifier designed to improve the classification accuracy of winter rape seedlings and weeds in fields. The proposed VGG-SVM model, based on the VGG network model, takes in a true color image of rape/weeds as input and achieves an average accuracy of 99.2% in training and 92.1% in testing. When compared to five other methods, the VGG-SVM model showed higher classification accuracy, robustness, and real-time performance [14].

G. Implementation of The CNN-SVM Hybrid Model For Leukocoria Detection (This Research)

In this research, two models will be combined, namely CNN-SVM where in the feature extraction process the eye image input comes from the leukocoria dataset using the CNN model. The studied features will be forwarded by the SVM model for the process of classifying normal and leukocoria eye images. SVM aims to represent multi-dimensional datasets in a space where data elements belonging to different classes are separated by hyperplanes. In this study, the SVM model is used as a binary classifier and replaces the final layer in the CNN architecture.

III. PROPOSED METHODE

A. Convolutional Neural Network.

In deep learning, convolutional neural networks (CNN, or ConvNet) are a class of artificial neural networks (ANN), most commonly applied to analyze visual images [15]. In this proposed model, CNN will be used for feature extraction from the input image. The convolution layer will extract features (feature maps) from the image data using filters with the calculation below.

$$y_{m,n} = g(\sum_i \sum_k x[j,k] W[m-n, n-k] + b) \quad (1)$$

$y_{m,n}$ is the output of the convolutional layer, g is the activation function, $x[j,k]$ is an element in the input data (such as an image) at position j and k , $W[m-n, n-k]$ is a weight coefficient, and b is a bias term. This formula represents the calculation of a single element in the output of the convolutional layer, by taking the sum of the element-wise product of the weights and the input data and adding the bias term. The activation function g is applied element-wise to the result of this sum to produce the output $y(m,n)$. This output value can be seen as a feature (or a combination of features) extracted from the input data by the convolutional layer.

B. Support Vector Machine

In this study support vector machine (SVM) was used as a final classification layer replacing sigmoid activation function. SVM is used to make predictions from the extracted features from SVM. Implementation of the SVM final classification layer in deep learning architecture is done by using the hinge loss function

$$L(w) = \sum_{i=1}^p \max(0, 1 - (y' i(w^T x_i + b))) + \lambda \|w\|_2^2 \quad (2)$$

The hinge loss function penalizes the predicted class if it is not confident enough. If the predicted class is correct, then the score ($w^T x_i + b$) will be positive and the hinge loss will be 0. If the predicted class is incorrect, then the score will be negative and the hinge loss will be the absolute value of the score, which increases as the score becomes more negative. The λ regularization term helps prevent overfitting by penalizing large weight values which in this study we used the L2 regularization. The L2 regularization term adds a penalty to the loss function that is proportional to the sum of the squares of the weights. This encourages the model to have small, non-zero weights, which can help reduce overfitting.

In addition to using hinge loss as the loss function, to implement support vector machine (SVM) as the final classifier, we also utilize a linear activation function in the last layer of the model.

$$\min \left(\frac{1}{2} \|w\|_2^2 \right) \quad (3)$$

which when combined with hinge loss in equation 2 will be as follows

$$\min \left(\frac{1}{2} \|w\|_2^2 \right) + \sum_{i=1}^p \max(0, 1 - (y' i(w^T x_i + b))) + \lambda \|w\|_2^2 \quad (4)$$

Linear activation allows the SVM classifier to make predictions based on the linear combination of the input features and the weights learned during training. The use of a linear activation function in the classifier layer is essential for SVM to function properly, as it allows the model to make decisions based on the hyperplane that maximally separates the different classes in the feature space.

Finally, to predict the class label from the test data x the classification layer will calculate it as follows:

$$\arg \max_{y_i} (w^T x) y' \quad (5)$$

This equation is trying to find the label y'_i that results in the highest dot product between the vectors w and x , which can be used as a prediction for the class of the i th example in the dataset.

C. Hybrid Model CNN-SVM.

In this study, we are going to combine the strengths of both CNN and SVM algorithms into one architecture. This model is called Hybrid CNN-SVM, in a hybrid model CNN-SVM, the CNN is used to extract features from the input image, and the SVM is used to classify the image based on these features. This approach is particularly useful for tasks where the input data is high-dimensional and complex, such as image classification. The CNN can extract relevant features from the data, while the SVM can use these features to accurately classify the data. The proposed model architecture can be seen in Figure 1.

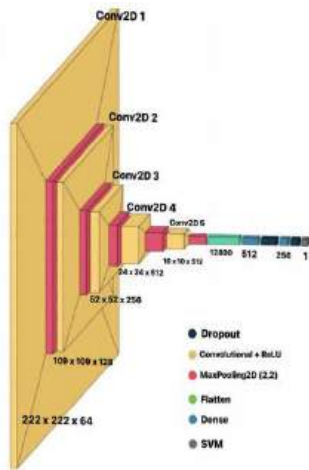


Fig. 1. Proposed Method Architecture

IV. EXPERIMENTS AND RESULTS

A. Datasets

This study aims to produce a model that can classify between normal eyes and eyes with leukocoria based on photographic images, so the data used in this study are images of normal eyes and images of eyes with leukocoria. To get image data for normal eyes and leukocoria, we searched with the keywords “eye”, “healthy eye”, “leukocoria”, and “retinoblastoma”. From the search results, a total of 118 normal eye images and 132 images were collected where the data will be divided into 3 subsets, namely the training data subset, validation data, and testing data with the following details

Datasets details:

- Training data: 102 leukocoria eye pictures and 82 normal eye picture
- Validation data: 20 leukocoria eye pictures and 20 normal eye picture
- Testing data: 10 leukocoria eye pictures and 16 normal eye pictures

sample of the dataset can be seen in Figure 2

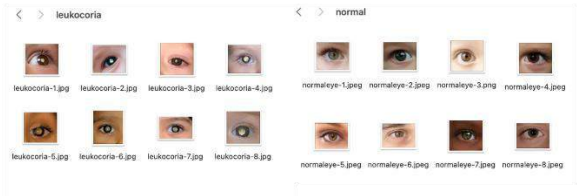


Fig. 2. Dataset Sample

B. Data Preprocessing

To increase the amount of data available for deep learning and improve performance, data augmentation is used in the data preprocessing process. Data augmentation involves generating new data from existing data, which can also help prevent overfitting. This is particularly important in this case due to the small size of the dataset. The stages of data augmentation carried out include:

Data augmentation steps:

- Rescaling the RGB image pixel data from a range of 0-255 to 0-1
- Rotating the image by a random degree (up to 20)
- Applying shear and zoom.
- and Flipping the image horizontally and vertically

These steps help facilitate the training process and improve performance.

C. Modeling

The model architecture will consist of a Convolutional Layer, a Fully Connected Layer, and SVM Classifier Layer. A convolutional Layer will be used for feature extraction of the input images. The first layer of the Convolutional Layer processes the input image (224x224 pixels and 3 RGB channels) using 64 filters, a 3x3 kernel, valid padding, and ReLU activation. The image is then reduced in size by 50% through max pooling with a filter size of 2x2 and stride of (2,2). The next four convolution layers use 128, 256, 512, and 512 filters respectively, along with a 3x3 kernel, valid padding, and ReLU activation. Each convolution layer is followed by max pooling with a filter size of 2x2 and stride of (2,2), resulting in images with sizes of 54x54, 26x26, 12x12, and finally 5x5 pixels. The CNN architecture can be seen in Table 1.

TABLE I. CNN ARCHITECTURE

No	Layer (type)	Output Shape	Number of trainable parameters
1	conv2d (Conv2D)	(None, 222, 222, 64)	1792
2	max_pooling2d (MaxPooling2D)	(None, 111, 111, 64)	0
3	conv2d_1 (Conv2D)	(None, 109, 109, 128)	7386
4	max_pooling2d_1 (MaxPooling2D)	(None, 54, 54, 128)	0
5	conv2d_2 (Conv2D)	(None, 52, 52, 256)	295168
6	max_pooling2d_2 (MaxPooling2D)	(None, 26, 26, 256)	0

7	conv2d_3 (Conv2D)	(None, 24, 24, 512)	1180160
8	max_pooling2d_3 (MaxPooling2D)	(None, 12, 12, 512)	0
9	conv2d_4 (Conv2D)	(None, 10, 10, 512)	2359808
10	max_pooling2d_3 (MaxPooling 2D)	(None, 5, 5, 512)	0

After passing through the convolution, the Flatten() layer converts the parameters from the convolution layer into a one-dimensional vector, which is then input into the fully connected layer. The fully connected layer consists of two Dense layers with 512 and 256 units and ReLU activation, each followed by a Dropout layer that drops 2% of the input units.

TABLE II. FULLY CONNECTED LAYER ARCHITECTURE

No	Layer (type)	Output Shape	Number of trainable parameters
11	flatten (Flatten)	(None, 12800)	0
12	dense (Dense)	(None, 512)	6554112
13	dropout (Dropout)	(None, 512)	0
14	dense_1 (Dense)	(None, 256)	13128
15	dropout_1 (Dropout)	(None, 256)	0

Last is the classification layer. The SVM algorithm is used in the classification layer. The classification layer uses a Dense layer with an l2 regularizer of 0.01 and a linear activation function and is set for binary classification with one output unit. The model is configured with a hinge loss function and Adam optimization, and is evaluated using the accuracy metric.

Before training, the model must be configured with a loss function, optimization method, and evaluation metric. In this case, the model is set for binary classification using the SVM algorithm and a hinge loss function. Adam optimization is chosen for its ability to adjust the learning rate based on features from the RMSprop and Adagrad optimizers. The model's performance will be measured using the accuracy metric in every epoch.

D. Training and Evaluation

At this stage, The model will be trained using a portion of the available training data and evaluated using a portion of the validation data. This process will be repeated for a total of 25 epochs, during which the loss value and accuracy of the model will be evaluated at each iteration. By training the model on a subset of the data and evaluating it on a different subset, we can monitor the model's performance and adjust it as needed to improve its accuracy. The results of the training model can be seen in the following figure 3.

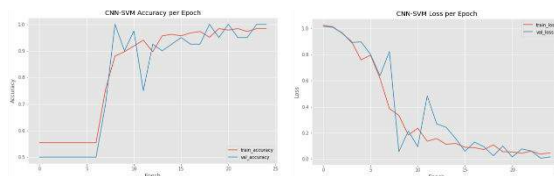


Fig. 3. Accuracy and Loss per Epochs Plot

The accuracy graph shows good results, with both the training and validation data consistently achieving accuracy above 90% after the 12th iteration. The graph also does not exhibit any signs of overfitting, where the training data's accuracy would continue to increase while the validation data's accuracy remains stagnant. Instead, both sets of data show similar patterns of improvement throughout the training process. Like the accuracy graph, the loss model graph does not show any signs of overfitting. Despite its fluctuating appearance, the graph demonstrates that the model was successful in minimizing loss over the 25 iterations of the training process.

E. Testing

During the testing phase, the created model will be used to make predictions on images in the test data subset. The prediction results will be evaluated using four metrics: recall, precision, accuracy, and f1-score. These metrics will provide insights into the model's performance and allow us to determine how well it can accurately classify the images in the test dataset.

As shown in Table III, the model correctly predicted 10 out of 10 positive cases and 15 out of 16 negative cases, with only 1 incorrect prediction in the negative class. The model's performance was then evaluated using various metrics, which are presented in Table IV.

TABLE III. CONFUSION MATRIX

		Actual Condition	
		Positive (10)	Negative (16)
	Total Population (26)		
Predicted Condition	Positive (12)	True Positive (TP) 10	False Positive (FP) 1
	Negative (15)	False Negative (FN) 0	True Negative (TN) 15

TABLE IV. ACCURACY, PRECISION, RECALL, AND F1-SCORE

Metrics			
Accuracy	Precision	Recall	F1-Score
96.2%	90.9%	100.0%	95.2%

The calculation of the metrics shows that the model has good performance, as indicated by its high accuracy of 96.2%. Additionally, the model has high values for precision, recall, and F1-score at 90.9%, 100.0%, and 95.2% respectively. The recall value of 100% indicates that the model can accurately classify positive cases (leukocoria), which is particularly important in detecting the symptoms of a disease. A high recall value minimizes the chances of false positive predictions. On the other hand, the precision of 90.9% means that the model is correct 90.9% of the time when predicting a positive case.

Based on the research results that have been obtained, performance comparisons with previous studies using similar methods and models have been carried out. The results of the comparison can be seen in the following table V.

TABLE V. COMPARISON OF THE PERFORMANCE OF THE PROPOSED MODEL WITH EACH DATASET

Method	Dataset	Metrics			
		Accuracy	Precision	Recall	F1-Score
CNN-SVM (This Research)	Data: 250 Normal: 118 Leukocoria: 132	96.2%	90.9%	100.0%	95.2%
Soft Fusion [13]	Data: 114 Normal: 90 Leukocoria: 54	88.1%	62.1%	92.8%	90.2%
CNN transfer learning	Data: 124	98.5%	98.3%	97.5%	89.5%

VGG16 [12]	Normal: 89 Leukocoria: 35				
CNN transfer learning ResNet50 [16]	Data: 4356 Normal: 4079 Leukocoria: 277	75.9%	43.2%	87.2%	57.7%

Based on the comparison results in Table V, the model proposed in this study, namely the CNN-SVM hybrid model, has superior recall and F1-score metrics scores compared to the three previous models. The CNN transfer learning VGG16 model is the model that produces the best accuracy and precision metrics compared to other models with a value of 98.5% for accuracy and 98.3% for precision.

With the results of these comparisons, the researcher compared the performance of the proposed model with models that had been offered in previous studies that had been trained using the dataset in this study. The previously proposed model implemented in this study by the provisions in respective research papers or journals, parameters that are not explained in the paper or journal will be adjusted to the rules of thumb applicable. Table VI represents the architecture of each model.

TABLE VI. PROPOSED MODEL ARCHITECTURE

Component	Model			
	CNN-SVM (this research)	CNN [17]	CNN transfer learning VGG16 [12]	CNN transfer learning ResNet50 [16]
Input Layers	224 x 224	40 x 40	64 x 64	224 x 224
Activation Function	- Feature Extraction: ReLU - Classification: Linear	- Feature Extraction: Hyperbolic Tangent - Classification: Softmax	- Feature Extraction: ReLU - Classification: Sigmoid	- Feature Extraction: ReLU - Classification: Sigmoid
Loss Function	- Hinge	- Categorical Cross entropy	- Binary Crossentropy	- Binary Crossentropy
Architecture	- 5 Convolution layers with Units (64, 128, 256, 512, 512) 3x3 Filters - Max Pooling 2x2 - 2 layers Fully Connected with units 512 and 256 - SVM classifier layer	- 3 Convolution layers with Unit (7, 7, 7) Filter 5x5 - Max Pooling 2x2 - 1 layer Fully Connected with 5 units - Softmax classifier layer	- 64 x 64 Input Layers - Transfer Learning VGG 16 - 2 layers fully connected with units 256 and 512 - Sigmoid classifier layer	- 64 x 64 Input Layers - Transfer Learning ResNet50 - Sigmoid classifier layer
Optimizer	Adam	RMSPROP	Adam	Adam

The testing process will use a subset of test data consisting of 16 normal eye images and 10 eye images with leukocoria. The test results will be measured using predetermined metrics, a comparison of the performance of the proposed model with models that have been proposed in previous studies can be seen in Table VII.

TABLE VII. COMPARISON OF THE PERFORMANCE OF THE PROPOSED MODEL WITH THE SAME DATASET

Metrics	Model			
	CNN-SVM (this research)	CNN [17]	CNN transfer learning VGG16 [12]	CNN transfer learning ResNet50 [16]
Accuracy	96.2%	76.5%	53.8%	61.5%
Precision	90.9%	62.5%	45.0%	0%
Recall	100.0%	100.0%	90.0%	0%
F1-Score	95.2%	76.9%	60.0%	0%

Based on the results of this comparison, the CNN-SVM hybrid method can outperform the three previous methods using the same dataset. The model with the CNN SVM hybrid method succeeded in correcting the deficiencies that existed in the CNN model which was not optimal in classifying the negative (normal) class. The CNN-SVM model can obtain an accuracy value of 96.2%, a precision of 90.9%, and an F1-score of 95.2% which indicates the CNN-SVM model is capable of classifying both positive and negative classes. From the test results on the previous proposed model, it is suspected that there are factors that affect the performance of each model, including the CNN model which has a simple architecture, so the performance is not as good as CNN-SVM. Models with Transfer learning ResNet50 are unable to classify because the model is too complex and the available data is not sufficient. Meanwhile, the model with VGG16 transfer learning may experience negative transfer which causes poor performance. By using a combination of CNN as a feature extractor in the image and also SVM as the final classifier it seems capable of carrying out the classification process well.

V. CONCLUSION

In this paper, we produced a CNN-SVM model that combines two algorithms, namely CNN as a feature extractor and SVM as a final classifier that is capable of classifying leukocoria symptoms in photographic eye images using the CNN-SVM hybrid method. Where from the results of the tests carried out, the proposed CNN-SVM hybrid model obtained an accuracy value of 96.2%, a recall of 100.0%, a precision of 90.9%, and an F1-Score of 95.2%. From the research results obtained for future research, the authors suggest further research to be able to reproduce the data in the dataset to get a more varied representation of normal eyes and leukocoria eyes to train the model to be more accurate.

REFERENCES

- [1] Abramson, D. H., Beaverson, K. *et al.*, "Screening for retinoblastoma: presenting signs as prognosticators of patient and ocular survival," *Pediatrics*, vol. 112, no. 5, pp. 1248–1255, 2003.
- [2] G. L. Chantada and C. A. Leal, "Retinoblastoma: An international perspective. In Clinical Ophthalmic Oncology," *Springer*, pp. 57–65, 2019.
- [3] R. Chauhan, K. K. Ghanshala, and R. C. Joshi, "Convolutional Neural Network (CNN) for Image Detection and Recognition," *ICSCCC 2018 - 1st Int. Conf. Secur. Cyber Comput. Commun.*, pp. 278–282, 2018, doi: 10.1109/ICSCCC.2018.8703316.
- [4] S. U. Masruroh, A. Fiade, M. I. Tanggok, R. A. Putri, and L. A. Pratiwi, "Convolutional Neural Network for Colorization of Black and White Photos," *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 22, no. 2, pp. 393–404, 2023, doi: 10.30812/matrik.v22i2.2652.
- [5] S. U. Masruroh, M. F. Syahid, F. Munthaha, A. T. Muharram, and R. A. Putri, "Deep Convolutional Neural Networks Transfer Learning Comparison on Arabic Handwriting Recognition System," *Int. J. Informatics Vis.*, vol. 7, no. 2, pp. 330–337, 2023, doi: 10.30630/ijov.7.2.1605.
- [6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017, doi: 10.1145/3065386.
- [7] P. Mishra, "Why are Convolutional Neural Networks good for image classification? | by Praful Mishra | DataDrivenInvestor," *DataDrivenInvestor*, 2019, <https://medium.datadriveninvestor.com/why-are-convolutional-neural-networks-good-for-image-classification-146ec6e865e8> (accessed Jul. 26, 2023).
- [8] M. Alkhaleefah and C. C. Wu, "A Hybrid CNN and RBF-Based SVM Approach for Breast Cancer Classification in Mammograms," *Proc. - 2018 IEEE Int. Conf. Syst. Man, Cybern. SMC 2018*, pp. 894–899, 2019, doi: 10.1109/SMC.2018.00159.
- [9] S. Ahlawat and A. Choudhary, "Hybrid CNN-SVM Classifier for Handwritten Digit Recognition," *Procedia Comput. Sci.*, vol. 167, no. 2019, pp. 2554–2560, 2020, doi: 10.1016/j.procs.2020.03.309.
- [10] A. Anand, "Brain Tumor Segmentation using Watershed Technique and Self Organizing Maps," *Indian J. Sci. Technol.*, vol. 10, no. 44, pp. 1–6, 2017, doi: 10.17485/ijst/2017/v10i44/120574.
- [11] M. C. Munson *et al.*, "Autonomous early detection of eye disease in childhood photographs," *Sci. Adv.*, vol. 5, no. 10, 2019, doi: 10.1126/sciadv.aax6363.
- [12] B. S. Rao, "Accurate leukocoria predictor based on deep VGG-net CNN technique," *IET Image Process.*, vol. 14, no. 10, pp. 2241–2248, 2020, doi: 10.1049/iet-ipr.2018.6656.
- [13] P. Rivas-Perea, E. Baker, G. Hamerly, and B. F. Shaw, "Detection of leukocoria using a soft fusion of expert classifiers under non-clinical settings," *BMC Ophthalmol.*, vol. 14, no. 1, 2014, doi: 10.1186/1471-2415-14-110.
- [14] T. Tao and X. Wei, "A hybrid CNN-SVM classifier for weed recognition in winter rape field," *Plant Methods*, vol. 18, no. 1, pp. 1–12, 2022, doi: 10.1186/s13007-022-00869-z.
- [15] M. V. Valueva, N. N. Nagomov, P. A. Lyakhov, G. V. Valuev, and N. I. Chervyakov, "Application of the residue number system to reduce hardware costs of the convolutional neural network implementation," *Math. Comput. Simul.*, vol. 177, pp. 232–243, 2020, doi: 10.1016/j.matcom.2020.04.031.
- [16] A. Bernard *et al.*, "EyeScreen: Development and Potential of a Novel Machine Learning Application to Detect Leukocoria," *Ophthalmol. Sci.*, vol. 2, no. 3, pp. 1–7, 2022, doi: 10.1016/j.xops.2022.100158.
- [17] R. Henning, P. Rivas-Perea, B. Shaw, and G. Hamerly, "A Convolutional Neural Network approach for classifying leukocoria," *Proc. IEEE Southwest Symp. Image Anal. Interpret.*, no. 222, pp. 9–12, 2014, doi: 10.1109/SSIAL.2014.6806016.

Reinforcement Learning-based Contention Window Adjustment for Wireless Body Area Networks

Jung-Hyok Kwon

Smart Computing Laboratory
Hallym University
Chuncheon, South Korea
jhkwon@hallym.ac.kr

Dongwan Kim

Department of Electronic Engineering
Dong-A University
Busan, South Korea
dongwankim@dau.ac.kr

Eui-Jik Kim*

Division of Software
Hallym University
Chuncheon, South Korea
ejkim32@hallym.ac.kr

Abstract—This paper presents a reinforcement learning-based contention window adjustment for wireless body area networks (WBANs), abbreviated RL-CWA. The RL-CWA uses Q-learning to adaptively adjust the contention window according to the acknowledgment reception, thereby reducing the number of collisions. In addition, RL-CWA maintains multiple Q-tables to take into account minimum and maximum contention windows that vary depending on the user priority (UP) of the traffic. To verify the superiority of RL-CWA, an experimental simulation was conducted. The results showed that the RL-CWA obtains better performance than the existing IEEE 802.15.6 MAC.

Keywords—contention window adjustment, IEEE 802.15.6, Internet of medical things, reinforcement learning, wireless body area networks

I. INTRODUCTION

Recently, with the growing interest in connected and personalized healthcare, the demand for the Internet of medical things (IoMT) services such as remote patient monitoring, medication management, and remote diagnostics are rapidly increasing [1,2]. In IoMT, users are normally equipped with multiple sensor devices to measure vital signs of the human body. The sensor devices form a wireless body area network (WBAN) and exchange the measured biomedical data via the wireless links of the WBAN [3]. However, the sensor device generates and transmits the data only with a limited power source [4]. In addition, some of the biomedical data may include fatal information requiring early access. Therefore, the efficient design of medium access protocol (MAC) is considered essential for reliable WBAN [5].

IEEE 802.15.6 MAC is standardized to address the requirements and challenges of WBANs [6, 7]. The standard provides mechanisms to support different traffic classes, prioritizing critical and delay-sensitive data, and ensuring reliable and timely delivery. Specifically, it classifies the traffic into eight classes according to the respective priority and differentiates the minimum and maximum contention windows (i.e., CW_{min} and CW_{max}) for each traffic class. In order to minimize energy consumption, it employs a carrier sense multiple access with collision avoidance (CSMA/CA)-based channel access mechanism in which the contention window (CW) is doubled only when an even number of transmission failures [8]. However, the existing standard may suffer from frequent collisions especially when the number of devices increases. This is because CW_{min} is always used on the first transmission attempt, and CW is maintained for an odd

number of transmission failures, regardless of the number of devices.

In order to address the problem, a number of studies have been conducted to minimize the number of collisions in WBANs. Bradai *et al.* proposed a priority MAC (PMAC) that used a superframe structure dividing into uplink and downlink phases [9]. In order to ensure the priority of the emergency data, they used two inter-frame spaces (IFSs): emergency IFS (EIFS) and normal IFS (NIFS). Yuan *et al.* and Zia *et al.* proposed adaptive MAC (A-MAC) and group-based medium access control (G-MAC) protocols, respectively [10, 11]. These MAC protocols proposed a new superframe structure including a contention access phase (CAP) including multiple sub-phase. In each sub-phase, only the designated traffics can be transmitted data using the existing CSMA/CA-based channel access mechanism. However, these MAC protocols just used a CSMA/CA-based channel access mechanism specified in IEEE 802.15.6 MAC; thus, frequent collisions may occur due to the inadequate size of CW. Furthermore, they have lower compatibility with the existing standard since they modify the superframe structure.

In this paper, we propose a reinforcement learning-based contention window adjustment for WBANs (RL-CWA), which adaptively adjusts the size of CW using the Q-learning [12, 13]. In RL-CWA, the sensor device uses a Q-agent to adjust CW according to acknowledgment (ACK) reception. The Q-agent maintains multiple Q-tables and a Q-function. Each Q-table is specified differently for each user priority (UP) of the traffic and lists the Q-values for varying states (i.e., CW) and actions (i.e., Decrease, Keep, Increase). The Q-function updates the Q-value depending on the reward determined based on the result of ACK reception. In RL-CWA, the Q-agent of the sensor device selects one of the Q-tables referring to the UP of the traffic to be transmitted. Thereafter, it determines the current state considering the action of the previous state and selects the action of the current state by comparing the Q-values. After transmission, the Q-agent updates the Q-value for the current state and action according to the result of the ACK reception (i.e., reward). To evaluate the RL-CWA performance, we conducted the experimental simulation. The results showed that RL-CWA obtains better performance in terms of the number of collisions and aggregate throughput compared to the existing IEEE 802.15.6 MAC [14].

The remainder of this paper is organized as follows. Section II presents the system model considered in this paper. In Section III, we describe the design of RL-CWA in detail. The simulation results are given in Section IV. Finally, Section V concludes this paper.

* Corresponding Author.

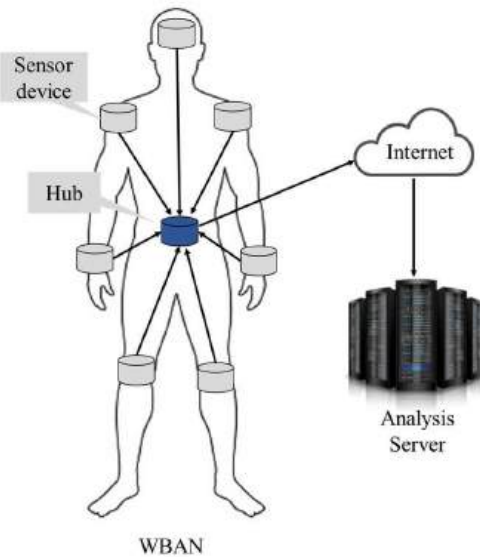


Fig. 1. System architecture for RL-CWA.

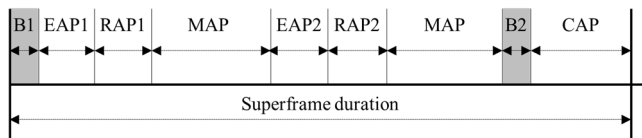


Fig. 2. Superframe structure.

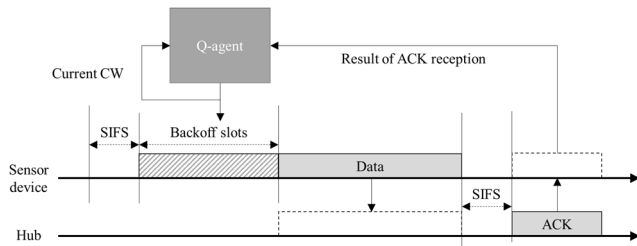


Fig. 3. Overall operation of RL-CWA.

II. SYSTEM MODEL

A. System Architecture

Fig. 1 shows the overall system architecture for RL-CWA, which consists of multiple sensor devices and a hub. The sensor devices periodically measure vital signs. Then, they generate and transmit biomedical data to the hub using RL-CWA. We assume that the sensor devices are connected to the hub in a single hop. We further assume that the data generated by the sensor devices contains biomedical information. In other words, we only consider the three UPs of traffic. Table I shows the CW_{min} and CW_{max} for each UP.

TABLE I. MINIMUM AND MAXIMUM CONTENTION WINDOW FOR EACH USER PRIORITY

UP	Traffic designation	CW_{min}	CW_{max}
5	Medical data or network control	4	8
6	High-priority data or network control	2	8
7	Emergency or medical implant event report	1	4

The hub acts as a central coordinator that plays a critical role in managing and controlling the operations of the WBAN. Specifically, it constructs the superframe and informs sensor

devices of the superframe structure. In addition, it manages the allocation of access rights and determines which sensor devices have permission to transmit data during specific time slots or access phases. The receives the data from all sensor devices and transmits the analysis server via the Internet.

B. Superframe Structure

We consider the beacon mode with a superframe that uses the superframe structure as illustrated in Fig. 2. The superframe is partitioned into beacon phase (i.e., B), exclusive access phase (i.e., EAP1 and EAP2), random access phase (i.e., RAP1 and RAP2), managed access phase (i.e., MAP), and contention access phase (CAP). B1 and B2 are used to broadcast beacon frames at the beginning of the superframe and before the start of the CAP, respectively. EAPs are used for high-priority and emergency traffic, but RAPs and CAP are used for all types of traffic. In EAP, RAP, and CAP, CSMA/CA-based access mechanism is used. For high-priority or emergency traffic, the sensor device considers that the adjacent EAP and RAP are treated as a single EAP. Finally, MAP is used to allocate dedicated slots for contention-free access. In our work, it is assumed that only EAPs and RAPs are used for data transmission. Specifically, if the UP of traffic to be transmitted is 6 or 7, adjacent EAP and RAP are used as single EAP. However, only RAP is used when the UP of traffic to be transmitted is 5.

III. DESIGN OF RL-CWA

Fig. 3 shows the overall operation of RL-CWA. When the sensor device has the data to be transmitted, it first checks the UP of the corresponding data. After short inter-frame space (SIFS), it randomly selects the number of backoff slots (i.e., randomly sets the backoff counter) in the range of $[0, CW]$ and waits until the backoff counter reaches zero. The CW for the initial transmission attempt (i.e., initial CW) is equal to the CW_{min} listed in TABLE I. That is, the initial CW varies depending on the UP of the data. When the backoff counter reaches zero, the sensor device transmits data and waits for the ACK from Hub after SIFS.

The Q-agent running on the sensor device adjusts the CW considering the result of ACK reception. To this end, the Q-agent maintains multiple Q-tables and a Q-function. Each Q-table is designated for each UP, and the number of Q-tables is the same as the number of UPs. The Q-table lists the Q-values for varying states (i.e., CWs) and actions (i.e., Decrease, Keep, Increase). Note that the Q-value in the Q-table for a specific UP is represented as $Q_{UP}(s, a)$, where s is the state index and a is the action index. The initial action is set to Keep. $Q_{UP}(0, 0)$ and $Q_{UP}(CW_{max}-CW_{min}, 2)$ are initialized to -100 , and the other Q-values are initialized to zero. Tables II, III, and IV show the Q-tables for UP=5, UP=6, and UP=7, respectively.

TABLE II. Q-TABLE FOR UP=5

State	Action		
	Decrease ($a=0$)	Keep ($a=1$)	Increase ($a=2$)
$CW=4 (s=0)$	$Q_5(0,0)$	$Q_5(0,1)$	$Q_5(0,2)$
$CW=5 (s=1)$	$Q_5(1,0)$	$Q_5(1,1)$	$Q_5(1,2)$
$CW=6 (s=2)$	$Q_5(2,0)$	$Q_5(2,1)$	$Q_5(2,2)$
$CW=7 (s=3)$	$Q_5(3,0)$	$Q_5(3,1)$	$Q_5(3,2)$
$CW=8 (s=4)$	$Q_5(4,0)$	$Q_5(4,1)$	$Q_5(4,2)$

TABLE III. Q-TABLE FOR UP=6

State	Action		
	Decrease ($a=0$)	Keep ($a=1$)	Increase ($a=2$)
CW=2 ($s=0$)	$Q_\delta(0,0)$	$Q_\delta(0,1)$	$Q_\delta(0,2)$
CW=3 ($s=1$)	$Q_\delta(1,0)$	$Q_\delta(1,1)$	$Q_\delta(1,2)$
CW=4 ($s=2$)	$Q_\delta(2,0)$	$Q_\delta(2,1)$	$Q_\delta(2,2)$
CW=5 ($s=3$)	$Q_\delta(3,0)$	$Q_\delta(3,1)$	$Q_\delta(3,2)$
CW=6 ($s=4$)	$Q_\delta(4,0)$	$Q_\delta(4,1)$	$Q_\delta(4,2)$
CW=7 ($s=5$)	$Q_\delta(5,0)$	$Q_\delta(5,1)$	$Q_\delta(5,2)$
CW=8 ($s=6$)	$Q_\delta(6,0)$	$Q_\delta(6,1)$	$Q_\delta(6,2)$

TABLE IV. Q-TABLE FOR UP=7

State	Action		
	Decrease ($a=0$)	Keep ($a=1$)	Increase ($a=2$)
CW=1 ($s=0$)	$Q_\gamma(0,0)$	$Q_\gamma(0,1)$	$Q_\gamma(0,2)$
CW=2 ($s=1$)	$Q_\gamma(1,0)$	$Q_\gamma(1,1)$	$Q_\gamma(1,2)$
CW=3 ($s=2$)	$Q_\gamma(2,0)$	$Q_\gamma(2,1)$	$Q_\gamma(2,2)$
CW=4 ($s=3$)	$Q_\gamma(3,0)$	$Q_\gamma(3,1)$	$Q_\gamma(3,2)$

The Q-function is used to update the Q-value for the current state and action. Equation (1) presents the Q-function.

$$Q_{UP}(s_t, a_t) \leftarrow (1 - \alpha) \cdot Q_{UP}(s_t, a_t) + \alpha[r_t + \gamma \cdot \max_a Q_{UP}(s_{t+1}, a)] \quad (1)$$

where the left side of $Q_{UP}(s_t, a_t)$ is the updated Q-value for the t -th transmission, α is the learning rate that determined in the range of $[0, 1]$, the right side of $Q_{UP}(s_t, a_t)$ is the old Q-value for the t -th transmission, r_t is the reward for the state and action of the t -th transmission, γ is the discount factor, and $\max_a Q_{UP}(s_{t+1}, a)$ is the estimate of optimal future Q-value.

In (1), r_t is determined according to the result of ACK reception. Specifically, the reward can be obtained by (2).

$$r_t = \begin{cases} 1, & \text{If ACK is received} \\ -1, & \text{If ACK is not received} \end{cases} \quad (2)$$

After updating the Q-value, the Q-agent selects one of the actions to determine the next state. Specifically, it selects the action whose Q-value is the greatest. In the case that multiple actions are selected at the same time, one of them is chosen randomly. If the selected action is Decrease or Increase, the CW for the next transmission is decreased or increased by one. That is, in this case, the state index changes. Otherwise, the CW is maintained as it is.

IV. SIMULATION RESULTS

To evaluate the RL-CWA performance, experimental simulation was conducted. The simulation results of RL-CWA were compared with that of the existing IEEE 802.15.6 MAC. In the simulation, the sensor device was set to always have the data to be transmitted (i.e., saturation condition), and the UP of the data was set to be randomly selected. The

superframe length was configured to 255 slots, and the length of EAP1, RAP1, EAP2, and RAP2 were 30 slots, respectively. The detailed simulation parameters are listed in Table V.

TABLE V. SIMULATION PARAMETERS

Parameter	Value	Parameter	Value
Data rate	151.8 kbps	PLCP preamble	90 bits
Superframe	255 slots	PLCP header	31 bits
Slot length	376 μ s	MAC header	56 bits
SIFS	75 μ s	MAC body	64 bits
mACK	1,336 μ s	FCS	16 bits

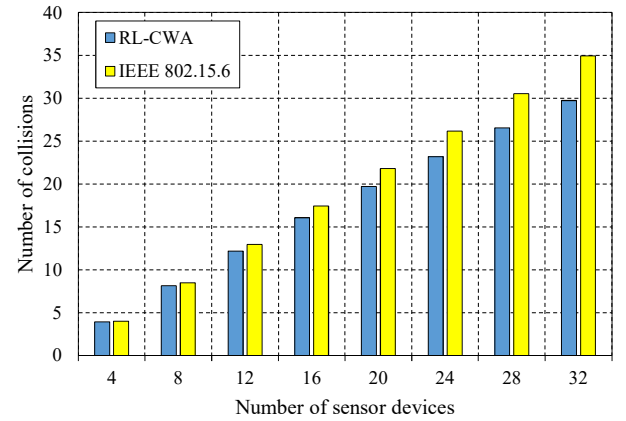


Fig. 4. Number of collisions.

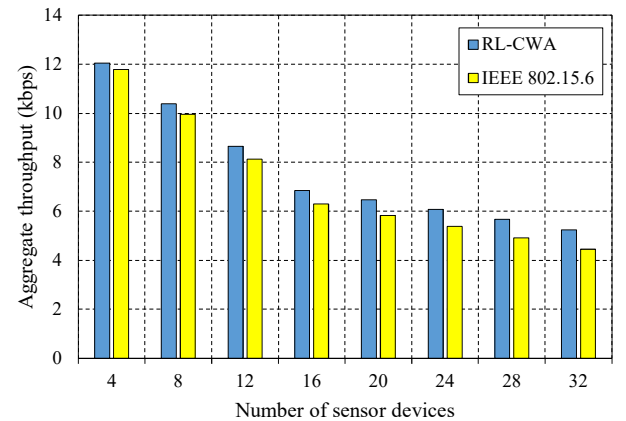


Fig. 5. Aggregate throughput.

Fig. 4 illustrates the relationship between the number of collisions and the number of sensor devices. As depicted, there is an increase in the number of collisions with an increasing number of sensor devices. This is because as the number of sensor devices increases, the probability that multiple devices select the same backoff counter increases. In the figure, RL-CWA exhibited fewer number of collisions than the existing IEEE 802.15.6 MAC. This improved performance of RL-CWA is attributed to its ability to finely adjust the CW using a Q-agent, as well as its avoidance of initializing the CW even after successful transmission. On average, the RL-CWA obtained 10.76% fewer number of collisions than the existing IEEE 802.15.6 MAC.

Fig. 5 shows the variation of the aggregate throughput for the varying number of sensor devices. The aggregate throughput decreases when the number of sensor devices increases, due to the increase in the number of collisions. Since the RL-CWA has a lower collision probability compared to the IEEE 802.15.6 MAC, it obtains higher aggregate throughput. Quantitatively, the RL-CWA had an 8.20% higher aggregate throughput than the IEEE 802.15.6 MAC.

V. CONCLUSION

This paper proposes an RL-CWA, a method for adjusting the CW in WBANs using reinforcement learning. RL-CWA utilizes Q-learning to dynamically adapt the CW size based on ACK reception, aiming to minimize collisions. In RL-CWA, the sensor device maintains multiple Q-tables to consider varying minimum and maximum CWs depending on the UP of the traffic. To validate the effectiveness of RL-CWA, an experimental simulation was performed. The results showed that RL-CWA has superior performance with respect to the number of collisions and aggregate throughput compared to existing IEEE 802.15.6 MAC.

ACKNOWLEDGMENT

This research was supported by Semiconductor R&D Support Project through the Gangwon Technopark (GWTP) funded by Gangwon Province (No. GWTP 2023-027). This work was supported by the NRF grant funded by the Korea government (MSIT) (NRF-2021R1C1C2095696). This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2022-0-00194, Analysis and research about 5G standard patent).

REFERENCES

- [1] A. Ghubaish, T. Salman, M. Zolanvari, D. Unal, A. Al-Ali and R. Jain, "Recent Advances in the Internet-of-Medical-Things (IoMT) Systems Security," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 8707–8718, 1 Jun. 2021.
- [2] M. Kumar, Kavita, S. Verma, A. Kumar, M. F. Ijaz and D. B. Rawat, "ANAF-IoMT: A Novel Architectural Framework for IoMT-Enabled Smart Healthcare System by Enhancing Security Based on RECC-VC," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 12, pp. 8936–8943, Dec. 2022.
- [3] G. Cai, Y. Fang, J. Wen, G. Han and X. Yang, "QoS-Aware Buffer-Aided Relaying Implant WBAN for Healthcare IoT: Opportunities and Challenges," *IEEE Network*, vol. 33, no. 4, pp. 96–103, Jul./Aug. 2019.
- [4] F. Hu, X. Liu, M. Shao, D. Sui and L. Wang, "Wireless Energy and Information Transfer in WBAN: An Overview," *IEEE Network*, vol. 31, no. 3, pp. 90–96, May/Jun. 2017.
- [5] M. Mohamed, B. J. Maiseli, Y. Ai, K. Mkocho and A. Al-Saman, "In-Body Sensor Communication: Trends and Challenges," *IEEE Electromagnetic Compatibility Magazine*, vol. 10, no. 2, pp. 47–52, 2nd Quarter 2021.
- [6] S. Misra, P. K. Bishoyi and S. Sarkar, "i-MAC: In-Body Sensor MAC in Wireless Body Area Networks for Healthcare IoT," *IEEE Systems Journal*, vol. 15, no. 3, pp. 4413–4420, Sep. 2021.
- [7] S. Sarkar, S. Misra, B. Bandyopadhyay, C. Chakraborty and M. S. Obaidat, "Performance Analysis of IEEE 802.15.6 MAC Protocol under Non-Ideal Channel Conditions and Saturated Traffic Regime," in *IEEE Transactions on Computers*, vol. 64, no. 10, pp. 2912–2925, 1 Oct. 2015.
- [8] N. B. N. Ranjan and J. G. A. "Performance Analysis of Prioritization and Contention Control Algorithm in Wireless Body Area Networks," in *The Computer Journal*, vol. 64, no. 1, pp. 211–223, Nov. 2019.
- [9] N. Bradai, L. C. Fourati, S. Boudjit and L. Kamoun, "New priority MAC protocol for wireless body area networks," *Proceedings of the 3rd ACM MobiHoc workshop on Pervasive wireless healthcare*, pp. 1–6, 2013.
- [10] D. Yuan, G. Zheng, H. Ma, J. Shang and J. Li, "An adaptive MAC protocol based on IEEE 802.15.6 for wireless body area networks," *Wireless Communication. Mobile Computing*, vol. 5, pp. 1–9, Feb. 2019.
- [11] Y. Zia, F. Bashir and K. N. Qureshi, "Dynamic superframe adaptation using group-based media access control for handling traffic heterogeneity in wireless body area networks," *International Journal of Distributed Sensor Networks*, vol. 16, no. 8, 2020.
- [12] H. Van Hasselt, A. Guez and D. Silver, "Deep reinforcement learning with double Q-learning," *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pp. 2094–2100, 2016.
- [13] M. Lin, Z. Sun, Y. Xia and J. Zhang, "Reinforcement Learning-Based Model Predictive Control for Discrete-Time Systems," *IEEE Transactions on Neural Networks and Learning Systems* (Early Access), pp. 1–13, May 2023.
- [14] IEEE, 802.15.6-2012, "IEEE Standard for Local and Metropolitan Area Networks—Part 15.6: Wireless Body Area Networks," 2012.

Generating Combination of Biblical Baby Names using Recurrent Neural Network (RNN) and Optimization Comparison

1st Samuel Ady Sanjaya

Faculty of Technology and Informatics
Universitas Multimedia Nusantara
Tangerang, Indonesia
samuel.ady@umn.ac.id

2nd Suyoto

Faculty of Technology and Industry
Universitas Atma Jaya Yogyakarta
Yogyakarta, Indonesia
suyoto@uajy.ac.id

Abstract— Giving names to newborns in every country has its own uniqueness based on ethnicity, religion, race, culture, and names from past history. All parents in the world should give their baby a name with a good meaning because there is a saying that a name is a prayer for the child. However, these names have been used repeatedly based on trends every year, so it is very possible that humans on Earth have the same name. In this research, an approach will be made to generate names based on names from the Bible to make combinations of characters into new names. This Bible-based name is used because it is the most common source of names for Christians around the world. Names will be generated using a Recurrent Neural Network (RNN) with a comparison of several Optimizers such as Adadelta, Adagard, SGD, Adam, Adamax, and RMSProp. This comparison found that the 3 optimizers with the best loss function were Adam at 1,755, RMSProp at 1,814, and Adamax at 2,064. The optimizer with the best loss function can produce data with a good combination of consonants and vowels, but the optimizer with a bad loss function will produce a large number of combinations of consonants. This research resulted in a new approach for generating names from the RNN model generation with the best optimizer that can be used as inspiration for baby names.

Keywords—Biblical names, loss function, RNN, optimizer algorithm

I. INTRODUCTION

The name given by the parents is very meaningful, in some areas including Indonesia, it is stated that the name is a prayer. Parents will try to give the best name to their children. The giving of names is influenced by various ethnic, cultural, and religious factors and even references from singers or film actors. Most modern people in some areas have thought of a creative child's name to differentiate as other children's names across cultures or countries. In Indonesia, now many people adopt names that come from outside their area, even abroad to get a name that is unique and not the same as their surroundings. However, it cannot be denied that giving names in an area is also usually carried out from generation to generation, especially for last names or family names. In Catholic beliefs, it is also customary to use a baptismal name for a person's first name. Moreover, if the population of Christians in the world increases, it is recorded that there is a growth in Christian beliefs from 1900 as much as 9.1% to 59.1% in 2022 [1]. So if someone usually has three to five words for their full name, it's getting narrower to create their name. Moreover, coupled with the names in the Bible, the number is limited. Apart from naming because of belief, naming can also be based on culture as in the example of

research to see naming phenomena in Javanese culture. [2]. Cultures in other places, such as Japanese culture and Mexican culture, also influence the evolution of baby names in the future. [3][4]. Merging names is also a solution to obtain a unique name, for example, combining parents' names or combining two popular names. Based on this inspiration, it was found that the purpose of this research was to generate names using one of the deep learning methods. The use of deep learning can also be used to identify Arabic name recognition with a combination of several methods such as RBF, CNN, BiLSTM, or the use of B-GRU [5], [6]. There is also the use of deep learning to process text by considering a network embedding scheme with a text merging [7]. As for the use of another deep learning method that is good for text processing, namely Recurrent Neural Network (RNN). RNN is one of the best RNN models that can be used in various study cases, especially text processing [8]. Some examples of the use of RNN in text modeling are sentiment analysis, miss-information detection, text summarization, character-level language models, and text classification [9]–[12]. This research will also use the text processing approach to generate new names based on existing patterns, especially names in the Bible. Determining the best model is still not enough, so this research will also compare the best optimizer to be combined with RNN.

II. METHODS

A. Data Collection and Preprocessing

The dataset is taken from a collection of Biblical names consisting of the names of prophets, people, kings, and angels totaling 3668 names [13]. This dataset also has several attributes which explain the proper name to use, gender, date, and place of birth and death:

- 1) Name
- 2) IsProperName
- 3) Gender
- 4) BirthYear
- 5) DeathYear
- 6) BirthPlace
- 7) DeathPlace

From this dataset, information on each name regarding being born and died can be obtained, although there are still some names information is not complete. Not all names in the Bible are good for use as name references because the names are irrelevant or too long, so filtering will be done first according to the IsProperName attribute. The name attribute will also be cleaned beforehand by deleting special characters

and spaces, including names that have more than 2 words, the goal is that no special characters will be combined into a new name [14][15]. Furthermore, after the data is cleaned, it is followed by tokenization. Tokenization is the process of converting words or letters into numbers. After that, the tokenized name pool is then converted into a matrix so that all name pools can be processed by the model.

B. Modeling

Name generation will be carried out using a semi-supervised learning approach because even though there is no labeled data or class classification, the model still requires data training to be able to learn patterns from existing names. This model will focus on studying the patterns and structure of names in the Bible, then it will try to do generation based on existing names using one of the deep learning models, namely RNN. The data to be modeled by the RNN must first exist in the form of a name tokenization result matrix to be processed. This RNN model will be run in several epochs to get the optimal value [9]. The preparation of several hidden layers is also considered to get the appropriate evaluation value. An overview of the deep learning process using RNN is shown in Figure 1.

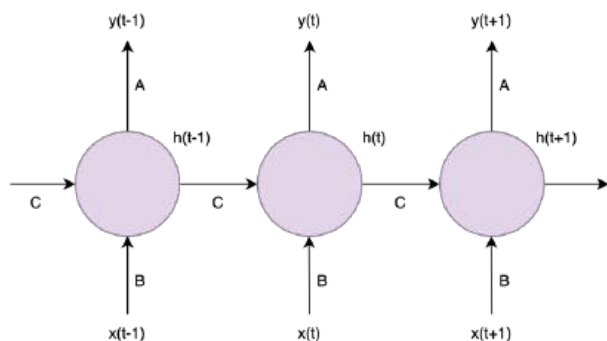


Fig. 1. RNN Schema

with equations,

$$h(t) = f_c(h(t-1), x(t))$$

where,

- $h(t)$: new state
- f_c : function with parameter c
- $h(t-1)$: old state
- $x(t)$: input vector at time step t

So x is the input layer, h is the hidden layer and y is the hidden layer. A , B , and C are the network parameters used to improve the output of the model. At any given time t , the current input is a combination of inputs at $x(t)$ and $x(t-1)$. The output at any given time is fetched back to the network to improve the output.

C. Evaluation of Different Optimizers

The RNN model will be evaluated using a loss function to get good evaluation results for a model. Loss functions are also commonly used for several other neural network-based deep learning models to be a good benchmark. The loss function is used to compare the target and predicted value with the aim of minimizing the loss between the predicted

and target output. So the value of the loss function will be better if the value is the minimum, lower better. Several evaluation metric comparisons for loss function also ensure that loss function is an evaluation metric that can be used for text processing or computer vision [16], [17]. Several related studies also use the loss function as the main metric for evaluating the model [18], [19]. After the evaluation metric is determined, the next step is to determine several optimizers that will be paired with the RNN with several comparisons according to Table 1 [20].

Table 1. Optimization Algorithm for Deep Learning

No	Optimizer		
	Algorithm	Advantages	Disadvantages
1	Adam	Fast and converges rapidly, high learning rate	Has probability to suffer a weight decay problem
2	Adadelata	Learning rate is not decay and the training is not stop	High computation cost
3	RMSProp	Has pseudo-curvature information and robust optimizer	Learning rate still manual
4	Stochastic Gradient Descent (SGD)	Run in less memory and less time to converges	High variance in model parameters

Based on Table 1, there are advantages and disadvantages of each optimizer which will be tested in this study with the RNN model. Besides the four optimizer algorithms, two other optimizer algorithms will be added, namely: Adamax and Adagard. So that later there will be a total of six optimizer algorithms used. From this comparison of optimizer algorithms, it will be tested which one has the best loss function and will also see how the performance of the name that has been generated from a combination of RNN and several optimizer algorithms will be seen.

III. RESULT AND DISCUSSIONS

A. Data Collection and Preprocessing

Based on the data that has been collected as many as 3668 lists of names derived from the Bible to be used. After selecting the data based on the eligibility of the name, the length of the name and also the name that does not contain special characters, 1789 names are obtained which will be processed. The name to be used has entered the transformation process to lower case, removing special characters and also filtering based on proper names. The following is an example of the dataset in Table 2 which consists of several attributes such as name, gender and place of birth.

Table 2. Dataset example

No	Attribute			
	Name	IsProperName	Gender	Birth Place
1.	Aaron	1	M	Egypt
2.	Abel	1	M	Eden
3.	Eve	1	F	Eden
4.	Nehusta	1	F	null
5.	Samuel	1	M	null

Even though there are several attributes, in this study only use name variables to make name combinations. Later the model will learn to do naming based on the names in the Bible. The distribution of names in this dataset has various name length variations, in Figure 2 it can be seen that most names have a character length between 4 to 8. Names with a total length of more than 12 characters are rarely used. One of the requirements that the name is proper is not too long, so names that are too long will be eliminated.

Name Length

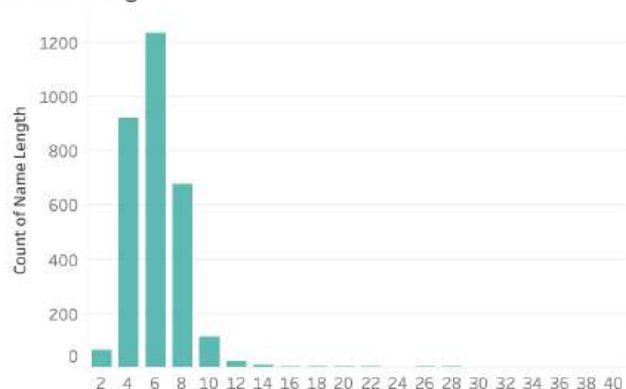


Fig. 2. Name Length Distribution

The list of names that will be used will then be tokenized to change the letters in each name into a token. The name that has been changed to lowercase, then tokenized, then the number of tokens is 26 letters, plus 3 characters. One letter will be represented by one number, then one name will be made into a number matrix which is a transformation from the previous tokenization.

```

aaron
arvadite
jaakobah
dinah
hamuel
kadmial
nahash
sallu
theudas
[[ 0  3  3 20 17 16  1  1  1]
 [ 0  3 20 24  3  6 11 22  7]
 [ 0 12  3  3 13 17  4  3 10]
 [ 0  6 11 16  3 10  1  1  1]
 [ 0 10  3 15 23  7 14  1  1]
 [ 0 13  3  6 15 11  7 14  1]
 [ 0 16  3 10  3 21 10  1  1]
 [ 0 21  3 14 14 23  1  1  1]
 [ 0 22 10  7 23  6  3 21  1]]

```

Fig. 3. Tokenization and Matrix Transformation

The results of the tokenization and matrix transformation are depicted in Figure 3. The name that has been changed to this matrix will be processed into the RNN model for training. An example of tokenization in Figure 3 is the name "aaron" will be changed to token "3 3 20 17 16". The number 1 in the matrix is the result of conversion to fill in the blank space so that the length is the same as the other names.

B. Modeling

Modeling with RNN is done using the Tensorflow library developed by Google. In Tensorflow, there is an RNN model that can be compiled with a combination of layers to get good results. In this study, there are 5 layers arranged to form an RNN. Starting from the first input layer with an embedding constraint of 16 and with an embedding initializer of 11. Then the RNN layer is formed into 512 units with relu activation. In the next three layers added dense with various kinds of activation with relu. The summary model used can be seen in Table 3 which consists of several layers.

Table 3. RNN model build layers

Layer	Layers		
	Layer (type)	Output Shape	Param#
1	Embedding	11, 16	464
2	RNN	11, 512	270848
3	Dense I	11, 128	65664
4	Dense II	11, 64	8256
5	Dense III	11, 29	1885

So the model obtained a total of 347,117 parameters and trainable parameters. This model is run with a batch size and epoch of 50. In the next stage, this model will be run several times using a different optimizer to get the best model based on loss function validation.

C. Evaluation of Different Optimizers

The evaluation carried out for this model is by softmax loss, or in other words, it is a combination of softmax activation combined with categorical cross-entropy with the following equation:

Softmax equation,

$$f(s)_i = \frac{e^{s_i}}{\sum_j^c e^{s_j}}$$

Cross Entropy equation,

$$CE = - \sum_i^c t_i \log(f(s)_i)$$

Categorical cross-entropy is used because the data uses multiple classes or the output can be in various letter combinations. The input will be converted first with encoding as in Fig 3.

Based on Table 4, there are three optimizers that have a loss function value that tends to be constant at around 3.36 compared to the epoch value. During the 50th epoch, the loss function results were still the same, so there was no significant loss function improvement. The Adadelta, Adagrad and SGD optimizer algorithms do not have a good learning rate because they fail to learn the pattern of constructing names.

Table 4. Loss functions with bad score

Epoch	Optimizer Algorithm		
	Adadelta	Adagard	SGD
10	3.36601591	3.36432672	3.3652246
20	3.36567068	3.36284447	3.36269403
30	3.36538386	3.36158252	3.36027384
40	3.37230325	3.36048651	3.35790443
50	3.3721447	3.35946655	3.35554481

Meanwhile, based on Table 5, there are three other optimizers that get good loss function values at the end of 50 epochs, namely Adam, Adamax and RMSProp. Adamax will look less good if the iteration is up to 30, but in the next epoch up to 50 it has a good result at number 2. Meanwhile the best performance is obtained by Adam optimizer with a number of 1.75 at the 50th epoch. For RMSProp it is also good at 1.81 but RMSProp has a good loss function value if only a few epochs are executed. This is evident from the first 10 epochs, which has the best loss function is RMSProp. So RMSProp can be an alternative to Adam if there aren't many epochs being run.

Table 5. Loss functions with good score

Epoch	Optimizer Algorithm		
	Adam	Adamax	RMSProp
10	2.04107976	3.37183929	1.94330382
20	1.91443384	3.3716104	2.07952929
30	1.85466015	3.37136817	1.94642556
40	1.80001903	2.12565208	1.86542249
50	1.75560939	2.06411362	1.81419039

The graph of the overall loss function comparison to the RNN model and various types of optimizers can be seen in Figure 4. From the figure, there are three constant models according to Table 4. In general, Adam is a good optimizer because it reduces loss function faster compared to RMSProp and also Adamax.

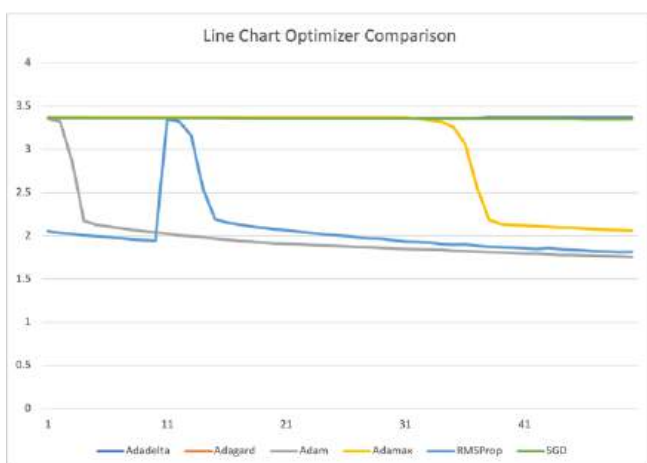


Fig. 4. Optimizer Comparison by Epochs

D. Result Example

The loss function table of the six comparisons has obtained the best based on Table 4 and Table 5. Next is the test of the

name generation results which have been grouped based on the loss function values in Table 6 and Table 7.

Table 6. Result example with bad loss function score

No	Optimizer Algorithm		
	Adadelta	Adagard	SGD
1.	xvairwokmq	qpaaevvi	cjbvpvskn
2.	hdampgptop	ecdoitmh	oyuaayyqtp
3.	pcroic	owpzuijlry	mgnlouylhc
4.	vaoswlfyee	ageuiyrjc	ybxxfpjqnqq
5.	opihsnd	reekeg	grubunedgb

Table 6 is a sample selection from the results of the running model and it is found that the names that are generated are not good because they are difficult to read because there are too many compositions of consonants compared to vowels. From the example in Table 6, it means that the value of the loss function is in line with the results of the sample name generation. The higher the value of the loss function, the more difficult it is for the model to be able to learn the name patterns in the dataset.

Table 7. Result example with good loss function score

No	Optimizer Algorithm		
	Adam	Adamax	RMSProp
1.	ehaci	balra	zahas
2.	qikeia	zacone	xumataa
3.	mleinahe	zhuala	eumaa
4.	lanonoa	zojoar	zaiare
5.	ucasaha	iaana	magra

Different results are shown in Table 7, where the generated names are easier to read because the composition of the consonants and vowels matches. However, for some people this name still feels foreign because the names in the Bible come from Greece, Israel, Egypt or other Middle Eastern countries. With the results in Table 7, the aim of this research has been achieved to generate new baby names using the right RNN and optimizer algorithm combination.

IV. CONCLUSIONS

Based on RNN modeling with a combination of several optimizers, it is found that the optimizer is very influential on the generation of new names. Optimizers with low loss function scores will be able to form names that are more suitable for the composition of vowels and consonants, such as the optimizer Adam, Adamax, RMSProp. But with a large loss function, it will also have an effect that names become unreadable because there are too many consonant letter compositions that are close together, such as in the Adadelta, Adagard, SGD optimizer. So that there is indeed a strong relationship between the loss function and the resulting name generation results. The smaller the loss function, the better the name generation results, and vice versa. The conclusion of this study, the best composition of the comparison of six optimizers is the application of RNN using the Adam optimizer with a loss function score of 1.755 in the 50th

epoch, followed by the RMSProp optimizer with a value of 1.814 and Adamax 2.064.

ACKNOWLEDGMENT

This research was supported by the Institution of Research and Community Services of Universitas Multimedia Nusantara. We thank our colleagues from Big Data Laboratory of Universitas Multimedia Nusantara part of the Information Systems Department who provided insight and expertise that greatly assisted the research.

REFERENCES

- [1] G. A. Zurlo, T. M. Johnson, and P. F. Crossing, "World Christianity and Religions 2022: A Complicated Relationship," *Int. Bull. Mission Res.*, vol. 46, no. 1, pp. 71–80, Jan. 2022, doi: 10.1177/23969393211046993.
- [2] E. K. Aribowo and N. Herawati, "Trends in Naming System on Javanese Society: A Shift From Javanese to Arabic," *Ling. Cult.*, vol. 10, no. 2, p. 117, Nov. 2016, doi: 10.21512/lc.v10i2.1730.
- [3] G. Unser-Schutz, "For Whom and by Whom Children Are Named: Family Involvement in Contemporary Japanese Naming Practices," *Genealogy*, vol. 3, no. 2, p. 29, May 2019, doi: 10.3390/genealogy3020029.
- [4] J. Thurtell and E. K. Merchant, "Gender-Differentiated Tarascan Surnames in Michoacán," *J. Interdiscip. Hist.*, vol. 48, no. 4, pp. 465–483, Feb. 2018, doi: 10.1162/JINH_a_01195.
- [5] N. Alsaaran and M. Alrabiah, "Arabic Named Entity Recognition: A BERT-BGRU Approach," *Comput. Mater. Contin.*, vol. 68, no. 1, pp. 471–485, 2021, doi: 10.32604/cmc.2021.016054.
- [6] A. Mousa, I. Shahin, A. B. Nassif, and A. Elnagar, "Cascaded RBF-CBiLSTM for Arabic Named Entity Recognition," in *2020 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI)*, Nov. 2020, pp. 1–5, doi: 10.1109/CCCI49893.2020.9256638.
- [7] H. Jiang, S. Fan, N. Zhang, and B. Zhu, "Deep learning for predicting patent application outcome: The fusion of text and network embeddings," *J. Informetr.*, vol. 17, no. 2, p. 101402, May 2023, doi: 10.1016/j.joi.2023.101402.
- [8] F. Nakhaei and M. Irannajad, "Application and comparison of RNN, RBFNN and MNLN approaches on prediction of flotation column performance," *Int. J. Min. Sci. Technol.*, vol. 25, no. 6, pp. 983–990, Nov. 2015, doi: 10.1016/j.ijmst.2015.09.016.
- [9] X. Dong and L. Qian, "Semi-supervised bidirectional RNN for misinformation detection," *Mach. Learn. with Appl.*, vol. 10, p. 100428, Dec. 2022, doi: 10.1016/j.mlwa.2022.100428.
- [10] I. Banerjee *et al.*, "Comparative effectiveness of convolutional neural network (CNN) and recurrent neural network (RNN) architectures for radiology text report classification," *Artif. Intell. Med.*, vol. 97, pp. 79–88, Jun. 2019, doi: 10.1016/j.artmed.2018.11.004.
- [11] M. Bani-Almarjeh and M.-B. Kurdy, "Arabic abstractive text summarization using RNN-based and transformer-based architectures," *Inf. Process. Manag.*, vol. 60, no. 2, p. 103227, Mar. 2023, doi: 10.1016/j.ipm.2022.103227.
- [12] S. A. Ali Shah, W. M. S. Yafooz, A.-H. M. Emara, and H. Sajid, "Generating Muslim Name using Character-Level Language Model in Deep Learning," in *2023 International Conference on Smart Computing and Application (ICSCA)*, Feb. 2023, pp. 1–6, doi: 10.1109/ICSCA57840.2023.10087673.
- [13] MetaV, "KJV Bible database with metadata MetaV," 2016, <https://github.com/arthurmilliken/KJV-bible-database-with-metadata-MetaV-> (accessed May 23, 2023).
- [14] D. A. Kristiyanti, A. H. Umam, M. Wahyudi, R. Amin, and L. Marlinda, "Comparison of SVM & Naive Bayes Algorithm for Sentiment Analysis Toward West Java Governor Candidate Period 2018-2023 Based on Public Opinion on Twitter," in *2018 6th International Conference on Cyber and IT Service Management (CITSM)*, Aug. 2018, pp. 1–6, doi: 10.1109/CITSM.2018.8674352.
- [15] D. A. Kristiyanti and M. Wahyudi, "Feature selection based on Genetic algorithm, particle swarm optimization and principal component analysis for opinion mining cosmetic product review," in *2017 5th International Conference on Cyber and IT Service Management (CITSM)*, Aug. 2017, pp. 1–6, doi: 10.1109/CITSM.2017.8089278.
- [16] Y. Tian, D. Su, S. Lauria, and X. Liu, "Recent advances on loss functions in deep learning for computer vision," *Neurocomputing*, vol. 497, pp. 129–158, Aug. 2022, doi: 10.1016/j.neucom.2022.04.127.
- [17] S. H. Rudy and T. P. Sapsis, "Output-weighted and relative entropy loss functions for deep learning precursors of extreme events," *Phys. D Nonlinear Phenom.*, vol. 443, p. 133570, Jan. 2023, doi: 10.1016/j.physd.2022.133570.
- [18] Y. Li, Y. Liu, and D. S. Williamson, "On Loss Functions for Deep-Learning Based T60 Estimation," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Jun. 2021, pp. 486–490, doi: 10.1109/ICASSP39728.2021.9414442.
- [19] Q. Gao, Y. Jiang, and S. Wang, "Improvement to Naive Loss Functions with Outlier Identifier," in *2020 IEEE Conference on Telecommunications, Optics and Computer Science (TOCS)*, Dec. 2020, pp. 203–205, doi: 10.1109/TOCS50858.2020.9339696.
- [20] N. Kumari, N. Sharma, and P. Singh, "Performance of Optimizers in Text Summarization for News Articles," *Procedia Comput. Sci.*, vol. 218, pp. 2430–2437, 2023, doi: 10.1016/j.procs.2023.01.218.

The Online Communication Station in City of Learning: Management of City Learning for Senior Citizen

^[1]Pastraporn Thipayasothorn, ^[2]Amornchai Chaichana

^[1,2]School of Industrial Education and Technology, King Mongkut's Institute of Technology Ladkrabang

^[1]pastraporn.th@kmitl.ac.th, ^[2]kcamornc@kmitl.ac.th,

Abstract— *The main aims of this study are threefold, that is, (i) exploring patterns of developing smart city and healthy city community, (ii) designing online communication station to support local-level city planning community for the elderly, and (iii) proposing policy for management of city planning for the elderly through developing online communication station based on survey conducted in Ladkrabang Sub-district, Ladkrabang District, Bangkok. This qualitative study was designed within three phases which included Phase One; focusing on survey on patterns of city development and community management of healthy city, Phase Two; designing of online communication station for city of learning, and Phase Three; offering policy-making for city planning management of the elderly which integrated an implication of online communication station and evaluating the outcome of the designed online communication station for city of learning in Phase Two. The results from this research showed significant implications for policy-making of city planning management especially for the target elderly, were also informative for stakeholders to utilize as practical guidelines in developing smart city and management of healthy city that required preparation for management and promote of activities to become city of learning.*

Index Terms—*Elderly community, learning city, management of community planning, online communication, online station*

I. INTRODUCTION

The concept of development for “smart city” and “city’s livability” has been recently paid considerable attention and taken into an account to be core principle in the notion of city development. Apart from numerous smart development systems, physical development plays a pivotal role in driving positive changes for Sustainable Development Goals (SDGs). Regarding this, United Nations identifies that establishment of sustainable cities and communities [1] as well as city’s livability that have various key indicators which many organisations around the world have focused and used for development such as security, public health, environment and culture, education, and infrastructure, etc.

Besides that, the trend of city development, smart design of city planning, and resilience can resolve problems occurred in cities and increase cities’ efficacy, and that could provide balance in the cities, enable us to use land more effectively in different ways such as mixed-use of area and buildings, transit-oriented development (TOD), support of resources for economics growth, resources of job opportunities [2] as well as public transportation system. From the aforementioned idea, riverside city in Ladkrabang sub-district, Ladkrabang district in Bangkok is the good example used the concept of learning city which is important for all Thai people in all regions across the country. The Thailand government takes this premise as one of national agendas which hopes to minimize equality and decentralize development to all regions of Thailand in order to uplift the nation’s development. Over the past decades, the government has encouraged and supported “City of Learning” continuously and sustainably.

City planning for community of the elderly has urged people in different cities to develop their cities to become livable, evolve infrastructure, facilitate learning which can more authentically respond to cities’ residents. Integrating the use of technology in supporting city of learning [3] and patterns for developing smart city and livable community management to manage city planning for the elderly at local level requires the need of research project on online communication station for learning city. Sequentially, it is expected to make policy for city planning to the elderly exploiting online communication station in learning city for city and communication development in the future.

II. RESEARCH OBJECTIVES

This research of creating online communication station in learning city for management of city planning among the elderly has the following aims including:

- 1) to explore patterns for smart city and community management of livable city;
- 2) to design online communication station in learning city in city planning for the elderly at local level; and
- 3) to make policy for city planning management for the elderly based on online communication station in city of learning.

III. CONCEPTUAL DESIGN

“Daily Activities” is highlighted as the activities for making daily happiness among the elderly which considers important focus on the elderly’s lives in reliving loneliness, creating happiness, and eliminating stress [4]. The activities should not only be supportive for the elderly to have healthy life in terms of physical and mental health, but also suits for

their physical limitation. Because of the assertion and the need of physical movement, family members and relatives of the elderly should, therefore, encourage them to do some free time activities. However, it is necessary to pay attention to safety, personal interests, and benefits to the elderly as well [5].

Development of local community can move forwards in any ways greatly needs essential components such as fundamental concept of local community development and dynamics of development underlying upon the current changes in economics, society, politics, environment as well as globalization.

IV. RESEARCH METHODOLOGY

The present study was carried out in the community of Ladkrabang sub-district, Ladkrabang district in Bangkok where was the second largest Eastern suburb (i.e., 77,406.1 Rai; the first largest area is Nongchok district) with a medium density of residence. Most are basin plain consisting of wide and large field surrounded with numerous canals. Majority of population living in the areas work as employees, sellers, farmers, and others.

The area in the communities of Ladkrabang sub-district, Ladkrabang district in Bangkok is located in the river banks of Khlong Prawet Buri Rom, covers 10.823 square kilometers with 30,601 residents. Most of them are Buddhist while other religions are minority. Some demographic information of population and samples who were the elderly in Ladkrabang sub-district from 13 communities are presented in Table 1.

Communities	Number of populations	Number of the elderly of ≤ 60
1. Luang Phrot and Than Liam	859	86
2. Kheha Nakhorn 2	3,000	296
3. Wat Sangkha Racha	1,190	120
4. Ruam Chai Phatthana	1,156	110
5. Pracha Ruam Chai	1,276	130
6. Mit Plook Sattha	2,455	240
7. Soi Thammanun	4,200	416
8. Lang Wat Lan Boon	619	62
9. Lang Boon Village	921	92
10. Somnuek Village	350	34
11. Thurd Satsana	390	40
12. Thawi Panya Rak	1,968	195
13. Rungruang Phatthana	337	33
Total	18,721	1,854

Source: <https://webportal.bangkok.go.th/ladkrabang>

The data in Table 1 indicates that there is 10.66% of the elderly in different communities in the target research site. For clear understanding information of the target research site, Fig. 1 below also purposively shows graphic locations of 13 communities on the map.



V.

Fig. 1 Ladkrabang Communities' Locations on Map

This study was designed with qualitative research approach using a case study which concentrated 13 communities as the target research site to develop innovation structure of administrative and managing community particularly to the elderly. The data obtained from this research were real in nature and embedded in methodology and instruments for a case study. Therefore, it is important to identify concise characters of key informants in order to shape the researchers' understanding of behaviours and social interaction in the context to explore patterns for social welfare management for the elderly and patterns for developing smart cities and managing livable cities. This study was well planned by the experts in the field of Research and Development who live outside the target research site to seek the current content knowledge and new content knowledge, implications of knowledge for analysis and innovation development, design smart application program for daily communication, and online communication station of city of learning. Two main activities undertaken from a step to a step were explained in the following explanation.

Activity One was conducted to understand the development model of an organized city and a livable community. There were 47 experts working in government sectors and the samples of the elderly in the communities:

- 1) Two experts in managing activities for the elderly from Ladkrabang district office;
- 2) Two experts in managing activities for the elderly from the Division of Community Development, Ladkrabang District Office;
- 3) Two experts in managing activities for the elderly from the Department of Older Persons (DOP), Ministry of Social Development and Human Security;
- 4) Two committees of the elderly club from Ladkrabang hospital; and
- 5) Thirty-nine representatives including community leaders, male and female elderly from 13 communities in Ladkrabang sub-district.

Activity Two designed an online communication station in the city of learning in the community for the elderly at the local level. This activity used research instruments based on case study to collect data with the target population. There were five experts both in the fields of information and technology and application program development comprising:

- 1) Two staff in the department of information from Ladkrabang district office; and
- 2) Three experts in application program development from King Mongkut's Institute of Technology Ladkrabang.

The qualitative data was analysed based on the interviews with the experts in information and technology and patterns for smart city development as well as livable city management, together with the information about environment and security in Ladkrabang sub-district, Ladkrabang district in Bangkok. Compilation of information about model instruments of the smart application program for daily communication and online communication station of city of learning is necessary for community management for the elderly in Ladkrabang sub-district [6].

V. RESEARCH FINDINGS

The data in the study of designing of online communication station (i.e., website) for city of learning to manage city planning for the elderly were taken from various resources such as the interviews with the experts in information and technology, the analyses of patterns for smart city development and city planning management for livable cities. Organizing of daily activities for the elderly are central to engagement from members and staff in the department of community development and social welfare in Ladkrabang sub-district, Ladkrabang district who were the main people in charge of data update on the designed website and policy-making for community planning management for the elderly [6]. The procedures in sending and receiving of data on the website are displayed in Fig. 2.

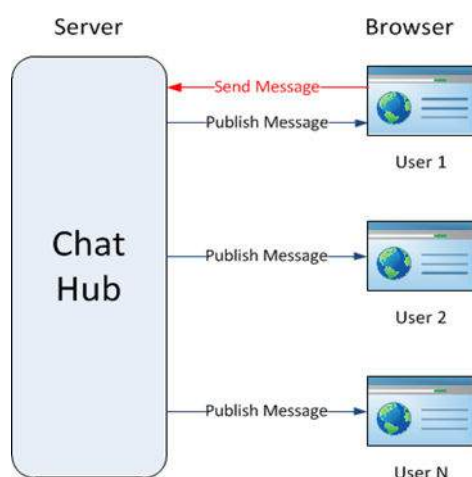


Fig. 2 Functioning Communication into the System

Consideration of the process in Fig. 2 based on the data from the research site with the target samples and research instruments basically enabled the researchers to conceptualisation of policy-making to deal with community planning management for the elderly with the use of online communication station in city of learning at local level.

VI. CONCLUSION

In this study, the survey from the communities could infer that Thailand is stepping into the aged society since there are more than 10% of senior citizen from total number of Thai populations. This phenomenon occurs due to advancement of medicine and public health service and decrease of birth rate. Hence, the elderly has received good and effective care and treatment. Also, some elderly who have good health condition can work and do some activities like other different aged-group individuals [7]. The development of appropriate online communication station (i.e., website) that responds to the needs of the elderly to use mobile for facilitating their routines and different daily activities. For the elderly those who are uncomfortable and face difficulties to make physical movement, the online communication station can prevent them from state of isolation and promote

sense of social belongings [8]. Consequently, the design of website for all individuals regardless of their disabilities and age can provide barrier-free for usage which represents the meaning beyond accessibility design on the principle of universal design which encompasses with the following components:

- 1) Equitable use
- 2) Flexibility in use
- 3) Simple and intuitive
- 4) Perceptible information
- 5) Low physical effort
- 6) Size and space for approach and use

The example of universal design are building's ramp and the use of door opener with close lever handle [7]. Regarding on the website, some features based on universal design include windows with description of pictures, video scripts, controlling functions of play, pause, and stop using keyboard. The term of universal design has been not only widely found in some academic disciplines in environment design, place, and objects, but also in the search for working people, the elderly, people with disabilities, the underprivileged with some limitation to use and gain an access to environment, place, and normal tools.

It can be concluded that the universal design suits for all people with some limitation. Because of this feature, this concept should be applied in design of product and environment. The universal design facilitates everyone to use products as independently as possible irrespective of their age and physical limitation. The key notion of universal design is greatly beneficial for the elderly and the disabled. Therefore, understanding of the elderly's activities significantly needs the concept of universal to consider designing online communication station or website to them.

VII. RECOMMENDATION

The data analysis in the present study which aimed to design online communication station or website for city of learning to manage community planning of the elderly on computer networking system could support and encourage the elderly to do recreational activities which is consistent with the information about activities in the elements on website, information receiving of users, and place for activities. It is essential to manipulate security with sending and receiving information. To conclude, it is important to consider some components to create online communication station or website in city of learning such as feature of usage of specific users, screen image of website, components in working system, and directions for development to keep up with changes of activities. The guidelines in developing in this study contributes to quality innovation and create added value to stakeholders relating to recreational activities among the elderly. The designs on the developed website in shown different ways in Fig.3 to Fig.7 below illustrated develop an efficacy communication structure integrating with integral concepts. The effectiveness of this innovation for effective communication is beneficial to connect communication of sending and receiving information on website for administrative center system in order to manage communities of the elderly who have lived in Ladkrabang district and other urban areas.



Fig. 3 Website's Home Page



Fig. 4 Activities' Locations in Ladkrabang Communities



Fig. 5 Locations on Google Map



Fig. 6 Activities for the Elderly in Ladkrabang Communities



Fig. 7 Activities' Plans of the Current Situations

REFERENCES

- [1] K. Malithong, "Educational technology and innovation," Chulalongkorn University Press, 2016.
- [2] J. Klaisang, "Media production and systematic use for learning in 21st century," Chulalongkorn University Press, 2017.
- [3] N. Saipin, "The role of communication infographics in Thai society," Dhurakij Pundit Communication Arts Journal, vol. 11, no. 2, pp. 145-179, 2017.
- [4] N. Wongyai, "A guide to developing digital literacy skills of digital native," Veridian Ejournal, Sipakorn University, vol. 10, no. 2, pp.1630-1642, May 2017.
- [5] Digital Government Development Agency, "Government website standard," Ministry of Digital Economy and Society, 2018.
- [6] P. Chaiwut, "Quality of life of elderly in Nongchangkhun sub-district, Mueang district, Lamphun province (Independent Study)," Chiang Mai University, 2013.
- [7] P. Thipayasothorn, A. Chaichana, T. Ruangrit, A. Phetsinchorn, and P. Munkong, "Design of aging health services surrounding Siridhorn hospital Bangkok," Paper presented at 6th PSPARN, Burapha University, 2022.
- [8] P. Youngyu, "Factors affecting life quality among the elderly in Donyang sub-district, Patiew district, Chomphon province (Unpublished Master's degree thesis)," Rajabhat Maha Sarakham University, 2019.

How Smart Technology Artificial Intelligence Robotics Algorithms (STARA) Awareness Influence Employee Perception on Companies in East Java, Indonesia?

Imam Mukhlis

Departement of Economic Development, Faculty of Economics and Business
Universitas Negeri Malang
Malang, Indonesia
Email: imam.mukhlis.fe@um.ac.id

Hariyono

Departement of History, Faculty of Social Science
Universitas Negeri Malang
Malang, Indonesia
Email: hariyono.fis@um.ac.id

Dediek Tri Kurniawan*

Departement of Management, Faculty of Economics and Business
Universitas Negeri Malang
Malang, Indonesia
Email: dediek.kurniawan.fe@um.ac.id

Rizky Firmansyah

Departement of Accountiong, Faculty of Economics and Business
Universitas Negeri Malang
Malang, Indonesia
Email: rizky.firmansyah.fe@um.ac.id

Adelia Shabrina Prameka

Departement of Departement, Faculty of Economics and Business
Universitas Brawijaya
Malang, Indonesia
Email: adelia.shabrina@ub.ac.id

*)Corresponding author

Abstract— STARA (smart technology, artificial intelligence, robotics and algorithms) technology is capable of replacing a broader variety of high-level skilled workers in non-routine cognitive tasks, not just low-skilled and unskilled human labor. STARA Awareness makes employees feel threatened by their work environment. This has an impact on the career goals of future. This research examines the effect of STARA awareness and job insecurity on turnover intention in profit-based organizations (companies) in East Java, Indonesia. East Java is the second largest economic contributor in Java Island in Indonesia. This study used the employees who work in profit-based-organization both state-owned-companies or private companies in East Java Province, Indonesia. To analyze the research data using Structural Equation Modeling - Partial Least Square (SEM-PLS) with SmartPLS software starting from validity and reliability tests, hypothesis testing, and mediation tests (indirect effects). This study is able to show that STARA Awareness has a significant effect on Job Insecurity and turnover intention in companies in East Java. However, STARA awareness does not have a significant effect on job satisfaction in companies in East Java, Indonesia. Job Insecurity has also been shown to have a significant effect on turnover intention and job satisfaction.

Keywords— STARA, job insecurity, East Java

I. INTRODUCTION

McKinsey's survey report entitled The State of AI in 2022, there are 50% of respondents from 1,492 global survey respondents who have adopted Artificial Intelligence (AI) in at least one of their business units. Stephen Hawking and Bill Gates warned that the number of unemployed is increasing due to the emergence of smart technology, artificial intelligence, robotics and algorithms or what is commonly called STARA [1], [2]. Experts also estimate that one in three jobs could be taken over by STARA by 2025 [3] Frey et al (2013); Thibodeau (2014). STARA can help businesses with capacity building, decision making, enhancing efficiency, and increasing production. According to [4],

STARA technology is capable of replacing a broader variety of high-level skilled workers in non-routine cognitive tasks, not just low-skilled and unskilled human labor. Several technologies have resulted in automation, reducing the importance of humans in the workplace. In the next one or two decades, most workers in transportation and logistics, office support and administration, as well as workers in high-risk production will be replaced by STARA [4].

Brougham et al (2018) said that STARA Awareness makes employees feel threatened by their work environment. This has an impact on the career goals of future and STARA Awareness had a negative impact on employees. Research conducted by [5] stated that STARA Awareness has a significant positive relationship on job insecurity. This is also supported by the findings of [6] which found that there is a positive relationship between STARA Awareness and job insecurity. According to [1] employees feel that STARA can replace their jobs. According to [2] STARA poses a big threat of unemployment to the world of work. Various studies on STARA awareness are being carried out. Several studies have been conducted in several countries, such as the United States, India, New Zealand, Australia, and China [1], [6]–[8]. Lingmont & Alexiou (2020) found that STARA awareness is able to increase employee perceptions of job insecurity or job insecurity. In addition, high STARA awareness will have an impact on employee turnover intention [1]. Li et al. (2019) explained that employee turnover intention will increase when they have high STARA awareness. In this case, employees feel that their job is threatened by the presence of STARA, so their intention to move from their current job or organization will increase. However, research on STARA awareness in Indonesia is arguably still not much done. Even though in Indonesia, STARA is not a new thing because many companies or governments in Indonesia have optimized its use.

In line with the situation above, research examining the effect of STARA awareness and job insecurity on turnover intention in profit-based organizations (companies) in East Java, Indonesia. East Java is the second largest economic contributor in Java Island in Indonesia with a contribution of 24.99% [9]. It is recorded that Jakarta is the largest contributor to the economy in Java with a contribution of 29.6%. Furthermore, West Java was 22.35%, Central Java was 14.54%, Banten was 6.96%, and Yogyakarta was 1.55% (Statistics Centre in East Java)[10]. East Java also has many industrial centers such as *Pasuruan Industrial Estate Rembang*, *Surabaya Industrial Estate Rungkut*, *Gresik Industrial Estate*, *Sidoarjo Industrial Estate Berbek*, and others.

II. LITERATURE REVIEW

Smart Technology Artificial Intelligence Robotics Algorithms (STARA) Awareness

Smart Technology, Artificial Intelligence, Robotics, and Algorithms or STARA awareness is a person's perception of a technological and information development in the form of intelligent technology, artificial intelligence, robotics, and algorithms that can replace human work more effectively and efficiently. The existence of STARA awareness makes employees think that STARA is able to replace their jobs [1]. To measure STARA awareness in this study, researchers used items developed by [1]. The STARA awareness item was developed based on job insecurity measurements by which were based on two indicators by [11] is a perceived threat of job loss and a sense of powerlessness to counteract the threat.

Turnover Intention

Turnover intention is an action taken by employees to resign permanently and voluntarily or involuntarily [12]. In this case, turnover can be resignation, transfer of division or units within one organization, and dismissal or death of member of organization. To measure turnover intention in this study, researchers used items developed by [13]. The turnover intention item is measured by four items developed by [14] and [15].

Job Insecurity

Job insecurity is a person's concern about his future work[16]. Job insecurity is not just about fear of losing a job, but is a subjective assessment of a lack of confidence in job stability and potential future job loss [17]. To measure job insecurity in this research object, researchers used items developed by [13]. This job insecurity item is adapted based on the measurement scale by [18] with cognitive and affective job insecurity indicators by [19].

III. RESEARCH METHOD

This research is classified as explanatory research in which this research will explain the causal relationship between the variables that will be examined through hypothesis testing [20]. In this study, researchers tried to explain the effect of STARA awareness (independent variable) on turnover intention (dependent variable) through job insecurity and job satisfaction (intervening variable). This study also adds expectation retraining as a moderator variable. This study used the employees who work in profit-based-organization both state-owned-companies or private companies in East Java Province, Indonesia. In collecting

data, an online questionnaire was distributed via a google form. The targeted respondents are employees from private companies in East Java Province. To analyze the research data using Structural Equation Modeling - Partial Least Square (SEM-PLS) with SmartPLS software starting from validity and reliability tests, hypothesis testing, and mediation tests (indirect effects).

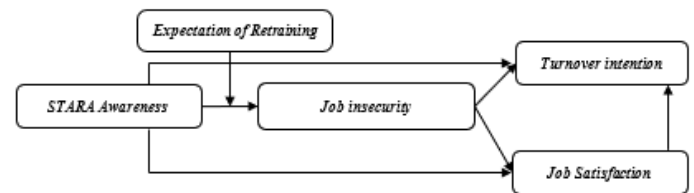


Figure 1. Research Method

Hypotheses:

- H1. STARA awareness significantly influenced on job insecurity
- H2. STARA awareness significantly influenced on job satisfaction
- H3. STARA awareness significantly influenced on turnover intention
- H4. job insecurity significantly influenced on turnover intention
- H5. job insecurity significantly influenced on job satisfaction
- H6. job satisfaction significantly influenced on turnover intention
- H7. Expectation of retraining can moderate the effect of STARA awareness on job insecurity
- H8. Job insecurity can mediate the effect of STARA awareness and turnover intention
- H9. Job insecurity can mediate the effect of STARA awareness and job satisfaction

In measuring the STARA variable, the researchers used four items from [1]. Job insecurity and turnover intention will be measured the items developed by [13]. Expectations of retraining are measured using items developed by [6]. Job satisfaction is measured using items developed by [21]. all variables used 6-likert scale with "1=strongly disagree" and "6=strongly agree".

IV. RESULT AND DISCUSSION

A. Respondent Demographics

This study used 536 respondents who consists of 54.1% male and 45.9% female. The respondents work in private companies about 73.1% and state-owned-enterprise about 26.9%. The majority of respondents have unmarried (single) status as much as 62.9%, 36% are married, and 1.1% are widow/widower. Based on working period, 43.7% of respondents have worked in their company about 1 - 3 years, 21.5% for less than 1 year 25.2% for more than 5 years, and 9.7% for 4-5 years.

B. Validity and Reliability

Based on validity test, four items (JIS4, JIS5, JIS6, JIS7) in job insecurity must remove from model because loading factor score below than 0.7. One item in jobs satisfaction (JS4) also have loading factor lower than 0.7. Table 2 showed that all of the item loadings were greater than the recommended value of 0.7 and were acceptable for further

analysis. From Table 2, composite reliability estimates were greater than the suggested threshold of 0.70 (ranging from 0.74 to 0.93), which indicated that the measurements were reliable. To assess the reliability of the measures, we calculated the composite reliability (CR) and average variance extracted (AVE). All estimated CR indices were greater than the recommended threshold of 0.7. The suggested threshold value of 0.5 for the AVE values also appeared exceeded by all constructs. Furthermore, the Cronbach's α values of all scales were acceptable and exceeded the recommended level of 0.7.

TABLE I. CRONBACH'S ALPHA, COMPOSITE RELIABILITY, AVERAGE VARIANCE EXTRACTED

	Cronbach's Alpha	CR	AVE	Decision
ER	0,852	0,910	0,771	Reliable
JIS	0,792	0,877	0,705	Reliable
JS	0,832	0,899	0,749	Reliable
STARA	0,896	0,928	0,763	Reliable
TI	0,875	0,915	0,730	Reliable

Sources: Data Processed by SmartPLS; Note: ER= Expected Retraining, JI= Job Insecurity, JS= Job Satisfaction, STARA=Smart Technology, Artificial Intelligence, Robotics and Algorithms, TI=Turnover Intention, CR=Composite Reliability; AVE= Average Variance Extracted (AVE)

C. Statistic Descriptive

Table 2 showed the results obtained for the entire sample. It is observed that job satisfaction ($M = 5.022$, $SD = 0.828$) had the highest mean scores, followed by expectation of retraining ($M = 4.757$, $SD = 0.951$), STARA ($M = 3.768$, $SD = 1.198$), turnover intention ($M = 3.055$, $SD = 1.330$), and job insecurity ($M = 2.187$, $SD = 0.888$) as the lowest mean score.

TABLE II. STATISTICS DESCRIPTIVE

Variable	Minimum	Maximum	Mean	Std. Deviation
STARA	1.00	6.00	3.768	1.198
ER	1.00	6.00	4.757	0.951
JIS	1.00	6.00	2.187	0.888
JS	1.00	6.00	5.022	0.828
TI	1.00	6.00	3.055	1.330

Sources: Data Processed by SPSS Note: ER= Expected Retraining, JI= Job Insecurity, JS= Job Satisfaction, OC= Organizational Commitment, STARA=Smart Technology, Artificial Intelligence, Robotics and Algorithms, TI=Turnover Intention

D. Hypotheses Test

In testing the measurement models, this study evaluated the explanatory power of the structural model with R^2 . R^2 value of the endogenous latent variables as a measure of model fit, because it provides information about the amount of variance of an endogenous latent variable explained by the model. The R^2 on job insecurity can show on Table 3.

TABLE III. R SQUARE

	R Square	R Square Adjusted
Job Insecurity	0,171	0,166
Job Satisfaction	0,394	0,391
Turnover Intention	0,245	0,241

Sources: Data Processed by SmartPLS

The results demonstrated on Table 4. This study confirmed the effect of STARA awareness on job insecurity ($\beta=0.110$, $t=2.277$, $p<0.05$, H1 accepted) and turnover intention in East Java companies ($\beta=0.281$, $t=6.607$, $p<0.05$, H3 accepted). In contrast, this finding cannot confirm the effect of STARA on job satisfaction ($\beta=-0.001$, $t=0.016$, $p>0.05$, H2 rejected). However, job insecurity had the

positive effect on turnover intention ($\beta=0.294$, $t=5.681$, $p<0.01$, H4 accepted) and negative effect on job satisfaction ($\beta=-0.627$, $t=20.555$, $p<0.01$, H5 accepted). In last, job satisfaction negatively influenced on turnover intention ($\beta=-0.154$, $t=3.196$, $p<0.01$, H6 accepted)

TABLE IV. HYPOTHESES TEST (DIRECT EFFECT)

	Path coefficient	t-value	P-value	Decision
STARA -> JIS	0,110	2,277	0,023	H1 accepted
STARA -> JS	-0,001	0,016	0,987	H2 rejected
STARA -> TI	0,281	6,607	0,000	H3 accepted
JIS -> TI	0,294	5,681	0,000	H4 accepted
JIS -> JS	-0,627	20,555	0,000	H5 accepted
JS -> TI	-0,154	3,196	0,001	H6 accepted

Sources: Data Processed by SmartPLS; Note: * $p < 0.10$, ** $p < 0.05$, *** $p < 0.01$, ER= Expected Retraining, JIS= Job Insecurity, JS= Job Satisfaction, OC= Organizational Commitment, STARA=Smart Technology, Artificial Intelligence, Robotics and Algorithms, TI=Turnover Intention

E. Moderation Test and Indirect Effect

To test the moderating effect, in Table 4, which postulated that expatriated training proved to moderate on the relationship between STARA and job insecurity, was not supported ($\beta = -0.135$, $t=2.416$, $p \leq 0.05$, H8 rejected).

TABLE V. MODERATION TEST

	Path coefficient	t-value	P-value	Decision
Moderating Effect 1 -> Job Insecurity	-0,135	2,416	0,016	H7 accepted

Sources: Data Processed by SmartPLS

From the indirect effect test, this study founded that job insecurity can mediate the effect of STARA on turnover intention ($\beta=0.032$, $t=2.124$, $p \leq 0.05$). In addition, job insecurity can also become a variable mediator for the relationship between STARA on job satisfaction ($\beta=-0.069$, $t=2.272$, $p \leq 0.05$).

TABLE VI. MEDIATION TEST

	Path coefficient	t-value	P-value	Decision
STARA -> JIS-> TI	0,032	2,124	0,034	H8 Accepted
STARA -> JS-> JS	-0,069	2,272	0,024	H9 Accepted

Sources: Data Processed by SmartPLS, JIS= Job Insecurity, JS= Job Satisfaction, STARA=Smart Technology, Artificial Intelligence, Robotics and Algorithms, TI=Turnover Intention

Discussion

STARA awareness will be positively associated with job insecurity

Statistical hypothesis test 1 in this study shows that STARA Awareness has a significant influence on job insecurity so that H1 is accepted. Based on these conditions, employees who have high STARA awareness will also have the perception that STARA can replace their jobs, which then creates job insecurity for employees [7]. This is in line with the findings [6] that STARA awareness can increase employee perceptions of job insecurity. In the next decade or two, most workers in transportation and logistics, office support and administration, and workers in high-risk production are replaced by STARA[4].

STARA awareness will be negatively associated with job satisfaction

This study failed to prove the negative effect of STARA on job satisfaction in companies in East Java so that H2 was rejected. This is contrary to the research of [1] explaining that STARA awareness is negatively related to job satisfaction. This happens because employees who have a higher awareness of STARA will cause low job satisfaction. Furthermore, in a study conducted by [5] said that STARA awareness is negatively related to job satisfaction, because the more employees feel the presence of STARA in their work, the level of job satisfaction that employees feel will be low. In research conducted by Meyer (2006) stated that STARA awareness has a significant effect on job satisfaction.

STARA awareness will be positively associated with turnover intention

Statistical hypothesis test 3 in this study shows that STARA Awareness has a significant effect on Turnover Intention so that H3 is accepted. High awareness of STARA will also have a high effect on turnover intention [8]. So when employees have awareness of the presence of STARA in their jobs, their intention to change jobs or turnover intention will also increase. This is in line with research [1] that the presence of Smart Technology, Artificial Intelligence, Robotics, and Algorithms or STARA will influence employees decisions to stay at their current jobs because they feel their jobs will be replaced by STARA. In addition, research [7] also found that STARA awareness significantly affects employee turnover intention.

Job insecurity will be positively associated with turnover intention

Statistical hypothesis test 4 in this study shows that Job Insecurity has a significant effect on turnover intention so that H4 is accepted. Based on research [7] there is an influence between job insecurity and turnover intention. Where the increasing threat of job insecurity will affect employee intention to look for work elsewhere or turnover intention. In line with research [22], [23] which shows the moderate level of influence between job insecurity and turnover intention. In addition, [5] also found that job insecurity has a positive influence on turnover intention, the findings show that employees who feel their jobs are not safe will intend to leave their companies. Thus, job insecurity is very likely to affect turnover intention.

Job insecurity will be negatively associated with job satisfaction

Hypothesis test 5 in this study shows that job insecurity has a significant effect on job satisfaction. These results are consistent with several previous studies that say that job insecurity has a significant effect on job satisfaction. As in research conducted by [24], said job insecurity has a negative influence on job satisfaction because if the level of feeling of job insecurity is high, the level of job satisfaction will decrease and vice versa. This statement is reinforced by another study by [25] which revealed that job insecurity has a negative and significant influence on job satisfaction, it is explained if the sense of job satisfaction in a company is low meaning that the level of job insecurity in a company is high. Furthermore, research conducted by [26] explained that job insecurity has a negative and significant effect on job satisfaction, which means that companies must be able to increase the level of job satisfaction by reducing job insecurity in the company. This statement is further

strengthened by research from [27] which states that job insecurity has a significant and negative influence on job satisfaction.

Expectation of retraining can moderate STARA awareness will be associated with job insecurity

Hypothesis test 7 in this study shows that Expected Retraining can moderate STARA Awareness of job insecurity. According to research data obtained by researchers, there is a significant effect showing that expected retraining can inhibit or support the emergence of job insecurity when employees feel good about STARA awareness. This is opposite with research conducted by [6] explaining that there is no direct influence between expected retraining and job insecurity. This happens because some employees expect retraining not only because they feel STARA awareness, but because they just want to prepare themselves more thoroughly for work. Therefore, it is said that there is interaction effect of STARA awareness and expected retraining on job insecurity. In the literature of [28][4] said the benefits of retraining and readiness are useful in new jobs or tasks that they will get in connection with job automation. The solution to reduce job insecurity in employees is by retraining [29].

Job insecurity can mediate the relationship between STARA awareness and turnover intention

Hypothesis test 8 in this study shows that STARA Awareness has an influence on Turnover Intention through Job Insecurity. This result is in line with several previous studies that say that STARA Awareness has an influence on Turnover Intention through Job Insecurity. As research conducted by [30] said that the STARA Awareness phenomenon could take over the work of employees in banking. In a study conducted by [31] it is explained that STARA Awareness has an influence on turnover intention through job insecurity, because employees who have a high skill level do not feel the threat (job insecurity) from the STARA phenomenon and while employees with low skill levels are more susceptible to experiencing a sense of desire to move (turnover intention). Roos et al (2017) revealed that STARA Awareness in companies will make the level of turnover intention high, this is because employees feel threatened (job insecurity) about the demand for smart technology is increasing [32].

Job insecurity can mediate the relationship between STARA awareness and turnover intention

Hypothesis test 9 in this study shows that STARA awareness has an influence on job satisfaction through job insecurity. According previous studies, the result say that STARA awareness has an influence on job satisfaction through job insecurity. Research by [6] explained that STARA awareness has the potential to have an effect on job insecurity and job satisfaction. Employees who have a low level of skills and education will feel worry (job insecurity) about their jobs in the future for fear of being replaced by new technology, besides age is also one of the reasons employees feel anxious and afraid. If employees feel job insecurity while working, then the level of job satisfaction felt by employees will also be high. Brougham et al (2018)

explained that employees who feel the presence of STARA Awareness will experience a low level of job satisfaction and this study did not find facts that say the level of job insecurity is high.

V. CONCLUSION

This study is able to show that STARA Awareness has a significant effect on job insecurity and turnover intention in companies in East Java. However, STARA awareness does not have a significant effect on job satisfaction in companies in East Java. Job Insecurity has also been shown to have a significant effect on turnover intention and job satisfaction. Except that, Expected Retraining is proven to moderate the influence between STARA Awareness and Job Insecurity. In testing mediator variables, job insecurity was proven to be able to mediate the relationship between STARA Awareness to turnover intention and STARA Awareness to job satisfaction. The development of technology in the business world is a natural thing, because this will greatly help companies to work effectively and efficiently. In addition, the company must be able to compete with its competitors and must not be left behind. This technological advancement can still be overcome, if employees are able to adapt to the technology that develops in the company.

VI. REFERENCES

- [1] D. Brougham and J. Haar, "Smart technology, artificial intelligence, robotics, and algorithms (STARA): Employees' perceptions of our future workplace," *J. Manag. Organ.*, vol. 24, no. 2, pp. 239–257, 2018.
- [2] J. Bort, "Bill Gates: People don't realize how many jobs will soon be replaced by Software Bots," *Bus. Insid.*, vol. 13, 2014.
- [3] P. Thibodeau, "One in three jobs will be taken by software or robots by 2025," *Computerworld*, 2014.
- [4] C. B. Frey and M. A. Osborne, "The future of employment: How susceptible are jobs to computerisation?," *Technol. Forecast. Soc. Change*, vol. 114, pp. 254–280, 2017.
- [5] D. T. Kurniawan, A. S. Prameka, and H. Praktikto, "STARA is in Sight: The Effect of STARA on Job Outcome through Job Insecurity Perception about Future Workplace on Private Companies in East Java, Indonesia," in *2022 3rd International Conference on Big Data Analytics and Practices (IBDAP)*, 2022, pp. 52–56.
- [6] D. N. J. Lingmont and A. Alexiou, "The contingent effect of job automating technology awareness on perceived job insecurity: Exploring the moderating role of organizational culture," *Technol. Forecast. Soc. Change*, vol. 161, p. 120302, 2020, doi: <https://doi.org/10.1016/j.techfore.2020.120302>.
- [7] D. Brougham and J. Haar, "Technological disruption and employment: The influence on job insecurity and turnover intentions: A multi-country study," *Technol. Forecast. Soc. Change*, vol. 161, p. 120276, 2020.
- [8] J. J. Li, M. A. Bonn, and B. H. Ye, "Hotel employee's artificial intelligence and robotics awareness and its impact on turnover intention: The moderating roles of perceived organizational support and competitive psychological climate," *Tour. Manag.*, vol. 73, pp. 172–181, 2019.
- [9] "Kominfo Jawa Timur," <https://kominfo.jatimprov.go.id/berita/jatim-sumbang-perekonomian-terbesar-kedua-di-jawa-sebesar-24-99>.
- [10] "www.bps.go.id."
- [11] L. Greenhalgh and Z. Rosenblatt, "Job insecurity: Toward conceptual clarity," *Acad. Manag. Rev.*, vol. 9, no. 3, pp. 438–448, 1984.
- [12] S. Robbins and T. Judge, *Organizational Behavior*. 2013.
- [13] H. S. Jung, Y. S. Jung, and H. H. Yoon, "COVID-19: The effects of job insecurity on the job engagement and turnover intent of deluxe hotel employees and the moderating role of generational characteristics," *Int. J. Hosp. Manag.*, vol. 92, p. 102703, 2021.
- [14] C. Cammann, M. Fichman, D. Jenkins, and J. Klesh, "The Michigan organizational assessment questionnaire," *Unpubl. manuscript, Univ. Michigan, Ann Arbor*, vol. 71, p. 138, 1979.
- [15] S. E. Seashore, E. E. Lawler, P. Mirvis, and C. Cammann, "Observing and measuring organizational change: A guide to field practice." New York: Wiley, 1982.
- [16] L. Greenhalgh and Z. Rosenblatt, "Evolution of research on job insecurity," *Int. Stud. Manag. Organ.*, vol. 40, no. 1, pp. 6–19, 2010.
- [17] X. Zheng, I. Diaz, N. Tang, and K. Tang, "Job insecurity and job satisfaction: The interactively moderating effects of optimism and person-supervisor deep-level similarity," *Career Dev. Int.*, vol. 19, no. 4, pp. 426–446, 2014.
- [18] J. Pienaar, H. De Witte, J. Hellgren, and M. Sverke, "The cognitive/affective distinction of job insecurity: Validation and differential relations," *South. African Bus. Rev.*, vol. 17, no. 2, pp. 1–22, 2013.
- [19] H. De Witte, "Arbeidsethos en jobonzekerheid: meting en gevolgen voor welzijn, tevredenheid en inzet op het werk," in *Van groep naar gemeenschap. liber amicorum prof. dr. leo lagrou*, Garant, 2000, pp. 325–350.
- [20] Sugiyono, *Metode Penelitian Bisnis (Pendekatan Kuantitatif, Kualitatif, dan R&D)*. Bandung: Alfabeta, 2014.
- [21] P. He, S. K. Murrmann, and R. R. Perdue, "An investigation of the relationships among employee empowerment, employee perceived service quality, and employee job satisfaction in a US hospitality organization," *J. Foodserv. Bus. Res.*, vol. 13, no. 1, pp. 36–50, 2010.
- [22] M. Sverke and J. Hellgren, "The nature of job insecurity: Understanding employment uncertainty on the brink of a new millennium," *Appl. Psychol.*, vol. 51, no. 1, pp. 23–42, 2002.
- [23] G. H. Cheng and D. K. Chan, "Who suffers more from job insecurity? A meta-analytic review," *Appl. Psychol.*, vol. 57, no. 2, pp. 272–303, 2008.
- [24] N. L. M. S. Devi and G. A. Sudibia, "Analisis pengaruh job insecurity, dan kompensasi finansial terhadap kepuasan kerja dan turnover intention: (Studi pada karyawan kontrak di bali dynasty resort)." Udayana University, 2015.
- [25] I. Setiawan and M. S. Putra, "Pengaruh Job Insecurity Terhadap Kepuasan Kerja Dan Turnover Intention Pada Karyawan Legian Village Hotel," *E-Jurnal Manaj. Univ. Udayana*, vol. 5, no. 8, p. 252971, 2016.
- [26] U. Udriyah, R. Riyadi, and R. R. B. Utaminingsy, "Pengaruh beban kerja, work-family conflict dan job insecurity terhadap kepuasan kerja serta dampaknya terhadap turnover intention di PT Binabusana Internusa Semarang," *Admisi Dan Bisnis*, vol. 18, no. 3, pp. 163–182w, 2018.
- [27] S. Ridho and A. R. Syamsuri, "Analisis Pengaruh Job Insecurity Dengan Kepuasan Kerja Dan Komitmen Organisasi Sebagai Variabel Intervening Terhadap Intensi Turnover," *INFORMATIKA*, vol. 6, no. 1, pp. 73–81, 2018.
- [28] M. Arntz, T. Gregory, and U. Zierahn, "The risk of automation for jobs in OECD countries: A comparative analysis," 2016.
- [29] O. Çınar, F. Karcıoğlu, and İ. Aslan, "The relationships among organizational cynicism, job insecurity and turnover intention: A survey study in Erzurum/Turkey," *Procedia-Social Behav. Sci.*, vol. 150, pp. 429–437, 2014.
- [30] L. Lestari and I. Djastuti, "Implementation of smart technology, artificial intelligence, robotics, and algorithms (STARA): A threat or opportunity for workers' future," *Rev. Manag. Entrep.*, vol. 4, no. 2, pp. 149–166, 2020.
- [31] H. Yu, C. Shum, M. Alcorn, J. Sun, and Z. He, "Robots can't take my job: antecedents and outcomes of Gen Z employees' service robot risk awareness," *Int. J. Contemp. Hosp. Manag.*, vol. 34, no. 8, pp. 2971–2988, 2022.
- [32] G. Roos and Z. Shroff, "What will happen to the jobs? Technology-enabled productivity improvement—good for some, bad for others," *Labour Ind. a J. Soc. Econ. relations Work*, vol. 27, no. 3, pp. 165–192, 2017.

Data-To-Question Generation Using Deep Learning

1st Nicole Rachel Koshy, 2nd Anshuman Dixit, 3rd Siddhi Shrikant Jadhav, 4th Arun V Penmatsa, 5th Sagar V Samanthapudi, 6th Mothi Gowtham Ashok Kumar, 7th Sydney Oghenetega Anuyah, 8th Gourav Vemula, 9th Patricia Snell Herzog, 10th Davide Bolchini

(nickoshy@iu.edu¹, anshumandixit1996@gmail.com², ssjadhav@iu.edu³, rpenmats@iu.edu⁴, svsamant@iu.edu⁵, moashok@iu.edu⁶, sanuyah@iu.edu⁷, gvemula@iu.edu⁸, pshezog@iupui.edu⁹, dbolchin@iupui.edu¹⁰)

Luddy School of Informatics, Computing and Engineering
Indiana University Purdue University Indianapolis
Indianapolis, United States

Abstract— Many publicly available datasets exist that can provide factual answers to a wide range of questions that benefit the public. Indeed, datasets created by governmental and non-governmental organizations often have a mandate to share data with the public. However, these datasets are often underutilized by knowledge workers due to the cumbersome amount of expertise and embedded implicit information needed for everyday users to access, analyze, and utilize their information. To seek solutions to this problem, this paper discusses the design of an automated process for generating questions that provide insight into a dataset. Given a relational dataset, our prototype system architecture follows a five-step process from data extraction, cleaning, pre-processing, entity recognition using deep learning, and questions formulation. Through examples of our results, we show that the questions generated by our approach are similar and, in some cases, more accurate than the ones generated by an AI engine like ChatGPT, whose question outputs while more fluent, are often not true to the facts represented in the original data. We discuss key limitations of our approach and the work to be done to bring to life a fully generalized pipeline that can take any data set and automatically provide the user with factual questions that the data can answer.

Keywords— *question generation, data analytics, semantic typing, meta categories, Sherlock, Spacy, semantic distance calculation, ChatGPT, LDA, NLP, deep learning, knowledge extraction, topic modeling*

I. INTRODUCTION

Since the advent of Natural Language Processing (NLP), accurate semantic typing of data attributes has been a crucial part of automating deep learning approaches in the exploration and identification of important characteristics of datasets. While subject matter experts can delve into most domain specific datasets with minimal assistance, big data and the ease of connecting disparate datasets into meaningful supersets has compounded the challenges we face with extracting meaningful information from such data in a timely manner.

The paradigm of generating questions to aid in the understanding of scientific datasets is a natural language problem that requires the use of a plethora of deep learning models for semantic typing, topic modeling and semantic distance calculation to determine the most relevant questions that can aid in the exploration of a dataset. Designing templates that can tackle a variety of data requires combining variables with differing semantic attributes within a dataset, along with their associated operators, before they can be converted into meaningful questions. This conversion process can follow a few methods. One method is to generate questions using only column names. Another combines column names and associated values to generate questions whose answers come directly from the data.

In this context, this project seeks answers to the question: How do we generate the maximum number of insights from a given dataset without spending an inordinate amount of time understanding and exploring the data? Can we develop pipelines that can generate preformulated queries whose answers can serve as the starting point in the exploration of a new dataset? Our work addresses the broader issues involved in this paradigm and uses datasets within the Indiana Datahub to build a program that can take a scientific dataset with its data dictionary and description, to generate questions that aid in understanding the data.

To this end, we previously worked on a question generation program that could automatically generate meaningful questions from official statistic datasets with a geo-spatial emphasis by using a semantic parser [1] to identify meta-categories, a Latent Dirichlet Allocation (LDA) model to identify domain specific keywords and a randomized query generator to generate natural language factual questions.

In that model, we were limited in the scope of the data we could analyze by the semantic parser we had created. To address this problem, in the current iteration of our work, we took this a step further by combining our approach with Sherlock, a deep-learning neural network model for semantic detection that has been trained on 686, 765 data attributes to identify and assign one of 78 semantic meta-categories for a given data column [2].

The core of our Question Generation from Datasets (QGD) pipeline is split into 5 phases that handle data extraction and cleaning, entity recognition, semantic categorization, similarity index calculation for correlating identified meta-categories with domain specific keywords and a question generation module that formulates natural language questions by semantically transforming a question string generated by combining related columns in the dataset with associated operators and dataset values. We have used the Indiana Hub [3] data repository as the data source for our project. Figure 1 provides an outline of the workflow of our pipeline.

One limitation of our program is Sherlock's ability to correctly detect semantic types. Being a pre-trained model that cannot take inputs, our ability to generate questions is dependent on being able to combine correlated column data that together would create a logical question whose answer can be found within the dataset.

Currently, we use a combination of pre-defined hardcoded templates that perform this operation by preferentially combining columns with location, categorical, datetime data with numeric data, so that the questions generated center around the non-numeric data. As an added exercise, we have also compared the output of our program against ChatGPT when it was provided with the same datasets to determine the differences in our approaches.

This research is based on work supported by the National Science Foundation under Grant IIS #1909845. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the NSF.

We briefly describe the functioning of our modules before discussing a comparative analysis of our output versus ChatGPT [4]. Our work can be found at <https://github.com/NicoleK286/Automated-Question-Generation>.

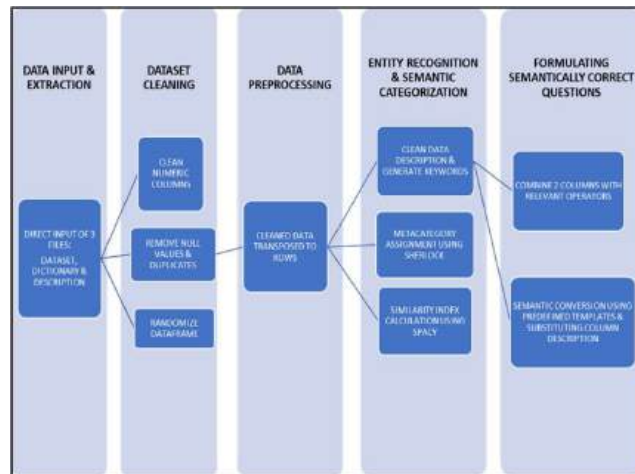


Fig 1: Flowchart of the functioning of the QGD pipeline

II. METHODOLOGY

A. Data Input

Our pipeline was designed using publicly available statistical datasets from Indiana Hub as use cases. For a given dataset, we input 3 files: the raw data, a data dictionary and a dataset description file. The preferred file format for raw data and the dictionary is *.csv and *.txt for description files.

B. Dataset Cleaning and Pre-Processing

The raw data file is subjected to a series of steps for cleaning. Since most statistical datasets would require cleaning of mainly numeric data values that contain suppressed or other non-numeric data, we focused our module on this premise.

One of the challenges we initially faced was correct datatype detection since any column with non-numeric data is treated as an object, in python. To circumvent this, we devised a technique to correctly classify numeric data by basing detection on the number of rows with numeric data in a column. If this count exceeds the alphanumeric count, the column is then reclassified as numeric. We then split the dataset into numeric and non-numeric columns and remove rows with non-numeric data from the relevant columns. A left join is performed to concatenate the amended dataset together before subjecting the entire data to null value and duplicates removal. In edge cases, where a numeric column has excessive suppressed or non-numeric data, the dataset will not be cleaned.

For input into Sherlock, the cleaned data is then stripped of column names and all rows in a column are transposed to a horizontal list of values. This transposed data is stored in a separate file, which is now ready for meta category identification.

C. Cleaning the Data Description File

The data description file is cleaned to replace all separators with spaces and then subjected to stopwords removal to eliminate common words that would hinder semantic distance

calculation. Stopwords removal is performed using the gensim module.

D. Semantic Categorization: Keyword Generation

The data description file is used to generate keywords that are compared with column names to calculate semantic distance indices which are later used in question prioritization. Topic modeling is performed on the cleaned wordlist obtained from the previous step using LDA [5] and gensim [6] to generate keywords, that are stored in a list for later use. Before distance calculation, all “ ” separators present in column names are removed so that individual words in each field can be compared with the keywords.

E. Entity Recognition: Sherlock Meta Category Assignment

The Sherlock model takes a horizontal list of values as input. From each row, meta category assignment is done by extracting and analyzing 1588 features ranging from cardinality, unique identifiers, semantic content and character distributions [2]. The output for each column is then used to look up associated operators which can be combined with the variables to generate questions.

The role of these operators is to provide added semantic context to a variable. For example, when formulating a question to a geographic location like a city or a particular county, like how many flu shots were administered within a region, the logical operator associated with the field should be “COUNT” which would translate into the “number of doses” on semantic conversion.

F. Semantic Distance Calculation for Question Prioritization Using Spacy

Semantic distance is a metric that is used to define the contextual likeness or similarity between two sets of words. The Spacy model in Python uses several techniques including part-of-speech (POS) tagging, Lemmatization, Inflection Morphology, and syntactic dependency parsing [7] to generate a similarity index for a pair of words. The higher the score for a pair of words, the more contextually similar they are considered.

In our program, we leverage these features to determine the relevance of each column by calculating the similarity index for each column with respect to all the keywords generated from the data description. The output stores the highest similarity index obtained for each column and the associated keywords in a list that is referenced for question prioritization later in the program.

G. Query Space Initiation and Question Formulation

The final module of our pipeline generates common language questions in two steps. First, we create a question string that concatenates 2 columns from the dataset together along with any one of the logical operators associated with the assigned meta category.

Second, we convert this string into a natural language question by replacing each column name with its description from the data dictionary and choosing the conversion template based on the operator associated with the second column. This is discussed in detail ahead.

H. Using Structured Statement Strings to Generate Question Combinations

The structured string that is used to generate the question series is designed in the form of a SELECT SQL statement using the following rules.

The first column selected either has the highest similarity index or one of the following associated meta-categories: Address, Location, county, age, gender, race, collection, category, result, city, club, year, day. The second column is randomly selected from the remaining list.

The meta category of the second column is checked, and an operator is randomly selected from the list of operators relevant to its semantic type. For example, if the second column has a meta category of "address", the operator selected could be one of the following – not equal to, equal to, count. Finally, a data value is randomly selected from the second column.

I. Semantic Conversion of Questions

The final phase in the pipeline is the conversion of the question string into semantically correct, common language questions. This involves first, substituting the column name with the column description, followed by selecting the question template for conversion by looping through a permutation of hardcoded combinations, based on the operator and meta category type.

The program can generate a series of up to 20 questions by randomly varying the question string parameters. Questions are generated in the following order of priority. The first two questions in a set will contain the column name with the highest similarity index. Next, cases where the second column is a numeric meta-category (rank, age, birth date, day, region, symbol) that have the comparison operators (min, max, average, count, sum) are given priority.

A set of 5 questions are generated in a single run with a total of 20 questions that can be created, for a single dataset. We are also testing various paraphrasers, such as Pegasus [8], to improve the quality of our output.

Examples of what our current output looks are shown below:

For a dataset that reports the "Prescription Related Claims of Mothers with Substance Use by Recipient County" [9].

Names of columns within the dataset: ['TOTAL PRESCRIPTIONS', 'TOTAL PRESCRIPTION COST', 'TOTAL PRESCRIPTIONS PREBIRTHING EVENT', 'PRESCRIPTION COST PREBIRTHING EVENT', 'PREBIRTHING MEDIAN COST', 'TOTAL PRESCRIPTIONS POSTBIRTHING EVENT', 'PRESCRIPTION COST POSTBIRTHING EVENT', 'POSTBIRTHING MEDIAN COST', 'COUNTY']

The meta categories predicted by Sherlock in order of columns are: ['code', 'address', 'day', 'address', 'symbol', 'code', 'address', 'depth', 'county'].

The question strings would have this pattern:

1st column: COUNTY

2nd column: POSTBIRTHING_MEDIAN_COST.

Question String 1: ['SELECT', 'COUNTY', 'FROM', 'test', 'WHERE', 'POSTBIRTHING_MEDIAN_COST', 'Minimum', 12.57]

Semantic Question 1: "What is the minimum Median cost of a prescription related to mothers two years after the delivery of the child with a live birth during the time period 07/2016 to 07/2017 and have a substance use related claim within 2 years prior to the delivery among all Mother's recorded county of residence as indicated closest to the delivery date?"

Paraphrased Question 1: "What is the minimum cost of a prescription related to mothers two years after the delivery of the child with a live birth during the time period of July 7, 2016 to July 7, 2017, and have a substance use related claim within 2 years prior to the delivery among all Mother's recorded county of residence".

1st column:

PRESCRIPTION_COST POSTBIRTHING_EVENT

2nd column: PREBIRTHING_MEDIAN_COST

Question String 2: ['SELECT', 'PRESCRIPTION_COST POSTBIRTHING_EVENT', 'FROM', 'test', 'WHERE', 'PREBIRTHING_MEDIAN_COST', 'Minimum', 11.92].

Semantic Question 2: "What is the minimum Median cost of a prescription related to mothers two years prior to the delivery of the child with a live birth during the time period 07/2016 to 07/2017 and have a substance use related claim within 2 years prior to the delivery among all Total cost of prescriptions prescribed to mothers two years after the delivery of the child with a live birth during the time period 07/2016 to 07/2017 and have a substance use related claim within 2 years prior to the delivery?"

Paraphrased Question 2: "What is the minimum cost of a prescription related to mothers two years prior to the delivery of the child with a live birth, and have a substance use related claim within 2 years prior to the delivery, among all Total cost of prescriptions prescribed to mothers two years?"

When we input a sample of the dataset into ChatGPT and ask it to generate questions, the result was a set of questions generated by combining a paraphrased likeness of the column names and associated values.

- Which county has the highest prescription cost per postbirthing event?
- How many counties are there in the dataset?
- What is the median cost of prebirthing medication in Blackford County?

However, when we input the dataset along with the data dictionary, the format of the questions shifts to focus on the text heavy descriptions and the questions start to resemble those generated by our module:

- What is the total number of prescriptions for mothers with a live birth during the time period 07/2016 to 07/2017 and have a substance use related claim within 2 years prior to the delivery?
- What is the overall total cost of prescriptions of mothers with a live birth during the time period 07/2016 to 07/2017 and have a substance use related claim within 2 years prior to the delivery?

- How many prescriptions were prescribed to mothers two years prior to the delivery of the child with a live birth during the time period 07/2016 to 07/2017 and have a substance use related claim within 2 years prior to the delivery?

In another example about “Hoosier Health and Well-being by County and Demographics” [10] where the column names are coded, the importance of the column description in interpreting the data becomes evident.

Keywords: ['demographics', 'health', 'county', 'hoosier'].

Column names: ['FEMALE COUNT', 'MALE COUNT', '18-22 COUNT', '23-27 COUNT', '28-32 COUNT', '33-37 COUNT', '38-42 COUNT', '43-47 COUNT', '48-52 COUNT', '53-57 COUNT', '58-62 COUNT', 'MARRIED COUNT', 'SEPARATED COUNT', 'DIVORCED COUNT', 'SINGLE COUNT', 'WIDOWED COUNT', 'BLACK COUNT', 'WHITE COUNT', 'CITIZEN COUNT', 'NO FORMAL ED COUNT', 'NO HS DIPLOMA COUNT', 'HS DIPLOMA COUNT', 'ATTENDING SCHOOL COUNT', 'QUESTION NUM', 'QUESTION COUNT', 'COUNTY COUNT', 'COUNTY', 'QUESTION LONG DESC', 'QUESTION SHORT DESC', 'ETL RUN TIMESTAMP'].

Meta categories predicted: ['address', 'day', 'ranking', 'ranking', 'day', 'day', 'ranking', 'ranking', 'ranking', 'ranking', 'ranking', 'ranking', 'address', 'ranking', 'position', 'address', 'address', 'ranking', 'code', 'day', 'rank', 'position', 'address', 'address', 'county', 'notes', 'name', 'address'].

Question Strings:

1st column: SINGLE_COUNT

2nd column: 28-32_COUNT

Question String 1: ['SELECT', 'SINGLE_COUNT', 'FROM', 'test', 'WHERE', '28-32_COUNT', 'Minimum', 351.0]

Semantic Question 1: “What is the minimum Count of survey respondents grouped by age as derived from provided date of birth among all Count of survey respondents grouped by provided marital status?”

Paraphrased Question 1: “What is the minimum Count of survey respondents grouped by age as a result of the provided date of birth among all respondents?”

1st column: FEMALE_COUNT

2nd column: ATTENDING_SCHOOL_COUNT

Question String 2: ['SELECT', 'FEMALE_COUNT', 'FROM', 'test', 'WHERE', 'ATTENDING_SCHOOL_COUNT', 'Maximum', 494.0]

Semantic Question 2: “Which Count of survey records where the respondent indicated gender was female has the highest Count of survey respondents grouped by self-reported as attending school at the time of the survey?”

Paraphrased Question 2: “Which Count of survey records has the highest number of respondents who said they were attending school at the time of the survey?”

For this example, ChatGPT generated only broad questions that could be formulated using easily readable column names and some descriptive attributes in the values:

- Which county has the highest number of males?

- What is the total number of people in the dataset?
- What is the total number of people who experienced not having enough money for food in the last 12 months?
- What is the total number of people who are citizens in the dataset?

Even after including the column names, most of the questions were very generalized, although a few provided some insight into the dataset:

- What is the meaning of the column 58-62_COUNT?
- What is the meaning of the column CITIZEN_COUNT?
- What is the range of the age group represented in the dataset?
- What is the total number of people represented in Bartholomew County?
- How many people in Blackford County have not completed high school?

A slightly different example is a financial dataset about resource allocation related to “COVID-19 Funds Transparency” [11].

Keywords: ['transparency', 'covid', 'funds'].

Column names: ['FUND SOURCE', 'BUSINESS UNIT', 'AWARD', 'EXPENSE', 'REMAINING', 'DESCRIPTION', 'FEDERAL AWARDING AGENCY', 'STATE AGENCY NAME'].

Meta categories predicted: ['day', 'symbol', 'elevation', 'elevation', 'elevation', 'collection', 'collection', 'collection']

Question strings:

1st column: FEDERAL AWARDING AGENCY

2nd column: EXPENSE

Question String 1: ['SELECT', 'FEDERAL AWARDING AGENCY', 'FROM', 'test', 'WHERE', 'EXPENSE', 'Minimum', 0.0]

Semantic Question 1: “What is the minimum \$ amount for transaction among all Federal aid agency name?”

Paraphrased Question 1: “What is the minimum amount for a transaction among all Federal aid agency names?”

1st column: STATE AGENCY NAME

2nd column: AWARD

Question String 2: ['SELECT', 'STATE AGENCY NAME', 'FROM', 'test', 'WHERE', 'AWARD', 'Minimum', 1068762.0]

Semantic Question 2: “Which Agency name has the least Spend limit against funding source for BU/project?”

Paraphrased Question 2: “Which agency has the lowest spend limit against funding sources?”

In this case, ChatGPT delved heavily into the column values to generate questions from the dataset alone.

- Which federal awarding agency provided funds for the 2020 TIII-Congregate Meals?
- How much was awarded under the 2020 Pandemic Unemployment Assistance Implementation Grants Admin (PUA)?
- Which state agency received funds under the 2020 Cooperative Agreement for Emergency Response: Public Health Crisis Response?

Including the column description, did not change the pattern of the questions in this case.

- What is the total amount of funding received by the State Department of Health for emergency response in 2020?
- How much money was allocated for Congregate Meals under the Families First Coronavirus Response Act?
- What is the purpose of the Pandemic Unemployment Assistance Implementation Grants Admin?

From the results, we can see that our program consistently produces factual natural language questions that shed light on the information within a dataset. Comparing our results with ChatGPT, we can see several similarities in the patterns. Since ChatGPT focuses more heavily on the text heavy section of a submission, the pattern shifts from using paraphrased column names to focusing on dataset values or column descriptors when they have larger text content, that can be used to independently formulate grammatically sound sentences.

III. DISCUSSION, LIMITATIONS, FUTURE WORK

In this paper, we have discussed an approach that enables the automated generation of factual questions from a given dataset. We take a dataset, its data dictionary and description as inputs and generate a prioritized set of questions by combining columns within the dataset with semantically relevant operators and converting this string into a natural language sentence using predefined templates and substituting column names with their descriptions. We also use a paraphraser to improve contextual quality. Our approach produced relevant questions for several public health datasets, whose answers could be found within the data.

Comparison with ChatGPT shows how the AI adopts different approaches depending on the content. However, as we have seen in the above examples, this has the disadvantage of generating vague questions when the fields definitions are not very enlightening from a descriptive standpoint. Since many statistical datasets have coded or abbreviated column names, this was the primary logic behind our rationale of substituting field descriptions for column names during the semantic conversion process.

A contribution of our approach is attending to a particularly important data type for public questions: geo-identifiers. Our application of the Sherlock package indicated that it improved upon prior models by automating detection of semantic types by including a greater number of data columns with high performance in predicting meaningful classifications. Interestingly, since most datasets available in the public domain include zip codes as a geographic identifier, Sherlock is particularly successful in identifying this geographic identifier, which can be meaningful in considering event attendance rates, for events such as for sporting events. However, since zip codes were developed to aid postal workers in delivering mail to physical addresses, this identifier is not particularly relevant in official datasets.

Most official statistics are reported within geographic units other than zip codes. Smaller geographic units are designed to approximate social and relational spaces, such as neighborhoods, and larger units align with geopolitical units, such as counties, metropolitan areas, and states [12]. To broaden the applications of semantic type detection in

advancing big data analytics, this project paid particular attention to the geographic identifiers in official statistics by initially advancing a separate module for Federal Information Processing Series (FIPS) data identifiers, independent of Sherlock. As a particular instance of a broader set of administratively constructed systems, the FIPS classification was developed to aggregate smaller geographies into large units without duplication [13]. Therefore, the FIPS units are key in harnessing the insights that public datasets can offer to citizens since geopolitical units have governance, elected officials, and administrative units related to the issues embedded in the data, e.g., public health.

A. Limitations

While the applications of our program are promising, there are some limitations in the current prototype that need to be addressed. First, our approach can handle several datasets within a particular public data hub, since our questions template was designed with these use cases, but we are limited in the scope of datasets we can process.

Secondly, the accuracy of the questions our program generates depends on several factors. Sherlock is a pre-trained heuristic neural network model that depends on repeated input to improve its predictions. We depend on this prediction to correctly pair operators with columns, which in turn affects our ability to correctly combine diverse semantic types. However, Sherlock is still limited in the variety of data it can correctly identify. We have observed several instances of it misidentifying important datatypes, such as datetime and float based numeric values being classified as addresses. Additionally, Sherlock is also inconsistent in its prediction of FIPS and other numeric data. With different datasets, FIPS was classified as an address or a symbol, depending on whether the datatype detected was a float or an integer. Additionally, there were several instances where numeric data was misclassified as addresses if the bulk of the data were 3-5 digit numbers that matched geo-identifiers.

Finally, the overall quality of our semantic conversion needs improvement. Our initial results with Pegasus showed that the ability of a paraphraser to improve the quality of our questions is dependent on the readability of the field descriptions available in the data dictionary. The more technical the description, the more grammatical and logical construction of the final question is affected. In such cases, all paraphrasers can do is rearrange the sentence, which may or may not improve its overall quality. To this end, we will be expanding the sentence libraries we work with, to determine if this can help improve the final output. Since ChatGPT encounters similar difficulties, as is evident from the second example we discussed, this may not be a problem that can be completely resolved.

B. Future Work

To mitigate the first limitation in future work, we are working towards building a dynamic template set that would be able to combine specific sets of semantic types together and expand on the associated range of operators to generate meaningful questions. This would facilitate scaling to include a larger set of publicly available datasets in future phases of this project. This includes developing a user interface that will facilitate knowledge workers in accessing datasets, while also collecting data regarding the inquiries requested. In the long-term, these data could be utilized to provide users with ratings regarding the most frequently asked questions, by

recommending the type of questions asked by specific types of users, and so on.

Second, future steps in this project are to continue testing the inaccuracies that Sherlock generates to find solutions to circumvent its mistyping of important meta categories such as datetime and addresses. To circumvent this, we are in the process of implementing regex checks for date formats so that we can convert such values into formats recognizable by Sherlock. Our initial attempts have shown that even after date conversion, Sherlock is unable to correctly recognize a converted date in all instances. For FIPS codes, we also plan to implement a conversion snippet which will check for county level FIPS codes and convert these to County names. Since Sherlock does a better job of recognizing County names correctly, we hope that this will improve the accuracy of our predictions. Future work in this direction will include expanding the diversity of the datasets we test and devising additional feasible solutions that can be integrated into future iterations of our pipeline.

As an added step, we may also consider experimenting with models such as SATO [14] which looks at column names along with their values to further contextualize semantic type detection.

In conclusion, improving the overall quality of the output of our program will be a multistep process that will require improving semantic type recognition, defining more combination of semantically diverse datatypes can be combined to generate logically correct questions and expanding the templates that are used for conversion. We have discussed several approaches we are exploring to this end, in this manuscript. Another aspect to consider is determining whether cleaning a dataset to remove non-numeric values from numeric columns is necessary. In our study of ChatGPT's methodology of generating questions, we also used datasets with null and suppressed values to observe how this would affect question generation since such data could be used to identify areas of low incidence and/or statistical significance. Our results showed that ChatGPT was able to successfully generate questions using such data which proved to be factually relevant in the exploration of the dataset. Therefore, another potential approach we could consider as our program evolves, is defining branch-off points where users may be given a choice to decide if they would like to receive results using a cleaned dataset or the original version. This would also require us to significantly reconstruct the current structure of our question templates. Success in this endeavor however, may result in a significant augmentation of our ability to delve into a dataset to generate statistically relevant questions.

C. Broader Significance

This project contributes to broader efforts to advance concept learning from semantic categories using natural language parsing and finds further support for NLP advancing generalization and accessibility through supervised deep learning [1]. Many curiosities held by knowledge workers and everyday people are advanced through the prompting of related inquiries. For example, a knowledge worker interested in public health issues could begin their inquiry by exploring the costs of prescription medications in relation to birth rates within specific locations, such as counties. After examining an available dataset, and the questions the dataset is prepared to answer, this knowledge worker could be prompted to ask

related questions, such as what the costs of prescription medications were in the years prior to and after the selected year, or in locations that are contingent to the selected county. To respond to this inquiry with fact-based information, it is necessary to parse data within relevant and semantically near categories. Additionally, the project advances on the prior work which developed the deep learning tools, Sherlock and Sato [2; 14], by applying this tool in service to big data analytics that are generated by the public sector and have the capability to inform knowledge workers and the public more generally [15].

REFERENCES

- [1] S. Srivastava, I. Labutov, and T. Mitchell, "Joint Concept Learning and Semantic Parsing from Natural Language Explanations," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 1527–1536. doi: 10.18653/v1/D17-1161.
- [2] M. Hulsebos et al., "Sherlock: A Deep Learning Approach to Semantic Data Type Detection," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, in KDD '19. New York, NY, USA: Association for Computing Machinery, Jul. 2019, pp. 1500–1508. doi: 10.1145/3292500.3330993.
- [3] "Indiana Data Hub." <https://hub.mph.in.gov/> (accessed Apr. 28, 2023).
- [4] ChatGPT, response to author query. OpenAI [Online]. <https://chatgpt.pro/> (accessed Apr. 28, 2023).
- [5] M. Kelechava, "Using LDA Topic Models as a Classification Model Input," Medium, Aug. 06, 2020. <https://towardsdatascience.com/unsupervised-nlp-topic-models-as-a-supervised-learning-input-cf8ee9e5cf28> (accessed Apr. 28, 2023).
- [6] R. Rehurek, "gensim: Python framework for fast Vector Space Modelling." [OS Independent]. Available: <https://pypi.org/project/gensim/> (accessed: Apr. 28, 2023).
- [7] "spaCy Linguistic Features." (n.d.) <https://spacy.io/usage/linguistic-features> (accessed Apr. 28, 2023).
- [8] "Pegasus." https://huggingface.co/docs/transformers/model_doc/pegasus (accessed Apr. 28, 2023).
- [9] "Prescription Related Claims of Mothers with Substance Use by Recipient County." The Indiana Data Hub. <https://hub.mph.in.gov/dataset/prescription-related-claims-of-mothers-with-substance-use-by-recipient-county>. (accessed on April 28, 2023).
- [10] "Hoosier Health and Well-being By County and Demographics." <https://hub.mph.in.gov/dataset/hoosier-health-and-well-being-by-county-and-demographics>. (accessed on April 28, 2023).
- [11] "COVID-19 Funds Transparency." <https://hub.mph.in.gov/dataset/covid-19-funds-transparency/resource/ecd4f142-6568-4f2b-88d3-73d5cd39f2a7>. (accessed on April 28, 2023).
- [12] US Census Bureau. (2021). "Understanding Geographic Identifiers (GEOIDs): What geographic identifiers are, how they are formed and what they are used for plus details on the differences between FIPS and GNIS codes," United States Census Bureau, <https://www.census.gov/programs-surveys/geography/guidance/geo-identifiers.html> (accessed Apr. 28, 2023).
- [13] "US Census Bureau Geographic Entities and Concepts," [Online]. Available: <https://www.census.gov/content/dam/Census/data/developers/geodata/concepts.pdf> (accessed Apr. 28, 2023).
- [14] D. Zhang, M. Hulsebos, Y. Suhara, Ç. Demiralp, J. Li, and W.-C. Tan, "Sato: contextual semantic type detection in tables," *Proc. VLDB Endow.*, vol. 13, no. 12, pp. 1835–1848, Aug. 2020, doi: 10.14778/3407790.3407793.
- [15] R. Connelly, C. J. Playford, V. Gayle, and C. Dibben, "The role of administrative data in the big data revolution in social science research," *Social Science Research*, vol. 59, pp. 1–12, Sep. 2016, doi: 10.1016/j.ssresearch.2016.04.015.

Can Herd Mentality be Negative? The Data Analysis

Shiu-Wan Hung

Department of Business Administration, National Central University
(Professor)
Tao-Yuan, Taiwan
shiuwan@mgt.ncu.edu.tw

Jyun-Hao Jian

Department of Business Administration, National Central University
(Doctoral Student)
Tao-Yuan, Taiwan
janhao753951@gmail.com

Abstract—There are consumers who purchase goods or services not out of necessity, but rather to conform with the majority. This behavior is known as herd mentality, where people blindly follow trends. The study conducted structural equation modeling (SEM) to analyze the data and to test the model. The study reveals that imitation is crucial for herd behavior and significantly affects economic reward, perceived enjoyment, and negative expectation disconfirmation. Enjoyment has a positive impact on one's intention to continue using uber, while economic reward has a negative effect. Negative expectation disconfirmation negatively impacts satisfaction, which, in turn, positively influences the consumer's intention to continue using uber. By providing detailed explanations of the enabler and expectation disconfirmation dimensions, this study contributes to the literature on technology acceptance research.

Keywords—herd behaviour, perceived value, expectation disconfirmation theory, structural equation modeling (sem)

I. INTRODUCTION

While consumers have gradually developed habits of online consumption, before placing an order they often use network evaluation as a basis for purchase decisions. When a product receives greater attention or wide popularity in society, more people will purchase it, and this swarming consumerist phenomenon can often be found on e-commerce platforms. This purchase motivation was driven by psychological induction, meaning consumers tended to follow trends with their purchases just to follow the consumer majority. The phenomenon of people blindly following trends prompts herd behavior.

The “herd behavior” refers to the tendency of people to make the same choices as others when making decisions [3]. [11] pointed out that consumers might not purchase products or services purely because they like or need them, meaning their purchase motivation is driven by peer pressure, and consumers may only reflect on their purchasing decisions after it is too late. On the other hand, some studies have pointed out that herd behavior helps people make money-saving decisions. By following the choices of others, they can save on decision-making costs, such as time and money spent on collecting information [19]. In addition, the choices of most people may reflect the wisdom of the crowd, and making the same decisions as others can make people feel more at ease [10].

However, with the advancements in information technology, new services are always being introduced. Compared with traditional industries, collaborative consumption is regarded as an economic behavior emerging from technological developments [6]. The costs of exchanging resources and purchasing goods decrease as people interact directly through peer-to-peer (P2P) networking [21]. P2P networking also helps to assimilate products and services within the B2B and B2C models [8], for example, Airbnb, which provides accommodation rental

services, and uber, which provides rideshare matching services.

Uber is a world-renowned collaborative consumption company that offers services for ridesharing through internet platforms that meet the demands of P2P network users. Unlike traditional taxis, uber fares are determined by the current supply and demand situation for rides; when the demand is high, the fare may be higher. When booking a vehicle, uber will provide ride information, driving route, estimated time of arrival, pickup location, and driver's personal information, as well as other passengers' previous ratings of the driver's service. After the trip, passengers can rate the driver according to their ride experience, and the system will review all opinions and evaluate the driver based on the driver's most recent trips.

Unlike the traditional transaction model, collaborative consumption has a ternary structure, which includes the platform provider, the peer service provider, and the customer. In the process of consumers' participation in “sharing,” they obtain economic reward, enjoyment, and value, which are all important factors that attract consumers to use such services [14]; [15]. However, this study asks whether there is herd behavior when consumers face such complex, emerging service types. In addition, this study examines the perceived value or negative expectancy disconfirmation that may be prompted by herd behavior, and whether it will affect the consumer's intention to use the service. Therefore, based on herd behavior, this research integrates the viewpoints of negative expectancy disconfirmation and perceived value, among others, and explores consumers' intent to continue using uber.

II. LITERATURE REVIEW AND HYPOTHESES DEVELOPMENT

A. Herd Behavior

Important dimensions of herd behavior are discounting one's own knowledge and beliefs in imitating others, while antecedents of herd behavior are observation of others because of incomplete information [11]. [5] argue that when an individual has incomplete information on a subject or faces uncertainty of choice, he/she may distrust their own judgment and follow the behavior of others. Before the request is confirmed, consumers only know the uber's price of the ride rather than types of the cars, ratings of the drivers, or wait time before the car arrives. Hence,

H1a: Incomplete information influences discounting one's own information.

H1b: Incomplete information influences imitating others.

[18] discussed behaviors of people who use information systems and found that people tend to discount their own judgment and knowledge in favor of what experienced premium users because they perceive premium users to have better information than others. Therefore, when an individual

is able to observe purchase behavior or intention of prior purchasers easily, he/she are prone to discount his/her own information and imitate others. Hence,

H2a: Observing others influences discounting one's own information.

H2b: Observing others influences imitating others.

B. Herd Behavior and Perceived Value

[10] studied mobile social applications and found that new users' perceived value of a mobile social application may be affected by regular users because they follow the opinions of other users and are more prone to take them seriously. Uber not only offers a ride service but also has a platform system. Users of uber have to request a ride through the mobile application, so other users' experiences and opinions of uber's mobile application may affect attitudes toward the platform. Hence,

H3a: Discounting one's own information influences platform quality.

H4a: Imitating others influences platform quality.

Enjoyment refers to the pleasure an individual senses when participating in a specific activity or indulging in a specific behavior [7]. [24] study the phenomenon of teenagers shopping with friends and discuss whether teenagers' attitudes toward retail stores. Their findings show that opinions and decisions of friends significantly impact a teenager's consumer choices about which products to buy and which shops to visit: teenagers are more likely to imitate their peers. However, making decisions similar to those many others have made helps to avoid unpleasant consumer experiences. Hence:

H3b: Discounting one's own information influences enjoyment.

H4b: Imitating others influences enjoyment.

Discounting one's own information and imitating others help consumers to overcome consumer uncertainty and to save time and money involved in collecting information [27]. Therefore, listening to opinions of others or imitating others helps consumers to make decisions quickly and efficiently and decreases the amount of time and money spent to experience an unfamiliar product or service; opinions and behaviors of others can influence consumers' decision making [5]. Hence:

H3c: Discounting one's own information influences economic reward.

H4c: Imitating others influences economic reward.

C. Perceived Value and Intention to Continue Using

[9] investigate users' intention to continue using e-learning services by conducting decomposed expectation disconfirmation theory. Their results suggest that perceived quality of e-service affects satisfaction, influencing intentions to continue using. Uber offers services that match a driver and a rider and fulfill the rider's transit needs. When consumers think the platform is able to fulfill their transit needs, they will become more willing to use uber in the future. Hence:

H5: Platform quality influences intention to continue using.

Enjoyment refers to the pleasure an individual receives when participating in a specific activity or engaging in a specific behavior [7]. Collaborative consumptions fulfill practical but also psychological needs. Prior research has

indicated that participating in collaborative consumption is a kind of hedonism, as consumers would continue participating in collaborative consumption, which allows them pleasure and playfulness [14]. Hence,

H6: Enjoyment influences intention to continue using.

[12] conduct empirical examinations of car2go, which provides a B2C car-sharing service, and airbnb, which provides accommodation-sharing service. Their studies indicate that cost saving positively influences satisfaction with a P2P system in the future and drives intentions to continue using. Furthermore, uber offers occasional discounts for consumers to save on costs, which enhances their willingness to use uber. Hence:

H7: Economic reward influences intention to continue using.

D. Herd behavior and Negative Expectation Disconfirmation

Expectation disconfirmation refers to the inconsistency between the expectation and the observed performance of a product or a service [1]. [26] pointed out that exposure to information about product performance characteristics will lead to the formation of consumers' specific beliefs or expectations of the product. With the expansion of uber services in various countries, consumers are quick to believe that its service quality is good. When consumers are affected by herd behavior and begin to discount their own information and imitate others, expectation disconfirmation will ensue. Hence:

H8a: Discounting one's own information influences negative expectation disconfirmation.

H8b: Imitating others influences negative expectation disconfirmation.

E. Negative Expectation Disconfirmation and Satisfaction

Prior research has found a relationship among expectation disconfirmation, satisfaction, and intention to continue using [1]. [25] discuss the online shopping environment and propose a model integrating expectation disconfirmation theory and users' satisfaction with purchasing goods online. The results suggest that expectation disconfirmation influences users' satisfaction. However, there are many aspects of uber that are problematic. For example, the service provided by uber is related to both demand and supply, so a request for a ride may not be confirmed if the demand is more than the supply, which may lead to a poor experience. Hence:

H9: Negative expectation disconfirmation influences users' satisfaction.

F. Satisfaction and Intention to Continue Using

[2] expectation-confirmation model of information system continuance, the intention to continue using information systems is highly associated with satisfaction. The more a user is satisfied with an information system, the more possible he/she will continue using it. [25] found that user satisfaction affects the willingness to participate in social platforms, and it is also regarded as an important indicator for measuring the success of social platforms. Uber is an innovative information technology platform. If consumers are satisfied with the uber experience, they will likely use it again. Hence:

H10: Users' satisfaction influences intention to continue using.

As described in the previous sections, the study integrates herd behavior, perceived value, and negative expectation disconfirmation and discusses the influence of herd behavior on consumers' satisfaction and intention to continue using. The research model is illustrated in Figure 1.

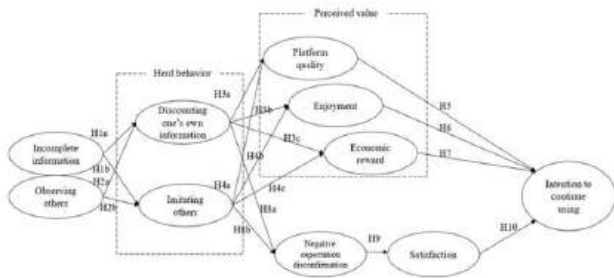


Figure 1. Research model

III. DATA ANALYSIS

The study collected samples from uber users who have ridden in uber cars in Taiwan. uber currently provides twenty-four-hour uber ride-hailing services in several major cities such as taipei, new taipei, and kaohsiung. Since Uber's platforms are applications of portable devices and websites, this study collected samples through a google online questionnaire. The study applies 7-point likert scales. Before the formal survey, the study conducted a pretest on 10 interviewers who are professors and students of department of executive master of business administration in a national university. The results of which show the quality of the questionnaire design and measurements to be acceptable. In the formal survey, this study collected 483 samples in total, yielding 421 completed questionnaires. Effective recovery ratio of questionnaire is 87%.

IV. DATA ANALYSIS AND RESULTS

A. Descriptive Statistics

Of the 421 valid samples collected for the study, 193 were male (45.8%) and 228 were female (54.2%). The age distribution of the samples with 21-30 combined are 58.43% of the sample, nearly half of the sample disposable income per month under 10,000 NT dollar (49.64%), the last time riding less than one month before are 46.8% of the sample.

B. Measurement Model

The study conducted confirmatory factor analysis (CFA) to test the internal consistency of the scales. [4] suggest factor loadings should exceed the value of 0.5, and Cronbach α should exceed the value of 0.7. Factor loadings of the constructs in the study ranged from 0.53 to 0.94, which exceed the recommended value of 0.5 [13]. Cronbach α coefficient of the latent constructs ranged from 0.71 to 0.94, which exceed the recommended value of 0.7. In addition, composite reliabilities (CR) exceeded value of 0.6 and the average variance extracted (AVE) for all constructs exceeded value of 0.5. These results suggest internal consistency and convergent validity of the constructs included in the proposed model are satisfied. The reliability measurement was illustrated in Table 1.

Table 1. Measurement model

Construct	Items	Factor loading	Cronbach's α	CR	AVE
1. Incomplete information	1	0.72	0.74	0.77	0.53
	2	0.78			
	3	0.67			

Construct	Items	Factor loading	Cronbach's α	CR	AVE
2. Observing others	1	0.82	0.77	0.85	0.59
	2	0.65			
	3	0.87			
	4	0.72			
3. Discounting one's own information	1	0.94	0.71	0.81	0.59
	2	0.62			
	3	0.71			
4. Imitating others	1	0.63	0.75	0.76	0.52
	2	0.77			
	3	0.75			
5. Economic reward	1	0.88	0.79	0.81	0.53
	2	0.85			
	3	0.57			
	4	0.53			
6. Enjoyment	1	0.57	0.90	0.89	0.64
	2	0.65			
	3	0.92			
	4	0.92			
	5	0.86			
7. Platform quality	1	0.80	0.91	0.91	0.67
	2	0.82			
	3	0.85			
	4	0.79			
	5	0.84			
8. Negative expectation disconfirmation	1	0.74	0.81	0.88	0.59
	2	0.83			
	3	0.79			
	4	0.63			
	5	0.82			
9. Satisfaction	1	0.92	0.87	0.92	0.74
	2	0.93			
	3	0.80			
	4	0.77			
10. Intention to continue using	1	0.90	0.94	0.94	0.77
	2	0.92			
	3	0.88			
	4	0.90			
	5	0.77			

According to [4], the scale accesses adequate discriminant validity when the correlation coefficients among construct pairs of latent variables were less than the square roots of AVEs. Measurements of discriminant validity were shown in Figure 2, showing enough discriminant validity of the analysis.

Variables	1	2	3	4	5	6	7	8	9	10
1	0.73									
2	-0.09	0.77								
3	-0.41	0.11	0.77							
4	-0.11	0.51	0.27	0.72						
5	-0.11	0.46	0.30	0.48	0.73					
6	-0.29	0.57	0.47	0.53	0.62	0.80				
7	-0.36	0.45	0.46	0.49	0.58	0.73	0.82			
8	0.58	-0.51	-0.44	-0.45	-0.55	-0.67	-0.70	0.77		
9	-0.28	0.54	0.49	0.58	0.62	0.77	0.77	-0.73	0.86	
10	-0.31	0.46	0.58	0.57	0.59	0.76	0.73	-0.71	0.83	0.88

The elements on the diagonal are the average variable variance extracted (AVE). Off-diagonal elements are the correlations among constructs. For discriminant validity, AVE should be larger than squared correlation between any pair of constructs; therefore, diagonal elements should be larger than off-diagonal elements.

Figure 2. Correlations matrix

C. Structural Model

The study conducted commercially available AMOS and SPSS to analyze the data and to test the model. Figure 3 illustrates results of structural equation model analysis, showing the model fit indices are acceptable ($\chi^2/df = 2.34$, GFI = 0.83, CFI = 0.92, RMSEA = 0.06, PGFI = 0.78). The model fitness detection indicators and results in Table 3.

Table 3. Model fitness detection indicators and results

Index	Metrics	Measurement model	References
χ^2/df	≤ 5	2.34	[16]
RMSEA	≤ 0.08	0.06	[22]
GFI	>0.8	0.83	[23]
NNFI	>0.9	0.92	[20]
PGFI	>0.5	0.78	[17]

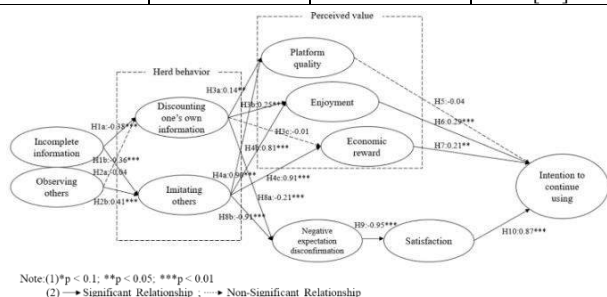


Figure 3. Results of structural equation model analysis

V. RESULTS AND CONSLUCIONS

This study focused on consumer intention to continue using the car-ride services of uber by employing herd behavior as our theoretical framework. The results suggest that enjoyment and negative expectation disconfirmation are important factors influencing consumer intention to continue using, while herd behavior has a positive impact on these two factors. Herd behavior occurs mainly when consumers know little about a product or a service and observes and trusts others' experience with the product or service. Incomplete information leads to the negative impact on consumers discounting their own information and positive impact on imitating others. The study explains both theoretical and practical implications.

This research contributes to the emerging collaborative consumption and consumer behavior literatures in a variety of ways. A sharing platform allows individuals to share their resources with others. The key to the success of a sharing platform is attracting enough users. Our study expands on a model that considers both the positive and negative factors of herd behavior and explores the intentions of consumers' continuous usage behavior. We emphasize the importance of herd behavior theory and clarify the key factors that users value. The results can provide specific strategic suggestions for sharing platforms such as uber, airbnb, foodpanda and further serve as an important reference for co-creating value.

ACKNOWLEDGEMENT

National Science and Technology Council, Taiwan R.O.C. under the grant NSTC 112-2410-H-008-015-MY3, has supported this work financially.

REFERENCES

- [1] A. Bhattacharjee, and G. Premkumar, "Understanding changes in belief and attitude toward information technology usage: A theoretical model and longitudinal test," *MIS quarterly*, vol.28, no.2, pp. 229–254, 2004.
- [2] A. Bhattacharjee, "Understanding information systems continuance: an expectation-confirmation model," *MIS quarterly*, vol.25, no.3, pp. 351–370, 2001.
- [3] A. V. Banerjee, "A simple model of herd behavior," *The quarterly journal of economics*, vol. 107, no.3, pp. 797–817, 1992.
- [4] C.Fornell, and D. F. Larcker, "Structural equation models with unobservable variables and measurement error: Algebra and statistics," *Journal of Marketing Research*, vol.18, no.3, pp. 382–388, 1981.

- [5] E. A. Walden, , and G. J. Browne, "Sequential adoption theory: a theory for understanding herding behavior in early adoption of novel technologies," *Journal of the Association for Information Systems*, vol.10, no.1, pp.31–62, 2009.
- [6] E. B. Swanson, "How information systems came to rule the world: Reflections on the information systems field," *The Information Society*, vol.36, no.2, pp. 109–123, 2020.
- [7] F. D. Davis, R. P. Bagozzi, and P. R. Warshaw, "Extrinsic and intrinsic motivation to use computers in the workplace," *Journal of Applied Social Psychology*, vol.22, pp. 1111–1132, 1992.
- [8] F. W. Horton Jr, "Why information management is an international issue," *The Information Society*, vol.8, no.2, pp. 119–126, 1992.
- [9] G. G. Lee, and H. F. Lin, "Customer perceptions of e-service quality in online shopping," *International Journal of Retail & Distribution Management*, vol.33, no.2, pp. 161–176, 2005.
- [10] H. Hong, M. Cao, and G. A. Wang, "The effects of network externalities and herding on user satisfaction with mobile social apps," *Journal of Electronic Commerce Research*, vol.18, no.1, pp. 18–31, 2017
- [11] H. Sun, "A longitudinal study of herd behavior in the adoption and continued use of technology," *Mis Quarterly*, vol.37, no.4, pp. 1013–1041, 2013.
- [12] I. P. Tussyadiah, and J. Pesonen, "Impacts of peer-to-peer accommodation use on travel patterns," *Journal of travel Research*, vol.55, no.8, pp. 1022–1040, 2016.
- [13] J. F. Hair, R. E. Anderson, R. L. Tatham, and W. C. Black, "Multivariate Data Analysis with Readings. (5th Ed.)," Englewood Cliffs, NJ: Prentice Hall, 1998.
- [14] J. Hamari, M. Sjöklint, and A. Ukkonen, "The sharing economy: Why people participate in collaborative consumption," *Journal of the Association for Information Science and Technology*, vol.67, no.9, pp.2047–2059, 2016.
- [15] J. Hwang, and M. A. Griffiths, "Share more, drive less: Millennials value perception and behavioral intent in using collaborative consumption services," *Journal of Consumer Marketing*, vol.34, no.2, pp. 132–146, 2017.
- [16] K. A. Bollen, "A new incremental fit index for general structural equation models," *Sociological methods & research*, vol.17, no.3, pp. 303–316, 1989.
- [17] L. T. Hu, and P. M. Bentler, "Cutoff criteria for fit indexes in covariance structure analysis: Conventional criteria versus new alternatives," *Structural Equation Modeling: A Multidisciplinary Journal*, vol.6, no.1, pp. 1–55, 1999.
- [18] M. C. Boudreau, and D. Robey, "Enacting integrated information technology: A human agency perspective," *Organization science*, vol.16, no.1, pp. 3–18, 2005.
- [19] M. Pingle, and R. H. Day, "Modes of economizing behavior: Experimental evidence," *Journal of Economic Behavior & Organization*, vol.29, no.2, pp. 191–209, 1996.
- [20] M. W. Browne, and R. Cudeck, "Alternative Ways of Assessing Model Fit," *Sociological methods & research*, vo;21, no.2, pp.230–258, 1993.
- [21] R. Belk, "You are what you can access: Sharing and collaborative consumption online," *Journal of business research*, vol.67, no.8, pp. 1595–1600, 2014.
- [22] R. P. Bagozzi, and Y. Yi, "On the evaluation of structural equation models," *Journal of the academy of marketing science*, vol.16, no.1, pp. 74–94, 1988.
- [23] S. A. Mulaik, L. R. James, J. Van Alstine, N. Bennett, S. Lind, and C. D. Stilwell, "Evaluation of goodness-of-fit indices for structural equation models. *Psychological Bulletin*," vol.105, no.3, pp. 430–445, 1989.
- [24] T. F. Mangleburg, P. M. Doney, and T. Bristol, "Shopping with friends and teens' susceptibility to peer influence," *Journal of retailing*, vol.80, no.2, pp. 101–116, 2004.
- [25] V. McKinney, K. Yoon, and F. M. Zahedi, "The measurement of web-customer satisfaction: An expectation and disconfirmation approach," *Information Systems Research*, vol.13, no.3, pp. 296–315, 2002.
- [26] V. Venkatesh, and S. Goyal, "Expectation Disconfirmation and Technology Adoption: Polynomial Modeling and Response Surface Analysis," *MIS Quarterly*, vol.34, pp. 281–303, 2010.
- [27] W. Duan, B. Gu, and A. B. Whinston, "Informational Cascades and Software Adoption on the Internet: An Empirical Investigation," *MIS quarterly*, vol.33, no.1, pp. 23–48, 2009.

Classification and Prediction of Pineapple Quality using Deep Learning

1st Zhehui Li
School of Computer Sciences
Universiti Sains Malaysia
Gelugor, Malaysia
lizhehui@student.usm.my

2nd Yung-Wey Chong
National Advanced IPv6 Centre
Universiti Sains Malaysia
Gelugor, Malaysia
chong@usm.my

3rd Mohd Nadhir Ab Wahab
School of Computer Sciences
Universiti Sains Malaysia
Gelugor, Malaysia
mohdnadhir@usm.my

4th Gin-Keat Lim
School of Chemical Sciences
Universiti Sains Malaysia
Gelugor, Malaysia
limgk@usm.my

5th Rahmad Dawood
Department of Informatics
Syiah Kuala University
Banda Aceh, Indonesia
rahmad.dawood@unsyiah.ac.id

Abstract—Pineapple, a popular tropical fruit with widespread consumption, necessitates precise classification and quality prediction to ensure its marketability and profitability. Prior research has explored various methods, including thermal imaging and acoustic spectroscopy, to evaluate pineapple quality, but these methods often require costly equipment. In this research, deep learning techniques are used for pineapple classification and quality prediction. The primary objective is to develop a model that integrates VGG16, LSTM, and GAN techniques to classify pineapple maturity based on color, shape, and texture while utilizing YOLO for grading. The dataset used in this study comprises images of MD2 pineapples obtained from Malaysian farms. The developed prototype extracts pertinent features from images captured by established farm standards in Malaysia. Through an automated process, the model effectively classifies the quality of these pineapples.

Index Terms—pineapple classification, artificial intelligence, smart agriculture, quality prediction

I. INTRODUCTION

Malaysia, a country situated in Southeast Asia, is known for its fertile soil and favorable climate, making it an optimal location for the cultivation of tropical fruits, including pineapple. Pineapple, a cholesterol-free and low-fat fruit holds significant demand in the global market [1]. In Southeast Asia, several countries such as the Philippines, Thailand, Malaysia, Laos, and Indonesia are prominent producers and suppliers of pineapple. According to the Food and Agriculture Organization (FAO), pineapple ranks as the third most popular tropical fruit worldwide [2].

The determination of pineapple quality is important due to customer preferences for sweet pineapple varieties over acidic ones. Various assessment methods, including sensory evaluation based on factors like smell, color, freshness, and texture, are used to measure pineapple quality. Typically, these methods require skilled personnel for manual execution. However, the manual classification of pineapples is prone to errors due to subjective judgments and the absence of

standardized criteria. The manual classification of fruits is affected by cost and labor constraints, prompting farmers to seek alternative solutions for fruit classification.

To address the aforementioned challenges, a deep learning model is proposed as a solution to assist farmers in the classification and prediction of pineapple quality, thereby ensuring consistent evaluations and minimizing human error [3]. The deep learning model encompasses various stages, including data acquisition, data pre-processing, feature selection, and model development. Once developed, the model can be implemented in an automatic sorting machine, enabling farmers to accurately predict the quality and grade of the fruits. In the context of pineapple grading using deep learning, different classifiers have been employed. For instance, the VGG16 model, a Convolutional Neural Network (CNN)-based model, is utilized for pineapple classification based on skin color, while YOLOv5 (You Only Look Once) is specifically designed for grading purposes.

II. RELATED WORK

In recent years, numerous studies have been conducted to automatically classify agricultural produce using various techniques. H. M. Zawbaa [4] proposes a machine learning method for automating fruit recognition, leveraging fruit shape and color features. The model consists of pre-processing, feature extraction, and classification stages. SIFT and color features are used for extraction, and K-NN and SVM are employed for classification. However, the model's efficacy on larger datasets has not been tested, and it does not address fruit grading and maturity stages, limiting its applicability for comprehensive MD2 pineapple evaluation.

[5] applies a deep learning approach using the VGG16 CNN architecture for pineapple type identification. However, the model's exceptional accuracy rate may indicate overfitting, and it does not account for the full diversity of pineapple

species or consider grading and maturity stages, which are essential for evaluating MD2 pineapples.

[6] aims to improve the generalization ability of fruit quality image classification by utilizing data augmentation through Conditional Generative Adversarial Networks (GANs). The authors construct a machine learning pipeline incorporating fine-tuning, transfer learning, and data augmentation with a generative model. They achieve promising results using a VGG16 model and data augmentation with synthetic images generated by a Conditional GAN.

While previous studies have made strides in using machine learning and deep learning techniques for fruit and flower classification, their focus has been scattered and not specifically targeted at the MD2 pineapple variety, particularly in terms of maturity and grading. Additionally, these studies have limitations such as potential overfitting, limited generalizability, and a narrow consideration of attributes that impact fruit quality. Despite these challenges, these studies offer valuable methodologies, including the use of CNN for classification tasks, and data augmentation using GANs to address data scarcity. Our research aims to bridge this gap by developing a comprehensive and accurate model for MD2 pineapple classification based on both maturity and grading.

III. METHODOLOGY

The pineapples used in this research are MD2 variety sourced from farms in Malaysia. The research is divided into four (4) stages namely data acquisition, data pre-processing, model development and testing. Figure 1 shows the research methodology. The research has separately trained two models for detecting the maturity and grade of pineapples. In data pre-processing, which utilized segmentation for handling maturity and labeling for a grade. The model creation process will be discussed in two separate subsections.

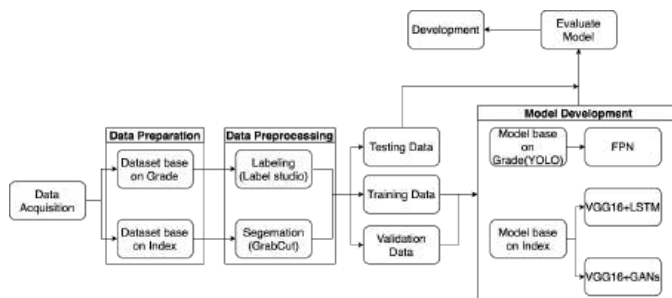


Fig. 1. Research methodology

A. Data acquisition

The pineapple sample is positioned on either the ground or a black box, and photographs are taken from above the sample using a camera. Natural light is used as the light source for capturing the images. Each pineapple specimen is photographed four times, with each shot taken at a 90-degree angle, allowing for a comprehensive 360-degree view of its scales. The images are captured using an iPhone 11 camera,

which has a focal length of $f = 26$ mm and an aperture of $f/1.8$. The image resolution is 3024 by 4032 pixels, and the images are stored locally, fully we take 2260 images. Figure 2 displays the images of the four sides of a pineapple fruit sample.



Fig. 2. Images of the pineapple fruit sample

The acquired dataset consists of 1431 MD2 pineapple images, representing different grades as listed below:

- Grade A: 500
- Grade B: 434
- Grade C: 420
- Rejected: 77

After segmentation, it was observed that some photos that met the grading criteria did not meet the requirements for maturity assessment. As a result, only 1300 photos were suitable for maturity training. Out of this dataset, there were 800 immature pineapple images and 500 mature images.

B. Data pre-processing

In this research GrabCut algorithm [7] is used for the segmentation of pineapple images. GrabCut is an image segmentation algorithm based on graph cut theory. It establishes a graph model within the image and utilizes an energy function minimization to separate the image into foreground and background regions. Each pixel is considered a node in the graph, and the edges between nodes represent the similarity between pixels. By minimizing the energy of the graph cut, accurate image segmentation can be achieved, making it suitable for this research.

Before applying the GrabCut algorithm, a region of interest (ROI) is defined for the image. This rectangular frame encompasses the target object, which in this case is the pineapple. By adjusting the position and size of the ROI, more precise object extraction can be obtained. Subsequently, a mask of the same size as the input image is created. This mask is used to label the foreground and background pixels in the image. Additionally, background and foreground models are constructed, which are continuously updated during the iterations of the GrabCut algorithm to improve the segmentation outcome.

The ROI, mask, background and foreground models are fed into the GrabCut algorithm. After multiple iterations, the values in the mask are updated to new values representing foreground and background regions. The mask values are modified to 0 or 1 and applied to the original image to

accomplish image segmentation. The segmented image is displayed as shown in Figure 3.

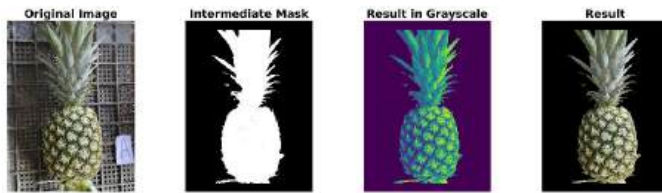


Fig. 3. Image segmentation process

After image segmentation, the subsequent step in our study is to classify the pineapples based on their maturity. The maturity standards are illustrated in Figure 4, which displays the assigned maturity levels for the general appearance determined by the scale color of the MD2 pineapple.

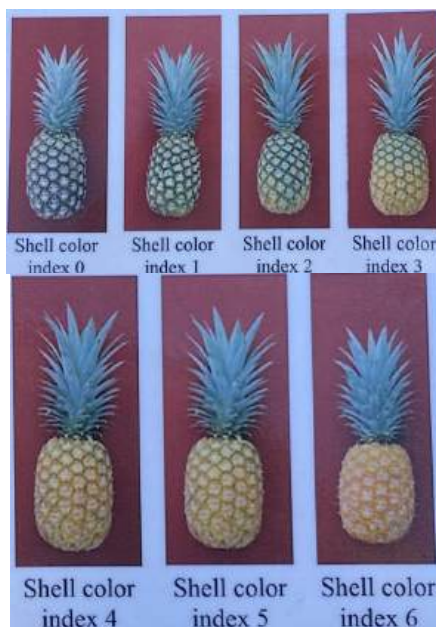


Fig. 4. Classification of maturity in MD2 pineapples

The maturity of pineapples is of significant importance as consumer demand is influenced by the desired time at which they would like to consume the fruit. The maturity of the pineapples is indicated by an index value ranging from 0 to 6. The classification is as follows (the classification standard base on the interview and feedback from local farmers) :

- **Immature:** Pineapples with an index ranging from 0 to 2. These fruits are considered not yet ready for consumption.
- **Mature:** Pineapples with an index ranging from 3 to 6. These fruits are considered ready for consumption.

The segmented images, accompanied by their respective maturity indices, compose the final dataset utilized for training our machine learning models. Notably, the dataset employs binary labels, classifying the pineapples as either mature or immature based on their indices. This binary classification

simplifies the problem at hand and facilitates the subsequent application of deep learning models such as YOLO and VGG16.

Subsequently, we proceed with image pre-processing for the dataset. The following operations are applied to the segmented images: RandomResizedCrop, which randomly crops and resizes the image to aid the model in learning features at different scales and perspectives, thereby enhancing its generalization ability; RandomRotation, involving random rotations of the image to enable the model to learn features at various rotation angles, thus improving its ability to handle rotation invariance; and RandomHorizontalFlip, which aids the model in learning features from horizontally flipped images, thus enhancing its capability to handle horizontal flip invariance.

For accurate labeling of these images, we utilized Label Studio, a flexible open-source data annotation tool. Each pineapple image was manually annotated by assigning a bounding box and a grade label: **A**, **B**, **C**, or **REJECTED**. The grade label is assigned based on the weight of the pineapple, as follows:

- **Grade A:** Represents pineapples of the highest quality, typically weighing between 1.5kg and 2kg.
- **Grade B:** Represents pineapples with good quality and weight, usually ranging from 1kg to 1.5kg.
- **Grade C:** Represents pineapples weighing less than 1kg, often exhibiting visible defects but still suitable for consumption.
- **REJECTED:** Signifies pineapples that fail to meet the minimum quality standards due to severe defects, poor overall condition, or weight not falling within the specified range.

An example of pineapple grade labels is depicted in Figure 5.

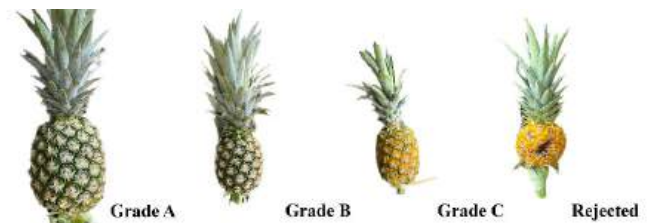


Fig. 5. An example of pineapple grade labels

A color code is assigned to each label to facilitate easy visual differentiation during annotation. The utilization of bounding boxes aids in delineating the specific area of the image that contains the pineapple, enabling the model to focus on relevant features while excluding irrelevant background information. The annotation process was carried out using Label Studio as illustrated in Figure 6.

Upon the completion of the annotation process, the dataset is split into training and validation sets. This annotated dataset is subsequently employed to train the YOLO models, enabling them to predict the grade of unseen pineapple images.

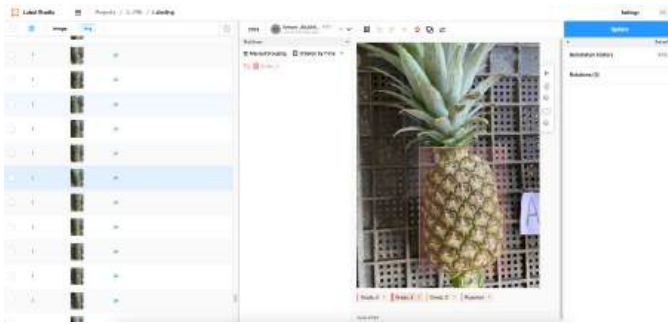


Fig. 6. The annotation process using Label Studio

C. Model Development

The model development is separated into two stages: firstly model to determine the grade and secondly model to determine the maturity.

1) *Model to determine the grade:* The model to determine the grade is built upon the YOLOv5 object detection algorithm, chosen for its superior speed and accuracy. The YOLOv5 model divides the input image into a grid and predicts bounding boxes and class probabilities for detected objects in each grid cell. The model was trained for a three-class task, with depth multiple and width multiple set to 0.33 and 0.50, respectively.

YOLOv5 consists of two parts: the backbone and the head. The backbone includes layers such as Focus, Conv, C3, and SPP modules to extract features from the input image. The head comprises layers like Conv, Upsample, Concat, and C3 modules to detect objects and assign classes using the extracted features. Figure 7 shows the structure of YOLOv5 utilized.

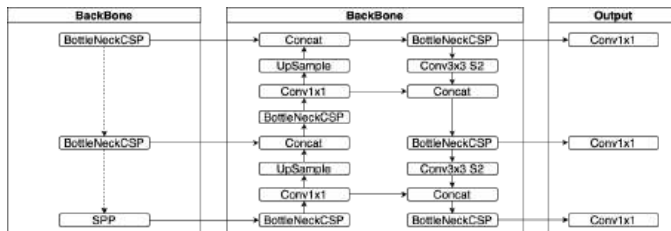


Fig. 7. Overview of YOLOv5 [8]

The model was trained using the Adam optimizer with a learning rate of 0.01 for 500 epochs. Three scales of anchors were used to handle objects of varying sizes across different layers of the feature pyramid:

- P3/8: [10,13, 16,30, 33,23]
- P4/16: [30,61, 62,45, 59,119]
- P5/32: [116,90, 156,198, 373,326]

2) *Model to determine the maturity:* A hybrid deep learning approach, combining CNN and Long Short-Term Memory(LSTM) networks, was implemented to determine the maturity of pineapples. This approach leveraged the spatial features extracted by the VGG16 model and the temporal dependencies handled by the LSTM model, effectively addressing

data volume limitations. Below is the specific implementation of the model:

- 1) **Data Augmentation:** Techniques such as RandomResizedCrop, RandomRotation, and RandomHorizontalFlip were used for data augmentation, enhancing the model's generalization ability.
- 2) **CNN (VGG16):** The original VGG16 model, consisting of 16 weight layers, was extended by adding a linear layer, ReLU activation, a dropout layer with a 0.5 probability, and another linear layer to output features.
- 3) **CNN with LSTM:** Building upon the modified VGG16 model, the VGG16 feature output served as input to an LSTM layer. Training the model involved utilizing a cross-entropy loss function and the Adam optimizer.

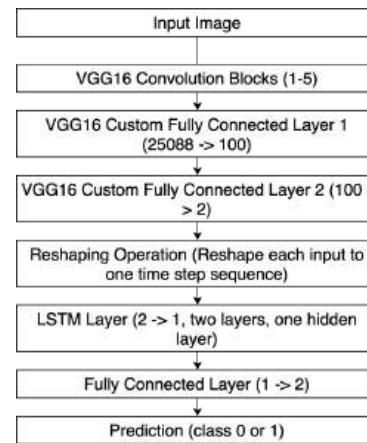


Fig. 8. VGG16+LSTM Architecture

- 4) **GAN:** GAN comprised a generator responsible for creating images from Gaussian-distributed random noise, indistinguishable from real pineapple images. The generator consisted of a deep convolutional neural network with transpose convolution layers and batch normalization layers. The input noise vectors were 100-dimensional, and the output images had a size of 224x224 with three color channels.

IV. IMPLEMENTATION

In this research several tools are used in the data pre-processing and model development namely OpenCV library in Python 3.8, Matplotlib and PyTorch framework. Once the data pre-processing was completed the obtained dataset was separated based on the grade and maturity. The data to train the model on the grade was divided as shown in Table I:

TABLE I
DISTRIBUTION OF GRADE DATASET

	Training Set	Validation Set	Test Cases
Grade A	350	100	50
Grade B	300	100	34
Grade C	290	100	10
Rejected	50	20	7

The maturity dataset consists of 1300 MD2 pineapple images, with an imbalance of 800 immature images and 500 mature images. After generating 300 maturity images using GANs, the datasets were balanced for the two categories. The distribution of images in the training and validation sets is presented in Table II.

TABLE II
DISTRIBUTION OF NEW MATURITY DATASET

	<i>Train</i>	<i>Validation</i>	<i>Test</i>
Immature	500	200	100
Mature (augmented)	500	200	100

The model's performance was assessed using precision, recall, and mAP (Mean Average Precision) metrics. Precision represents the ratio of accurately predicted positive samples to the total number of samples predicted as positive. Recall, also known as sensitivity or true positive rate, indicates the ratio of correctly predicted positive samples to the total number of actual positive samples. mAP is a widely utilized evaluation metric in object detection tasks. It calculates the average precision and recall values across various thresholds.

V. RESULTS AND DISCUSSION

Table III presents the performance of the model in classifying the grade of pineapple. The grading model, utilizing the YOLOv5 object detection algorithm, demonstrated exceptional performance with precision, recall, and mAP scores of 97.84%, 96.67%, and 96.97% respectively. These results indicate the model's strong ability to accurately detect and classify pineapples into their respective grades.

TABLE III
PERFORMANCE METRICS OF THE MODEL

Metric	Value (%)
Precision	97.84
Recall	96.67
mAP	96.97

Table IV illustrates the performance of the model in classifying the maturity of pineapples. The modified VGG16 model exhibited notable performance, achieving an accuracy of approximately 94.5%. However, when combining the VGG16 model with the LSTM layer, there was a decline in accuracy, resulting in a recorded accuracy of around 84.1%. Further analysis suggested that this decrease in accuracy could be attributed to the LSTM model processing an excessive number of feature values.

To address this issue, GAN is deployed for data augmentation instead of combining the model with the LSTM layer. This decision led to a significant improvement in performance, with the VGG16 + GANs model achieving an impressive accuracy rate of 96.9%. These results indicate that GANs can effectively enhance the model's classification accuracy by serving as a valuable tool for data augmentation in our pineapple recognition task.

TABLE IV
MODEL TRAINING RESULTS

Model	Accuracy (%)
Modified VGG16	94.5
VGG16 + LSTM	84.1
VGG16 + GANs	96.9

VI. CONCLUSION

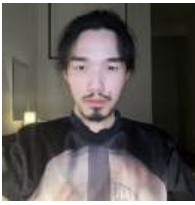
The research successfully developed two machine learning models to automate the classification of pineapples based on grades and maturity index. The grading model, utilizing the YOLOv5 object detection algorithm, demonstrated excellent performance with high precision, recall, and mAP scores. By using a dataset of 1431 MD2 pineapple images, the approach can enhance the efficiency and accuracy of grading in commercial settings. The maturity index model adopted a hybrid deep learning approach by combining CNN with GANs for data augmentation, addressing data imbalance and generating high-quality images. This model assists growers in optimizing harvest times, reducing wastage, and maximizing crop yield.

ACKNOWLEDGMENT

This research was supported by Keio University under Community Research Grant number 304/PNAV/6501371/K164. Special thanks to the Malaysian Pineapple Industry Board (LPNM) and Federal Agricultural Marketing Authority (FAMA) for their assistance in facilitating our visit to pineapple and guava farms and enabling us to capture the dataset.

REFERENCES

- [1] M. Mahmud, R. Abdullah, and J. Yaacob, "Effect of vermicompost amendment on nutritional status of sandy loam soil, growth performance, and yield of pineapple (ananas comosus var. MD2) under field conditions," *Agronomy*, vol. 8, no. 9, p. 183, 2018.
- [2] Home — Food and Agriculture Organization of the United Nations, <https://www.fao.org/3/cb9412en/cb9412en.pdf> (accessed Jul. 4, 2023).
- [3] M. Alirezaei, D. Zare, and S. M. Nassiri, "Application of computer vision for determining viscoelastic characteristics of date fruits," *Journal of Food Engineering*, vol. 118, no. 3, pp. 326-332, 2013, doi: 10.1016/j.jfoodeng.2013.04.023.
- [4] H. M. Zawbaa, M. Abbass, M. Hazman, and A. E. Hassenian, "Automatic Fruit Image Recognition System based on shape and color features," *Communications in Computer and Information Science*, pp. 278-290, 2014.
- [5] Amjad H. Alfarra, Lamis F. Samhan, Yasmin E. Aslem, Marah M. Almasawabe, Samy S, and Abu-Naser, Classification of pineapple using Deep Learning, 2021. [Online].
- [6] J. J. Bird, C. M. Barnes, L. J. Manso, A. Ekárt, and D. R. Faria, "Fruit Quality and defect image classification with conditional Gan Data augmentation," *Scientia Horticulturae*, vol. 293, p. 110684, 2022.
- [7] N. Xu, B. Price, S. Cohen, J. Yang, and T. Huang, "Deep GrabCut for object selection," *Proceedings of the British Machine Vision Conference 2017*, 2017.
- [8] Ultralytics, "Ultralytics/yolov5: Yolov5 in PyTorch" <https://github.com/ultralytics/yolov5> (accessed Jun. 30, 2023).



Zhehui Li is a Computer Science undergraduate student at Universiti Sains Malaysia (USM), specializing in Software Engineering. He is currently undertaking an internship at the National Advanced IPv6 Centre (NAv6). His academic and professional pursuits are anchored in machine learning, with an interest in the development of back-end and front-end systems.



Gin Keat Lim is a senior lecturer in the School of Chemical Sciences of Universiti Sains Malaysia (USM), where he joins as a faculty member since 2012. Prior to joining USM, he had various experiences working in other higher-learning institutions. His research interests are in Agricultural Farming Science, Natural Product Chemistry and Food Analysis.



Yung-Wey Chong is a senior lecturer in the National Advanced IPv6 Centre of Excellence at Universiti Sains Malaysia, where she has been a faculty member since 2012. Prior to joining USM, she had experience working in the telecommunications industry. Her research interests revolve around Industry Revolution 4.0 and encompass various areas such as embedded systems, wireless communication, cloud computing, and artificial intelligence.



Rahmad Dawood is an Assistant Professor in the Department of Electrical and Computer Engineering at Universitas Syiah Kuala (USK) and the founding director of the Telematics Research Centre at USK. Prior to joining USK, Rahmad worked as an IT specialist for several multinational corporations. Rahmad's research focuses on ICT4D, which employs a variety of computing technologies to promote development in developing countries, notably Embedded Systems, the Internet of Things, Artificial Intelligence, Human-Computer Interaction,

and Software Engineering.



Mohd Nadhir Ab Wahab, IEEE Member received the B.Eng. (Hons.) and M.Sc. degrees in Mechatronics Engineering from Universiti Malaysia Perlis, in 2010 and 2012, respectively, and the Ph.D. degree in robotics and automation system from the University of Salford, U.K., in 2017. He is currently a Senior Lecturer at the School of Computer Sciences, Universiti Sains Malaysia. He is also the Founder and the CEO of USM start-up company, Intech Art Sdn. Bhd. His research interests include mobile robotics, computer vision, machine learning, deep learning,

artificial intelligence, optimization, navigation, and path planning.

Masked Face Recognition Using Deep Learning Techniques

Yiyang Ma

Computer Engineering Department
King Mongkut's University of Technology Thonburi
Bangkok, Thailand
yiyang.ma@mail.kmutt.ac.th

Naruemon Wattanapongsakorn

Computer Engineering Department
King Mongkut's University of Technology Thonburi
Bangkok, Thailand
naruemon.wat@kmutt.ac.th

Abstract— Since the COVID-19 pandemic, people have been wearing masks in public, which poses a huge challenge to facial recognition. This paper introduces a new solution which is a masked face recognition method based on FaceNet and Inception-ResNet which are deep learning techniques. A well-known benchmark dataset (RWMFD) is employed to train the face recognition model. Furthermore, simulating masks and data augmentation are used to further improve the classification precision of the model. Our four experimental cases on the synthetic and the real-mask face datasets show that our method can effectively process face recognition and identification task with and without masks, with very high performance in terms of accuracy, precision, recall and F1-score.

Keywords—FaceNet, Masked face recognition, Data augmentation

I. INTRODUCTION

Since the COVID-19 pandemic, people have been wearing masks in public, which poses a huge challenge to facial recognition. Since Turk M. [1] proposed an EigenFace, face recognition has been the focus of research. Traditional facial recognition (FR) systems are usually based on the main facial features such as eyes, nose and mouth, i.e. the whole face features. Mask obstruction brings more challenges to facial recognition because the loss of two important facial features such as the nose and mouth renders more than half of the facial features unavailable and introduces a large amount of noise into the facial key point feature information. Therefore, mask face recognition becomes a major challenge for face recognition algorithms[2].

In this research, we developed a masked face recognition (MFR) model to solve this problem. Deep learning allows models to learn highly discriminative facial features for face recognition.

II. RELATED WORK

To solve the problem of mask face recognition using deep learning. A well-known approach is to improve the performance of existing algorithms for mask face recognition. Enhancement of the dataset images can be done. Q. Cao et al.[3] trained the model by mixing face datasets of people wearing masks and not wearing masks to improve the performance of the existing face recognition models. A. Anwar, et al.[4] fine-tuned the pre-trained model on a mixed dataset using triplet loss to further improve the accuracy of the face recognition model for masked face recognition. Y Li et al.[5] used cropping and attention modules to make the network focus more on the features around the eyes, which can effectively improve the accuracy of masked face

recognition but reduce the performance of face recognition in unobstructed scenes.

In this paper, we propose a face recognition method based on the FaceNet [6] model for masked faces, which can take into account both masked face recognition and unmasked face recognition tasks. The backbone feature extraction network uses a hybrid model of Inception-ResNet-v1, and the residual network module is introduced in the convolution module, which can effectively improve the accuracy and training speed of the model and extract more robust information for identity recognition. The method can combine the advantages of Inception and ResNet to improve the generalization ability and recognition accuracy of the model.

III. DEEP LEARNING MODEL

The mask face recognition model is based on FaceNet framework. It consists of four processes: data pre-processing, data modeling, saving the best model and model evaluation, as shown in Fig. 1.

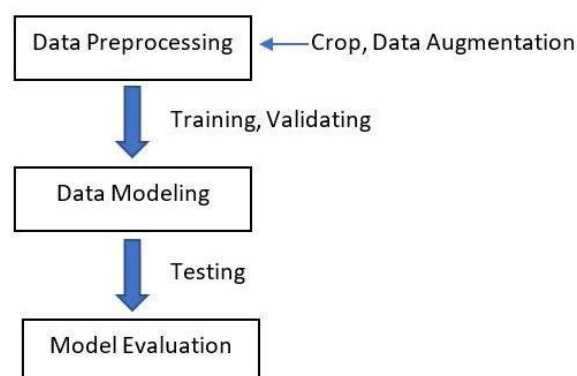


Fig. 1. Masked face recognition model.

FaceNet is a neural network architecture that uses a triplet loss function to train the model to produce embeddings, or numerical representations, of faces. The goal is to learn a mapping from images of faces to points in an embedding space, where the squared L2 distance corresponds to the similarities of these face. These embeddings can then be used to compare and identify faces in images and videos. Fig. 2 presents the framework of FaceNet [6].

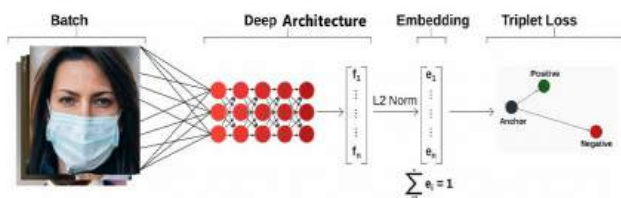


Fig. 2. The framework of FaceNet [6].

Inception-ResNet-v1 [7] combines both Inception and ResNet architectures and is used as our deep architecture. It consists of a series of Inception modules, each of which is composed of several parallel convolutional layers of different sizes. The output of each Inception module is then passed through a series of residual blocks, which use shortcut connections to improve information flow. Fig. 6 shows the block diagram of Inception-ResNet-v1 [7].

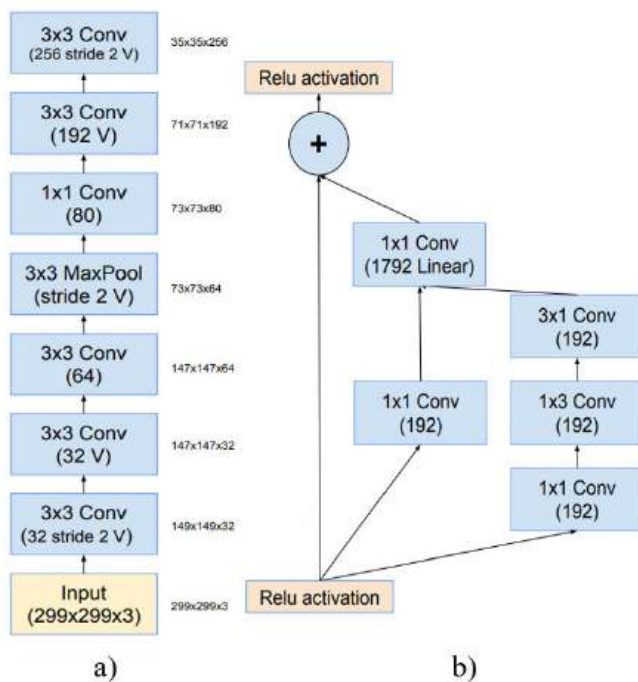


Fig. 3. Block diagram of Inception-ResNet-v1. a) stem b) module

The input to Inception-ResNet-v1 is a RGB image of size 299 x 299 pixels. The convolutional layers are arranged in a series of Inception modules, each of which contains multiple parallel branches with different filter sizes. It is used to extract features from the image at different scales and levels of abstraction. After each Inception modules, the network uses reduction module to reduce the spatial dimensions of the feature maps while increasing their depth. This helps to reduce the computational cost of the network while preserving important features. The output of Inception-ResNet-v1 is an embedding that represents the face image.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

This section describes the dataset used in this paper, the preprocessing of the dataset, the model training method, and then analyzes the experimental results to verify the effectiveness of the algorithm. The model is trained on Windows 11, using a single NVIDIA GTX1070 GPU, with a batch size of 196 and a learning rate 0.0001.

A. Dataset

A dataset is one of the most important parts in the machine learning process. The dataset in this work is obtained from <https://github.com/X-zhangyang/Real-World-Masked-Face-Dataset>. This RWMFD dataset, organized by National Engineering Research Center for Multimedia Software (NERCMS), School of Computer Science, Wuhan University, devoted mainly to improve the recognition performance of the existing face recognition technology on the masked faces during the COVID-19 pandemic.

1) Data Preprocessing

We noticed severe class imbalance in the dataset, the existing real-world face mask dataset is small in size and only suitable for model evaluation but not for training. In our experiments, masks and no-masks will be divided into multiple groups to be used as training sets. To make the data consistent, a tool for adding masks to standard faces is used in this paper. MaskTheFace effectively simulates adding a mask to a face by locating facial keypoints and calculating the facial angle. The large-scale face dataset is converted into a synthetic mask face dataset, as shown in Fig. 3. Before adding the mask, MTCNN is used to align and crop the face portion of the images in the dataset, which can avoid the failure of face keypoint detection due to mask occlusion.

Data augmentation can expand the dataset by generating more training data, increase the diversity of samples, and improve the generalization ability and robustness of the model. In this paper, we use data augmentation including adjustment, flip and rotation.

- Adjustment: Adjust images with random brightness, contrast, hue, saturation.
- Flip: Mirror image with horizontal flip.
- Rotation: Random rotation angle between 0-20 degrees.

2) Experimental Cases

The custom data sets are processed and divided into 4 experimental cases. Cases A and B each considers 10-person face images. Cases C and D each considers 100-person face images. Unmasked face recognition are provided in cases A and C, while the other two cases perform masked face recognition. The detailed information is presented in Table I.

TABLE I. EXPERIMENTAL CASES

Case	Sampled Dataset	Train (images)	Validate (images)	Test (images)
A	Unmasked faces of 10 persons, 100 images/person	700	100	200
B	Masked faces of 10 persons, 100 images/person	700	100	200
C	Unmasked faces of 100 persons, 100 images/person	7,000	1,000	2,000
D	Masked faces of 100 persons, 100 images/person	7,000	1,000	2,000

B. Model Construction

For each experimental case, we split the data into 70% for training, 10 % for validating and 20% for testing the FaceNet with InceptionResnet-v1 model. In each case, the model is

trained using the dataset presented in Table I, and using the same parameter setting and initial weights. The hyperparameter setting are shown in Table II.

TABLE II. HYPER PARAMETERS

Model	Inception-ResNet-v1
Batch size	196
Optimizer	Adam
Learning rate	0.0001
Epoch	80

C. Result Discussion

a) Simulation Results

We evaluate each experimental case with 5 metrics, which are confusion matrices, accuracy, precision, recall and F1-score. Fig. 4-7 present the training accuracy, training loss and validation accuracy for the experimental cases A, B, C and D, respectively. Fig. 8 presents the confusion matrices of cases A and B where ten persons are recognised and identified. It is shown that the results are very accurate most of the time.

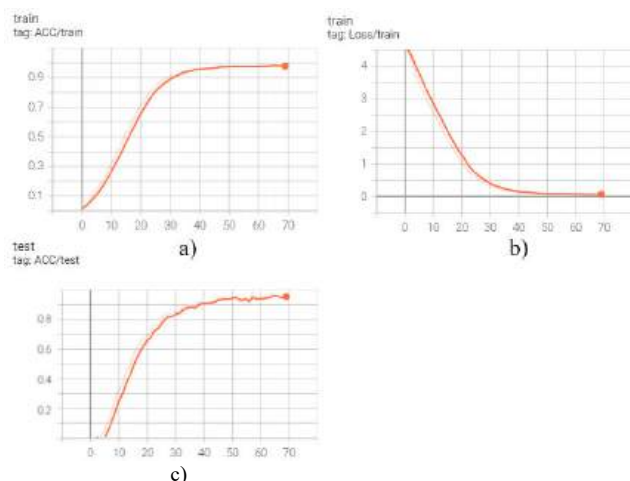


Fig. 4. Case A: a) training accuracy, b) training loss, c) validation accuracy

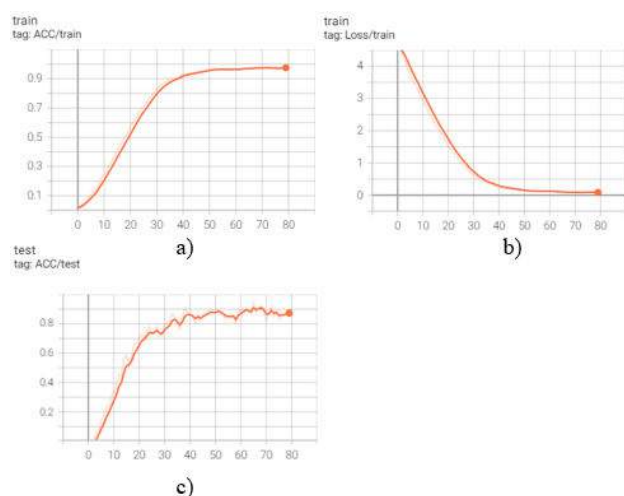


Fig. 5. Case B: a) training accuracy, b) training loss, c) validation accuracy

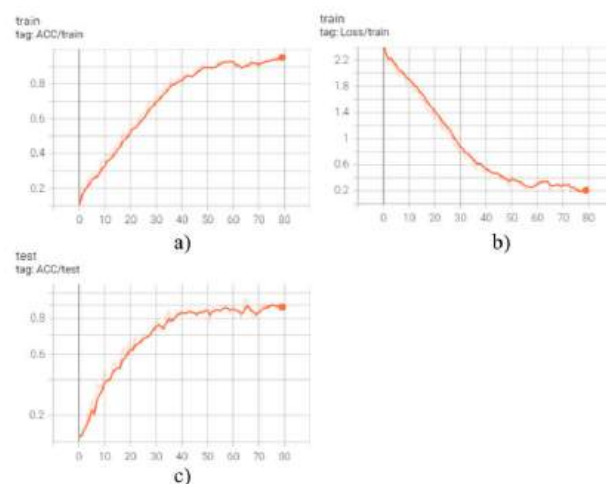


Fig. 6. Case C: a) training accuracy, b) training loss, c) validation accuracy

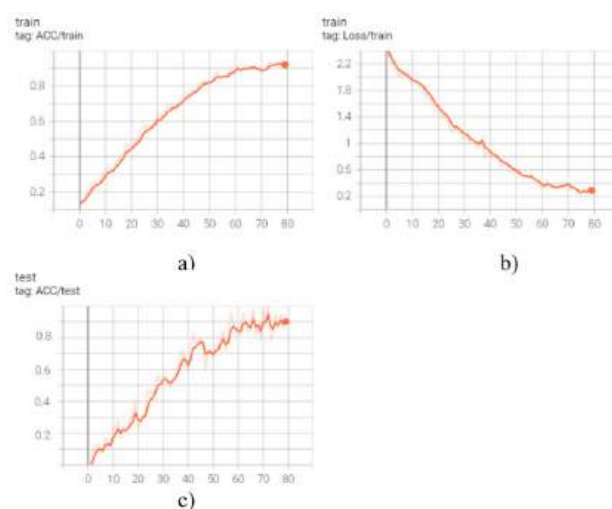


Fig. 7. Case D: a) training accuracy, b) training loss, c) validation accuracy

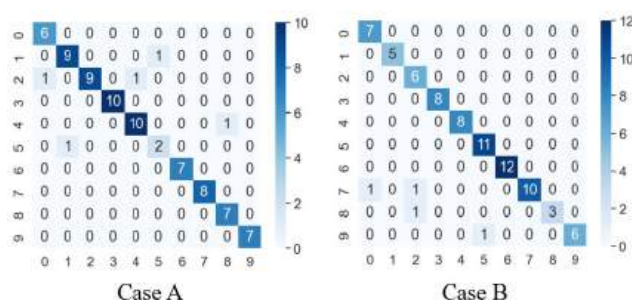


Fig. 8. The confusion matrices of cases A and B.

The performance of our face recognition model is clearly identified based on all experimental cases and shown in Table III.

TABLE III. PERFORMANCE OF THE FACE RECOGNITION MODEL

Experiment	Performance Metrics			
	Accuracy	Recall	Precision	F1-score
Case A: Unmasked	0.9575	0.9571	0.9581	0.9565
Case B: Masked	0.9545	0.9547	0.9544	0.9528
Case C: Unmasked	0.9300	0.9256	0.9343	0.9260
Case D: Masked	0.9250	0.9251	0.9329	0.9262

In our face recognition model with unmasked faces, we achieve 95.75% and 93% accuracy on Cases A and C, respectively. Our model also performs well with masked face datasets, offering 95.45% and 92.5% accuracy on cases B and D, respectively.

b) Comparison with other approaches

Golwalkar et al. [8] employed a self-developed FaceMaskNet-21 network using deep metric learning to recognize masked faces with FaceMaskNet-21 model giving an accuracy of 88.92%, while using 5000 images of 525 persons. Anwar et al. [4] proposed VGGFace2 model that obtained 95.99% accuracy on MRF2 dataset, where 269 images of 53 persons were considered. Ejaz et al. [9] presented MTCNN and FaceNet model, which offered 92.26% accuracy, while using 990 images of 45 persons. The accuracy comparison with our face recognition model is shown in Table IV, indicating a superior performance of our proposed model with FaceNet-InceptionResNet-v1. We consider recognizing 100 persons using 10,000 images from a well-known benchmark dataset (RWMFD).

TABLE IV. PERFORMANCE OF EACH MODEL

Model	Dataset	Accuracy
FaceMaskNet-21 [8]	5,000 images of 525 persons	88.22%
VGGFace2 [4]	269 images of 53 persons	95.99%
MTCNN & FaceNet [9]	990 images of 45 persons	92.96%
Ours	10,000 images of 100 persons	95.75 %

V. CONCLUSION

In this paper, we presented masked face recognition and identification techniques that can be applied to both masked and unmasked face recognition scenarios. We tested our model with 10 persons and 100 persons. Furthermore, based on the results of our four experimental cases, the best model is the fine-tuned FaceNet, with InceptionResNet-v1 as the deep layers, which is able to accommodate both masked and unmasked face recognition. In the model evaluation, the model achieves 95.75% accuracy on the test dataset.

REFERENCES

- [1] M. Turk and A. Pentland, "Eigenfaces for recognition", *Journal of Cognitive Neuroscience*, vol 3, no. 1, pp. 71–86, 1991.
- [2] Q. Duan and L. Zhang, "Look more into occlusion: realistic face frontalization and recognition with BoostGAN," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 214–228, 2020.
- [3] Q. Cao, L. Shen et al., "VGGFace2: A dataset for recognising faces across pose and age," *13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, Xi'an, China, 2018, pp. 67–74.
- [4] A. Anwar, A. Raychowdhury, "Masked face recognition for secure authentication," *arXiv preprint*, arXiv:2008.11104, 2020.
- [5] Y. Li, K. Guo et al., "Cropping and attention based approach for masked face recognition," *Applied Intelligence*, 51, pp. 3012–3025, 2021.
- [6] F. Schroff, D. Kalenichenko and J. Philbin, "FaceNet: a unified embedding for face recognition and clustering," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, 2015, pp. 815–823, doi: 10.1109/CVPR.2015.7298682.
- [7] C. Szegedy, S. Ioffe, V. Vanhoucke and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," *In Proceedings of the AAAI conference on artificial intelligence*, vol. 31, no. 1, 2017, pp. 4278–4284.
- [8] R. Golwalkar, N. Mehendale, "Masked-face recognition using deep metric learning and facemasknet-21", *Applied Intelligence*, 2022.
- [9] M. S. Ejaz and M. R. Islam, "Masked face recognition using convolutional neural network", *International Conference on Sustainable Technologies for Industry 4.0 (STI)*, 24–25 December 2019, Dhaka, Bangladesh, pp. 1–6.

Analysis, Visualization and Interpretation of Determinants of Degree Program Selection of the Senior High School Graduates Utilizing Classification and Clustering Algorithms

Loyd Soriano Echalar
loydechalar@lspu.edu.ph

Jonalyn Joy B. Labayne
jjlabayne@lspu.edu.ph

Merardo Camba
merardo.camba@lspu.edu.ph

Abstract—Preparing for the future life is one of the most important keys to success in life. The Republic of the Philippines issued the RA No. 10533 “An Act Enhancing the Philippine Basic Education System by Strengthening its Curriculum and Increasing the Number of Years for Basic Education” which results in the implementation of the K-12 Curriculum Program. The program aims to strengthen the basic education of the country by adding two more years in High School and integrating different specializations that prepare and guide the students to select a specialization that fits them in the future. Many cases occurred that senior high school graduates took a degree program that is not aligned with the specialization they took during senior high school. This research paper presents the analysis, visualization, and interpretation of determinants of degree program selection of senior high school graduates. Python and Jupyter Notebook was used to generate the results. Clustering and Classification Algorithms were used to analyze, visualize and interpret the data. Based on the results, it is found that the main reasons for the conflict are the passion of the students, the industry demand for specialization, and their family preferences. It is also found that the residence of the students affects choosing the degree program. As a recommendation, the education authority must review and revisit the rules and implement guidelines of the K-12 curriculum program to eliminate and avoid such confusion and conflicts in selecting degree programs.

Keywords—Specialization Track, Degree Program, Determinants, Python, Jupyter Notebook, Classification, Clustering

Introduction

Education is the foundation for all success in life. Education in the Philippines is one of the most valued sectors in society to address poverty and make everyone's life well. [1] This is the reason why the Philippine Government implemented a Republic Act which aims to strengthen basic education in the country. Republic Act 10533 also known as “An Act Enhancing the Philippine Basic Education by Strengthening its Curriculum and Increasing the Number of Years for Basic Education, Appropriating Funds Thereof and for Other Purposes” drives the Department of Education to implement the additional year for the basic education. [2] In September 2013 when the Department of Education releases a DepEd Order No. 43 s. 2013 an order to implement the K-12 Curriculum Program for basic education. [3] K-12 Curriculum Program aims to enhance the basic skills of the learners, produce competent graduates, and prepare graduates for their employment and their future. Additional two years were added to the curriculum known as the Senior High School which aims to help the learners to learn and improve in different areas and fields of specialization.

The program is composed of four main tracks namely Academic Track, Technical-Vocational-Livelihood Track, Sports Track, and Arts and Design Track. Each track is composed of a

different strand or specialization which the students are allowed to select and choose what they want or what they are capable of. [4]

After taking the specialization and graduating from senior high school, students are facing a new challenge in pursuing a degree course in college. Selecting a degree program is based on students' wants, skills, and capabilities and not based on the track or specialization they took during senior high school. When the students select a degree program that is not aligned with their specialization taken during senior high school, a bit of confusion, conundrum, and conflicts have occurred. It is also difficult for teachers to re-divert the knowledge and skills of the students that are knowledgeable and skilled in different areas of specialization. That is why it is important that the specialization taken during senior high school and the selected degree program are connected to have a synchronization of knowledge and skills and to prepare students for the betterment of their future.

This research paper presents the analysis, visualization, and interpretation of determinants of why the students who graduated from senior high school select a degree program that is not aligned and connected to the specialization they took during senior high school. This research aims to identify the reasons behind their decision in selecting degree programs and aims to eliminate the conundrum behind the said conflicts. Analysis was done through the use of Python and Jupyter Notebook, while the visualization and interpretation were done through the use of the classification and clustering algorithm.

I. EASE OF USE

This section includes the methodologies used in the analysis, visualization, and interpretation of determinants for degree program selection. Each phase of the method used was discussed to explain and understand the things done by the researcher.

The data used in this research paper are gathered through a survey with a series of questions that identifies some of the reasons for selecting a degree program and other related information that are significant to the research. The data includes the demographic profile of the students such as age, sex, status, and address; some other data are the track during senior high school, the degree program they took, the passion of the students, and the preferences in selecting the course.

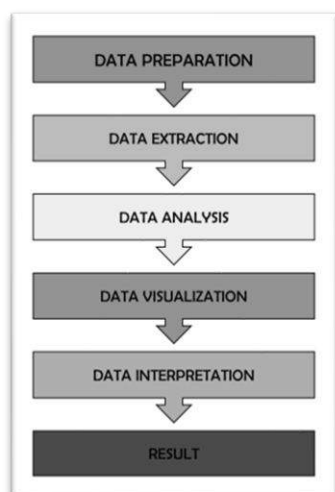


Figure 1. Data Mining Phases

Fig. 1 shows the phases for conducting data mining for the gathered data. The first phase will be the data preparation, followed by data extraction, then data analysis, data visualization, and data interpretation for a reliable result.

A. Data Preparation

In this phase, the data was encoded using MS Office Excel. The data is composed of different values, variables, shortcuts, and long texts, local languages, incomplete information, and empty cells. For the preparation of the data, the following was done: empty cells and incomplete information were deleted from the set of data; local languages were changed; long and shortcut texts were converted to standard texts, and different answers with the same meaning were categorized into general terminologies for standardization. These were done to avoid problems, difficulties, and inconsistencies during the analysis, visualization, and interpretation of the data. A survey was conducted on more than 750 students in the BS IT course.

B. Data Extraction

In this phase, the pre-process data were examined to get what specific information is needed before going through the analysis. All the information is important to generate a report. Columns and rows of the data set were examined and assessed to see what applicable columns and rows are helpful to identify the determinants. From the data, the columns address, passion, track, course, and reason are identified as significant to determine the possible causes of selecting a degree program that is not aligned and connected to the specialization they took during senior high school; and other information may extract from the selected data.

II. PREPARE YOUR PAPER BEFORE STYLING

Before you begin to format your paper, first write and save the content as a separate text file. Complete all content and organizational editing before formatting. Please note sections A-D below for more information on proofreading, spelling, and grammar.

Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs, and limit the use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads-the template will do that for you.

A. Data Analysis

For the analysis of data, Python and Jupyter Notebook was used. Python is an interpreted high-level programming language that is designed to emphasize code readability with the use of significant indentation; it also helps the programmer write clear and logical codes. [5] While the Jupyter Notebook is an open-source web application that is used to create and share documents that contain codes, equations, visualization, and text. [6] These two were used to program and analyze the set of data being loaded. Two different algorithms were also used to visualize and interpret the data namely the Classification Algorithm and the Clustering Algorithm. A classification algorithm is a supervised learning technique that identifies new observations into a number of classes or groups on the basis of training data. [7] While the clustering algorithm is an unsupervised machine learning task that involves automatically discovering natural grouping in the data. [8] These two algorithms were used to identify important contents and information of the data. It is also used to visualize the groupings of data, classify the data points, and also distribute of the data.

B. Data Visualization

To model and visualize the data for interpretation. Three model algorithms of classification algorithm and clustering algorithm were used. For the Classification algorithm, the Decision Tree model, K-Nearest Neighbors (kNN) model, and the Naïve Bayes model are used to easily visualize and observe the classes and groups in the data. While the clustering algorithm used the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) model, Hierarchical model, and k-Means model to discover the natural grouping of the analyzed data. These six model algorithms use different diagrams, illustrations, and figures to model and envision the content of the data and convert it into reliable information.

C. Data Interpretation

Through the use of different diagrams, figures, and illustrations of the model algorithms the content of the data is interpreted into reliable information. The distribution of the data, the plotting, and trends were used to extract information. The interpreted information was discussed to understand better the visualization and the result. For this phase, it is important that each of the models should be clear and accurate for convenient interpretation and discussion of the results.

III. RESULT AND DISCUSSIONS

This section includes the classification algorithm and clustering algorithm models' results and discussion. Each of the models is briefly discussed based on how the data were distributed and plotted.

The succeeding figures and discussions are the results of the Classification Algorithm done on the data.

A. Accuracy Against Various Parameters

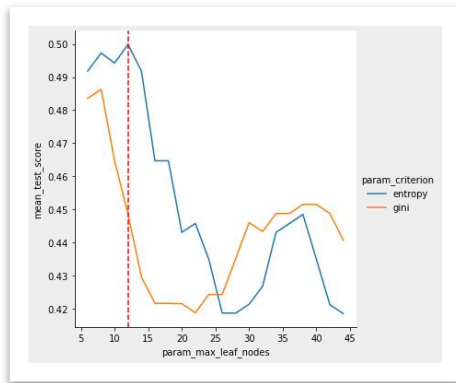


Figure 2. Plot Accuracy Against Various Parameters

Fig. 2 shows against various parameters can help to analyze the relationship between the performance of a model and different factors. The specific parameters to consider depend on the context and the nature of the problem stated.

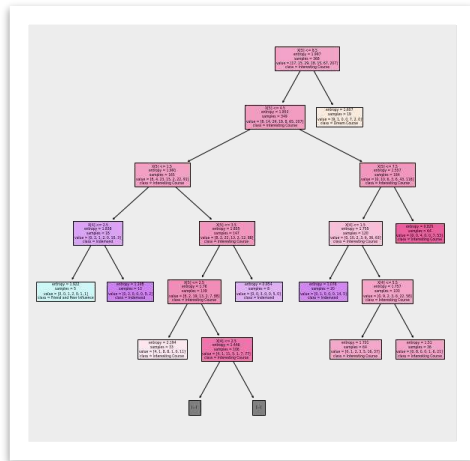


Figure 3. Decision Tree

Fig. 3 shows the parameters and plotting accuracy against the features. And also observe the decision tree's performance and identify the parameter value that yields the best accuracy.

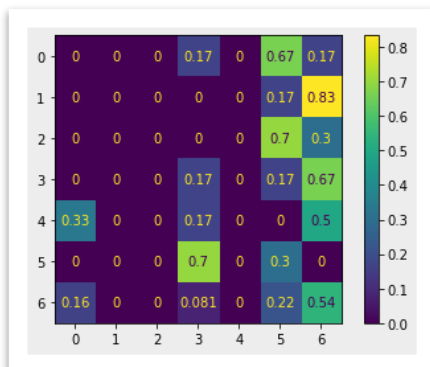


Figure 4. Decision Tree Confusion Matrix

Fig. 4 shows the summary of the predicted and actual class labels produced by the model. The figure also predicts the set of rules based on the given data.

The figures above are the results and visualization created using the Decision Tree Model Algorithm. The accuracy of the training data set is 81.96% and the accuracy for the testing data set is 59.67%.

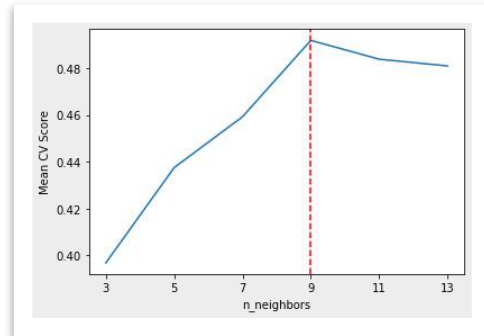


Figure 5. Plot the Accuracy by Neighbors Parameter

Fig. 5 shows that plotting the accuracy by the neighbor's parameter is a common visualization when working with k-nearest neighbors (KNN) algorithms and similar algorithms that use the number of neighbors as a hyperparameter.

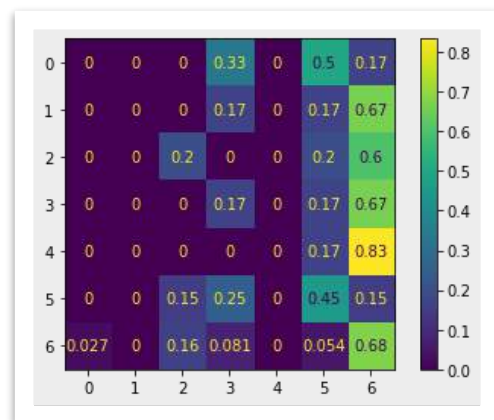


Figure 6. kNN Confusion Matrix

Fig. 6. shows the confusion matrix using the test data to give the fraction of true positives, true negatives, false positives, and false negatives.

The figures above are the results and visualization created using the k-Nearest Neighbors Model Algorithm. The accuracy of the training data set is 87.39% and the accuracy for the testing data set is 70.66%.

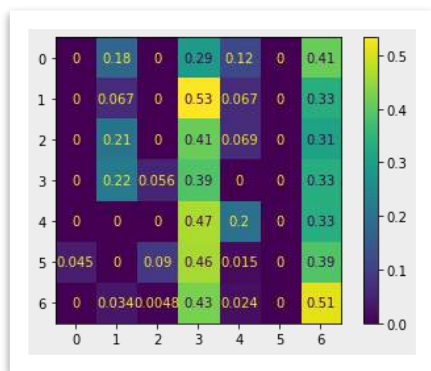


Figure 7. kNN Confusion Matrix

Fig. 7 shows the confusion matrix using the test data to give the fraction of true positives, true negatives, false positives, and false negatives.

The figures above are the results and visualization created using the C. Naïve Bayes Model Algorithm. The accuracy of the training data set is 56.26% and the accuracy for the testing data set is 51.79%.

Based on the models and visualizations, among the three model algorithms of the classification algorithm, the decision tree has a good visualization than the kNN and Naïve Bayes. But in terms of accuracy of the training data kNN has the higher accuracy rate followed by the decision tree and the naïve bayes. And in terms of accuracy of the testing data set, kNN also has the higher accuracy rate followed by the decision tree and the naïve bayes. The succeeding figures and discussions are the results of the Clustering Algorithm done on the data.

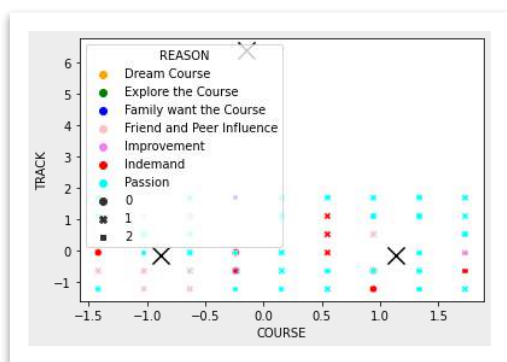


Figure 8. k-Means Cluster Visualization

Fig. 8 shows the cluster or groupings and distribution of the data points.

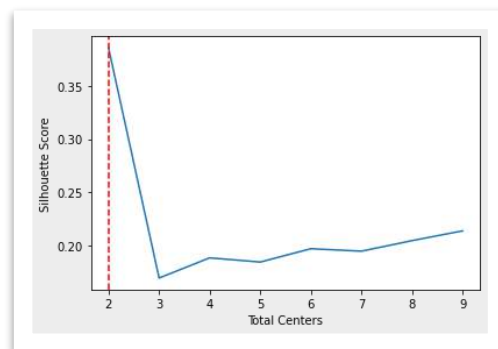


Figure 9. Plot of Scores for Different Number of Clusters

Fig. 9 shows the plotting and count of parameters of the data. The best model is also shown through the red line.

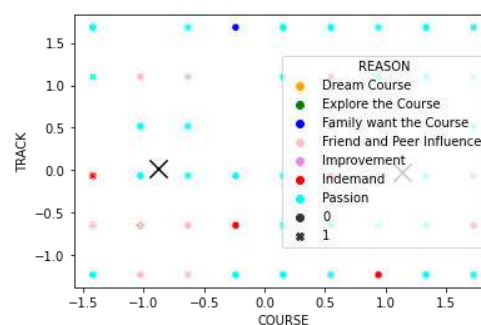


Figure 10. k-Means Cluster Visualization Best Model

Fig. 10 shows the best cluster or groupings and distribution of the data points.

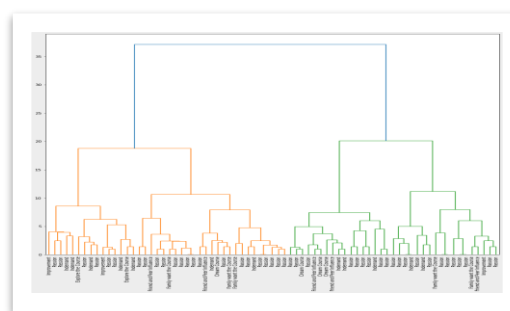


Figure 11. Hierarchical Cluster Visualization

Fig. 11 shows the hierarchical visualization of the data and displays the cluster.

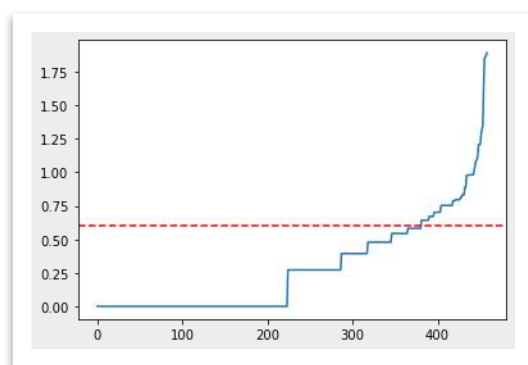


Figure 12. DBSCAN Parameter

Fig. 12 shows the plotting and count of parameters of the data. The best model is also shown through the red line.

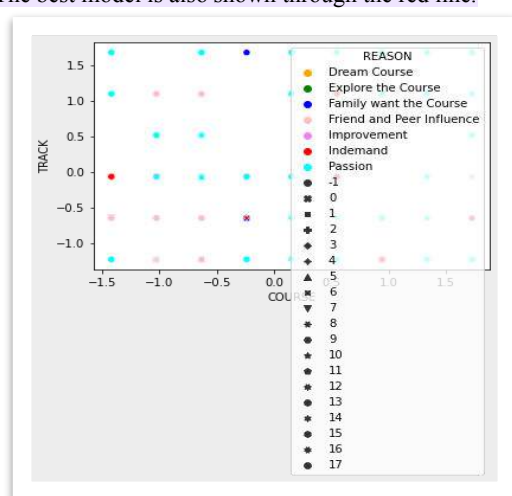


Figure 13. DBSCAN Cluster Visualization Model

Fig. 13 shows the best cluster or groupings and distribution of the data points.

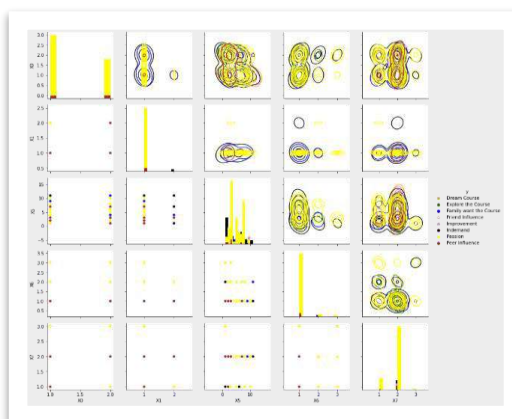


Figure 14. Scatter Plots and Density Plots for Feature Pairs

Fig. 14 shows the plotting of data points and the relationships between all of the features using PairGrid.

Based on the result of the visualization, among the three clustering algorithms, the k-Means shows the best visualization of data followed by the DBSCAN and the Hierarchical Clustering. All three models have the same interpretation and discussion of the results.

Based on the models and visualization output of the classification and clustering algorithm the main reasons why the students select a degree program that is not aligned and connected to the specialization track they took during their senior high school are the following: the passion that they realized throughout the track; the demand of the specialization to industry; and the preferences of their family members to the course. The residence of the students also affects the decision to take different specializations in the college. Some other reasons that students consider in selecting the degree program are dream courses from a young age; they want to explore new things in a different field; friends and other people influence them to take the course, and they want to improve their specialization to a particular field. The models and visualization in this section display different illustration but the same results and discussions.

IV. CONCLUSIONS

Based on the result of data analysis, visualization, and interpretation using classification and clustering algorithms and their different model algorithms, the following is concluded. The most determinants in degree program selection of the senior high school graduates are the passion that they realized throughout the track; the demand of the specialization to industry; and the preferences of their family members to the course among others. The residence of the students also affects in selecting the degree course as the resources of the course are close to their community and places. Like for example, a student who resides in the Barangays within the vicinity of Laguna de Bay took a Bachelor of Science in Fisheries Program. In terms of analyzing and visualizing the data, the classification algorithm and the clustering algorithm are best helpful in extracting significant information from the data set. Among the three model algorithms of the classification algorithm, the decision tree classification model algorithm gives a well and organized visualization of data, and the kNN classification model algorithm gives a higher accuracy rate among others. In clustering algorithm, among the three clustering model algorithms, the k-Means clustering algorithm gives the most accurate and organized visualization of the data.

V. RECOMMENDATIONS

As a recommendation based on the conclusions found on the data analysis, visualization, and interpretation. The education authority must review and revisit the rules and implementing guidelines of the K-12 curriculum program to eliminate and avoid such confusion and conflicts in selecting degree programs for senior high school graduates. In terms of the loading data in the application, it is a must that the data undergo pre-processing and cleaning to provide more accurate and reliable analysis, visualization, and result. Large data that are related to the research may add additional features and important data for more accurate and reliable results. In terms of model generation, it is recommended to use the different model algorithms in order to identify which are the best model algorithm must be used and to identify conflicts, difficulties, convenience, and significance to each of the model algorithms.

- 4 [1] Study Country, "Education in Philippines," Study Country, 2021. [Online]. Available: <https://www.studycountry.com/guide/PH-education.htm>. [Accessed 2021].
- [2] Congress of the Philippines, "Republic Act No. 10533," Republic of the Philippines, 15 May 2013. [Online]. Available: <https://mirror.officialgazette.gov.ph/2013/05/15/republic-act-no-10533/>. [Accessed 19 November 2021].
- 42 [3] Department of Education, "The K to 12 Curriculum and Its Support System," Department of Education, 4 November 2016. [Online]. Available: <https://www.deped.gov.ph/wp-content/uploads/2018/07/J.Andaya-Education-Summit.pdf>. [Accessed 19 November 2021].
- 40 [4] Department of Eduaction, "K to 12 Basic Education Curriculum," 28 June 2016. [Online]. Available: <https://www.deped.gov.ph/k-to-12/about/k-to-12-basic-education-curriculum/>. [Accessed 19 November 2021].
- 49 [5] Programiz, "Learn Python Programming," Programiz, [Online]. Available: <https://www.programiz.com/python-programming>. [Accessed 19 November 2021].
- [6] M. Driscoll, "Jupyter Notebook: An Introduction," Real Python, 2019. [Online]. Available: <https://realpython.com/jupyter-notebook-introduction/>. [Accessed 19 November 2021].
- [7] Java Point, "Classification Algorithm in Machine Learning," Java Point, 2019. [Online]. Available: <https://www.javatpoint.com/classification-algorithm-in-machine-learning>. [Accessed 19 November 2021].
- [8] J. Brownlee, "10 Clustering Algorithms With Python," Machine Learning Mastery, 6 April 2020. [Online]. Available: <https://machinelearningmastery.com/clustering-algorithms-with-python/>. [Accessed 19 November 2021].
- [9] M. A. T. S. Loyd S Echalar, "PaLife: a mobile application for palay (Rice) health condition classification utilizing image processing and pigment analysis towards sustainability of palay production," 2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), pp. 443-448, 2018.
- [10] S. B. S. C. F. G. Loyd S Echalar, "Maize plant nutrient deficiency recognition through deep neural network," Journal of Agriculture and Technology Management, vol. 24, no. 3, pp. 8-12, 2021.
- [11] S. B. S. L. S. E. Crisanto F Gulay, "Weight diary: An android application monitoring daily routine for body weight prediction through lifestyle mining," Journal of Agriculture and Technology Management, vol. 24, no. 3, pp. 13-17, 2021.
- [12] "Model development of students' scholarship status at first asia institute of technology and humanities (Faith)," Jonalyn Joy B Labayne, Lester L Mercado, Jheanel Espiritu Estrada, pp. 152-157, 2018.

Forecasting Air Passengers in Phuket, Thailand: An Application of Machine Learning Algorithms

APHINAN PHOTUN

Department of Mathematics, Faculty of Science
Naresuan University
Phitsanulok, Thailand
aphinanp62@nu.ac.th

THANAKORN THAMINKAEW

Big Data Institute (Public Organization)
Bangkok, Thailand
thanakorn.th@depa.or.th

Abstract—The COVID-19 pandemic occurred in 2020 had a significant impact on the global economy, resulting in a significant decrease in air passengers. As a result, time series trends experienced significant fluctuations that were difficult to predict. The objective of this research was to forecast the number of air passengers in Phuket, Thailand, from September to December 2022. Additional features, such as exchange rates and time-based properties, were included to enhance the accuracy of our predictions. We then compared the performance of statistical models and machine learning models using evaluation metrics such as mean absolute error (MAE), root mean squared error (RMSE), and mean absolute percentage error (MAPE). Our findings revealed that the SARIMAX model outperformed other models for air passenger forecasting. Nonetheless, machine learning models might be more appropriate for forecasting new data due to their stability with additional training data. This research emphasized the importance of selecting the appropriate model based on the available data and the particular needs of the forecasting problem.

Index Terms—air passengers, time series forecasting, exchange rates, machine learning models, vector autoregressive model, SARIMA model.

I. INTRODUCTION

The outbreak of the COVID-19 pandemic in 2020 had a significant impact on the global economy. One of the most affected industries was the tourism sector, which experienced a sharp decline in air passengers. The pandemic caused a global lockdown and travel restrictions, resulting in a decrease in air travel demand. The decline in air passenger numbers led to a decline in the economies of countries that rely heavily on tourism, such as Thailand.

Phuket, Thailand is heavily dependent on tourism, and the pandemic caused a significant decline in the number of visitors, leading to a sharp drop in revenue for hotels, restaurants, and other businesses that rely on tourism. Therefore, it is crucial to predict air passenger numbers in Phuket to help these affected businesses as well as the government to prepare for future tourists' demands. In 2020, S.V. Gudmunsson et al. [5] estimated the relationship between the strength of economic shocks and temporal recovery in the world air transport industry. The results showed that world passenger demand recovery to pre-COVID-19 levels was estimated to take 2.4 years (recovery by late 2022), and Asia Pacific had the shortest estimated average recovery time at 2.2 years.

This study used machine learning and statistical models to forecast air passenger numbers in Phuket, Thailand. While the

models used in this study are not new, their performance in situations characterized by significant fluctuations in time series trends, both positive and negative, had not been explored sufficiently. To improve the accuracy of the predictions, daily air passenger data collected from September to December 2022, as well as exchange rates and other relevant time-based properties, were considered. To assess the accuracy of the predictions, we used mean absolute error (MAE), root mean squared error (RMSE), and mean absolute percentage error (MAPE). The insights gained from this study were expected to provide useful information for the tourism industry in Phuket and support their efforts to recover from the pandemic's economic impact.

II. RELATED WORK AND BACKGROUND

Forecasting air passengers has been of interest to researchers for many decades. H. Song et al. [7] conducted a review of 211 key papers published between 1968 and 2018, which showed the evolution of methods used in forecasting tourism demand. The majority of studies in this field used traditional time series models, such as (S)ARIMA, (S)ARIMAX, and VAR models, with a few incorporating machine learning models such as Gaussian process regression (GPR) and Elman Network, etc. However, in recent years, researchers had begun incorporating newer techniques, such as Support Vector Regression (SVR) and Neural Network models. It's important to note that although air passenger forecasting and tourism demand forecasting are two distinct concepts, there can be a correlation between them.

J. Feng et al. [1] proposed a hybrid approach for short-term air passenger demand forecasting that consists of Variational Mode Decomposition (VMD), Autoregressive Moving Average (ARMA), and Kernel Extreme Learning Machine (KELM). The VMD algorithm was used to decompose the original time series into multiple mode functions. The stationary and non-stationary modes were then identified by the autocorrelation function test (ACF test). The ARMA models were used to forecast the stationary modes, and The KELM models were used to forecast the non-stationary modes. Finally, the forecast results of all the components were integrated using another KELM model. The experimental results demonstrated that this approach outperforms other benchmark models in air passenger demand forecasting.

G. Ulrich et al. [2] used data from the top 20 busiest airports in the world to forecast passenger numbers using a

Global Vector Autoregression (GVAR) model, in comparison with other benchmark models such as Vector autoregression (VAR), Neural Network Autoregressive (NNAR), and ARIMA models. The study found that the GVAR model performed the best for some of the airports but not for all of them, with other models such as VAR and NNAR showing better performance for most airports. The use of data from the top 20 busiest airports in the world added to the significance of the study, as it reflected a large sample size of passenger demand data from major international airports.

X. Gang et al. [3] proposed two hybrid approaches based on seasonal decomposition and least squares support vector regression (LSSVR) model for short-term forecasting of air passengers at Hong Kong International Airport. They performed seasonal decomposition to decompose the original time series into seasonal factors, trend cycles, and an irregular component. Then, the LSSVR model was used to predict the components independently, and these prediction results of the components were combined by multiplication. The method looked similar to J. Feng et al.'s method [1]. The experimental results showed that the proposed hybrid approaches were better than other benchmark models such as ARIMA, BPNN, EMD-LSSVR, X12-LSSVR, and TS-LSSVR models.

T. Wai Hong Kan et al. [4] used two different models to forecast airport passenger traffic at Hong Kong International Airport (HKIA): the Box-Jenkins Seasonal ARIMA (SARIMA) model and the ARIMAX model. Both models predicted a steady growth in future airport passenger traffic in Hong Kong, but the ARIMAX model outperformed the SARIMA model. In the ARIMAX model, the authors included explanatory variables such as GDP per capita, connecting traffic, and visitors by air transport (with transformed data using the natural logarithm), as well as other variables like SARS outbreak, cross-strait agreement, fuel prices, and individual visitor scheme.

The algorithms and notable features of each model are briefly described to provide context for the machine learning and statistical models used in our research.

A. Machine Learning Models

We begin by discussing linear regression and related models before moving on to more advanced techniques such as support vector regression, as we explore the application of machine learning algorithms in air passenger forecasting.

Linear regression, Ridge Regression, Lasso Regression, and Elastic Net are all supervised learning models that can be used for regression tasks. These models are commonly used for predicting a continuous numerical value, such as air passenger traffic. Linear regression is a simple and easy-to-interpret model that assumes a linear relationship between the dependent and independent variables. Ridge Regression [11] and Lasso Regression [9] are regularized versions of linear regression that add penalty terms to the loss function, which helps to prevent overfitting and improve model generalization. Elastic Net combines both L1 and L2 penalties to the cost function, offering a balance between the two models.

Support Vector Regression (SVR) is another popular regression model that can be used for air passenger forecasting.

SVR is based on finding the best-fitting line or hyperplane that maximizes the margin around the line, subject to a constraint that allows some points to fall outside the margin. The support vectors, which are the data points that lie on the boundary of the margin, are used to determine the location of the hyperplane [20], [21].

Decision Trees are a non-parametric supervised learning model that can be used for both classification and regression tasks. A decision tree is built by recursively splitting the data based on the feature that provides the maximum information gain [13]. Historical data on air passengers, along with relevant features such as time of year and day of the week, can be used to train a Decision Tree model. This model can then make predictions for future periods based on the same features used during training.

Random Forest is an ensemble learning model that combines multiple decision trees to improve performance and reduce overfitting. Random Forest generates a set of decision trees by bootstrap aggregating random subsets of the training data and splitting each node based on a random subset of features [15]. Random Forests can handle high-dimensional data with many features since each tree only uses a random subset of features at each split.

Gradient Boosting is a widely used ensemble technique that involves constructing a sequence of basic models, often decision trees, in a step-by-step manner. At each stage, a new model is trained on the residuals, or the difference between the actual and predicted values, of the previous model. The outputs of each model are then combined using weights to produce the final prediction. The main objective of Gradient Boosting is to reduce the loss function by iteratively adding new models to the ensemble [22].

B. Statistical models

Time series data can be accurately predicted using statistical models as they are able to capture trends and seasonal patterns. Among these models, the SARIMA models have been widely used in many studies. In our research, we utilized three of these well-known models. The SARIMA model is an extension of the classic ARIMA model for univariate time series forecasting. It is designed to handle time series data with recurring patterns over fixed intervals, such as monthly or quarterly data [23]. On the other hand, the SARIMAX model is similar to SARIMA but includes exogenous variables that are external factors affecting the time series. The exogenous variables can be added as a separate input series that is used in combination with the time series data for forecasting.

The VAR model, proposed by Sims (1980) [26], is a widely used approach for forecasting correlated time series systems. The model is known for its flexibility and practicality, as it is easy to estimate and fits the data well. The VAR method models a system by considering each endogenous variable as a function of lagged values of all endogenous variables, which is different from structural models.

We employed Python to construct our models, using Keras, Scikit-learn, and other packages for machine learning models. For the statistical models, we used the pmdarima package. Additionally, Scikit-learn was utilized to apply evaluation metrics.

III. METHODOLOGY

This section presents our methodology, starting with dataset details and experimental design, including data sources, variables, and preprocessing steps. We then describe our model optimization and time series cross-validation approach, ensuring accurate forecasting models.

A. Data description and experiment design

Phuket International Airport is the second busiest airport in Thailand. In 2016, the number of passengers arrived at the airport was 15.1 million, up 17.8 percent from the previous year [19]. The dataset for this study originated from the Phuket Immigration Office, which captured daily travel records from September 1 to December 20, 2022. The dataset comprises 732,853 entries, encompassing individual details like gender, birth date, nationality, visa type, and flight number. We transformed this dataset into a daily time series representing air passenger counts, as depicted in Fig. 1.

Analyzing passenger counts during this period revealed a consistent upward trend, attributed to Thailand's progressive reopening. Aside from the past air passenger data, we found other variables can provide useful information for forecasting. Consequently, we integrated multiple exogenous variables in our model, which includes date features and exchange rate lags. Date features such as day of the week, month, day of the year, day of the month, and week of the year helped capturing the seasonal patterns in air passenger numbers, such as increased travel during weekends or holidays. Meanwhile, the exchange rate, a crucial determinant of air passenger travel, indicated the purchasing power of the local currency and could significantly impact travel affordability. In our experiment, we included exchange rate lags ranging from one week to one year to reflect changes in the exchange rate between Thai baht and other currencies. Specifically, our analysis used the exchange rate between the Thai baht and the US dollar.

In our experiments, we split the data into two sets: the training set and the test set. The training set was used to determine the unknown parameters of the pre-defined models, while the test set was used to evaluate the forecasting performance. To

create one-step-ahead forecasts for each out-of-sample observation, the model was set based on the previous data, using the lag period determined by analyzing the autocorrelation and partial correlation of the time series.

To compare the forecasting performance of multiple models, we utilized mean absolute error (MAE), root mean squared error (RMSE), and mean absolute percentage error (MAPE). We defined the metrics as follows:

$$MAE = \frac{1}{n} \sum_{t=1}^n |y_t - \hat{y}_t| \quad (1)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2} \quad (2)$$

$$MAPE = \frac{1}{n} \sum_{t=1}^n \left| \frac{y_t - \hat{y}_t}{y_t} \right| \times 100 \quad (3)$$

where n is sample size, y_t is actual value of the observation and $\hat{y}_t$ is the corresponding forecast in the t th date.

It's worth emphasizing that lower values of MAE, RMSE, and MAPE indicate a higher level of forecasting performance, as they assess the difference between actual and predicted values.

B. Model Optimization and Time Series Cross Validation

Optimizing the model is a crucial aspect of the modeling process, as it involves finding the most suitable set of hyperparameters for optimal model performance. In our study, we utilized the grid search method to fine-tune the hyperparameters of our machine learning models, which include Linear Regression, Ridge Regression, Lasso Regression, Elastic Net, SVR, Decision Tree, Random Forest, and Gradient Boosting. For each model, we specified a range of values for each hyperparameter and tested all possible combinations using the grid search method, as outlined in Table I. We did not include the Linear Regression's hyperparameters because it solely calculated the intercept. We selected the best set of hyperparameters based on the least MAPE. Meanwhile, for statistical models, the optimal hyperparameters were determined based on AIC and BIC values [23]. Nonetheless, we chose the best model based on the lowest MAPE which is equivalent to choosing the optimal model based on AIC and BIC values. For SARIMAX models, we used the same approach as the machine learning models, as there is no implementation available for determining the optimal hyperparameters. To train SARIMAX models with exogenous variables, we used the order in the SARIMA model and feature engineered all possible combinations of those features.

To optimize model performance through hyperparameter tuning, cross-validation [18] is often used, especially with smaller datasets where creating representative training, validation, and test sets is challenging. Cross-validation involves splitting a single training dataset into subsets of training and test datasets. The model is scored on the held-out fold, and the scores are averaged across the splits to evaluate the model's performance. However, this approach is not suitable for time series forecasting [26] because it may use future information to forecast

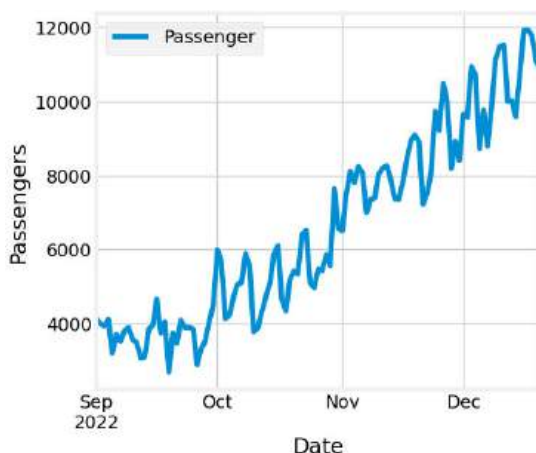


Fig. 1. Daily air passengers at the Phuket International Airport.

TABLE I
HYPERPARAMETERS OF VARIOUS MACHINE LEARNING MODELS.

Model Name	Parameters	Description	Number of samples	Range
Ridge Regression	alpha	Constant that multiplies the L2 term.	20	$[10^{-5}, \dots, 1]$
Lasso Regression	alpha	Constant that multiplies the L1 term.	20	$[10^{-5}, \dots, 1]$
Elastic Net	alpha	Constant that multiplies the penalty terms.	20	$[10^{-3}, \dots, 1]$
	L1_ratio	Mixing parameter between L1 and L2 penalties.	11	$[0, 1]$
SVR	kernel	The kernel function used for the decision function.	3	linear, poly, rbf
	C	Penalty parameter for the error term.	20	$[10^{-3}, \dots, 10^3]$
	epsilon	The size of the epsilon-tube used in the margin.	10	$[10^{-4}, \dots, 10]$
	gamma	Parameter for controls the shape of the boundary.	7	$[10^{-5}, \dots, 1, \text{scale}, \text{auto}]$
Decision Tree	max_depth	Maximum depth of the decision tree.	11	$[10, 20, \dots, 110]$
	min_samples_split	Minimum of samples required to split node.	5	$[2, \dots, 20]$
	min_samples_leaf	Minimum of samples required to be a leaf node.	5	$[1, \dots, 10]$
Random Forest	n_estimators	Number of trees in the forest.	10	$[200, \dots, 2000]$
	max_depth	Maximum depth of each decision tree in the forest.	11	$[10, 20, \dots, 110]$
	min_samples_split	Minimum of samples required to split node.	5	$[2, \dots, 20]$
	min_samples_leaf	Minimum of samples required to be a leaf node.	5	$[1, \dots, 10]$
	bootstrap	Whether to use bootstrap samples when building trees.	2	$[\text{True}, \text{False}]$
Gradient Boosting	n_estimators	Number of boosting stages to perform.	10	$[5, \dots, 500]$
	max_depth	Maximum depth of each decision tree in the ensemble.	10	$[1, 2, 3, \dots, 10]$
	learning_rate	Shrinkage parameter applied to the weights of each tree.	5	$[0.01, 0.1, 1, 10, 100]$

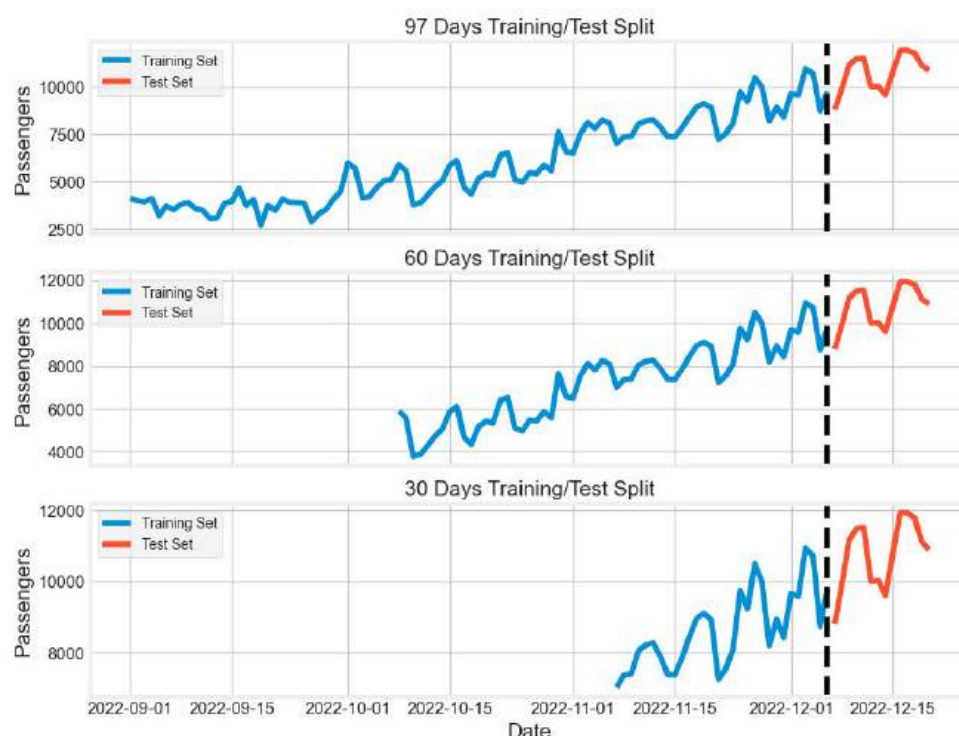


Fig. 2. Training and Test Data Split.

the past. Therefore, we used temporal folds and fix the test set at the same time when training each model to compare it with other models. We selected the number of training sets to train models, and in our study, we chose six periods of time, including 97 (full training set), 75, 60, 45, 30, and 28 days. Fig. 2 illustrates our training and test data split.

IV. RESULT

We used 85% of the daily air passenger data from the Phuket International Airport as the training set, consisting of 97 observations spanning from September 1 to December 6, 2022. The remaining 15% of the data, comprising 14 observations

from December 7 to December 20, 2022, used as the test set. We split the dataset in this manner because the dataset only covered a three-and-a-half-month period, making it necessary to include data from each month in the training process.

Following the experimental design, we evaluated the models based on the least MAPE as displayed in Table II. The results showed that SVR performed better than the other machine learning models, followed by Lasso regression, with a MAPE of more than 5% for the other models. In addition to MAPE, other evaluation metrics such as MAE and RMSE also demonstrated that SVR and Lasso regression outperformed the other models.

TABLE II
EXPERIMENTAL RESULTS OF VARIOUS MACHINE LEARNING MODELS.

Model Name	MAE	RMSE	MAPE
Linear Regression	684.2845	757.2031	0.0631
Ridge Regression	602.1751	681.4786	0.0559
Lasso Regression	526.8889	623.9476	0.0497
Elastic Net	586.3512	674.4896	0.0548
SVR	463.0979	581.9560	0.0444
Decision Tree	905.1428	1063.9288	0.0819
Random Forest	1079.0767	1230.9421	0.0968
Gradient Boosting	1049.6667	1223.4509	0.0956

Fig.3(a) illustrated the comparison between actual and predicted values of air passengers generated by different models. Our observation showed that the machine learning models were not effective in capturing the trend and seasonality of air passenger traffic. This is because the insufficient correlation between the exogenous variables and air passenger traffic, as well as the limited range of hyperparameters that we used from Table I. We also evaluated statistical models based on AIC and BIC, and selected the three models with the lowest values for further analysis. The results are shown in Table III.

TABLE III
AIC & BIC OF STATISTICAL MODELS.

Model Name	AIC	BIC
SARIMA(1, 1, 1)(0, 1, 1) ₇	312.046	315.033
SARIMAX(1, 1, 1)(0, 1, 1)₇	3.618	13.437
VAR(7)	540.465	586.704

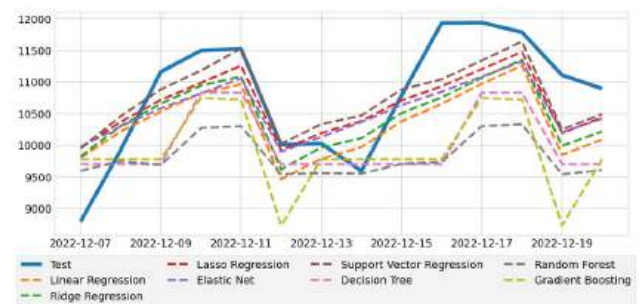
SARIMAX(1, 1, 1)(0, 1, 1)₇ had the lowest AIC and BIC values, and in Table IV, we can see that this model also had the lowest MAE, RMSE, and MAPE among the statistical models.

TABLE IV
EXPERIMENTAL RESULTS OF VARIOUS STATISTICAL MODELS.

Model Name	MAE	RMSE	MAPE
SARIMA(1, 1, 1)(0, 1, 1) ₇	409.5075	617.1881	0.0376
SARIMAX(1, 1, 1)(0, 1, 1)₇	380.6862	507.2117	0.0346
VAR(7)	851.3203	1077.8091	0.0789

The results showed that statistical models with the lowest AIC and BIC outperformed the machine learning models with the least MAPE. Fig. 4 displayed a comparison between the actual and predicted values of air passengers from the two best models, namely SVR and SARIMAX. Although both models were able to capture some aspects of the trend and seasonality, the SARIMAX model performed better than the SVR model. However, Table V revealed that the best SVR model was trained using 97 training data, while the best SARIMAX model was trained using only 30 training data. This implied that careful selection of the training data was necessary to obtain the best performance from the SARIMAX model, whereas the SVR model performed better with a larger amount of training data. In light of these observations, the advantages of SVR became evident in the context of air passengers forecasting in Phuket, Thailand. SVR demonstrated inherent strengths in capturing complex relationships, adapting to non-linear patterns, and resiliently handling high-dimensional data, all of which aligned well with the intricacies of the number of air passengers. Furthermore, its capability to navigate external influences, like

date features and exchange rate lags, enhanced its predictive abilities, rendering SVR a preferred option for achieving precise forecasts.



(a) Machine Learning models



(b) Statistical models

Fig. 3. Actual versus Predicted air passengers

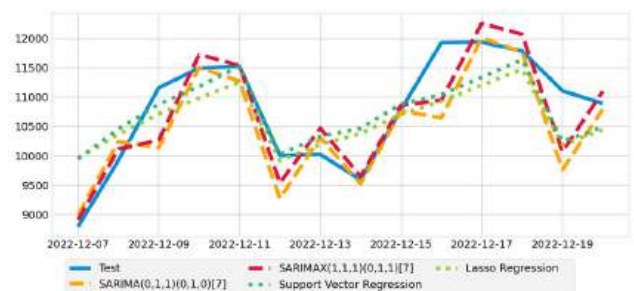


Fig. 4. Actual versus Predicted air passengers between Machine Learning and Statistical.

V. CONCLUSIONS AND FURTHER WORK

Our findings indicated that the SARIMAX model is the most suitable for air passenger forecasting in Phuket because of the significant fluctuations in trends caused by the pandemic. This was supported by the SARIMAX model's lowest MAE, RMSE, and MAPE values. However, the model's performance could be affected by the choice of training data, leading to instability in its predictions, as Table V demonstrated. In contrast, machine learning models became more stable with more training data, particularly in the future when pandemic effects were less significant and time series trends became more stable. Both statistical and machine learning models had their advantages and disadvantages, and the choice of model should be based on the available data and specific forecasting needs.

Our future work includes exploring other forecasting models (Neural Networks, LSTM etc.) to improve forecasting accuracy.

TABLE V
COMPARISON OF BEST PERFORMING MODELS FOR AIR PASSENGER FORECASTING.

Model Name	Training Size	MAE	RMSE	MAPE
SARIMA(1, 1, 1)(0, 1, 1) ₇	28	409.5075	617.1881	0.0376
SARIMAX(1, 1, 1)(0, 1, 1)₇	30	380.6862	507.2117	0.0346
Lasso Regression ['alpha': 10 ⁻⁶ , 'fit_intercept': True]	97	526.8889	623.9476	0.0497
SVR ['kernel': 'linear', 'epsilon': 10, 'C': 100]	97	463.0979	581.9560	0.0444

We plan to deploy our model on a chatbot platform like LINE to generate real-time air passenger predictions for the benefit of the local economy and tourism industry.

REFERENCES

- [1] J. Feng, L. Yongwu, S. Shaolong, L. Hongtao, "Forecasting air passenger demand with a new hybrid ensemble approach," *Journal of Air Transport Management*, vol. 83, 2020. DOI:10.1016/j.jairtraman.2019.101744.
- [2] G. Ulrich, Z. Bozana, "Forecasting air passenger numbers with a GVAR model," *Annals of Tourism Research*, vol. 89, 2021. DOI:10.1016/j.annals.2021.103252.
- [3] X. Gang, W. Shouyang, K. L. Kin, "Short-term forecasting of air passenger by using hybrid seasonal decomposition and least squares support vector regression approaches," *Journal of Air Transport Management*, vol. 37, pp. 20-26, 2014.
- [4] T. Wai Hong Kan, B. Hatice Ozer, G. Andrew, G. Hamish, "Forecasting of Hong Kong airport's passenger throughput," *Tourism Management*, vol. 42, pp. 62-76, 2014.
- [5] S.V. Gudmundsson, M. Cattaneo, R. Redondi, "Forecasting temporal world recovery in air transport markets in the presence of large economic shocks: The case of COVID-19," *Journal of Air Transport Management*, vol. 91, 2021. DOI:10.1016/j.jairtraman.2020.102007.
- [6] S. Xu, H. K. Chan, T. Zhang, "Forecasting the demand of the aviation industry using hybrid time series SARIMA-SVR approach," *Transportation Research Part E*, vol. 122, pp. 169-180, 2019.
- [7] H. Song, R. T.R. Qiu, J. Park, "A review of research on tourism demand forecasting: Launching the Annals of Tourism Research Curated Collection on tourism demand forecasting," *Annals of Tourism Research*, vol. 75, pp. 338-362, 2019.
- [8] A. E. Hoerl, R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 42(1), pp. 80-86, 1970.
- [9] R. Tibshirani, "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, pp. 267-288, 1996.
- [10] H. Zou, and T. Hastie, "Regularization and Variable Selection via the Elastic Net," *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, vol. 67(2), pp. 301-320, 2005.
- [11] DEEPA. "Linear regression based on the Regularization," accessed 10 February 2023, <https://learnerjoy.com/ridge-regression-lasso-regression-elastic-net-regression/>.
- [12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, pp. 2825-2830, 2011.
- [13] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, pp. 81-106, 1986.
- [14] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, pp. 123-140, 1996.
- [15] T. K. Ho, "Random decision forests," *Proceedings of 3rd international conference on document analysis and recognition*, vol. 1, pp. 278-282, 1995.
- [16] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55(1), pp. 119-139, 1997.
- [17] J. H. Friedman, "Stochastic gradient boosting," *Computational Statistics and Data Analysis*, vol. 38(4), pp. 367-378, 2002.
- [18] D. Berrar, "Cross-Validation," *Reference Module in Life Sciences*, 2018. Doi:10.1016/B978-0-12-809633-8.20349-X.
- [19] A. Maierbrugger, "Thailand, Singapore Seen As Top Expat Locations," accessed 21 February 2023, <http://investvine.com/thailand-singapore-seen-as-top-expat-locations/>.
- [20] H. Drucker, C. J. C. Burgers, L. Kaufman, A. Smola, V. Vapnik, "Support Vector Regression Machines," *Advances in Neural Information Processing Systems*, vol. 9, pp. 155-161, 1996.
- [21] A. Smola and B. Scholkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, pp. 199-222, 2004.
- [22] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv*, 2016, Doi:10.48550/arXiv.1609.04747.
- [23] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, G. M. Ljung, "Time series analysis: Forecasting and control, 5th Edition," Hoboken, NJ: Wiley, 2015.
- [24] P. Durka, S. Pastorekova, "ARIMA vs. ARIMAX - which approach is better analyze and forecast macroeconomic time series?," *Proceedings of 30th International Conference Mathematical Methods in Economics*, pp. 136-140, 2012.
- [25] A. Pankratz, "Forecasting with Dynamic Regression Models," Wiley-Interscience, 1991.
- [26] C. A. Sims, "Macroeconomics and Reality," *Econometrica*, vol. 48(1), pp. 1-48, 1980.
- [27] A. Amidon, "Don't Use K-fold Validation for Time Series Forecasting," *Towards Data Science*, accessed 14 February 2023, <https://towardsdatascience.com/dont-use-k-fold-validation-for-time-series-forecasting-30b724aaca64>

Machine Learning Models for Micro-bubble Image Detection in Mosquito Sprayer Quality Control: Addressing Class and Scale Imbalance

Suppasit Srisaeng¹, Pongsakorn Sadakorn¹, Kritchavat Ploddi¹, Darin Areechokechai¹, Worasait Suwannik²

1. Department of Disease Control, Ministry of Public Health Thailand

2. Department of Computer Science, Kasetsart University

Abstract— Introduction: Vector-borne diseases (VBDs) account for around 17% of global infectious diseases, with malaria and dengue being the most prevalent. Effective VBD prevention involves controlling mosquito populations by spraying micro-bubble (MB) insecticide. Quality control measures for MB size are essential, but few centers can assess MB size measurement. Machine learning (ML) is a powerful tool for automating and enhancing image analysis tasks, including measuring MB size. In this study, we train and evaluate ML models using an MB image dataset to assess their performance.

Methods: Our experimental study used an MB image dataset labeled by specialists, with images resized and randomly assigned to train, validate, and test datasets. We employed various evaluation metrics, such as recall, precision, and mean average precision (mAP). Image augmentation techniques were applied to increase the number of unqualified MBs. Pretrained models from Detectron2 (D2), MMDetection (MM), and YOLO libraries were used for training and evaluation.

Results: We identified severe class and scale imbalances in the dataset. Single-stage and two-stage CNN models demonstrated superior performance in MB detection compared to DCN and transformer-based models. D2 outperformed MM, and training Faster-RCNN R-50 on an augmented dataset improved performance for unqualified MB detection. Considering multiple evaluation metrics, YOLO v7 exhibited the highest accuracy and effectiveness in detecting MBs.

Conclusion: Our study revealed the effectiveness of CNN models in MB detection and the importance of selecting appropriate libraries and pretrained weights. Image augmentation improved unqualified MB detection while considering multiple evaluation metrics provided a more comprehensive understanding of model performance. Models like YOLO v7 and Faster-RCNN R-50 and X-101 are recommended for practical MB quality assessment tasks, as they demonstrate high precision and recall performance for qualified and unqualified MBs.

Keywords—Artificial intelligence, Object detection, Supervised Machine Learning, Vector Borne Diseases

I. INTRODUCTION

Vector-borne diseases (VBDs) account for around 17% of global infectious diseases, with malaria and dengue being the

most prevalent. Annually, malaria causes approximately 219 million cases and over 400,000 deaths, while dengue results in an estimated 96 million cases and 40,000 deaths[1]. In Thailand, dengue has an annual incidence of 50,000-130,000 cases, and malaria records 3,000-7,000 cases annually[2]. Thailand aims to eliminate malaria by 2026, with vector control (VC) as an urgent national strategy. VC measures to target both larval and adult mosquitoes. Spraying insecticide micro-bubble (MB) targets adult mosquitoes, reducing disease transmission risk[3] and also effective for dengue control[4]. Space spraying quality is based on the sprayer equipment, the applicator, and the chemical used. These sprayers must adhere to quality controls, especially regarding temperature, flow rate, and ensuring MB sizes stay below 30 μm [5].

In Thailand, 38 local VBD Centers control sprayer quality. Most centers can measure temperature and flow rate, but few can assess MB size measurement. The Vector-Borne Disease Division (VBDD) oversees MB qualification and size quality control. However, only three officers can do MB size measurements. Officers must use a microscope to measure over 600 MBs per sprayer, which takes around ten hours. It places a significant burden on officers, demanding skill and endurance, leading to workflow delays that can span months. That increases the risk of unchecked unqualified sprayers, compromising the efficacy of vector control efforts.

Machine learning (ML) offers a powerful tool for automating and enhancing various tasks, particularly image analysis. ML algorithms can accurately capture and detect image features, opening new possibilities for improving efficiency in image analysis tasks. However, image object detection remains challenging due to class, scale, and spatial imbalance[6]. Tailored approaches are needed to address these issues and fully harness ML's potential in imaging applications. In this study, we train and evaluate ML models using our MB image dataset to assess their performance.

II. METHODS

In our experimental study, droplets from sprayers were captured on magnesium oxide-coated slides from 38 local centers not exceeding a year from their collection date. Officers captured MB images using a high-quality microscope in the VBDD laboratory. MBs were labeled as either 'qualified' (non-overlapping with a glowing circular rim) or 'unqualified' (overlapping with a faded non-circular rim). Then the images were resized from 1600x1200 to 500x375 pixels, underwent color normalization, and were computerized randomly divided into training, validation, or testing datasets. Datasets were organized in YOLO and COCO formats. To increase the quantity of unqualified MBs, 1252 of the 2428 slides with

unqualified MBs images were duplicated and augmented into four sets. These included applying black or average color (gray) patches to qualified MBs, flipping, or rotating. There were two training processes: 1) For non-augmented training, the 2428 slide images were randomly divided into 1457 (60%) for training, 486 (20%) for validation, and 485 (20%) for testing. 2) For augmented training, the training set expanded to include an additional 1252 augmented images. Both the validation and testing phases utilized the same non-augmented images (Figure.1)

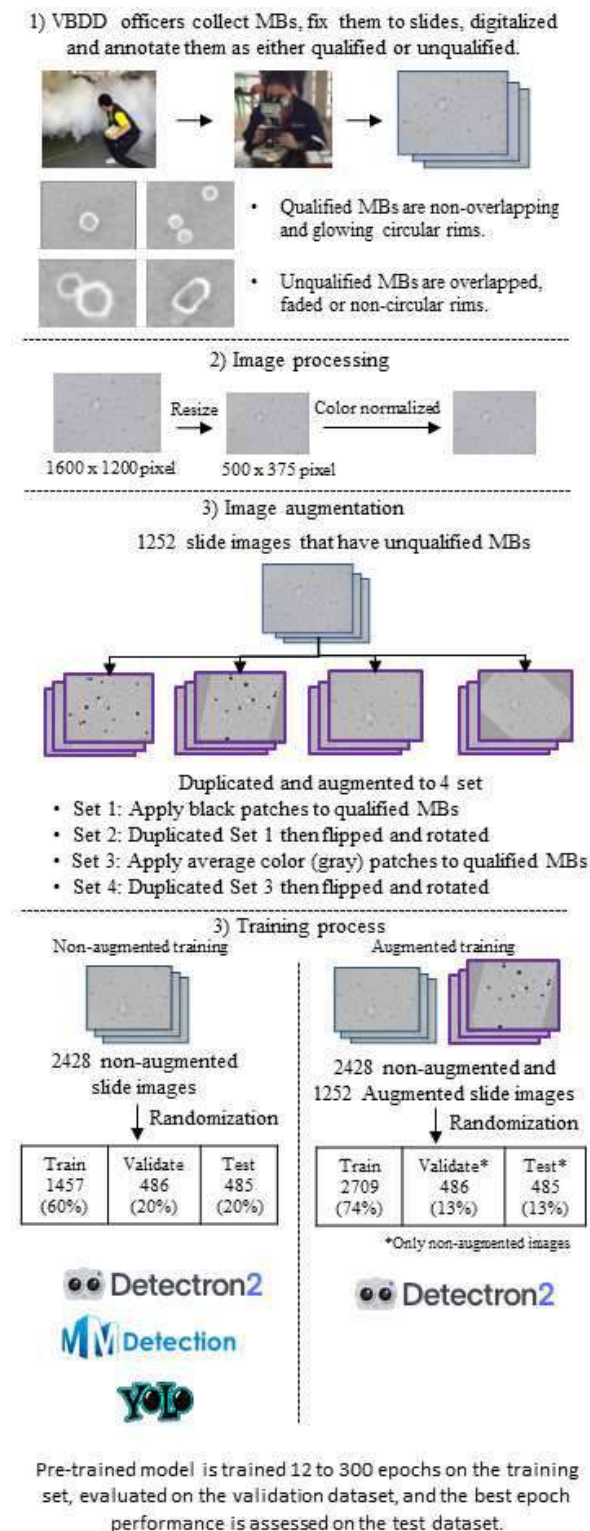


Fig. 1. Study workflow

Pretrained models from Detectron2 (D2), MMDetection (MM), and YOLO libraries were trained and validated over 12-300 epochs. The best performance epoch by mAP was evaluated on the test dataset. We used various models grouped as follows: Convolutional Neural Networks (CNN), including single-stage models like RetinaNet[7] and YOLO v7[8], and two-stage models like Faster-RCNN[9], Vision Transformer (ViT), including FocalNet[10] and Swin Transformer v2[11], and Deformable Convolutional Networks (DCN) likes InterImage[12]. The training was primarily on non-augmented datasets (Table.1). Only the Faster-RCNN pretrained by D2 was trained using the augmented datasets, given its shorter training time than larger models (Table.2).

We used Mann-Whitney U test and Chi-squared tests for comparison with a p-value < 0.05. Model metrics included Recall, Precision, max F1 score, PR curve, mAP, detection time, and median differences of label and prediction in the box's area and count. We used Python 3.9 on Google Colab's Tesla T4, with data stored on Google Drive.

III. RESULTS

There were 37,336 qualified and 2,923 unqualified MBs, with a ratio of 12.8:1 and a chi-square significant difference. Additionally, there was a difference in the median sizes, with qualified MBs having a median of 82 pixels (IQR 49-141) and unqualified MBs having a median of 160 pixels (IQR 51-335) with a significant difference.

Regarding the location of MBs in the images, the median label box in the horizontal (X-axis) for qualified MBs was at pixel 248 (IQR 129-366), while unqualified MBs were at pixel 254 (IQR 126-367). In the vertical (Y-axis), the median label box for qualified MBs was at pixel 186 (IQR 95-276), while unqualified MBs were at pixel 183 (IQR 97-272). Both axes had no significant difference. However, there was a foreground-background imbalance with the median sum area of label boxes 654 pixels (IQR 1,193-2,335) and a non-label area of 185,164 pixels (IQR 186,307-186,846) with a significant difference.

In the non-augmented dataset, models with mAP above 40 include pretrained D2's RetinaNet X-101, D2's Faster-RCNN R-50, D2's Faster-RCNN X-101, and YOLO v7. YOLO v7 achieved the highest mAP of 52.4 for all MB, while Faster-RCNN R-50 had the highest max F1 score of 88 for qualified MB. YOLO v7 also achieved the highest max F1 score of 47 for unqualified MB (Table.1).

Faster-RCNN R-50 showed an increase in unqualified MB AP from 17.7 on the non-augmented dataset to a range of 22.4-27.4 on the augmented dataset, with the highest performance achieved by technique of applying black patches to qualified MB (Table.2). Upon reviewing the predicted boxes, we observed that some, such as RetinaNet, FocalNet, and InternImage, produced duplicate predicted boxes in the same area and captured artifact features (Fig.2).

The PR curves of RetinaNet, Faster-RCNN, and YOLO v7 models indicate high performance for qualified MB as they approach the upper-right corner of the graph. However, none of the models achieved high performance for unqualified MB (Fig.3). For unqualified MB, all models displayed a median difference of one box count without any statistical difference. However, only Faster-RCNN R-50 and YOLO v7 showed a median box size difference of less than 35 pixels without statistical difference. For qualified MB, RetinaNet, Faster-RCNN, and YOLO v7 exhibited a median difference of one box count and a median box size difference of fewer than 26 pixels, with no statistical difference (Table.3)

IV. DISCUSSION

We identified class and scale imbalances in this study but did not find spatial imbalances. In the non-augmented dataset, YOLO v7 achieved the highest mAP for all MBs, while Faster-RCNN R-50 had the highest max F1 for qualified MBs. YOLO v7 also achieved the highest max F1 score for unqualified MBs. Faster-RCNN R-50 demonstrated an increase in unqualified MB AP when using the augmented dataset, with the best performance achieved using the augmentation technique of applying black patches to qualified MBs. RetinaNet, FocalNet, and InternImage produced duplicate predicted boxes and captured artifact features. The PR curves for RetinaNet, Faster-RCNN, and YOLO v7 models showed high performance for qualified MBs, but none achieved high performance for unqualified MBs. The study also highlighted the median differences in box counts and size for various models. Faster-RCNN R-50 and YOLO v7 show the smallest differences in both categories for unqualified MBs.

A. Handle imbalance issues

Class imbalances are prevalent in machine learning tasks and can negatively impact model performance. Data-level techniques, such as image augmentation, model-level techniques, like focal loss function from RetinaNet, and two-stage detectors, can address these imbalances[6], [13]. In this study, data-level techniques improved the Faster-RCNN R-50 model's performance in detecting unqualified MBs. Future research may explore additional models and techniques to address class and scale imbalances in MB detection tasks.

B. CNNs Excel in MB Detection Over DCN & ViT

We observed that single-stage and two-stage CNN models demonstrated superior performance in MB detection compared to DCN and transformer-based models. This superiority may be attributed to several factors: (1) Task suitability: CNN-based models are specifically designed for object detection tasks, making them more apt for this particular problem; (2) Network architecture: Single-stage and two-stage CNNs offer effective feature extraction and object localization capabilities[6]; (3) Dataset size and complexity: DCN and transformer-based models often require larger and more diverse datasets for optimal performance, which may not be readily available for this specific task. These findings suggest that leveraging the strengths of CNN models can improve the detection of qualified and unqualified MBs in class and scale imbalances.

C. Detectron2 Outshines MMDetection

We compared the performance of RetinaNet X-101 and Faster-RCNN X-101 models trained with two different libraries, MM and D2, while keeping the training setup identical. We found that Detectron2 yielded superior results, which could be attributed to several factors: (1) Default hyperparameters: Detectron2's default settings, such as learning rate and weight decay, may be better optimized for the specific task of MB detection; (2) Pretrained weight models: Both libraries utilize pretrained weight models on the ImageNet dataset but with different training setups. Detectron2's pretrained weights could offer a better starting

point for this task; (3) Transfer learning process: Variations in the pretrained weight models can impact the transfer learning process, leading to discrepancies in the final model performance. Our findings highlight the importance of selecting the appropriate library and pretrained weights when training deep-learning models for detecting qualified and unqualified MBs.

D. Augmented Data Enhances Unqualified Micro-Bubble Detection

Training Faster-RCNN R-50 on an augmented dataset improved performance for unqualified MB detection, with AP increasing from 17.7 to a range of 22.4-27.4. This improvement can be attributed to (1) Increased dataset diversity, which enables the model to learn various features and patterns, making it more robust to real-world scenarios; (2) Handling of class imbalance by oversampling underrepresented classes; (3) Improved object localization, as the model learns to identify objects with varying orientations and scales better[14]. However, the performance for the non-augmented class remained relatively stable (AP 68.8 to a range of 66.3-69.2).

E. Beyond AP: Comprehensive Evaluating

It is crucial to consider a range of metrics, including recall, precision, and F1 scores, instead of solely relying on Average Precision (AP) when evaluating MB detection performance. For instance, RetinaNet X-101 achieved a qualified MB AP of 70.5, but with a precision of 31, recall of 99, and max F1 of 47 due to multiple predicted bounding boxes for the same ground truth box. In contrast, YOLO v7 displayed a similar AP of 73.8 but outperformed with a precision of 93, recall of 95, and max F1 of 83. Assessing these additional metrics and reviewing result images allows for a more comprehensive evaluation, better-informing model selection for real-world tasks.

F. Simpler explanation matrix

A simpler explanation of model performances for laypersons could involve examining the differences in predicted and labeled bounding box sizes and counts for each image. For example, YOLO v7 exhibits a median sum of a predicted size mismatch of just 2 pixels and a predicted box count mismatch of 0 boxes per image, highlighting its accuracy and effectiveness in detecting MBs.

G. Selecting Suitable Models

For quality assessment tasks in MB detection, choosing models with high precision and recall performance for qualified and unqualified MBs is essential. Models like YOLO v7, Faster-RCNN R-50, and X-101 excel in these metrics, ensuring accurate classification and minimizing false positives. These models are recommended for practical MB quality assessment.

V. LIMITATION

This study used image data from a high-quality microscope in the VBDD laboratory, lacking data from low-quality microscope local centers, which might limit the generalizability of the results to local contexts. Additionally, the investigation did not explore the impact of various hyperparameters on model performance, including learning rate, batch size, or the number of epochs, which, if

optimized, could potentially enhance model performance. Furthermore, due to the cost and time associated with training on augmented datasets, we did not train and test multiple models, limiting the evaluation of other potential models in this study.

VI. CONCLUSION

We evaluate various ML models' performance in object detection of qualified and unqualified MB of mosquito sprayers. Our findings demonstrated that YOLO v7 and Faster-RCNN R-50 outperformed other models in detecting both qualified and unqualified MBs. The study also highlighted the challenges of class and scale imbalances and the importance of addressing them using data-level and model-level techniques. Furthermore, we emphasized the need for comprehensive evaluation metrics, such as recall, precision, and F1 scores, in addition to AP. The findings suggested that YOLO v7, Faster-RCNN R-50, and X-101 are suitable models for practical MB quality assessment tasks, given their high precision and recall performance.

REFERENCES

- [1] World Health Organization, "Vector-borne diseases," 2020. [www.who.int. https://www.who.int/news-room/fact-sheets/detail/vector-borne-diseases](https://www.who.int/news-room/fact-sheets/detail/vector-borne-diseases) (accessed Apr. 10, 2023).
- [2] D. Vector-Borne Disease Division, "Dengue report 2022 4 Jan 2023," 2023. <https://datastudio.google.com/reporting/84a91f26-9f99-4d85-aa9c-9f2f42205a77/page/cFWgC> (accessed Apr. 10, 2023).
- [3] H. Dahmana and O. Mediannikov, "Mosquito-Borne Diseases Emergence/Resurgence and How to Effectively Control It Biologically," *Pathogens*, vol. 9, no. 4, p. 310, Apr. 2020, doi: 10.3390/pathogens9040310.
- [4] N. G. Gratz, "The control of *Aedes aegypti* in South-east Asia and the Western Pacific.," *Bull World Health Organ*, vol. 36, no. 4, pp. 614–7, 1967.
- [5] P. and E. W. P. E. S. (WHOPES) Communicable Disease Control, Space spray application of insecticides for vector and public health pest control A practitioner's guide. Accessed: Apr. 10, 2023. [Online]. Available: https://apps.who.int/iris/bitstream/handle/10665/68057/WHO_CDS_WHOPE_S_GCDPP_2003.5.pdf
- [6] K. Oksuz, B. C. Cam, S. Kalkan, and E. Akbas, "Imbalance Problems in Object Detection: A Review," *IEEE Trans Pattern Anal Mach Intell*, vol. 43, no. 10, pp. 3388–3415, Oct. 2021, doi: 10.1109/TPAMI.2020.2981890.
- [7] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," Aug. 2017.
- [8] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," Jul. 2022.
- [9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," Jun. 2015.
- [10] J. Yang, C. Li, X. Dai, L. Yuan, and J. Gao, "Focal Modulation Networks," Mar. 2022.
- [11] Z. Liu et al., "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," in 2021 IEEE/CVF International Conference on Computer Vision (ICCV), IEEE, Oct. 2021, pp. 9992–10002. doi: 10.1109/ICCV48922.2021.00986.
- [12] W. Wang et al., "InternImage: Exploring Large-Scale Vision Foundation Models with Deformable Convolutions," Nov. 2022.
- [13] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *J Big Data*, vol. 6, no. 1, p. 27, Dec. 2019, doi: 10.1186/s40537-019-0192-5.
- [14] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *J Big Data*, vol. 6, no. 1, p. 60, Dec. 2019, doi: 10.1186/s40537-019-0197-0.

TABLE I. MODELS ON NON-AUGMENT DATASET TRAINING

Model	Backbone	Library	Epoch	Unqualified MB				Qualified MB				mAP	Time per image
				Precision	Recall	Max F1	AP	Precision	Recall	Max F1	AP		
RetinaNet	X-101	D2	12	4	91	7	31.2	31	99	47	70.5	50.8	0.14 s
		MM	12	2	23	4	8.0	16	94	28	31.8	16.3	0.24 s
Faster-RCNN	R-50	D2	12	34	30	32	17.7	83	95	88	68.8	43.2	0.10 s
	X-101	D2	12	38	32	35	26.5	96	76	85	73.3	49.9	0.26 s
		MM	12	37	46	41	23.9	89	29	44	23.8	24.0	0.10 s
YOLO v7		YOLO	300	54	42	47	31.0	93	95	83	73.8	52.4	0.08 s
Swin TransformerV2	RCNN	MM	12	27	42	33	20.0	85	28	42	22.7	21.4	0.26 s
FocalNet	RCNN	MM	12	14	43	21	13.5	80	28	42	21.3	17.4	0.25 s
InternImage	Cascade-RCNN	MM	12	18	50	26	25.5	84	34	49	28.4	21.6	0.42 s

TABLE II. MODELS ON AUGMENT DATASET TRAINING AND BASELINE PERFORMANCE

Model	Library	Epoch	Data	Unqualified MB				Qualified MB				mAP	Time per image
				Precision	Recall	Max F1	AP	Precision	Recall	Max F1	AP		
Faster-RCNN With R-50	D2	12	Non-augment	34	30	32	17.7	83	95	88	68.8	43.2	0.10s
			Gray patch	22	52	31	25.5	87	89	88	69.0	47.3	0.10s
			Gray Flip Rotate Crop	24	53	33	26.3	93	88	90	66.3	46.3	0.10s
			Black patch	27	48	35	27.4	82	93	87	69.2	48.3	0.10s
			Black Flip Rotate Crop	33	37	35	22.4	87	94	90	68.6	45.5	0.10s

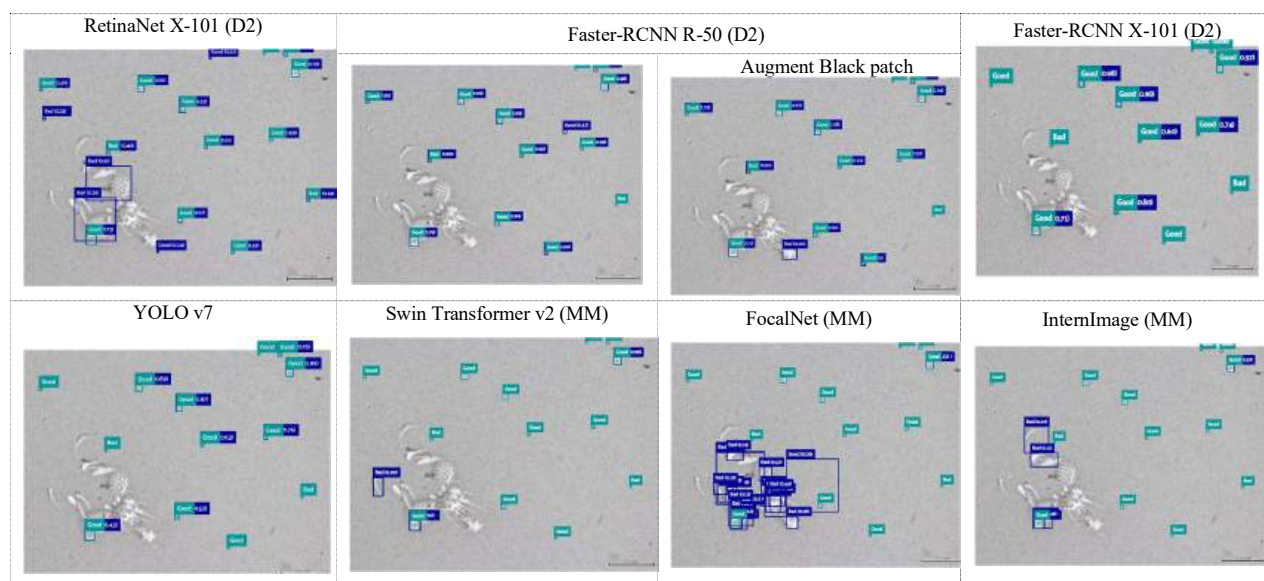


Fig. 2. Example of model predicted box (Blue box) and labeled box (Green box)

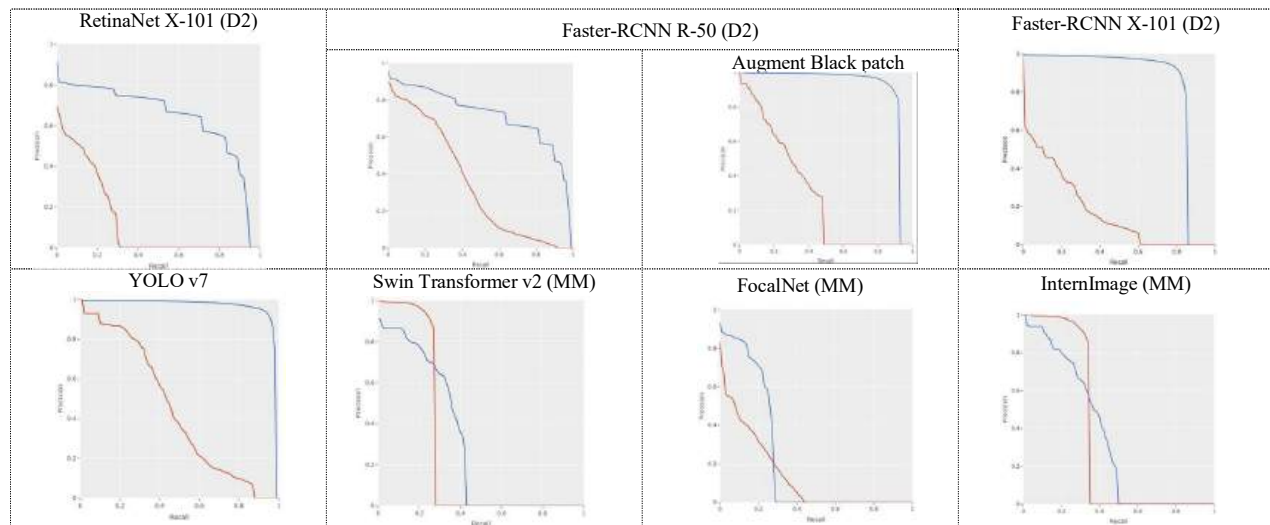


Fig. 3. Model precision-recall curve (Blueline is Qualified MB, Redline is Unqualified MB)

TABLE III. DIFFERENCE BETWEEN PREDICTED AND LABELED BOX AREA AND COUNT PER IMAGE

Model	Backbone	Library	Unqualified MB		Qualified MB	
			Median difference between predicted box area and labeled box area per image	Median difference between predicted box count and labeled box count per image	Median difference between predicted box area and labeled box area per image	Median difference between predicted box count and labeled box count per image
RetinaNet	X-101	M2	145 pixels (IQR -30, 908) *	0 box (IQR -1, 2.25)	17 pixels (IQR -28, 158)	1 box (IQR 0, 2)
Faster-RCNN	R-50	M2	-35 pixels (IQR -113, 44)	-1 box (IQR -2, 0)	-4 pixel (IQR -54, 73)	1 box (IQR 0, 2)
	Augment Black patch	M2	-15 pixels (IQR -57, 206) *	-1 box (IQR -1, 0)	21 pixel (IQR -20, 107)	0 box (IQR 0, 1)
	X-101	M2	-26 pixel (IQR -74, 216) *	-1 box (IQR -1, 0)	-26 pixel (IQR -74, 216)	-1 box (IQR -1, 0)
YOLO v7		YOLO	-34 pixel (IQR -113, 23)	-1 box (IQR -1, 0)	-2 pixel (IQR -54, 90)	0 box (IQR -1, 1)
Swin Transformer v2	RCNN	MM	-34 pixel (IQR -97, 129)	-1 box (IQR -1, 0)	-444 pixel (IQR -1018, -201) *	-8 box (IQR -16, -4) **
FocalNet	RCNN	MM	-22 pixel (IQR -74, 461) *	-1 box (IQR -1, 0)	-464 pixel (IQR -968, -175) *	-9 box (IQR -16, -4) **
InternImage	Cascade- RCNN	MM	-23 pixel (IQR -72, 272) *	-1 box (IQR -1, 0)	-390 pixel (IQR -826, -160) *	-8 box (IQR -15, -4) **

*Unpair T-Test p-value < 0.05, **Chi-square test p-value < 0.05

Real-time prediction of free lime in cement clinker using support vector machine algorithm

Rathimala Kannan

Department of Information Technology
Faculty of Management
Multimedia University
Cyberjaya, Selangor, Malaysia
rathimala.kannan@mmu.edu.my

Kannan Ramakrishnan

Faculty of Computing and Informatics
Multimedia University
Cyberjaya, Selangor, Malaysia
kannan.ramakrishnan@mmu.edu.my

Ahmad Hanif Ghanim

MBA Student,
IPMI International Business School
Jakarta, Indonesia
ahmad.ghanim@ipmi.ac.id

Satesh More

Head, Performance Management
Technical Directorate
PT Cemindo Gemilang Tbk
Jakarta, Indonesia
satesh.more@cemindo.com

Abstract—Free lime content is an important quality parameter in the production of clinker, targeted between 0.5% to 1.5%. Existing studies have tried to predict absolute free lime content, using soft sensors data, with limited success due to the complexity of the clinker burning process. Subject matter experts believe that, instead of predicting the free lime absolute value, predicting free lime quality as good, over-burn, or under-burn is more practically beneficial. This study aims to predict the free lime clinker quality as good or under-burn by leveraging data mining methods and machine learning techniques. Seven months of hourly data pertinent to rotary kiln feed chemistry and operation parameters were collected from a real operational cement plant. Classification models were built using support vector machine (SVM). The SVM produced a sensitivity value of 0.998 for good clinker class and 0.865 for under-burn class with an accuracy of 96%. The availability of these predictions in real time can help plant operators to avoid under-burning and over-burning. Such insights will assist relevant cement plants to reduce off-specification products, coal usage, production cost, and carbon emissions.

Keywords—Cement industry, Classification algorithms, Clinker quality, Machine learning, Predictive models, Support vector machine

I. INTRODUCTION

Free lime content in cement clinker is a critical quality parameter [2], [10]. The desired target for the free lime content in the cement clinker is 0.5% - 1.5% [10] or below 2% [12]. Lower values of free lime indicates an over-burn, consuming more energy both in the clinker processing and subsequent grinding process [2]. Higher free lime content indicates under-burned clinker, which is also undesirable as the clinker would contain higher expansion factors which may lead to cracks in the construction [12]. This leads to off-specification products which require further processing to meet market specifications.

Currently, most cement production facilities measure free lime by collecting clinker samples after the whole process has been completed. This process takes place at the exit of the clinker cooler, either hourly or bi-hourly, followed by laboratory analysis. The time delay resulting from the processing and movement of materials has hindered real-

time clinker quality assessment. While there have been attempts to sample the clinker at the immediate exit from rotary kiln or inside the clinker cooler to reduce the time delays, with limited success due to the harsh environment there.

Kiln operators look at various quality indicators during operations, including measurements and visuals from various online sensors in the kiln. These indicators allow operators to make judgements about the control actions to ensure consistent clinker quality. Given this complexity, cement companies require experienced kiln operators. The extant level of instrumentation, however, still relies heavily on post-processing which takes place one to two hours after the production process. A real-time quality control system would allow kiln operators to perform adjustments while the production process is still underway, thereby keeping clinker quality within specified limits, managing energy consumption and environmental impact, and reducing the quantity of off-specification products.

The advancement of machine learning has led to various attempts to develop real time clinker free lime prediction models built using machine learning algorithms trained with historical data. The successful prediction will help in real time adjustments to operating parameters to avoid over/under burning, leading to reduced energy consumption, reduced CO₂ emissions, minimize off-specification products, and optimize production costs.

This research attempts to predict free lime content quality as good quality or under-burn by applying data mining methods and machine learning techniques. This paper contributes by developing machine learning models using SVM algorithm and identification of optimal model by varying the hyperparameter values. The findings were shared with the cement plant company, and they found them to be helpful. This study also contributes to the body of current literature on domain-driven data mining research by bridging the gap between theoretical research and practical implementations and converting the insights into business value and effect.

The remaining part of this article is arranged in the following ways. Section II describes SVM algorithms and related studies, section III presents the research methodology

and results. Section IV presents the results and discussion followed by section V conclusion.

II. SUPPORT VECTOR MACHINE

A. SVM architecture

Given the complex, multivariable nature of kiln operations, previous studies have employed SVM to predict free lime content, [10], [11], and [20]. Support Vector Machine (SVM) is a machine learning algorithm that belongs to the class of supervised learning methods [3]. It is primarily used for classification, regression, and outlier detection, and can handle both linear and non-linear data effectively [16]. SVM is known for its robustness, generalization capabilities, and ability to handle high-dimensional datasets [3], [16]. It has found applications in various fields, including image classification, text categorization, bioinformatics, and financial analysis. Support vector machine (SVM) was chosen from the literature review to develop the classification machine learning models to predict free lime quality as good / under-burn.

III. EXPERIMENT AND ANALYSIS

The production of clinker involves multiple parameters that fall into two main categories. The first is kiln operating parameters which include kiln feed rate, fuel feeding rate, main motor current, rotation speed, preheater speed, cooler speed, and gas inlets. The second category is raw meal quality, which includes lime saturation factor, aluminum modulus, and silica modulus. This study uses data from both kiln operating parameters and raw meal quality as inputs for the model, and the measured free lime content as output. This study employed KNIME Analytics Platform for all stages from data pre-processing to analysis.

A. Preprocessing of data

This study used operational data from seven months of plant operations. The initial data set contained 7,000 rows of data with eighteen variables. Initial preprocessing included handling missing data and outliers, and correlation testing of the various variables. Missing data were either imputed with the median values for input variables or removed altogether for the output variable. Outliers were found due to sensor failures, as confirmed by domain experts, and thus removed from further analysis. Correlation testing showed that thirteen variables had a strong relationship with free lime content. These were both from kiln operations and raw meal quality. The variables from kiln operations were kiln feed, kiln coal, calciner coal, kiln speed, kiln current, fan speed, calciner temperature, cooler grate speed, secondary air temperature, and tertiary air temperature. The variables for raw meal quality were kiln feed free lime, lime saturation factor, and silica modulus.

B. SVM classification model

Pre-processed data from an earlier stage is taken for the modeling purpose. There were 4123 rows and 15 variables in the processed dataset. The computation of a new binary variable was necessary for the classification of clinker quality into two categories. A new variable clinker quality was computed where free lime value over 1.5% labelled as under-burn and free lime value between 0.5 and 1.5% labelled as good clinker.

Afterwards, the data was partitioned into three sets: 'Training & validation' (modeling) and 'Test' data

(evaluation). To avoid data leakage problem, normalization of the data was done individually, after data partitioning for all three datasets. MinMax normalization method was applied to standardize all the variables between 0 to 1 value.

In the modelling phase, SVM was applied to train and validate the prediction models. Once the models were trained, the hyper parameters were tuned to identify the optimal parameters under which the model shows the best performance. The standard performance measures such as precision, sensitivity, f-measure, and accuracy were calculated for each of the models and the optimal model was determined based on the sensitivity measure. Table I depicts the training performance of the SVM model and hyperparameter tuning to identify the optimal model.

TABLE I. SVM MODEL TRAINING PERFORMANCE AND HYPERPARAMETER TUNING

Model ling: Evalu ation Data	Model ling Traini ng: Valida tion Data	Qualit y Measu re	Target: Clinker Quality	Precis ion	Sensit ivity	F-Measu re	Accura cy
50:50	70:30	Polyno mial	Good Quality	0.941	100%	0.969	0.955
			Under burn	1	0.839	0.912	
		Hypert angent	Good Quality	0.802	1	0.89	0.822
			Under burn	1	0.368	0.538	
		RBF	Good Quality	0.735	0.998	0.847	0.74
			Under burn	0.933	0.08	0.148	
	60:40	Polyno mial	Good Quality	0.947	1	0.973	0.96
			Under burn	1	0.857	0.923	
		Hypert angent	Good Quality	0.798	1	0.888	0.818
			Under burn	1	0.351	0.519	
		RBF	Good Quality	0.73	0.998	0.844	0.733
			Under burn	0.923	0.052	0.098	
	50:50	Polyno mial	Good Quality	0.937	0.996	0.965	0.949
			Under burn	0.988	0.827	0.9	
		Hypert angent	Good Quality	0.796	1	0.886	0.816
			Under burn	100%	34.3%	51%	
		RBF	Good Quality	0.73	0.996	0.843	0.732
			Under burn	0.842	0.055	0.104	
70:30	70:30	Polyno mial	Good Quality	0.934	1	0.966	0.949
			Under burn	1	0.819	0.9	
		Hypert angent	Good Quality	0.827	1	0.906	0.85
			Under burn	1	0.465	0.635	
		RBF	Good Quality	0.741	0.998	0.851	0.748
			Under burn	0.963	0.107	0.193	
	60:40	Polyno mial	Good Quality	0.943	1	0.971	0.957
			Under burn	1	0.846	0.916	
		Hypert	Good	0.815	1	0.898	0.836

		angent	Quality				
			Under burn	1	0.417	0.588	
		RBF	Good Quality	0.742	0.999	0.851	0.749
			Under burn	0.972	0.108	0.194	
	50:50	Polyno mial	Good Quality	0.946	1	0.972	0.959
			Under burn	1	0.854	0.921	
		Hypert angent	Good Quality	0.807	1	0.893	0.827
			Under burn	100	0.385	0.556	
		RBF	Good Quality	0.734	0.995	0.845	0.737
			Under burn	0.861	0.077	0.141	

As presented in Table II, the SVM model can accurately classify the good clinker quality with an accuracy rate of 0.96, an F-measure of 0.973, a precision of 0.95 and a sensitivity (recall) of 0.998, when the modelling and evaluation data split ratio is 70:30, and training and validation data split ratio is 60:40, with the quality measure polynomial.

TABLE II. SVM MODEL TESTING PERFORMANCE

Target Clinker	Precision	Sensitivity	F-Measure	Accuracy (%)
Good Clinker	0.95	0.998	0.973	0.96
Under burn	0.865	0.865	0.924	

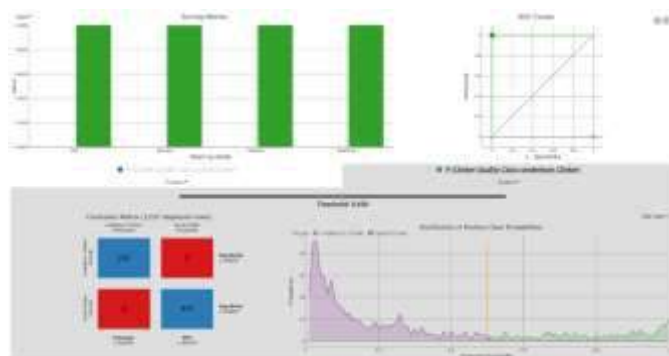


Fig. 1. SVM model performance

After the modelling process, the optimal model was evaluated against the test dataset, which was new to the model. The standard performance evaluation measures for binary classification models, such as precision, sensitivity (recall), f-measure and accuracy were used to assess the machine learning models. These measures can be derived from a confusion matrix, which consists of true positives, false positives, true negatives, and false negatives from the predictions of machine learning models [18]. Fig. 1 depicts the performance of SVM model in predicting the under-burn clinker class with prediction probability threshold value of 0.5.

IV. RESULTS AND DISCUSSIONS

The end-to-end cement value chain has a significant environmental impact. International Energy Agency (2022) suggests that cement manufacturing is the third largest consumer of industrial energy globally with consumption of 7%, while also contributing global carbon emissions of 7%, placing it as the second highest carbon contributor [19]. High residual free lime in the finished clinker causes cement over expansion and cracking. Low residual free lime content indicates that the clinker is overburdened, implying that too much fuel was used in the process. The target range of free lime in the clinker from the cement plant under study is 0.5 - 1.5%. Free lime content is over-burning and under-burning when its value is less than 0.5% and greater than 1.5%, respectively. Free lime in the clinker is normally measured in the laboratory, using chemical methods or X-Ray Diffraction (XRD) analyzer. As the clinker sample is collected at the discharge of the clinker cooler and testing time takes place in the laboratory, free lime results are of little use to the kiln operator to adjust the operation parameters. It acts more of a confirmation of what happened in the kiln, typically an hour before. Having predictive capabilities would allow for real-time measurement and adjustments, thus controlling production quality, reducing off-specification products, while optimizing the fuel usage.

Significantly, the findings from this study motivate investigation into the classification models, by computing a new variable to classify the free lime in the clinker into two classes: good clinker or under-burn. According to this research, it is possible to predict the clinker class or quality of burn as under-burn or good clinker while still maintaining a significantly higher level of accuracy. These predictions will allow kiln operators to focus on the action that needs to be taken to maintain good quality clinker and to avoid under-burning. The findings showed that the SVM was capable of recalling 99.8% good clinker quality and 86.5% of under-burn free lime content quality.

This prediction performance was higher than the existing studies that proposed regression models. Importantly, similar study on prediction of free lime [2] reported that SVM outperformed when compared with other prediction models. Therefore, this study recommends building classification models to predict free lime content quality, using SVM algorithm. These findings' implications have been discussed with pertinent plant process managers who work in the cement industry, and they have concurred that classification models would be practically helpful and thus increase the business opportunities for plant operators and directors. One of the coauthors of this paper, who presently serves as the Head of Performance Management, Technical Directorate in a cement company and has 26 years of experience working in cement plants, concurs with the implications. The prediction models are developed using the data obtained from soft sensors. If soft sensor can predict the free lime in real time, fuel to kiln can be adjusted accordingly avoiding under/over burning. This can save fuel costs and maintain consistent quality.

V. CONCLUSIONS

This study was aimed to predict free lime quality in cement clinker by leveraging data mining method and machine learning techniques. The findings show that predicting free lime quality as good or under-burn would be

more useful to the cement industry rather than predicting the absolute free lime value. Hyperparameter tuning improved the prediction accuracy.

The absence of the overburn dataset is the limitation of this study. Another limitation is the use of KNIME data science software where the authors are constrained to use available hyper parameter tuning options. For future research in this area, this study suggests researchers to include over burning in the prediction classes so that the dependent variable will have three classes namely over burning, under burning and good quality. Another recommendation could be for future studies to deploy their models in live environment to check the real-life effect of the models on clinker production.

ACKNOWLEDGEMENT

The authors would like to thank the cement plant company for the dataset.

REFERENCES

- [1]. A. M. Ali, J. D. Tabares, and M. W. McGinley, "A machine learning approach for clinker quality prediction and nonlinear model predictive control design for a rotary cement kiln," *J Adv Manuf Process*, pp. 1–30, 2022.
- [2]. F. Alemayehu and S. Omprakash, "Minimization of Variation in Clinker Quality," *Advances in Materials*, vol. 2, no. 2, p. 23, 2013, doi: 10.11648/j.am.20130202.12.
- [3]. Bishop, C. M. (2007). *Pattern Recognition and Machine Learning*. Springer.
- [4]. N. V. S. Avula, S. K. Veeram, S. Behera, and S. Balasubramanian, "Building robust machine learning models for small chemical science data: the case of shear viscosity of fluids," *Mach Learn Sci Technol*, vol. 3, no. 4, Dec. 2022, doi: 10.1088/2632-2153/ACAC01.
- [5]. X. Hao, Z. Zhang, Q. Xu, G. Huang, and K. Wang, "Prediction of f-CaO content in cement clinker: A novel prediction method based on LightGBM and Bayesian optimization," *Chemometrics and Intelligent Laboratory Systems*, vol. 220, p. 104461, Jan. 2022, doi: 10.1016/J.CHEMOLAB.2021.104461.
- [6]. International Energy Agency, "Cement technology roadmap plots path to cutting CO2 emissions 24% by 2050 - News - IEA," International Energy Agency, 2022. <https://www.iea.org/news/cement-technology-roadmap-plots-path-to-cutting-co2-emissions-24-by-2050> (accessed Feb. 06, 2023).
- [7]. T. Kautz, B. M. Eskofier, and C. F. Pasluosta, "Generic performance measure for multiclass-classifiers," *Pattern Recognit*, vol. 68, pp. 111–125, Aug. 2017, doi: 10.1016/J.PATCOG.2017.03.008.
- [8]. W. Li, D. Wang, and T. Chai, "Multisource data ensemble modeling for clinker free lime content estimate in rotary kiln sintering processes," *IEEE Trans Syst Man Cybern Syst*, vol. 45, no. 2, pp. 303–314, Feb. 2015, doi: 10.1109/TSMC.2014.2332305.
- [9]. C. Liu, E. Fakharizadi, T. Xu, and P. S. Yu, "Recent advances in domain-driven data mining," *Int J Data Sci Anal*, Jan. 2022, doi: 10.1007/S41060-022-00378-1.
- [10]. X. Liu, J. Jin, W. Wu, and F. Herz, "A novel support vector machine ensemble model for estimation of free lime content in cement clinkers," *ISA Trans*, vol. 99, pp. 479–487, Apr. 2020, doi: 10.1016/J.ISATRA.2019.09.003.
- [11]. N.-O. E. Moses and S. B. Alabi, "Predictive Model for Cement Clinker Quality Parameters," *Journal of Materials Science and Chemical Engineering*, vol. 04, no. 07, pp. 84–100, 2016, doi: 10.4236/MSCE.2016.47012.
- [12]. N. H. Mtarfi, Z. Rais, and M. Taleb, "Effect of clinker free lime and cement fineness on the cement physicochemical properties," 2017.
- [13]. A. K. Pani, V. K. Vadlamudi, and H. K. Mohanta, "Development and comparison of neural network based soft sensors for online estimation of cement clinker quality," *ISA Trans*, vol. 52, no. 1, pp. 19–29, 2013, doi: 10.1016/J.ISATRA.2012.07.004.
- [14]. Paz, I. T., Hernández-Gress, N., & Gonzalez-Mendoza, M. (2014, September). Classification of Database by Using Parallelization of Algorithms Third Generation in a GPU. *Communications in Computer and Information Science*, (459), 25-38.
- [15]. C. Schröer, F. Kruse, and J. M. Gómez, "A Systematic Literature Review on Applying CRISP-DM Process Model," *Procedia Comput Sci*, vol. 181, pp. 526–534, Jan. 2021, doi: 10.1016/J.PROCS.2021.01.199.
- [16]. Scikit-learn: Machine Learning in Python, Pedregosa et al., *JMLR* 12, pp. 2825–2830, 2011.
- [17]. C. Shao, S. Zheng, C. Gu, Y. Hu, and X. Qin, "A novel outlier detection method for monitoring data in dam engineering," *Expert Syst Appl*, vol. 193, p. 116476, May 2022, doi: 10.1016/J.ESWA.2021.116476.
- [18]. P. Sharma, R. Rao, N. Mohapatra, and A. Saxena, "Development of Suitable Machine Learning Model for a Cement Plant Calciner," *International Journal of Engineering Applied Sciences and Technology*, vol. 6, 2021, Accessed: Feb. 06, 2023. [Online]. Available: <http://www.ijeast.com>
- [19]. J. Yuan, L. Zhong, H. Du, and H. Tao, "Prediction of free lime content in cement clinker based on RBF neural network," *Journal Wuhan University of Technology, Materials Science Edition*, vol. 27, no. 1, pp. 187–190, Feb. 2012, doi: 10.1007/s11595-012-0433-3.
- [20]. H. Yu, R. Zhang, Q. Meng and S. Lu, "Research on soft sensing of cement clinker f-CaO based on LS_SVM and burning zone temperature," 2018 Chinese Control And Decision Conference (CCDC), Shenyang, China, 2018, pp. 6366–6370, doi: 10.1109/CCDC.2018.8408248.

Performance Analysis of TSP Training Instance Generation Methods in Policy-Based Hyper-Heuristic Optimization Framework

Orachun Udomkasemsud

Department of Computer Engineering
King Mongkut's University of
Technology Thonburi
Bangkok, Thailand
orachun.udo@kmutt.ac.th

Booncharoen Sirinaovakul

Department of Computer Engineering
King Mongkut's University of
Technology Thonburi
Bangkok, Thailand
booncharoen.sir@kmutt.ac.th

Tiranee Achalakul

Government Big Data Institute
Bangkok, Thailand
tiranee.ac@depa.or.th

Abstract— The Policy-Based Hyper-Heuristic Optimization framework (PHH), which utilizes reinforcement learning to solve combinatorial optimization problems, has demonstrated promising results in prior studies. However, training PHH models with real problem instances can be resource-intensive and challenging to obtain. To address this issue, this paper explores the effects of different randomization methods for generating training instances in the context of the traveling salesman problem (TSP). Seven randomization methods, namely grid, uniform, square, circle, Gaussian, wave, and mixed, were employed to generate TSP instances. The PHH models were trained using these instances, and their performance was evaluated on unseen TSP instances from TSPLIB. Experimental results indicate that training with TSP instances generated by the square method consistently achieved the lowest travel distances across small, medium, and large test instances. This research contributes to the understanding of instance-generation techniques for combinatorial optimization problems and provides insights for designing efficient hyper-heuristic frameworks.

Keywords—hyper-heuristics, reinforcement learning, combinatorial optimization

I. INTRODUCTION

Combinatorial optimization problems present significant challenges across various domains, necessitating efficient algorithms to find optimal or near-optimal solutions. Meta-heuristics offer flexibility and can be customized to address diverse problem types. Over the past few years, numerous meta-heuristic approaches have been introduced to tackle combinatorial optimization problems [1][2][3]. However, heuristics, designed for specific problems, may require parameter adjustments and can be time-consuming when solving large complex problems, often failing to yield the global optimum within a reasonable timeframe [4]. Furthermore, the performance of heuristic methods may vary across different problem instances.

The research community recently has shown increasing interest in hyper-heuristics [5] as a potential solution to these challenges. Among such frameworks is the PHH (Policy-Based Hyper-Heuristic with Reinforcement Learning) **Error! Reference source not found.**, which leverages reinforcement learning (RL) techniques to tackle these problems. Specifically, the PHH framework has shown promising results in solving the Traveling Salesman Problem (TSP) [7], a classic combinatorial optimization problem with numerous practical applications.

However, as problem instances grow in size and complexity, the effectiveness of PHH may diminish. To address this limitation, PHH was designed to handle larger, unseen problems by training on smaller instances. While this approach offers a potential solution, obtaining real-world problem instances for training can be resource-intensive and time-consuming. As an alternative, this paper focuses on exploring methods to generate randomized TSP instances and investigates their impact on the performance of PHH when applied to solving larger, unseen TSP problem instances.

Through the study of different TSP instance generation methods, we aim to provide insights into the viability of using randomized instances for training RL-based hyper-heuristic frameworks, such as PHH, to tackle larger combinatorial optimization problems.

The rest of this paper is structured as follows: Section II provides a literature review, discussing our previous work on the PHH framework. Section III outlines the methodology, describing the experimental setup and the methods for generating randomized TSP instances. Additionally, it presents the training process employed for PHH using these instances and the metrics used to evaluate its performance. In Section IV, we present the results and analyze the effects of different randomized TSP instance generation methods. Finally, Section V concludes the paper by summarizing the research objective and highlighting the key contributions.

II. BACKGROUND AND RELATED WORKS

A. PHH: Policy-Based Hyper-Heuristic with Reinforcement Learning

Hyper-heuristics [5] have been developed as a means to generalize across different problem domains and create optimization methodologies that can adapt to various combinatorial optimization problems. However, the performance of hyper-heuristics may degrade when applied to unseen problem instances due to the lack of problem-specific knowledge. To address this limitation, reinforcement learning techniques have been incorporated into hyper-heuristics to enhance their performance while maintaining generalization capabilities.

The PHH **Error! Reference source not found.** is a constructive hyper-heuristic framework that utilizes policy-based reinforcement learning. Constructive hyper-heuristics involve the use of constructive low-level heuristics to build solution components incrementally, leading to an optimal solution. The Proximal Policy Optimization (PPO) [8]

algorithm serves as the RL algorithm within the PHH framework. PPO operates by optimizing a policy function to maximize a defined objective function, typically formulated as the expected cumulative reward in RL tasks. The PPO algorithm employs a loss function that includes a clipping mechanism to ensure stable and effective policy updates. This allows the PHH models to learn and refine their decision-making policies based on the observed rewards and states during training. The loss function is shown in (1), where $A_{\pi_\theta}(s_t, a_t)$ represents the advantage value function for taking an action a_t at state s_t , $Q_{\pi_\theta}(s_t, a_t)$ denotes the cumulative reward function for taking action a_t at state s_t , $V_{\pi_\theta}(s_t)$ is a state value function for state s_t , π_θ is a policy parameterized by θ .

$$\begin{aligned} \nabla_\theta J(\theta) &= \mathbb{E} \left[\min \left(\frac{\text{ratio}_t(\theta) A_{\pi_\theta}(s_t, a_t)}{\text{ratio}_t^{\text{clip}}(\theta, \epsilon) A_{\pi_\theta}(s_t, a_t)} \right) \right] \quad (1) \\ \text{ratio}_t(\theta) &= \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} \\ \text{ratio}_t^{\text{clip}}(\theta, \epsilon) &= \text{clip}(\text{ratio}_t(\theta), 1 - \epsilon, 1 + \epsilon) \\ A_{\pi_\theta}(s_t, a_t) &= Q_{\pi_\theta}(s_t, a_t) - V_{\pi_\theta}(s_t) \end{aligned}$$

The PHH (policy-based hyper-heuristics) framework stands out due to its incorporation of several elements that contribute to its ability to generalize across different problem domains. The framework introduces generalized constructive low-level heuristics (LLHs) responsible for creating solution components gradually, leading to optimal solutions. This adaptability is achieved by constructing solutions in a flexible and problem-specific manner, enabling PHH models to tackle various combinatorial optimization problems effectively.

Furthermore, the PHH framework incorporates a generalized state function and a generalized reward function. The generalized state function captures relevant information about the current problem instance, allowing models to perceive and analyze problem characteristics and constraints. This information empowers models to make informed decisions during the optimization process. The generalized reward function evaluates the quality of solutions generated by the models, providing feedback on the effectiveness of their decision-making policies. By employing generalized state and reward functions, the PHH framework ensures models can adapt to different problem instances and optimize their performance accordingly.

The integration of these components—generalized constructive LLHs, a generalized state function, and a generalized reward function—within the PHH framework enables it to address a wide range of combinatorial optimization problems while maintaining a high level of generalization. This flexibility allows models to learn effective decision-making policies that can be applied to various problem domains, enhancing the framework's versatility and potential for solving real-world optimization challenges.

In terms of training methodology, the PHH framework adopts an on-policy training approach. This means that models are trained using the most recent data generated from the current policy, ensuring that the training instances are consistent with the current policy's decision-making behavior. This on-policy training method helps models adapt and

improve their performance over time as they learn from the most up-to-date experiences.

Additionally, the PHH framework leverages an experience replay technique during training. Experience replay involves storing and sampling past experiences from a replay buffer to enhance the learning process. By randomly sampling experiences from the replay buffer, PHH models can learn from a diverse range of past experiences, reducing the potential for overfitting specific training instances and promoting better generalization to unseen problem instances.

The performance of the PHH framework has been extensively evaluated on benchmark problem instances, including the traveling salesman problem (TSP), bin packing problem, and capacitated vehicle routing problem. In these evaluations, the PHH framework has consistently demonstrated superior solution quality compared to traditional metaheuristics and online-training DQN-based hyper-heuristics, achieving improvements of up to 98%.

Moreover, the practical applicability of the PHH framework has been demonstrated by its successful application to a real-world problem: deadline-constrained workflow scheduling in hybrid cloud environments.

The success of the PHH framework in solving both benchmark problem instances and real-world scenarios showcases its effectiveness as a hyper-heuristic framework with reinforcement learning integration. It demonstrates its potential to address a wide range of optimization challenges and highlights its practical utility in real-world applications.

B. Traveling Salesman Problem

The Traveling Salesman Problem (TSP) [7] is a classic combinatorial optimization problem that has garnered significant attention in the fields of operations research and computer science. The problem can be stated as follows: given a set of cities and their locations represented as 2-D coordinates, the objective is to find the shortest possible route that visits each city exactly once and returns to the starting city.

Mathematically, the TSP can be formalized as an optimization model as follows. Let n be the number of cities, and x_{ij} indicates whether the route travels from city i to city j ($x_{ij} = 1$) or not ($x_{ij} = 0$). The objective is to minimize the total distance traveled, as shown in (2), subject to the constraints that each city is visited exactly once, and the route starts and ends at the same city:

$$\text{Minimize } d_{\text{route}} = \sum_{i=1}^n \sum_{j=1}^n (d_{ij} x_{ij}) \quad (2)$$

subject to

$$\begin{aligned} \sum_{j=1}^n x_{ij} &= 1, \text{ for all } i, \text{ except the starting city} \\ \sum_{i=1}^n x_{ij} &= 1, \text{ for all } j, \text{ except the starting city} \\ x_{ij} &\in \{0, 1\}, \text{ for all } i, j \end{aligned}$$

where d_{ij} is the distance between city i and city j .

The TSP is known to be an NP-hard problem, meaning that finding an optimal solution for large instances becomes computationally challenging and time-consuming. Therefore, developing efficient algorithms, such as the Policy-Based Hyper-Heuristic Optimization framework, to tackle TSP

instances is crucial for practical applications and real-world problem-solving.

III. METHODOLOGY

This section presents the methodology used in this study to investigate the effects of using randomized TSP instances for training the PHH framework and its subsequent performance in solving larger, unseen problems.

A. Generation of Randomized TSP Instances

To generate randomized TSP instances for training the PHH framework, various randomization methods were employed to offer different spatial distributions of city locations. Specifically, a set of $n=100$ city locations denoted as C with x_i and y_i coordinates within the range of 0 to 100 were randomly generated. The following randomization methods were considered:

- a) **Grid:** This method aligns the randomly generated cities in a grid pattern, resulting in a structured and evenly spaced distribution of city locations. The (x_i, y_i) coordinates can be generated using (3) and (4).

$$x_i = ((i - 1) \bmod s) \cdot \text{floor}(100/s) \quad (3)$$

$$y_i = \text{floor}((i - 1)/s) \quad (4)$$

$$s = \text{ceil}(\sqrt{n})$$

- b) **Uniform:** The uniform method randomizes the city locations uniformly across the given range, resulting in a scattered distribution of cities across the TSP instance. The (x_i, y_i) coordinates can be generated using (5) and (6).

$$x_i = \text{uniform}(0, 100) \quad (5)$$

$$y_i = \text{uniform}(0, 100) \quad (6)$$

- c) **Square:** City locations in this method are aligned along the edges of a square, creating a square-shaped distribution of cities. The (x_i, y_i) coordinates can be generated using (7) and (8).

$$x_i = \begin{cases} \text{uniform}(0, 100) & ; i \in \{0, 1, \dots, s\} \\ & \cup \{2s + 1, 2s + 2, \dots, 3s\} \\ 0 & ; i \in \{s + 1, s + 2, \dots, 2s\} \\ 100 & ; i \in \{3s + 1, 3s + 2, \dots, n\} \end{cases} \quad (7)$$

$$y_i = \begin{cases} 0 & ; i \in \{0, 1, \dots, s\} \\ \text{uniform}(0, 100) & ; i \in \{s + 1, s + 2, \dots, 2s\} \\ & \cup \{3s + 1, 3s + 2, \dots, n\} \\ 100 & ; i \in \{2s + 1, 2s + 2, \dots, 3s\} \end{cases} \quad (8)$$

$$s = \text{ceil}(n/4)$$

- d) **Circle:** The circle method aligns the city locations along the edges of a circle, resulting in a circular distribution of cities. The (x_i, y_i) coordinates can be generated using (9) and (10).

$$x_i = 50 \cos(\text{uniform}(0, 2\pi)) \quad (9)$$

$$y_i = 50 \sin(\text{uniform}(0, 2\pi)) \quad (10)$$

- e) **Gaussian:** With the Gaussian method, cities are more densely concentrated in the center of the TSP instance, gradually decreasing in density towards the

outer regions. The (x_i, y_i) coordinates can be generated using (11) and (12).

$$x_i = \text{clip}(\sigma(50, 25), 0, 100) \quad (11)$$

$$y_i = \text{clip}(\sigma(50, 25), 0, 100) \quad (12)$$

- f) **Wave:** This method aligns the city locations along a sinusoidal wave, resulting in a curved distribution pattern. The (x_i, y_i) coordinates can be generated using (13) and (14).

$$x_i = \text{uniform}(0, 100/n) \quad (13)$$

$$y_i = 50 + 50 \cos(\text{uniform}(0, 0.2n)) \quad (14)$$

- g) **Mixed:** This method randomly selects locations from all locations generated by all methods.

Examples of TSP instances generated by each method can be illustrated in Fig. 1.

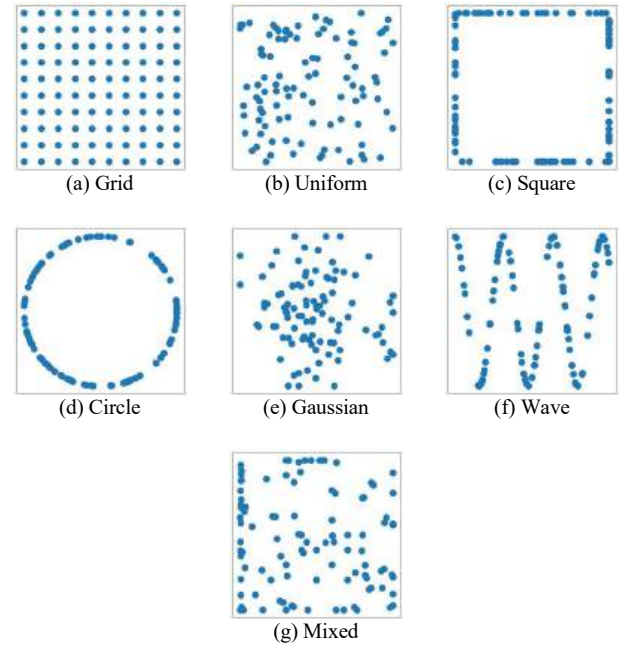


Fig. 1. Examples of TSP instances randomly generated by each method.

B. Experimental Setup

This subsection outlines the experimental process conducted to evaluate the effects of randomization methods for generating TSP training instances on the performance of the PHH framework on larger, unseen TSP instances.

The experiment starts with training the PHH models using different TSP instance randomization methods. Crucial hyper-parameters, including learning rates, exploration-exploitation trade-offs, and network architectures, were determined for the training process. These pre-defined hyper-parameters aimed to strike a balance between exploration and exploitation, ensuring effective training of the PHH models. The hyper-parameters for training the PHH models were selected based on empirical studies and are summarized in Table I.

During the training process, the training instances are re-generated at the beginning of each episode. This dynamic re-generation of training instances increases the diversity of the data, preventing the PHH models from overfitting to specific training instances. Additionally, to assess the models' performance and generalization capabilities, an evaluation

step is performed every 102,400 steps. During this evaluation step, the PHH models are tested on a randomly generated TSP instance using the same randomization method as in the training phase.

TABLE I. PARAMETER SETTINGS FOR PHH

Parameters	Values
Training frequency	Every 10,240 steps
Number of epochs	5
Learning rate	0.01x; x = training progress remaining from 1 to 0
Batch size	2048
Clip range	0.2
Discount factor	0.99
Policy network	3 dense layers with ReLU: 128x64x32
Value network	2 dense layers with ReLU: 64x32
Training timesteps	2E+7 steps
Evaluate frequency	Every 102,400 steps
Early stopping	Stop if not improved for 100 evaluations

The training process continues until a stopping criterion is met. If the evaluation performance does not improve for 100 consecutive evaluations, the training process is halted. Otherwise, the training continues until it reaches a maximum of 2E+7 steps.

Once the PHH models are trained, their effectiveness is evaluated on larger, unseen TSP instances sourced from the well-known TSPLIB benchmark [9]. These larger TSP instances are known to pose more complex and challenging optimization problems, allowing for a thorough assessment of the PHH framework's performance in solving real-world scenarios. The TSP benchmarking instances from TSPLIB used in this experiment are summarized in Table II.

TABLE II. SUMMARY OF TSP INSTANCES FROM TSPLIB USED IN THE EVALUATION

Size	Instance	Cities
small	ch150	150
small	d198	198
small	pr144	144
small	rat195	195
medium	p654	654
medium	rat575	575
medium	d657	657
medium	rd400	400
large	pr2392	2,392
large	fnl4461	4,461
large	pcb3038	3,038
large	rl5915	5,915

To evaluate the performance of the PHH models, we recorded the traveled distances obtained when the models solved the larger TSP instances. The traveled distances served as objective metrics for evaluating the solution quality achieved by the PHH framework. By analyzing the recorded traveled distances, we could measure the effectiveness of the PHH models in addressing the optimization challenges posed by the larger TSP instances.

The PHH framework was implemented using Python 3.10 and the Stable-Baselines3 library [10], which facilitated the application of reinforcement learning techniques to optimize the hyper-heuristic models. Stable-Baselines3 is a popular

library that provides powerful tools for implementing reinforcement learning algorithms, enabling efficient training and evaluation of the PHH models.

This experimental setup and training process allowed us to investigate the impact of the PHH framework, trained on smaller randomized TSP instances, on solving larger TSP instances from the TSPLIB benchmark. By evaluating the PHH models on these larger TSP instances, we could gain valuable insights into the framework's performance in handling scalability, complexity, and real-world combinatorial optimization challenges.

IV. RESULTS AND ANALYSIS

In this section, we present the results of our experiments and provide an in-depth analysis of the performance of the PHH framework when trained on randomized TSP instances and applied to larger, unseen TSP problems. The analysis aims to gain insights into the impact of different randomized instance generation methods on the performance of the PHH framework.

Table III shows the average route distances obtained from the PHH models when applied to small, medium, and large test TSP instances. The distances are recorded for models trained with TSP instances generated by different methods. These distances serve as objective metrics for evaluating the solution quality achieved by the PHH framework.

To assess the significance of the differences in performance among the different generation methods, statistical tests were conducted. Friedman's test was employed to test whether there are significant differences among the generation methods. The P-value from Friedman's test is 8.50E-09, indicating that there are indeed significant differences among the methods.

Nemenyi's test was used to further test the significance of each pair of generation methods. Table IV shows the pairwise P-values. The test reveals two pairs of methods with insignificant differences: wave vs. mixed and uniform vs. Gaussian. For all other pairs of methods, the P-values are less than 0.001, indicating significant differences in performance.

The experiment results indicate that training the PHH models with TSP instances generated using the square method consistently resulted in the lowest traveled distances across all small, medium, and large test instances. This suggests that the square method exhibited superior performance in terms of producing more optimal solutions for the PHH models.

The average ranks of distances obtained from PHH models trained with different generation methods are shown in Table V. The table presents the rank orders of the methods based on their performance on large, medium, and small test instances.

When examining the performance of the different randomization methods on large test instances, the average ranks of the traveled distances followed a specific order. The square method achieved the highest rank, indicating the lowest traveled distances, followed by the grid, circle, mixed, uniform, and wave methods. This ordering suggests that the square method was particularly effective in generating training instances that facilitated better performance when confronted with the challenges posed by the larger TSP instances.

Similar trends were observed when considering the medium test instances, where the square method again attained the highest average rank. Following the square method, the grid, uniform, circle, mixed, and wave methods ranked consecutively. These findings highlight the consistent advantages of the square method across different problem sizes, reinforcing its efficacy in generating training instances that promote improved performance for the PHH models.

For the small test instances, the average ranks of the traveled distances followed a distinct order. Once again, the square method secured the highest rank, suggesting its effectiveness in producing more favorable solutions. The grid and uniform methods followed in the rank order, followed by the Gaussian, wave, circle, and mixed methods. These results indicate that while the square method yielded the best outcomes, the grid and uniform methods also demonstrated competitive performance on the small test instances.

The superior performance achieved by training the PHH models with TSP instances generated by the square method can be attributed to the specific characteristics of this randomization approach. The square method aligns city locations along the edges of a square, resulting in a structured and evenly distributed arrangement of cities. This spatial distribution may offer inherent advantages in terms of facilitating better exploration and exploitation of the search space during training. The structured nature of the square method allows the PHH models to capture and generalize patterns more effectively, enabling them to learn optimal strategies for solving TSP instances. This increased ability to learn from the structured training instances translates into improved performance when faced with unseen test instances.

On the other hand, the circle method, while introducing diversity and deviation from the regular grid structure, may pose challenges for the PHH models in effectively exploring and exploiting the search space. The circular distribution lacks the clear, well-defined patterns and structure provided by the square method, making it harder for the models to identify optimal solution strategies.

The uniform and Gaussian methods did not demonstrate significant differences in performance. The uniform method's lack of structured patterns limits the models' ability to identify regularities or exploit specific solution strategies, hindering their capacity to find optimal solutions. Similarly, the

Gaussian method's concentration of city locations in the center creates a skewed spatial distribution that may restrict exploration and hinder the models' ability to generalize effective solution strategies.

The grid method's highly structured arrangement may limit the models' ability to explore alternative paths and search for optimal solutions, potentially leading to suboptimal performance. In contrast, the square method's structured arrangement strikes a better balance between regularity and diversity, enabling the PHH models to learn more robust and effective strategies for solving TSP instances.

The wave method's irregular arrangement may pose challenges for the PHH models in effectively exploring and exploiting the search space. The lack of clear patterns and structure in the wave distribution hinders the models' ability to identify optimal solution strategies and may lead to suboptimal performance.

The mixed method, while introducing diversity, may introduce additional complexities and challenges during the training process. The mixed method's less structured nature may make it harder for the PHH models to identify and exploit regular patterns or optimal solution strategies. As a result, the models may struggle to generalize effectively and achieve optimal performance when confronted with unseen test instances. The mixed method's higher distance highlights the potential drawbacks of utilizing more varied and irregular randomization methods, emphasizing the importance of striking a balance between diversity and structured patterns when generating training instances for combinatorial optimization problems. Further exploration and refinement of randomization methods may be necessary to enhance the performance of the PHH framework when using mixed instances.

Overall, the analysis of the results underscores the significance of the instance randomization method employed during training in determining the performance of the PHH in solving TSP. The square method consistently emerged as the most effective, generating training instances that facilitated superior solution quality across various problem sizes. The findings suggest that the spatial distribution of city locations influenced the performance of the PHH models, with the square method providing more favorable arrangements for tackling the complexities of the TSP instances.

TABLE III. AVERAGE ROUTE DISTANCE OF SOLUTIONS OBTAINED FROM THE PHH FRAMEWORK TRAINED ON TSP INSTANCES FROM EACH GENERATION METHOD. THE LOWEST AVERAGE DISTANCES AMONG ALL METHODS ARE IN BOLD.

Size	Instance	Training Instance Generation Method						
		Grid	Uniform	Square	Circle	Gaussian	Wave	Mixed
Small	ch150	1.98E+04	1.93E+04	1.58E+04	1.94E+04	2.25E+04	2.15E+04	2.09E+04
	d198	5.46E+04	5.18E+04	3.67E+04	4.96E+04	5.14E+04	5.59E+04	5.66E+04
	pr144	2.54E+05	2.51E+05	2.00E+05	2.90E+05	2.69E+05	2.91E+05	2.88E+05
	rat195	8.32E+03	7.44E+03	5.95E+03	7.94E+03	9.97E+03	7.59E+03	8.43E+03
Medium	p654	1.30E+05	1.36E+05	8.94E+04	1.19E+05	2.09E+05	1.40E+05	1.65E+05
	rat575	2.47E+04	2.23E+04	1.75E+04	2.45E+04	3.58E+04	2.49E+04	2.55E+04
	d657	1.71E+05	1.54E+05	1.23E+05	1.84E+05	2.09E+05	1.67E+05	1.74E+05
	rd400	4.68E+04	4.70E+04	3.87E+04	5.61E+04	6.18E+04	5.67E+04	5.44E+04
Large	pr2392	1.76E+06	1.66E+06	1.13E+06	1.83E+06	3.20E+06	1.49E+06	1.82E+06
	fnl4461	7.87E+05	6.04E+05	4.52E+05	7.71E+05	1.42E+06	6.30E+05	8.14E+05
	pcb3038	5.57E+05	4.41E+05	3.60E+05	5.86E+05	8.37E+05	4.93E+05	5.35E+05
	rl5915	3.44E+06	2.53E+06	1.92E+06	3.55E+06	5.76E+06	2.79E+06	3.42E+06

TABLE IV. PAIRWISE P-VALUES CALCULATED FROM THE EXPERIMENT RESULTS INDICATE WHETHER THE DIFFERENCE IN DISTANCES FROM EACH PAIR OF TEST INSTANCE GENERATION METHODS IS STATISTICALLY SIGNIFICANT AMONG ALL TEST INSTANCES. SIGNIFICANT DIFFERENCES ARE HIGHLIGHTED IN GREEN.

	Grid	Mixed	Square	Wave	Uniform	Gaussian	Circle
Grid	1.000	0.001	0.001	0.001	0.001	0.001	0.001
Mixed	0.001	1.000	0.001	0.191	0.001	0.001	0.001
Square	0.001	0.001	1.000	0.001	0.001	0.001	0.001
Wave	0.001	0.191	0.001	1.000	0.001	0.001	0.001
Uniform	0.001	0.001	0.001	0.001	1.000	0.900	0.004
Gaussian	0.001	0.001	0.001	0.001	0.900	1.000	0.001
Circle	0.001	0.001	0.001	0.001	0.004	0.001	1.000

TABLE V. AVERAGE PERFORMANCE RANK OF TEST INSTANCE GENERATION METHODS. THE HIGHEST RANKS ARE IN BOLD.

Size	Gaussian	Grid	Square	Uniform	Wave	Circle	Mixed
Small	4.25	2.50	1.00	3.75	5.25	5.50	5.75
Medium	3.25	2.75	1.00	4.00	7.00	4.75	5.25
Large	4.75	2.25	1.00	5.50	7.00	2.75	4.75

V. CONCLUSION

In this paper, we investigated the effects of different randomization methods for generating TSP training instances on the performance of the PHH framework when applied to larger, unseen TSP instances. Our experimental results shed light on the impact of instance generation on the PHH framework's ability to tackle optimization challenges in the TSP domain.

Through rigorous experimentation and analysis, we found that training the PHH models with small TSP instances generated by the square method consistently yielded the lowest traveled distances across small, medium, and large test instances. The square method's structured spatial distribution of city locations proved advantageous in promoting effective exploration and exploitation during training, leading to improved solution quality.

Our findings emphasize the importance of the instance generation process in the training of reinforcement learning-based hyper-heuristic frameworks like PHH. The spatial arrangement of city locations significantly impacts the performance and generalization capabilities of the models. The structured nature of instance generation methods, such as the square method, can offer advantages in terms of facilitating better exploration and exploitation of the search space.

This research has practical implications for designing efficient reinforcement learning-based hyper-heuristics for combinatorial optimization problems. The insights gained from our study can inform the development of improved instance generation methods that strike a balance between diversity and structured patterns, enabling enhanced generalization and solution quality.

Future research directions may involve exploring alternative randomization methods or investigating hybrid approaches that combine the strengths of different instance generation techniques. Furthermore, extending this study to other combinatorial optimization problems and evaluating the performance of PHH in real-world instances can provide further insights into the framework's effectiveness and potential applications.

REFERENCES

- [1] S. Muthuraman and V. Prasanna Venkatesan, "A Comprehensive Study on Hybrid Meta-Heuristic Approaches Used for Solving Combinatorial Optimization Problems," Proc. 2nd World Congress on Computing and Communication Technologies (WCCCT 2017), pp. 185-190, 2017.
- [2] A. Zhou, B. Y. Qu, H. Li, S. Z. Zhao, P. N. Suganthan, and Q. Zhangd, "Multiobjective evolutionary algorithms: A survey of the state of the art," Swarm and Evol. Comput., vol. 1, no. 1, pp. 32-49, 2011.
- [3] S. Voß, "Meta-heuristics: The state of the art," Lect. Notes Comput. Sci., vol. 2148, pp. 1-23, 2001.
- [4] E. G. Talbi, Metaheuristics: From design to implementation, John Wiley & Sons, 2009.
- [5] P. Ross, "Hyper-heuristics," in Search methodologies, Springer, pp. 611-638, 2014.
- [6] O. Udomkasemsub, B. Sirinaovakul and T. Achalakul, "PHH: Policy-Based Hyper-Heuristic With Reinforcement Learning," in IEEE Access, vol. 11, pp. 52026-52049, 2023.
- [7] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, "The traveling salesman problem: A guided tour of combinatorial optimization," John Wiley & Sons, 1985.
- [8] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," <http://arxiv.org/abs/1707.06347>, 2017.
- [9] G. Reinelt, "TSPLIB—A Traveling Salesman Problem Library," ORSA Journal on Computing, vol. 3, pp. 376-384, 1991.
- [10] R. Antonin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-Baselines3: Reliable Reinforcement Learning Implementations," J. Mach. Learn. Res., vol. 22, pp. 1–8, 2021.

Student Individual Performance in Online Learning: Technology Adoption and Culture

1st Oda I. B. Hariyanto

Tourism Department
Universitas International Batam
Batam, Indonesia 29426
oda@uib.ac.id

2nd Doni Purnama Alamsyah

Entrepreneurship Department
BINUS Business School
Bina Nusantara University
Jakarta, Indonesia 11480
doni.syah@binus.ac.id

Abstract—Technology adoption and e-learning user culture are interesting to study because they provide insights into academic performance. However, the continued use of e-learning is needed so that academic performance is achieved. In line with the importance of technology adoption, the purpose of this study is to examine the impact of continuance intention on student performance with the support of the quality, usefulness, and effectiveness of e-learning. The study was conducted by analyzing online user behavior from students, where 491 quantitative questionnaire data were collected to test the research model. The research method is carried out through a survey of experienced e-learning students. The analysis was carried out using the SEM and IPA approaches with the SmartPLS analysis tool. The research hypothesis was tested based on the model and it was found that continuance intention can be influenced by the quality, usability, and effectiveness of e-learning. It is further known that student performance increases when it is supported by continuance intention. The importance of continuance intention as part of the results of technology adoption on the concept of e-learning. A culture of independent learning in e-learning will be formed if continuance intention is achieved by students. Information in this research becomes an input for e-learning providers, especially in evaluating e-learning performance.

Keywords—Quality, Usefulness, Effectiveness, Continuance Intention, Student Performance.

I. INTRODUCTION

Technology adoption is currently a trend study, especially in online user behavior [1], [2]. The impact of a good adaptation of the online platform gives good performance to users, the concept of e-learning occurs in student academic performance [3]. Technology adoption is inseparable from culture, because adapting easily to online platforms can already become a habit [3]. Previous studies explained that students who are familiar with technology in their daily life have no problems learning online, because of the ease of adaptation [4]. Culture in using technology cannot be achieved easily, depending on the user, in this case the user's interest in adapting to technology [5], [6]. E-learning exists as a learning medium at the university level, this learning method exists because of the needs of students and as a result of technological developments [7]. The existence of e-learning is a must for students to adapt, but good habits are needed, especially for students who are not familiar with information technology. The purpose of online learning is to support student learning performance, where student performance is the final achievement of learning [8]. Growing user interest in e-learning can depend on user culture or habits [9], but good adaptation to technology is known to depend on the characteristics of e-learning that users value [10].

Online learning is seen as having many advantages, including being able to be carried out by users at different

times and places [11]. Moreover, the quality of e-learning is seen as having a greater role in the interest in the use of sustainable online learning [9], [12]. The quality of the online platform is related to the content, design, systems, IT support, and teaching support [13], [14]. In addition to the quality that is seen by users when adapting to online learning is usefulness, because it is considered to support students in learning, e-learning can be considered useful [15]. The value that is evaluated is more on how to use the online platform, ease of use is an important concern [16]. Ease of interacting with online platforms supports the use of sustainability in the future [17]. Each student has many subjects that need to be completed to determine graduation, meaning that many assignments need to be done. Online platforms are seen as the right media if they support many student activities [18]. So effectiveness in adapting to e-learning provides support for the use of online platforms [19]. Analysis of adaptation to e-learning reveals several views that are known to have an impact on the use of online platforms, including quality [13], usefulness [15], and effectiveness [3], [19]. A good adaptation is believed to become a student culture in using e-learning.

Based on the initial analysis, it is known the importance of a culture of adaptation in online learning, to support the use of sustainable e-learning. It is known that the continuance intention of students on online platforms can improve student performance, especially in academic learning [20]. The analysis of this study explains the position of the current research which focuses on examining the relationship between continuance intention and student performance through the quality, usefulness, and effectiveness of e-learning as assessed by students. The research position describes a good model for supporting student performance in online learning.

A. The Quality, Usefulness, and Effectiveness of Online Platform

The term online platform is widely used to describe technology adaptation by users [21]. This relates to e-learning that is used at the university level. E-learning itself is defined as an online platform in the form of a website or mobile application dedicated to student learning media that can be done anywhere [4], [18]. E-learning is evaluated by students in terms of quality because it has an impact on student learning comfort [13], [22]. The quality of the e-learning in question relates to all the good ratings of e-learning assessed by students compared to other platforms. E-learning quality is considered to support student performance if it is useful to use [23]. Evaluation of students as users on online platforms is related to content, design, management, IT support, and instructors [14].

E-learning is known as a learning medium that provides many conveniences, so the usefulness of online platforms

becomes student evaluation [15], [24]. Usefulness relates to students' assumptions that technology can be easily adapted without effort and supports user activities [25]. Evaluation of the ease of online learning is known from several views including assist, easier, and improve [16]. Effective learning media is the goal for students to adapt and use information technology [4]. Effectiveness relates to the notion that the use of e-learning is useful and supports users in achieving what they expect [19]. It is said to be effective if it is assessed from several sides, namely support for performance, increased motivation, and ease of completing tasks [4], [19].

B. Continuance Intention of User

The key to success in adapting information technology is increasing continuance intention [9]. An example of continuance intention is the impact on the sustainability of online learning implementation [17]. That's why many studies of the concept of online learning aim to examine continuance intention. In theory, it is stated that continuance intention is a desire and need for online platform users to continue to use it as a learning medium [26]. The intention of students as e-learning users develops well when it begins with a good adaptation of information technology [10], [14]. The culture of using information technology as part of activities will make it easier when interacting with online learning. This means that continuance intention requires not only experience in using but user habits in interacting with information technology.

E-learning is seen as a medium that facilitates student learning, but not for some students who are not familiar with online platforms [27], [28]. This means that many views are assessed by students in using e-learning, and the benefits obtained by users must exceed their expectations [29]. Continuance intention will develop well if all expectations from users when adapting go well and support student performance [30]. In line with these expectations, several assessments were made that were considered by students to be related to continuance intention, including interest in using, intensity in using, and plans to continue using online learning [18].

C. Student Performance

The purpose of learning both online and offline is to achieve learning performance, this performance is related to student academics [8]. Performance is a measure of a person's achievement in their activities, related to students, the performance in question is student achievements in learning better and in line with expectations at the beginning [31]. The learning in question is related to e-learning, which is chosen by students, of course, with the hope of being able to support student learning performance. Some of the assumptions from students regarding e-learning as a learning medium include those related to easy and fun learning, because it can be done according to the expectations of the users [20], [32]. The main goal of adaptation to e-learning is to achieve student performance so that learning outcomes occur [33]. Some of the evaluations that assessed student performance include those relating to increased performance, increased productivity, increased learning effectiveness, and in general an increase in academic performance [32].

D. Research Model and Hypothesis

This study describes and tests the research model based on the research hypothesis. Previous studies explained that student learning performance can be influenced by student

interest in learning [33]. Especially in the concept of e-learning that requires good adaptability so that it supports student academics [34], [35]. This explains that there is a good relationship between continuance intention in e-learning and student performance achievements. Other studies explain the continuation of the existence of continuance intention, which can be influenced by user judgments related to quality [13], usefulness [15], and effectiveness [3], [19]. The quality of e-learning provides opportunities for good adaptation of online learning concepts [13]. It is continued that the usefulness of e-learning is the main factor for the use of online learning [15]. It was also conveyed that the user's interest in using e-learning depends on the effectiveness provided by e-learning [19]. In line with previous studies explaining that there is support for the quality, usefulness, and effectiveness of online learning on student achievement continuance intentions. As well as support for continuance intention on student performance in learning. Based on the analysis of previous research studies, the research model and supporting hypotheses are presented (Fig. 1).

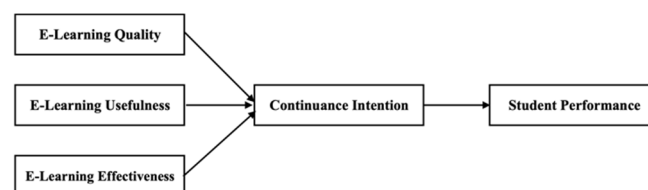


Fig. 1. Hypothesis Development.

- | | |
|--------------|--|
| Hypothesis 1 | Continuance intention can be well influenced through e-learning quality. |
| Hypothesis 2 | Continuance intention can be well influenced through e-learning usefulness. |
| Hypothesis 3 | Continuance intention can be well influenced through e-learning effectiveness. |
| Hypothesis 4 | Continuance intention can affect the improvement of student performance. |

II. RESEARCH DESIGN

A. Quantitative Research

In line with the research model, this study uses a qualitative data type with a research questionnaire. The method used is a survey of university-level online learning users. The questionnaire is made in an online form where the questions are based on the size of each variable. Each answer from the questionnaire has been determined by the Likert Scale approach, where a score of 1 to 5 determines the rating of the respondent's approval. The selected respondents are students who have experience with e-learning, where these students are in the city of Bandung one of the cities that has many universities in Indonesia.

B. Measurement

The size of this study is based on research variables, where each variable has a different number of sizes to be analyzed. Student performance is measured by four indicators, and continuance intention is measured by three indicators. Furthermore, e-learning quality is assessed by five indicators, and e-learning usefulness and effectiveness are measured by the same indicator, namely three indicators. All data from the size were submitted in the questionnaire, and the questionnaire data were analyzed with two tests, namely the PLS Algorithm and Bootstrapping [36]. The statistical analysis was carried

out on SmartPLS. To find out the gap in the mediation of continuance intention, an Importance Performance Map (IPA) test was carried out [37].

The analysis of the research was carried out based on the research model (Fig. 1) and the research hypothesis. However, previously carried out model testing to determine the Fit Model and research instruments.

III. RESEARCH FINDINGS AND EVALUATION

The results of the online questionnaire distribution obtained 491 tabulated data, where all data is valid because all questionnaire answers are filled. Then the data is processed through the first statistical test, namely the PLS Algorithm (Fig. 2). The test results show that all the relationships between variables have a positive value, which means they have a good relationship. The first statistical results are evaluated through the Fit Model (Table I), where it is known that the SRMR and NFI values are supported because they fall within the specified criteria. Followed by research instrument tests with the results shown in Table II. From the measurement of Cronbach's Alpha, Composite Reliability, and AVE it appears that all are accepted because all values are above 0.700 [36]. Based on the test results of the PLS Algorithm, the model (Fig. 2) can be accepted and analyzed further.

The second statistical result was carried out through the Bootstrapping Process test, where the results appear in Table III. It is known that all research constructs have a p-value below 0.000, which means that all measures are accepted. The results of the next bootstrapping are presented through the results of the research hypothesis test (Table IV) and are analyzed for further discussion.

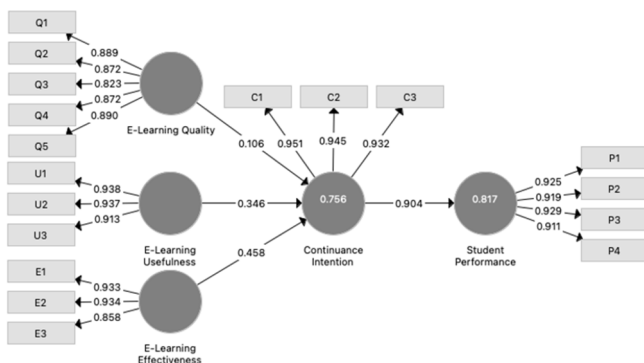


Fig. 2. PLS Algorithm Result.

TABLE I. FIT MODEL RESULT

Measurements	Saturated Model	Result
Standardized Root Mean Squared Residual (SRMR)	0,035	Support (<0,080)
Normed fit index (NFI)	0,912	Support (>0,900)

TABLE II. CONTRACTS RELIABILITY AND VALIDITY OF MODEL

Constructs	Cronbach's Alpha	Composite Reliability	Average Variance Extracted (AVE)
Continuance Intention	0,937	0,960	0,888
E-Learning Effectiveness	0,894	0,935	0,827
E-Learning Quality	0,919	0,939	0,756
E-Learning Usefulness	0,921	0,950	0,864
Student Performance	0,941	0,957	0,849

TABLE III. BOOTSTRAPPING RESULT

Constructs	T Statistics (O/STDEV)	P Values
C1 <- Continuance Intention	134,196	0,000
C2 <- Continuance Intention	110,543	0,000
C3 <- Continuance Intention	101,604	0,000
E1 <- E-Learning Effectiveness	130,504	0,000
E2 <- E-Learning Effectiveness	134,594	0,000
E3 <- E-Learning Effectiveness	44,143	0,000
P1 <- Student Performance	93,817	0,000
P2 <- Student Performance	94,180	0,000
P3 <- Student Performance	98,062	0,000
P4 <- Student Performance	83,303	0,000
Q1 <- E-Learning Quality	60,829	0,000
Q2 <- E-Learning Quality	55,241	0,000
Q3 <- E-Learning Quality	40,005	0,000
Q4 <- E-Learning Quality	64,546	0,000
Q5 <- E-Learning Quality	69,679	0,000
U1 <- E-Learning Usefulness	102,844	0,000
U2 <- E-Learning Usefulness	107,114	0,000
U3 <- E-Learning Usefulness	79,458	0,000

TABLE IV. P-VALUES OF HYPOTHESIS

Hypothesis		T Statistics	P-Values
E-Learning Quality -> Continuance Intention	H1	2,213	0,027
E-Learning Usefulness -> Continuance Intention	H2	4,486	0,000
E-Learning Effectiveness -> Continuance Intention	H3	6,443	0,000
Continuance Intention -> Student Performance	H4	72,043	0,000

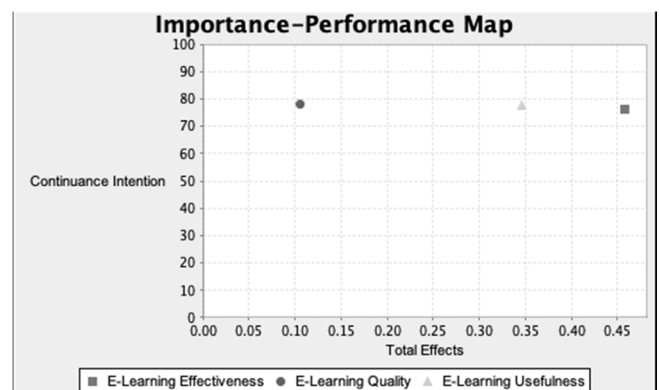


Fig. 3. Importance of Performance Map.

A. The Antecedent of Continuance Intention

The results of the test have been carried out and then a discussion is presented in line with the research hypothetical design (Fig. 1). In the first hypothesis (H1) it is stated that there is an impact of e-learning quality on the continuance intention of students who learn through online learning. The test results are known in Fig. 2 and the positive correlation value is 0.106. The test results show that there is a relationship but not too close, meaning that it has no significant impact. It is different from the second hypothesis (H2) which conveys the impact of e-learning usefulness on student continuance intention. The correlation value was obtained at 0.346 and it can be explained that there is a positive impact from e-learning usefulness in supporting student learning behavior through online learning. The findings from this study are in the third hypothesis (H3), based on the test results it is known that the correlation results are greater than the tests of other variables,

which is obtained at 0.458. This means that related to continuance intention, it is more influenced by e-learning effectiveness. Students assume that more effective online learning media will support learning performance. Effectiveness is studied from several sides including e-learning support on student performance, increasing student motivation due to e-learning, and the ease of completing assignments through e-learning. It is known through this research that the position of e-learning is assessed by students. The antecedents that have the most impact on influencing interest in using online learning are the first based on effectiveness, the second based on usefulness, and the third based on quality.

The results of this test are clarified again based on the hypothesis testing that has been summarized previously (Table IV). H1, H2, and H3 are accepted considering that the p-value is below 0.050. In line with previous research, the first is that effectiveness is a matter to be considered related to adaptation to online learning [3], [19]. On the other hand, it was also conveyed that the usefulness and quality assessed by students when interacting with online learning became a consideration of loyalty and further interest [13], [15]. The findings in this study are beneficial for universities as e-learning providers, where studies on user behavior are needed to maintain the user's ongoing interest in e-learning.

B. The Role of Continuance Intention

The main objective of this study is to explain the achievements of student performance. The analysis is presented based on the fourth hypothesis (H4), which is presented based on the test results (Fig. 2) that continuance intention has a positive impact. It can be seen from the correlation value of 0.904 which explains the very close relationship between variables. Studies on the relationship between user interest in online learning and academic performance have indeed been explained and mutually support each other. In line with this study, it seems consistent that the results of the research support previous research. The concept of online learning is important because of efforts to adapt to information technology, but cannot be separated from the culture formed by students [34], [35]. This culture is of course related to direct interaction in the e-learning concept. The culture that is formed results in the growth of continuance intention, further impacting student learning performance. So that the goal of having a choice of online learning models is achieved, namely supporting student academics. The results of the fourth hypothesis are clarified by evaluating the p-value (Table IV), where the value supports.

The results of testing all research hypotheses explain that a model can be formed from the role of continuance intention in student academic support. Considering continuance as a mediating variable that can be supported by effectiveness, usefulness, and quality, as well as the ability to support student performance. Explain the important position of continuance intention as behavior in online users. However, there is a problem gap in this study, where the true quality that has been formed in e-learning is not able to explain more about user interests. So the importance-performance map test was carried out (Fig. 3), where continuance intention was the goal of the test. The results of the IPA test show that the e-learning quality posts are in the top left matrix position (concentrate position) which explains that the existence of e-learning based on quality is highly expected by users, but users consider that the current quality delivered is still far from expectations. It seems

clear that there is a problem gap, namely the implementation of e-learning quality which is considered important by students is still not well received. This finding is an evaluation based on a sample of some students in Bandung so that the university can determine the direction of e-learning quality in line with student expectations.

IV. CONCLUSIONS

This research examines the online user behavior of students as e-learning users, where the research analysis focuses on the student performance model. Adaptation of important information technology is e-learning, where there is a user culture that is online learning. The results of the study found that continuance intention can be influenced by the quality, usefulness, and effectiveness of e-learning. Another finding is that continuance intention has a positive influence on student performance. In the performance of the importance map, it is explained that there is a problem gap in e-learning quality, meaning that this needs to be a university study in its implementation. These findings explain the importance of increasing continuance intention in the e-learning concept, with the aim of student academic performance. Continuance intention is related to user culture due to technology adoption in the e-learning concept. The information described in the research can be used as input for organizers of online learning concepts such as universities, to evaluate the implementation of e-learning. As well as this research has the opportunity to continue by examining other factors that explain continuance intention because the culture of online learning users has many factors including study habits and learning abilities. Because this is not a study in this study and is a limitation of this study.

REFERENCES

- [1] S. Nikou and I. Maslov, "Finnish university students' satisfaction with e-learning outcomes during the COVID-19 pandemic," *International Journal of Educational Management*, vol. 37, no. 1, 2023, doi: 10.1108/IJEM-04-2022-0166.
- [2] I. Y. Alyoussef, "Acceptance of e-learning in higher education: The role of task-technology fit with the information systems success model," *Heliyon*, vol. 9, no. 3, 2023, doi: 10.1016/j.heliyon.2023.e13751.
- [3] M. A. Alyahya, I. A. Elshaer, F. Abunasser, O. H. Mahmoud Hassan, and A. E. E. Sobaih, "E-Learning Experience in Higher Education amid COVID-19: Does Gender Really Matter in A Gender-Segregated Culture?," *Sustainability (Switzerland)*, vol. 14, no. 6, 2022, doi: 10.3390/su14063298.
- [4] A. Nuryatin, H. Mukhibad, and T. Tusyanah, "Effectiveness of Online Learning at Universities: Do Sociocultural Differences Matter?," *European Journal of Educational Research*, vol. 11, no. 4, 2022, doi: 10.12973/eu-jer.11.4.2153.
- [5] A. A. Alfalah, "Factors influencing students' adoption and use of mobile learning management systems (m-LMSs): A quantitative study of Saudi Arabia," *International Journal of Information Management Data Insights*, vol. 3, no. 1, 2023, doi: 10.1016/j.jjimei.2022.100143.
- [6] C.-Y. Lin and Y.-H. Ho, "Determinants of green practice adoption for logistics companies in China," *Journal of business ethics*, vol. 98, pp. 67–83, 2011.
- [7] E. Fernando, "Factors influence the success of e-learning systems for distance learning at the University," in *2020 International Conference on Information Management and Technology (ICIMTech)*, IEEE, 2020, pp. 294–299.
- [8] M. Kokoç, "Flexibility in e-Learning: Modelling Its Relation to Behavioural Engagement and Academic Performance.," *Themes in eLearning*, vol. 12, no. 12, pp. 1–16, 2019.
- [9] K.-M. Lin, "e-Learning continuance intention: Moderating effects of user e-learning experience," *Comput Educ*, vol. 56, no. 2, pp. 515–526, 2011.
- [10] S. M. Utomo, D. Purnama Alamsyah, and O. I. B. Hariyanto, "Continuance Intention of E-Learning: New Model of Technology Adoption," in *2022 3rd International Conference on Big Data*

Analytics and Practices (IBDAP), Sep. 2022, pp. 85–89. doi: 10.1109/IBDAP55587.2022.9907354.

- [11] H. M. Hsibollah and K. M. Idris, "E-learning adoption: The role of relative advantages, trialability and academic specialisation," *Campus-Wide Information Systems*, vol. 26, no. 1, pp. 54–70, 2009, doi: 10.1108/10650740910921564.
- [12] A. Rahmania, K. Jesslyn, A. Gui, Suryanto, Nelly, and S. A. Sari, "Understanding Higher Education Students Continuance Intention Towards e-Learning," in *2022 2nd International Conference on Information Technology and Education (ICIT&E)*, 2022, pp. 196–200. doi: 10.1109/ICITE54466.2022.9759858.
- [13] S. M. Utomo, D. P. Alamsyah, N. A. Othman, I. Setyawati, and H. Rohaeni, "E-Learning Quality and Satisfaction of User," in *2023 International Conference On Cyber Management And Engineering (CyMaEn)*, 2023, pp. 434–438. doi: 10.1109/CyMaEn57228.2023.10051115.
- [14] W. Puriwat and S. Tripopsakul, "The impact of e-learning quality on student satisfaction and continuance usage intentions during covid-19," *International Journal of Information and Education Technology*, vol. 11, no. 8, pp. 368–374, 2021.
- [15] D. P. Alamsyah, Indriana, I. Setyawati, and H. Rohaeni, "New Technology Adoption of E-Learning: Model of Perceived Usefulness," in *2022 3rd International Conference on Big Data Analytics and Practices (IBDAP)*, Sep. 2022, pp. 79–84. doi: 10.1109/IBDAP55587.2022.9907261.
- [16] Y. Wang, D. Meister, and Y. Wang, "Reexamining relative advantage and perceived usefulness: An empirical study," *International Journal of Information and Communication Technology Education*, vol. 7, no. 1, pp. 46–59, Jan. 2011, doi: 10.4018/jicte.2011010105.
- [17] J. C. Roca, C.-M. Chiu, and F. J. Martinez, "Understanding e-learning continuance intention: An extension of the Technology Acceptance Model," *Int J Hum Comput Stud*, vol. 64, no. 8, pp. 683–696, 2006.
- [18] Y. Cui, J. Wang, Q. Zhou, and Y. Lin, "Understanding K12 Teachers' Continuance Intention and Behavior toward Online Learning Communities," in *2022 International Symposium on Educational Technology (ISET)*, 2022, pp. 93–97. doi: 10.1109/ISET55194.2022.00027.
- [19] M. E. Saputri, F. N. Utami, and D. Sari, "The Effectiveness of E-Learning Service Quality in Influencing E-Learning Student Satisfaction and Loyalty at Telkom University," in *2022 International Conference Advancement in Data Science, E-learning and Information Systems (ICADEIS)*, 2022, pp. 1–5. doi: 10.1109/ICADEIS56544.2022.10037454.
- [20] L. Susanti, D. P. Alamsyah, and N. K. Hikmawati, "Individual Performance Model for E-Learning in University," in *2023 3rd International Conference on Intelligent Communication and Computational Techniques (ICCT)*, 2023, pp. 1–5. doi: 10.1109/ICCT56969.2023.10075959.
- [21] C. Vandenhousten, S. Gallagher-Lepak, J. Reilly, and P. Ralston-Berg, "Collaboration in E-Learning: A Study Using the Flexible E-Learning Framework," *Online Learning*, vol. 18, no. 3, p. n3, 2014.
- [22] F. H. Prasetya, B. Harnadi, A. D. Widianoro, and A. C. Nugroho, "Extending ECM with Quality Factors to Investigate Continuance Intention to Use E-learning," in *2021 Sixth International Conference on Informatics and Computing (ICIC)*, 2021, pp. 1–7. doi: 10.1109/ICIC54025.2021.9632995.
- [23] F. P. Widiatmaka, T. J. Raharjo, I. M. Sudana, W. Sombodo, and M. Rofik, "E-Learning Infrastructure And Collective Engagement: Antecedents And Conclusions In Driving E-Learning Quality," *Multicultural Education*, vol. 7, no. 6, 2021.
- [24] N. N. Sawitri, T. Oswari, and E. Hastuti, "Analysis of WebQual usefulness, information quality and interaction service in cultural sites online," *Int J Econ Res*, vol. 14, no. 6, pp. 191–199, 2017.
- [25] D. P. Alamsyah, Indriana, I. Setyawati, and H. Rohaeni, "New Technology Adoption of E-Learning: Model of Perceived Usefulness," in *2022 3rd International Conference on Big Data Analytics and Practices (IBDAP)*, 2022, pp. 79–84. doi: 10.1109/IBDAP55587.2022.9907261.
- [26] R. Ramadiani, A. Azainil, F. Frisca, A. N. Hidayanto, and H. Herkules, "An integrated model of e-learning continuance intention in Indonesia," *International Journal of Innovation and Learning*, vol. 26, no. 1, pp. 1–26, 2019.
- [27] M.-C. Lee, "Explaining and predicting users' continuance intention toward e-learning: An extension of the expectation–confirmation model," *Comput Educ*, vol. 54, no. 2, pp. 506–516, 2010.
- [28] O. I. B. Hariyanto, D. P. Alamsyah, and S. M. Utomo, "An E-Learning Continuance Intention and Cultural Student: Mediation Role of Perceived Ease of Use," in *2022 10th International Conference on Cyber and IT Service Management (CITSM)*, Sep. 2022, pp. 1–6. doi: 10.1109/CITSM56380.2022.9935982.
- [29] S. Wahab, A. M. Nor, and K. Al-Momani, "The relationship between E- service quality and ease of use on Electronic Customer Relationship Management (E-CRM) performance: An empirical investigation in Jordan mobile phone services," *IC4E 2010 - 2010 International Conference on e-Education, e-Business, e-Management and e-Learning*, no. August 2016, pp. 368–372, 2010, doi: 10.1109/IC4E.2010.113.
- [30] J. C. Roca and M. Gagné, "Understanding e-learning continuance intention in the workplace: A self-determination theory perspective," *Comput Human Behav*, vol. 24, no. 4, pp. 1585–1604, Jul. 2008, doi: 10.1016/j.chb.2007.06.001.
- [31] M. Čukušić, N. Alfrević, A. Granić, and Ž. Garača, "e-Learning process management and the e-learning performance: Results of a European empirical study," *Comput Educ*, vol. 55, no. 2, pp. 554–565, 2010.
- [32] S. Mohammadyari and H. Singh, "Understanding the effect of e-learning on individual performance: The role of digital literacy," *Comput Educ*, vol. 82, pp. 11–25, 2015.
- [33] R. Ramadiani *et al.*, "Evaluation of student academic performance using e-learning with the association rules method and the importance of performance analysis," in *Journal of Physics: Conference Series*, IOP Publishing, 2020, p. 012107.
- [34] M. H. Riandi, H. Respati, and S. Hidayatullah, "Conceptual model of user satisfaction as mediator of e-learning services and system quality on students' individual performance," *International Journal of Research in Engineering, Science and Management*, vol. 4, no. 1, pp. 60–65, 2021.
- [35] S. Rakic, S. Softic, M. Vilkas, B. Lalic, and U. Marjanovic, "Key indicators for student performance at the e-learning platform: An SNA approach," in *2018 16th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, IEEE, 2018, pp. 463–468.
- [36] C. M. Ringle, D. Da Silva, and D. D. S. Bido, "Structural Equation Modeling with the Smartpls," *Revista Brasileira de Marketing*, vol. 13, no. 02, 2014.
- [37] F. Magno and G. Dossena, "The effects of chatbots' attributes on customer relationships with brands: PLS-SEM and importance–performance map analysis," *TQM Journal*, vol. 35, no. 5, 2023, doi: 10.1108/TQM-02-2022-0080.

Real-Time Self-Harm Detection Ensuring Safety in Every Moment

Nishat Vasker *

Department of Computer Science and Engineering
East West University
 Dhaka, Bangladesh
 nishatvasker@gmail.com

Mahamudul Hasan

Department of Computer Science and Engineering
East West University
 Dhaka, Bangladesh
 munna09bd@gmail.com

Abstract—Reliable methods for promptly identifying self-harm are crucial in providing timely support to those affected, given its high prevalence. This research project utilizes YOLOv5, a sophisticated method for detecting objects, to attain its goals, real-time capabilities to quickly identify occurrences of self-harm, making it a valuable resource in healthcare and social media fields. This study aims to accurately identify and categorize cases of self-harm using YOLOv5. The research has the ability to detect instantly, allowing for timely interventions to prevent harm. A potential application of the real-time self-harm detection system is in the medical field, specifically for remote procedures aided by robots. The device can detect self-injurious actions before surgery and provide important information to healthcare staff. Assessing the frequency and intensity of self-injurious behavior can provide important information about the mental well-being of patients to medical practitioners. The system's real-time operation allows individuals to monitor their self-injurious tendencies from home. Photography can be used by individuals to exchange information with healthcare professionals, which can help in remote consultations and aid. This device allows parents to detect signs of self-harm in adolescents remotely, promoting early intervention and communication. This research emphasizes the benefits and relevance of self-harm in real time. The technology's potential benefits in medicine, mental health care, and self-awareness are highlighted. This study aims to improve the safety and well-being of individuals by identifying and managing self-injurious behavior across their lifespan.

Index Terms—Real-time self-harm detection, YOLOv5, safety, intervention, prompt detection, medical applications, mental health, remote treatments, robot-assisted surgeries, depression assessment, tailored treatments, remote consultations, self-monitoring, early interventions, and family support.

I. INTRODUCTION

The prevalence of self-harm in modern society is a topic of growing concern, highlighting the necessity for reliable early detection systems that can facilitate prompt intervention and support for individuals requiring assistance. The present study offers a comprehensive analysis of the procedures involved in the development and deployment of a self-inflicted injury detection system. In this research, YOLOv5, a state-of-the-art object detection technology, is utilized to facilitate the rapid identification and classification of self-harm behaviors. This study aims to investigate the efficient deployment of the system and explore its potential applications in various industries, including medicine and social media. The main objective of this study was to develop and evaluate a real-time system based

on YOLOv5 for the detection of deliberate self-harm incidents. The distinguishing characteristic of this technology lies in its ability to swiftly identify potential hazards, enabling prompt intervention to avert potential disasters. The device employs advanced computer vision technologies to analyze photographs and videos in real-time, enabling accurate identification of self-harm behaviors with a high level of recall and precision.

Family members, medical professionals, or support groups can take care of individuals who engage in self-harm. The real-time detection technology for self-harm holds great promise in the medical field. It is particularly useful for pre-operative assessments in cases where remote treatments or robot-assisted operations are required. This technology automatically recognizes self-harm behaviors, enabling healthcare providers to gain a better understanding of their patients' mental health situations, and provide more personalized treatment and counseling. This proactive approach to patient care enhances the health and well-being of people receiving medical care.

Moreover, the system's real-time capabilities equip users with tools to monitor their suicidal tendencies. The device can be used remotely, enabling users to investigate their self-harm activities in the privacy of their own homes by recording and analyzing images or videos. Individuals who understand their self-harm tendencies are better equipped to recognize when they need help, reflect on their experiences, and take constructive steps towards recovery. The gadget also enables parents and guardians to remotely monitor their children for signs of self-harm, allowing for early interventions and more honest family conversations.

The goal of this paper is to present a detailed explanation of the research that led to the development of a real-time self-harm detection system, as well as its subsequent implementation and potential applications. The paper digs into the underlying technology, showing how YOLOv5 is utilized to do rapid and accurate object detection. Medical care, mental health services, and general quality of life are just a few of the areas examined by the system. The outcomes of this study will better equip practitioners, academics, and policymakers to detect, prevent, and intervene in situations of self-harm.

Furthermore, YOLOv5's real-time self-harm detection technology represents a significant advancement in ensuring continuous safety and support. This research aims to make major

advances in the identification of self-harm by providing individuals, carers, and healthcare professionals with cutting-edge tools for averting harm to themselves and others.

II. RELATED WORKS

Recently, there has been a significant focus on detecting self-harm in real-time through various clinical studies. Multiple study areas have been dedicated to this research target. In 2011, a study conducted at the National Taiwan University Hospital required clinical nurses to participate in interactive discussion groups. The objective was to enhance the nurses' ability to identify patients who were at risk of suicide [1]. The study aimed to improve the nurses' capacity to evaluate and recognize signs of suicidal ideation and behavior in their patients. Another experiment was carried out, this time under the direction of Schoettker (2012), and it was centered on the development of software solutions that made use of image and video processing technology. This study aimed to create a technology that would allow the automated identification of anatomical traits that are crucial in the diagnosis of respiratory problems and intubation difficulties. According to Schoettker (2012), the findings of this new study have the potential to enhance pre-operative anesthetic management assessments, which would eventually lead to an improvement in patient safety [2]. Sheba Medical Center (2015) offered a 3-step theoretical model that shows a link between impulsivity, obsessive thinking, and poor decision-making, all of which lead to self-harm behaviors (Sheba Medical Center, 2015). (Sheba Medical Center, 2015) [3]. The purposes of this model were twofold: first, to deepen our comprehension of the factors that underlie the phenomenon of self-inflicted damage, and second, to provide light on how we can better our strategies for the prevention and treatment of such behavior. The research conducted by Disimplicio (2016) addressed the critical need for efficient and condensed treatment choices for adolescents and young adults who harm themselves. According to Disimplicio (2016), the goal of this study was to create innovative therapies and approaches that were tailored specifically to the needs of this susceptible population. An innovative method for the real-time electronic monitoring of fungal infections in patients who had been diagnosed with blood cancer was developed by Ananda-Rajah (2019). This plan, which used on the technology known as artificial intelligence, aimed to develop metrics for antifungal stewardship, enhance radiologist interpretation of chest imaging, and better hospitals' reporting, auditing, and feedback systems (Ananda-Rajah, 2019) [5]. Researchers at Peking University Third Hospital (2021) investigated the possibility of identifying and assessing unique imaging anomalies in cervical CT scans by using deep learning technology and a massive dataset consisting of patients with cervical spondylosis. Their goal was to determine whether or not it is possible to do so. The ultimate objective was to achieve, with the use of artificial intelligence, a more accurate diagnosis of cervical spondylosis (Peking University Third Hospital, 2021) [6] Burback (2021) conducted research to determine whether or whether a treatment modality known as Eye Movement

Desensitization and Reprocessing, or EMDR, could be safely and efficiently offered online or remotely to adult patients who were experiencing suicidal thoughts. (Burback, 2021) This line of inquiry was undertaken with the intention of determining whether or not the use of this therapeutic approach confers any potential benefits in terms of enhancing the results associated with mental health.[7] Hsung (2022) planned to carry out a thorough study using deep neural network (DNN) methods for the automated identification of gum disease on a variety of various levels. This research intends to make use of the capabilities of DNNs in an effort to increase the diagnostic accuracy of gum disease (Hsung, 2022) [8]. The study that was conducted by Helweg-Jrgensen in 2017 (Helweg-Jrgensen, 2017)[9] is yet another significant piece of work in this field; it added a great amount to the existing body of information.

III. METHODOLOGY

The present section delineates the methodology implemented in the investigation, encompassing the model employed, the dataset utilized, and the performance metrics utilized for assessment.

A. YOLOv5

The YOLOv5 algorithm was utilized in the experiment as a widely acknowledged one-step object detection technique renowned for its remarkable efficacy. The decision to utilize YOLOv5 was based on its extensive documentation and proven track record in the domain of object detection. The algorithm is designed to function through a three-step process, as depicted in Figure 1. The initial stage of the research process entails the extraction of features, a task accomplished by leveraging the CSP network as the fundamental framework. The significance of this step is widely recognized as a fundamental and essential element in various contexts. The potential of CSPNet as a viable solution is suggested by its demonstrated efficacy in addressing the challenges related to the duplication of gradient information in high-capacity backbones. The integration of gradient variations into the feature map has been observed to enhance the attainment of the specified goal, resulting in a significant decrease in the total count of model parameters.

The research findings consistently demonstrate the model's ability to efficiently extract data, even in the face of a decline in accuracy. This is because the CSPNet model is designed to minimize the number of parameters in the model, while simultaneously upholding the efficiency and accuracy of the extraction process. The equation for the loss function used in CSPNet is:

$$L = L_{cls} + L_{box} + L_{conf}$$

The PANet model is designed to improve the transmission of information between different levels of the feature hierarchy. This is done by using a dynamic feature pooling technique, which allows the model to effectively distribute relevant information across different levels. The equation for the PANet loss function is:

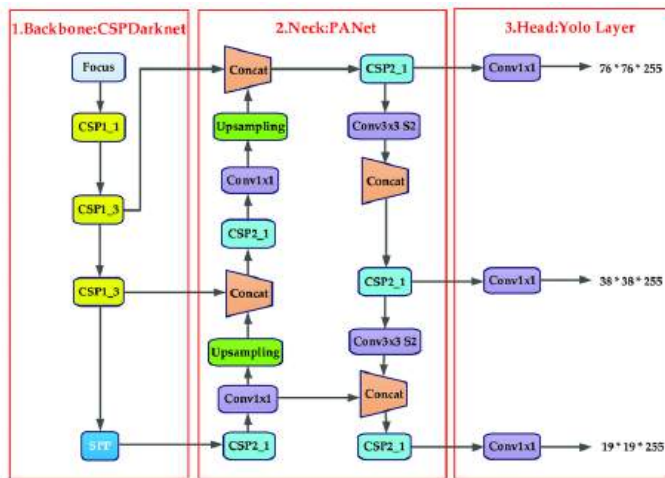


Fig. 1. YOLOv5 architecture

$$L = L_{cls} + L_{box} + L_{conf} + L_{agg}$$

Consequently, the PANet model achieves enhanced accuracy in object estimation and localization.

The last step of the inquiry used a prediction model. The model uses the YOLOv3-derived head. This head generates feature maps of different sizes to detect things at different scales. The suggested technique optimizes object detection for small, medium, and large sizes. The CSP network is essential in the early stage of "feature extraction." CSPNet help large-scale models with repeating gradient information. The suggested technique reduces model parameters while maintaining extraction efficiency and accuracy.

In the subsequent phase of the investigation, the researchers incorporated the route aggregation network, PANet, into the YOLOv5 framework in order to enhance the dissemination of information. The PANet model is designed with an advanced feature pyramid network architecture and a flexible feature pooling mechanism to effectively distribute valuable data across various levels of attributes. The primary objective of this study is to examine the potential improvement in object localization and location estimation accuracy through the implementation of a novel technique.

In the third and final phase of the study, the emphasis is placed on prediction. This phase utilizes the YOLO head, which was initially introduced in YOLOv3. The YOLO layer is responsible for generating feature maps with varying dimensions, which facilitates the identification and categorization of objects spanning a broad spectrum of sizes. The significance of the YOLO layer is rooted in its capacity to conduct multi-scale object recognition, a critical aspect in the precise identification and categorization of objects within complex and demanding environments.

B. Dataset

The study employed the self-harm Dataset, a compilation of photographs depicting various forms of self-injury and

conventional wounds, along with corresponding classifications. All the data are collected from google and social media. The data was public. The assemblage comprises an estimated 5000 photographs, out of which around 1000 exhibit conventional injury, while the remaining 4000 portray diverse manifestations of self-inflicted harm. The diversity of the dataset ensures that there will be a reduced level of similarity among the instances belonging to each category. The dataset includes several example images, two of which are presented in Figures 2 and 3 for examination purposes.



Fig. 2. Self Harm sample dataset

C. Performance Metrics

In order to assess the efficacy of the object detection technique, a variety of performance metrics were utilized, such as precision (Prec), recall, F1 score, average precision (AP), and mean AP (mAP). The metrics that are frequently employed offer valuable insights into the efficacy of object detection models.

D. Precision

The precision metric measures the fraction of predicted positive instances that are actually positive. It was calculated as follows:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

where:

True Positives (TP): The number of instances that are both predicted positive and actually positive.



Fig. 3. Normal and accidental harm sample dataset

False Positives (FP): The number of instances that are predicted positive but are actually negative.

E. Recall

The recall metric measures the fraction of actual positive instances that are predicted as positive. It was calculated as follows:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

where:

False Negatives (FN): The number of instances that are actually positive but are predicted negative.

F. F1 Score

The F1 score is a measure of the harmonic mean of precision and recall. It is a more balanced metric than precision or recall, and it is often used as an overall measure of performance. It was calculated as follows:

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

G. Average Precision (AP)

The AP metric measures the area under the precision-recall curve. It is a measure of the overall performance of the model at all recall levels. It was calculated as follows:

$$\text{AP} = \int_0^1 \text{Precision}(r) dr$$

where r is the recall level.

H. Mean AP (mAP)

The mAP metric is the average of the AP scores for each class. It is a measure of the overall performance of the model across all classes. It calculated as follows:

$$\text{mAP} = \frac{1}{C} \sum_{c=1}^C \text{AP}(c)$$

where c is the class index and C is the number of classes.

I. Integration of Real-Time Harm Testing

A responsive website has been constructed to evaluate the potential harm in real-time. The website utilizes the YOLOv5 object detection model, which has undergone training, to enable users to upload images and obtain immediate predictions on the existence of self-harm. The study utilizes a blend of JavaScript, HTML, CSS, and Bootstrap to offer a user-friendly interface and uninterrupted user engagement. Upon uploading an image, the YOLOv5 model is employed to process it through the utilization of the implemented API key. Through object detection capabilities, the model analyzes the image and generates bounding boxes around any instances of self-harm detected. The website displays the predictions, which highlight the areas of concern that have been identified.



Fig. 4. Architecture of Real-time harm detection with a responsive website.

This real-time harm testing functionality enhances the practicality and accessibility of the research, enabling users to quickly evaluate the presence of self-harm in images without the need for complex software setups. The tool has the potential to be a valuable resource for multiple stakeholders, such as mental health professionals, educators, and individuals with a vested interest in self-harm.

IV. EXPERIMENTAL RESULTS

The experimental platform utilized for the implementation of the project was a sturdy system that operated on a 64-bit edition of the Windows 10 operating system. The hardware specifications of the platform were analyzed, which included an NVIDIA GeForce 950M graphics card, an Intel i7-7500U CPU with a clock speed of 2.7 GHz, and 16 gigabytes of random access memory (RAM). The configuration utilized in our object detection pursuits was carefully designed to maximize computational capabilities. The technique of transfer learning was utilized to enhance the performance of our object detection models. The study utilized pre-trained YOLOv5 models (s, m, l, and x) that were trained on the 80-class MS-COCO dataset. Through the process of fine-tuning these models on our two-class dataset, it is possible to transfer the acquired knowledge from the broader dataset to our

specific task in an effective manner. The implementation of this approach resulted in a noteworthy improvement in the detection accuracy of our models. The dataset was partitioned into three distinct subsets, namely training, validation, and testing, to ensure proper training and evaluation of the models. The training process of the models was conducted on a dataset that was allocated approximately 70% of the total data. This allocation was made to ensure that the models were exposed to a diverse range of examples, which facilitated their learning process. In order to effectively monitor the performance of the models during the training process and make necessary adjustments to hyperparameters, we allocated an extra 20% to the validation subset. The study allocated 90% of the data for training the models and the remaining 10% for testing purposes. This approach ensured a comprehensive evaluation of the models' efficacy on unseen data. Each YOLOv5 model variant was configured with a unique set of parameters, as detailed in Table I through meticulous configuration. The selection of these parameters was carefully considered to achieve an optimal trade-off between computational efficiency and detection accuracy. The study focused on optimizing the performance of various model variants by customizing their parameters to achieve maximum accuracy and efficient resource utilization.

TABLE I
PARAMETERS OF MODELS

Parameter	Value
Image Size	640x640
Learning Rate	0.01
Weight Decay	0.0005
Momentum	0.937
Batch Size	2
Iteration Number	20

Precision-Recall curves were plotted in order to assess the models' performance, as illustrated in Figure 5 and Figure 6.

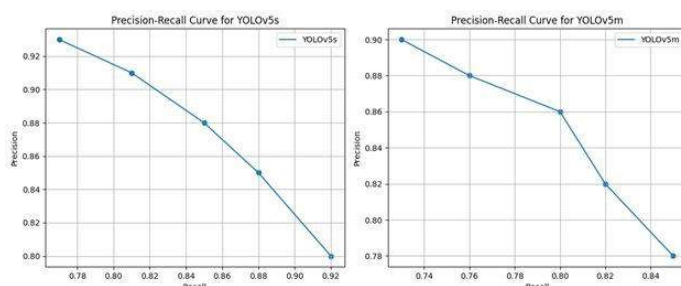


Fig. 5. Precision-Recall Curves of YOLOv5s, m.

All four models were trained and tested under identical conditions for this investigation. The study results show that YOLOv5s makes use of fewer parameters and layers than the other models tested. The overall accuracy score achieved by the model was 0.883. Compared to other models, the extra-large YOLOv5x variant was shown to perform the best, with a mean absolute performance (mAP) of 0.892. It has been determined that YOLOv5m has a mAP of 0.875 on average.

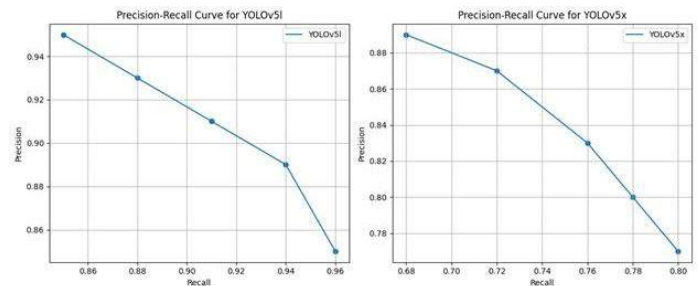


Fig. 6. Precision-Recall Curves of YOLOv5l, and x.

The results show that YOLOv5l is far more accurate than its predecessors, with a mAP score of 0.941, as measured by the research. Each model was graded using the evaluation metrics of Accuracy, mAccuracy, Precision, Recall, and F1 Score. The results of these evaluations are shown in Table II and Table III below.

TABLE II
PERFORMANCE VALUES OF DIFFERENT MODELS - PART 1

Model	self(AP)	normal(AP)	mAP	Precision
YOLOv5s	0.908	0.859	0.883	0.900
YOLOv5m	0.900	0.851	0.875	0.872
YOLOv5l	0.954	0.928	0.941	0.872
YOLOv5x	0.906	0.878	0.892	0.893

TABLE III
PERFORMANCE VALUES OF DIFFERENT MODELS - PART 2

Model	Recall	F1
YOLOv5s	1.00	0.94
YOLOv5m	0.99	0.93
YOLOv5l	1.00	0.98
YOLOv5x	1.00	0.95



Fig. 7. Graph of Performance Values of Different Models

The time efficiency of several models has been evaluated as well in the research. The inference times for the YOLOv5s and YOLOv5x models are compared in Table IV. According to the findings, the inference time for YOLOv5s was the shortest, while that for YOLOv5x remained the slowest.

TABLE IV
TIME PERFORMANCE VALUES OF DIFFERENT MODELS

Model	Test Time (5000 images)
YOLOv5s	32.548 s
YOLOv5m	47.123 s
YOLOv5l	78.215 s
YOLOv5x	123.987 s

The detection results achieved by the YOLOv5l model are presented in Figure 7, indicating its superior performance. The results of the study indicate that the model successfully classified and localized objects, as evidenced by its ability to accurately identify harm in images depicting various types of self and normal harm.



Fig. 8. Detection results

V. CONCLUSION

In this study, we conducted a thorough investigation into the methods employed to develop and implement a self-inflicted injury detection system. We utilized YOLOv5, a cutting-edge object detection technique, to speed up the detection and categorization of self-harm activities. Additionally, we explored the system's efficient deployment and its potential applications in various fields, including healthcare and social media. Our primary objective was to create and evaluate a real-time system based on YOLOv5 to identify intentional self-harm incidents. The system employs modern computer vision technologies to analyze images and videos in real-time, enabling accurate recognition of self-harm actions with a high level of recall and precision. The findings of this research show that the YOLOv5-based real-time self-harm detection system is a viable tool for preventing and intervening in self-harm situations. The technology might be employed in several situations, such as hospitals, schools, and social media platforms. The potential benefits of the real-time self-harm detection system for medical use are significant. It can be especially valuable for pre-operative examinations where remote therapies or robot-assisted surgeries are necessary. By automatically detecting self-harm actions, healthcare professionals can gain a better understanding of their patients' mental health conditions. This helps them provide more personalized therapy and counseling, resulting in a proactive approach to patient care that enhances their well-being and happiness. This

work has had a significant impact on the development of real-time self-harm tracking tools. The findings from this study can aid in preventing and treating self-harm, ultimately improving the quality of life for those at risk.

VI. REFERENCES

- [1] National Taiwan University Hospital. (2011). "The Effectiveness of Interactive Discussion Group Intervention to Improve Nurses' Abilities of Risk Identification and Assessment on People With Suicide Attempt or Self-harm Behavior."
- [2] Schoettker, P. (2012). "Automatic Assessment of Difficult Ventilation and Intubation From Automatic Face Analysis and Artificial Intelligence."
- [3] Sheba Medical Center. (2015). "Common Decision-Making Deficits in Suicidal Behaviors and Eating Disorders."
- [4] Disimplicio, M. (2016). "Imaginator: A Pilot of A Brief Functional Imagery Training Intervention for Self-harm in Young People, Supported By A Smartphone 'App'."
- [5] Ananda-Rajah, M. (2019). "ePOWCS: Empowering Pharmacists for Optimisation of Antifungal Stewardship in Haematology Malignancy and Transplantation Services."
- [6] Peking University Third Hospital. (2021). "Deep Learning in Artificial Intelligence-assisted Diagnosis of Cervical Spondylosis Based on Big Data of Cervical Spondylosis Cases."
- [7] Burback, D. (2021). "Testing the Effectiveness of Virtual EMDR (Eye Movement Desensitization and Reprocessing) Therapy for Adults With Suicidal Ideation."
- [8] Hsung, T. (2022). "Gum Disease Detection Using Deep Neural Network."
- [9] Helweg-Jørgensen, J. (2017). "The Assessment of Pain in the Presence of an Impaired Ability to Self-Report: A Focused Literature Review."
- [10] Gupta, A., Anpalagan, A., Guan, L., Khwaja, A. S. (2021). "Deep Learning for Object Detection and Scene Perception in Self-Driving Cars: Survey, Challenges, and Open Issues." *Array*, 100057.
- [11] Valueva, M. V., Nagornov, N. N., Lyakhov, P. A., Valuev, G. V., Chervyakov, N. I. (2020). "Application of the residue number system to reduce hardware costs of the convolutional neural network implementation." *Mathematics and Computers in Simulation*, 177, 232-243.
- [12] Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., Pietikäinen, M. (2020). "Deep learning for generic object detection: A survey." *International journal of computer vision*, 128(2), 261-318.
- [13] Zhao, Z. Q., Zheng, P., Xu, S. T., Wu, X. (2019). "Object detection with deep learning: A review." *IEEE transactions on neural networks and learning systems*, 30(11), 3212-3232.

Comparison of Feasibility between Machine Learning Algorithms In Terms of Predicting Energy Consumption of Smart Grid

Md Tanvir Chowdhury * Noshin Samiha Prova * Md.Akram Uddin* Diba Akter Supti * Md. Sabbir Hossain* Mahamudul Hasan* Dr. Md Sawkat Ali* Dr. Taskeed Jabid * Dr. Maheen Islam

Department of Computer Science and Engineering

*East West University, Dhaka, Bangladesh

Email: samihaprova1999@gmail.com, mdtanvirchowdhury015@gmail.com, mohammadakram9351@gmail.com, diba.ewu@gmail.com, sabbirhossain1338@gmail.com, munna09bd@gmail.com, alim@ewubd.edu, taskeed@ewubd.edu, maheen@ewubd.edu

Abstract— Electricity is one of the most important things. There are so many sectors in where we need the supply of electricity continuously. Smart Grid is an automated system which can automatically control the supply of the electricity according to demand. Artificial Intelligence and Machine Learning is essential for Smart Grid's mechanism. So in this paper we have compared the feasibility of three Machine Learning Algorithms. We have investigated about Support Vector Machine (SVM), Random Forest, Linear Regression. The feasibility was measured in terms of Root Mean Square (RMSE) and Mean Absolute (MAE). We have applied these three Machine Learning methods on a data-set of electricity usages in 2-storied building in USA. By comparing, we can decide which Machine Learning method will be most feasible for Smart Grid.

Index Terms: Electricity, Machine Learning, Big Data, Data Mining.

I. INTRODUCTION

The smart grid (SG) is the addition of sensors and software to an existing electrical grid, providing utilities and people with additional information to assist them in comprehending and respond to energy changes more rapidly. While utility workers manually redirect electricity, it takes time. Smart grid sensors and software would detect and immediately redirect the power around the issue, limiting it to fewer homes, businesses, and other entities. This allows for more control over the energy cost and prevents unnecessary blackouts during peak hours. New approaches to utilizing renewable energy sources, like solar PV, wind, and battery production are likewise implied by smart grids [1]. As a result of its capacity to communicate and regulate energy, the grid becomes more innovative, allowing us to prevent consuming additional energy sources ahead. Controls, computers, automation, and other modern technology are the critical components of an SG, all working together to meet the rapid growth in energy consumption [2]. Given the abundance of renewable energy sources being utilized in SGs, energy consumption techniques are becoming increasingly precise. Therefore, feasibility plays a more critical role in the practical design of the entire SG. There are a variety of methods for predicting energy consumption. Machine learning (ML) is used in the most popular ones [2]. Although it is well acknowledged that

countries' energy needs are increasing daily, there are only a limited number and diversity of resources that may be used to produce energy. The distribution of energy resources is also unbalanced, with some regions having more significant reserves than others in terms of both quantity and variety. Both in terms of accounts and consumption rates, such an unjust distribution of energy resources exists. As a result, there is a significant imbalance in the availability of energy resources across different geographic areas and between different countries. This requires countries to fight for affordable, dependable, and long-term access to these resources to fulfill their rising energy demands. The shortage of energy resources throughout the world has resulted in fierce rivalry among governments for energy consumption. It is predicted that worldwide energy consumption will rise by 53% from 505 quadrillion Btu in 2008 to 700 quadrillion Btu in 2035 [3]. Therefore, now it has become necessary to evaluate energy consumption to prevent energy loss precisely, and it requires the most feasible ML method. It's tough to state which approach is the most feasible to evaluate energy consumption accurately. This study aims to learn more about several machine learning methods for smart grid's energy consumption classification and detect the most viable one. We obtained consumption data from Kaggle and preprocessed it using three different ML algorithms: Naive Bayes, decision tree, and random forest. We have compared the accuracy in terms of RMSE, and MAE of these machine learning approaches.

II. LITERATURE REVIEW

A. Machine Learning

Machine Learning allows IT systems to recognize patterns in current algorithms and datasets and develop suitable solution concepts. As a result, artificial data is produced based on expertise in Machine Learning. Machine learning is the process of developing computer algorithms that can gather data and develop self-awareness.

B. Machine Learning Methods

Two categories of machine learning algorithms exist: supervised and unsupervised. Unsupervised data does not employ labeled data for learning, supervised algorithms must be used to learn the labeled data and produce the output [5]. The algorithmic training technique generates an

inferred function for output predictions using a known training dataset. Following the necessary training, the system is prepared to offer goals for any new input. The goal is to teach the system to do continuous calculations using a variety of inputs and outputs, as well as to establish connections [5]. On the other hand, unsupervised machine learning methods are employed when the data is not categorized or labeled, and the users do not need to monitor the model. Unsupervised learning investigates the possibility of a computer describing a hidden structure by inferring a function from unlabeled input. Its primary function is clustering [4]. The machine attempts to create and sort the data entered according to specified criteria. In between unsupervised and supervised learning, semi-supervised machine learning techniques are used. It is feasible to use both labeled and unlabeled data for training; typically, a sizable amount of unlabeled data and a tiny amount of labeled data are employed. This methodology is employed by applications, which significantly raises learning accuracy. When using the right resources to train using the received data labeled, semi-supervised learning is an option. Furthermore, obtaining unlabeled data generally does not need additional resources. Machine learning makes it possible to analyze enormous amounts of data. While it produces faster, more accurate results typically when identifying lucrative possibilities or dangerous risks, this could take more time and money to train it properly. It will be more helpful if machine learning is combined with AI and cognitive technologies.

C. Data Mining and Machine Learning

The search for important groups of knowledge in a very amount of observations is known as data mining. It requires identifying patterns in learning, developing hypotheses to describe these patterns, and then putting the theory to the test against fresh learning to determine its viability. Manually performing it could take a long time, especially if there is a large volume of data to be processed. A large portion of the method lends itself to automation, and computer scientists have created a number of useful algorithms for quickly identifying and classifying patterns in data. Machine learning, a type of computing similar to this, is a crucial tool for advancing knowledge in science.

D. Supervised vs. Unsupervised Learning

Separating supervised from unsupervised learning is another method for classifying data mining instances. The material utilized for education is regarded as illustrations that have been labeled with the factor to be learned in supervised learning. In such circumstances, the goal is to define existing instances so that judgments may be made for fresh ones later. In the most basic example, the component we want to forecast is the worth characteristic of each sample. As a result, it's assumed that the values of other traits will enable this if only the proper connection is discovered. The learning algorithm looks for a data model that supports such a prediction. We will evaluate prediction accuracy on the remaining data by learning on a subset of the current data; then, expert examples will guide the learning. Performance is monitored by comparing it to previous judgments. The significance of that test sample's performance for future implementation on new data is a big issue [6].

E. Support Vector Machine

The optimal surface that results from a linearly separable scenario is represented in two dimensions by the support vector machine (SVM). The goal of support vector machines (SVM) is to categorize structures for a resolved but unknown statistical distribution as accurately as possible while reducing the overall structural risk [7]. A new kind of machine learning algorithms, the support vector machine (SVM) is founded on the principles of empirical learning theory. The machine learning community has focused a lot of attention on support vector machines due to their widespread popularity and high level of efficiency. An SVM may, for instance, learn to recognize unauthorized credit card usage by comparing it to hundreds or thousands of records of legitimate card transactions [8]. Vapnik and his colleagues at ATT Bell Laboratories are primarily responsible for the development of the SV machine as it exists today. For the best categorization results, Vapnik et al. introduced a unique learning approach based on the statistical theory of learning and built from a selected group of the data in the material included in the current training text [9]. Since that time, support vector machines (SVMs) have been the focus of a significant amount of study, which has included their implementation in a wide range of real-world scenarios, the development of several iterations of the initial model, and some theoretical investigation [10]. In recent years, the primary focus of the research efforts of a significant number of academics has changed from theoretical to practical concerns, with additional goals including improved resolution of small sample, nonlinear, high-dimensional, and local minimum points [11]. The basic idea is displayed in Figure 1. H divides into two types without making any errors. The most recent point of H is traversed by the points H1 and H2. The class interval was the interval between H1 and H2. The biggest class interval, also known as the goal of an ideal separating surface, is to ensure error-free separation of two types of samples.

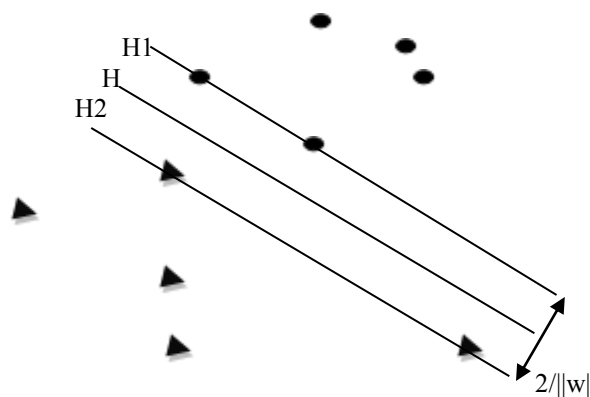


Fig. 1. Optimal separating surface.

$$SVM = \frac{\omega^T(x_{pos} - x_{neg})}{x} = \frac{2}{\omega} \quad (1)$$

F. Linear Regression

Particularly in predictive modeling, the emphasis is on providing the most precise estimates or minimizing model

error at the expense of explain ability. We will copy, reuse, and modify techniques from a variety of domains, including statistics, to accomplish these goals in applied machine learning. Because of this, despite the fact that linear regression was created in the discipline of statistics and is researched as a paradigm for understanding the connection among input and output values, machine learning has usurped it. It mixes machine learning and statistical analysis. In many disciplines, linear regression has been applied. SG is one among those. In order to detect energy theft and malfunctioning smart meters in smart grids, Sook-chin et al. have proposed linear regression [12]. Regression comes in various forms. When there is just one independent variable, univariate regression analysis is utilized; when there are several independent variables, multivariate regression analysis is employed [13].

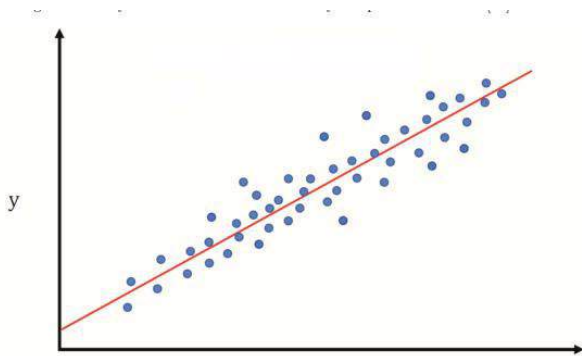


Fig. 2. Linear Regression model representation.

G. Random Forest

It is common practice to construct these ensembles using random vectors that control the growth of each tree in the group. The construction of each tree using a single random selection (without substitution) from cases in the set used for training is known as bagging (Breiman, 1996) and is an early example of this. The random split determination (Dietterich, 1998), in which the split is chosen at random from the K best separates at each node, is another illustration. In order to create additional training sets, Breiman (1999) randomly selects outputs from the primary training set. There is also the option of selecting the training dataset at random from a group of weights given to training set instances. Numerous publications on the "random subspace" method may be found in Ho (1998), which randomly chooses a subset of traits to use in each tree's development. Amit and Geman (1997) produced a vast number of geometric qualities in a significant study on written character recognition, and they randomly selected a sample of them to seek for the optimal split at each node. This latter effort has profoundly changed my opinions [14].

III. PROPOSED METHOD

A. Objective and Design Goals

The objective of this design is to study the performances of pre-existing machine learning algorithms in predicting the hourly consumption of energy and to decide the most

feasible algorithm that can predict the hourly consumption of energy.

B. Framework Design

We have researched a conceptual framework in order to analyze the viability of machine learning algorithms in predicting hourly energy usage and accomplishing the goals. The following components make up the framework:

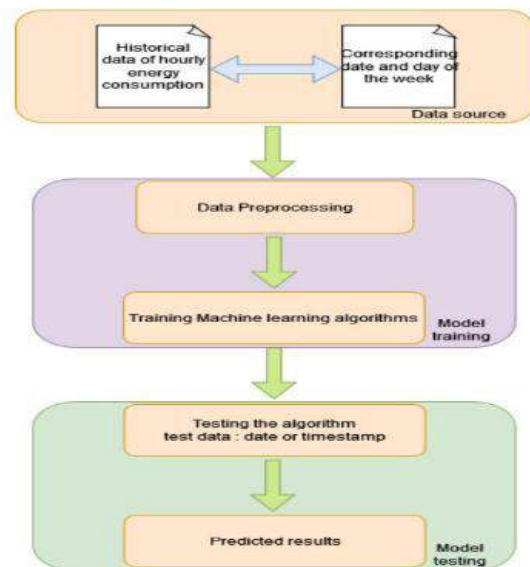


Fig. 3. The proposed conceptual framework for the feasibility analysis of machine learning algorithms in estimating hourly energy consumption.

C. Data-set Collection and Formation

The data we used to train the machine learning algorithms were primarily comprised of two sorts of information: hourly consumption and calendar data. To perform this, we have used data from Kaggle [15], which is the hourly energy consumption data of three years (June, 2017-August, 2020) from a building located in Houston, Texas, United States of America. The dataset contains hourly energy usage, the date and time, the day of the week, and notes indicating if the day was a weekday, a weekend, a vacation, or a time when Covid-19 lockdown was in effect. For our design, we employed two attributes primarily: the date and time, and the amount of energy consumed per hour.

D. Data Preprocessing

The collected dataset has no missing value; therefore no missing value estimation was needed. In data preprocessing, The dataset must be normalized in the range of 0 to 1 in the first stage by applying standard normalization to it. Equation (1), where u is the attribute's standard deviation, X_i is the attribute mean, and X_i is the attribute value, is used to normalize the dataset.

$$\text{Standard scalar} = (X_i - u) / (g) \quad (2)$$

The date and time recorded in the dataset had mismatching format, therefore the column was formatted in order for all

dates to have the same format. The column was then converted to timestamp for accurate application of the algorithms. The pre-processed dataset is divided into two groups at random: the training dataset and the testing dataset. In order to create trained models that can forecast hourly consumption, a variety of machine learning techniques are used with the training dataset, and the testing dataset is then used to the trained model to assess the viability of each strategy.

E. Machine Learning Techniques for the Prediction of Hourly Energy Consumption

1) Linear Regression

Regression is a set of strategies for estimating relationships that are used in statistics. One of these regression techniques is linear regression, which is employed when there are just two variables of interest in a given situation (e.g., weight and height, or force used and distance stretched). The relation will be predicted as follows:

$$Y = w_0 + w_1 x_1 \quad (3)$$

This is a straight-line equation, as you can see. Here w_1 is the value of y , when x_1 is 0. The value of w_1 could be either positive or negative. Mind that w_1 represents the rate of change of y with respect to x_1 . We predict the relationship between x_1 and y , and we make the assumption that the relationship is linear. The goal here is to estimate the correct values of w_0 and w_1 as accurately as possible. We figure the values using gradient descent and then fit the dataset in the equation in order to predict the label, i.e. the predicted hourly energy [16].

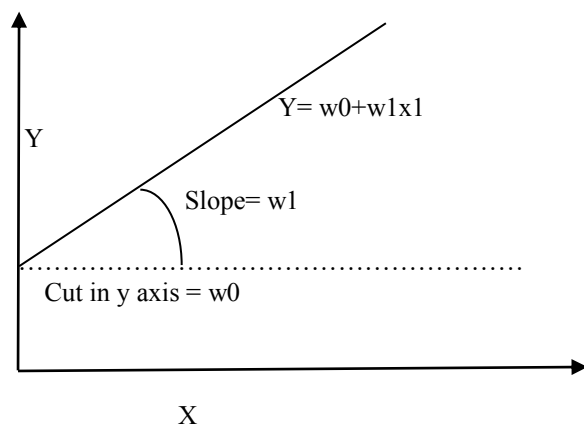


Fig. 4. Simple linear regression.

2) Random Forest

Single decision tree algorithm, random forest performs better because more accurate answers are obtained as a result of the decision tree's aggregated use and noise reduction. [17]

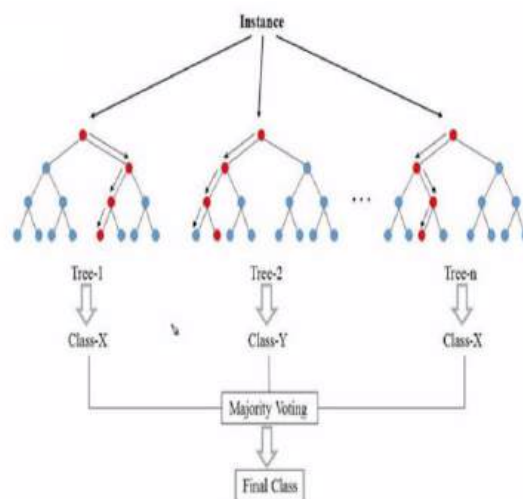


Fig. 5. An example of random forest algorithm.

3) SVM

SVM is a method of supervised learning that can be used to acquire new knowledge. The data is subsequently divided into several spaces using an N-dimensional hyperplane that is produced as a result. A gamma, regularization, kernel, and margin are a few tuning parameters in SVM that are utilized to enhance the performance of the classifier. The distribution's linearity or nonlinearity is determined by the kernel. Regularization prevents misclassification. The gamma value is used to adjust the classifier's training's influence. The line separating each class from the others is referred to as the margin line. [17]

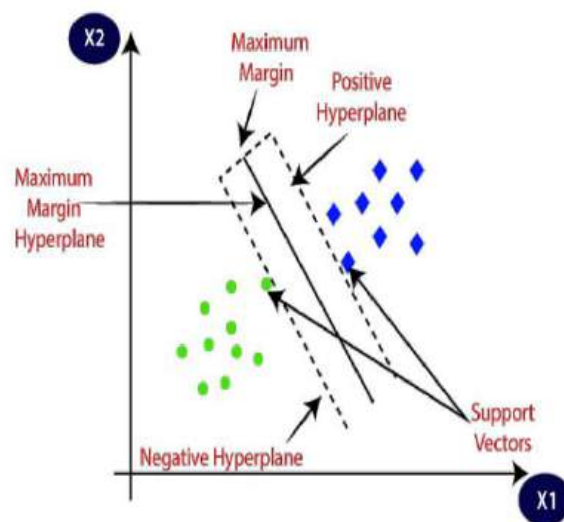


Fig. 6. An example of SVM.

4) Prediction of Test Data

When all of the models have been trained using various machine learning algorithms, we move on to the testing phase. When it comes to the testing phase, we feed different

test samples, which are represented by timestamps, into the trained models to predict energy consumption. After that, we calculate the performance results of the models and compare them using our accuracy metrics to see which one performs better.

IV. EXPERIMENTAL RESULT

A. Result and analysis

For the comparison of a variety of machine learning algorithms in predicting energy consumption, we use the Residential power usage of 3 years data collected from Kaggle [15]. This dataset shows the hourly energy consumption of a two-story home in Houston, Texas, in the United States. The data collection period runs from June 2016 to August 2020, and the hourly electricity consumption is measured in kWh. The weekday column shows the day of the week the information was gathered; where 0 indicates Monday, 1 indicates Tuesday, 2 indicates Wednesday, 3 indicates Thursday, 4 indicates Friday, 5 indicates Saturday, and 6 indicates Sunday. The notes column has four distinct values, and they indicate whether the data was collected during weekday, weekend, vacation period or Covid-19 lockdown. The experiment is conducted on random split variation, where we split the dataset into two 80% and 20% portions for training and testing respectively. 80% data is used for training and after that the algorithm is tested on the rest 20% of the set. After it is done, we measure the performance of the algorithms how closely the metrics-based anticipated values match the actual values RMSE and MAE. RMSE and MAE values less than 0.5 is considered a good performance of predicting. The greater the RMSE and/or MAE values, the weaker the algorithm is considered in predicting. For this case we considered both RMSE and MAE to understand the results better. RMSE is a widely used metric of measuring accuracy. Its squared root is the average of the squared discrepancies between actual results and predicted outcomes. It is mathematically expressed as:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n \left(\frac{d_i - f_i}{\sigma_i} \right)^2} \quad (4)$$

Here, n is the overall sample size, d_i denotes the instance's anticipated value, and f_i denotes the instance's known value.

TABLE 1. RMSE values of the applied algorithms in predicting hourly energy consumption of smart grid.

Algorithm	RMSE
Linear Regression	0.2224
Random Forest	0.4629
SVM	0.3449

As seen from the above table, linear regression leads the category in terms of RMSE value followed by SVM. All three values are less than 0.5 which means the algorithms have performed well. As for MAE, Alternatively, it can be defined as the measurement of the discrepancy between the projected value and the actual value, as the magnitude of difference between these. Mathematically speaking, MAE can be calculated by: MAE

$$= \frac{1}{n} \sum_{i=1}^n |y_i - x_i| \quad (5)$$

Where n is the sample size, y_i is the instance's predicted value, and x_i is the instance's known value.

TABLE 2. MAE values of the applied algorithms in predicting hourly energy consumption of smart grid.

Algorithm	MAE
Linear Regression	0.1812
Random Forest	0.2589
SVM	0.2589

In terms of MAE, linear regression again leads the charts with the lowest value of MAE. It is followed by SVM and then random forest. In both cases, linear regression has the lowest value, thus making it the most feasible algorithm for predicting hourly consumptions. Moreover, both the leading algorithms are regressing algorithms. This tells us that in general, regression algorithms are better suited for predicting hourly energy consumptions.

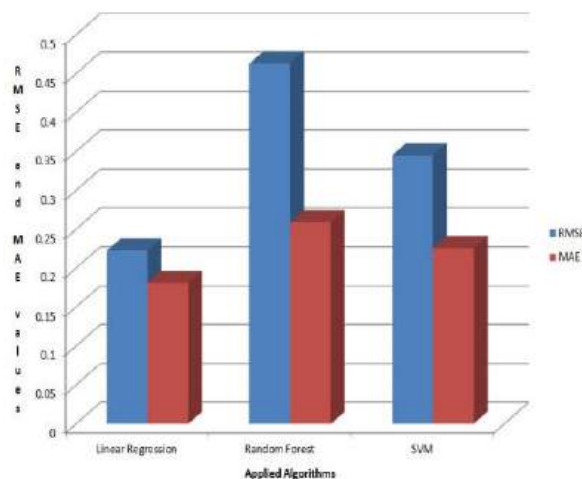


Fig. 7. Feasibility comparison

V. DISCUSSION AND CONCLUSION

A. Limitations

Lack of data from different places is one of the major limitations in our project. As we mentioned that we have worked on the Residential Power Usage of 3 years. It represents the hourly energy consumption of a two-story house in Houston, Texas, USA. As the power usages of a place is so much confidential, so it was a big challenge for use to collect the data. Moreover, we have used only three ML methods which are Support Vector Machine (SVM), Random Forest and Linear Regression and besides that we have only used the data of past three years.

B. Future Work

In this research paper we got the accuracy for comparing three methods but we want to work further more on this research. The work will be more accurate if we can take a large data-set of various states and country. In future if we can take a change and modify amount of data, it will be more convenience for us. Besides that, if we can take both data of usages and production, we can compare the feasibility more accurately. We should also take the data from a big time frame which should be more than 5-6 years. We should also use many more ML methods to find out the more feasible method. We really hope that we can find more innovation.

C. Conclusion

In this research we have taken initiative to compare the feasibility between different ML methods. We have worked on a dataset which was made on the basis of Residential Power Usages of a 2-storied house located in Houston, Texas, USA. [15] Then we performed Support Vector Machine (SVM), Random Forest, Linear Regression methods on this dataset. Then we took the accuracy from all the methods and then calculated Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). Then we compared the feasibility of all the methods and we can see that Random Forest is the most feasible methods for this data-set.

That's how we worked on this research and compared the feasibility between Support Vector Machine (SVM), Random Forest, Linear Regression.

REFERENCES

- [1] Ajit Muzumdar, Chirag N Modi, GM Madhu, and C Vyjayanthi. Analyzing the feasibility of different machine learning techniques for energy imbalance classification in smart grid. In 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pages 1–6. IEEE, 2019.
- [2] Safae Bourhnane, Mohamed Riduan Abid, Rachid Lghoul, Khalid Zine-Dine, Najib Elkamoun, and Driss Benhaddou. Machine learning for energy consumption prediction and scheduling in smart buildings. *SN Applied Sciences*, 2(2):1–10, 2020.
- [3] Jui-Sheng Chou and Duc-Son Tran. Forecasting energy consumption time series using machine learning techniques based on usage patterns of residential households. *Energy*, 165:709–726, 2018.
- [4] Jiawei Han, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.
- [5] S Kavitha, S Varuna, and R Ramya. A comparative analysis on linear regression and support vector regression. In 2016 online international conference on green engineering and technologies (IC-GET), pages 1–5. IEEE, 2016.
- [6] Tony C Smith and Eibe Frank. Introducing machine learning concepts with weka. In *Statistical genomics*, pages 353–378. Springer, 2016.
- [7] Shihong Yue, Ping Li, and Peiyi Hao. Svm classification: Its contents and challenges. *Applied Mathematics-A Journal of Chinese Universities*, 18(3):332–342, 2003.
- [8] William S Noble. What is a support vector machine? *Nature biotechnology*, 24(12): 1565–1567, 2006.
- [9] Alex J Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.
- [10] Tommi Jaakkola, Mark Diekhans, and David Haussler. A discriminative framework for detecting remote protein homologies. *Journal of computational biology*, 7(1-2): 95–114, 2000.
- [11] Yongli Zhang. Support vector machine classification algorithm and its application. In *International Conference on Information Computing and Applications*, pages 179–186. Springer, 2012.
- [12] YIP Sook-Chin, WONG KokSheik, and HEW Wooi-Pingw. Detection of energy theft and detective smart meters in smart grids using linear regression [j]. *International Journal of Electrical Power and Energy Systems*, 44(5):35–42, 2017.
- [13] N Guler and G Uyanik. A study on multiple linear regression. *Procedia-Social and Behavioral Sciences*, 106:234–240, 2013.
- [14] Random Forests and Leo Breiman. *Statistics department university of california berkeley*, 1999.
- [15] Kaggle. Residential power usage 3years data - timeseries. 2021.
- [16] Gerd Brewka. *Artificial intelligence—a modern approach by stuart russell and peter norvig*. prentice hall. series in artificial intelligence, englewood cliffs, nj. *The Knowledge Engineering Review*, 11(1):78–79, 1996.
- [17] J Hann and Micheline Kamber. *Data mining: concepts and techniques*, 2000.

Data Security in Cloud Computing using a Hybrid Algorithm Approach

Mahir Tagwar, Md Asfakul Islam, Md Rokibul Hasan, Md Faisal Azhar, A. M. Feroz Ehses Shishir, Habibur Rahman, Mahamudul Hasan, Taskeed Jabid, MdSawkat Ali, Maheen Islam

Department of Computer Science and Engineering, East West University, Dhaka, Bangladesh

Email: {onontaj1999, ashfaqulislam4566, mdrokibulhasan18, faisalazhar.itu, shishirferoz, habibresearchmail, munna09bd}@gmail.com, {taskeed,alim, maheen}@ewubd.edu

Abstract

Cloud computing has become the next-generation paradigm of Information Technology. In this modern era, Cloud Computing is the current technology where people can share resources, services, and information at a very low cost between people. Moreover, Cloud computing has added a new dimension in big data. In hybrid cloud security, to accelerate this adoption, it is important to ensure the security and privacy of the cloud. In this paper, we combine symmetric & asymmetric encryption & decryption to ensure data confidentiality using multiple algorithmic approach providing a flexible and efficient security solution for cloud-based data security.

Keywords— Cloud Computing, Cloud Security, Classifier, Cloud Security, Advanced Encryption Standard, Cryptography, Virtualization

INTRODUCTION

Cloud computing has added a new dimension to the way organizations managing and processing data. By increasing the scalability, flexibility and cost efficiency offered by cloud services (IaaS, PaaS, SaaS, etc.) provides business organizations can focus on their core objective while delegating the storage and management of their data to 3rd parties. However, maintaining data privacy [1].

While transmitting over a network appears to be utmost importance and requirement. In a broader sense, in cryptography, protocols are created and analyzed to make it impossible for outsiders or the public to access private messages [2].

This research paper proposed an extremely secure and efficient algorithm with the combination of symmetric and asymmetric algorithm for both encryption and decryption for distribution of keys. Utilizing both the quick performance of symmetric encryption and the dependable security of asymmetric algorithm encryption, the merging allows users to access data only in ways that are approved [3].

By combining the strengths of AES, ECC, and SHA algorithms, providing a more secure and efficient solution for data encryption. This has the potential to benefit a wide range of applications, from secure communication and financial transactions to medical records and government databases. The increased security provided by our algorithm can help prevent data breaches, financial losses, and reputational damage, ultimately contributing to a safer and more trustworthy digital environment. The AES key is first hashed using SHA-512 to obtain a fixed-length digest, which is then signed using the private key of the ECC key pair to obtain a digital signature [4]. The AES key is then encrypted using the public key of the ECC key pair to obtain a cipher text, which is transmitted to the intended recipient along with the digital signature. The recipient verifies the digital signature and decrypts the cipher text using the private key of the ECC key pair to obtain the original AES key, which is used to encrypt and decrypt the message using the AES algorithm [7].

BACKGROUND AND RELATED WORK

The computer program might act as the starting point for a big new industry [3]. The development of cloud computing has taken many years, starting with the introduction of the first computers. From the centralized mainframe age through the distributed client-server period made possible by the invention of personal computers to the Internet era, in which the business was able to link to the rest of the

world via a global computer network, is a logical progression [6]. There are two different kinds of models used in cloud computing: the deployment model and the service model. There are three different types of service models: infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS). Every type provides a different level of convenience of service.

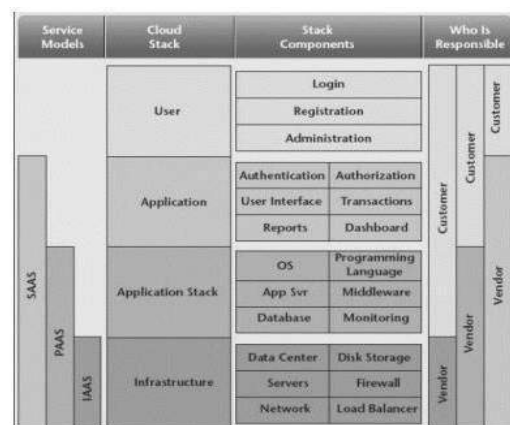


Fig.1. Levels of IaaS, PaaS and SaaS service model in cloud computing [5].

(IaaS) Infrastructure as a Service IaaS is defined as "delivering computer infrastructure (typically a platform virtualization environment) as a service, along with raw storage and networking," by the Cloud Security Alliance (CSA), a cloud security standards organization. The most well-known and established IaaS cloud service provider is Amazon Web Services (AWS) [15]. IaaS, PaaS, and SaaS service model levels in cloud computing are shown in Figure 1 [5]. PaaS (Platform as a Service) According to the CSA, PaaS refers to the provision of a platform and solution stack as a service. PaaS services make it possible to deploy applications without the expense and difficulty of purchasing, administering, and setting up the supporting hardware, software, and hosting resources. Vendors of PaaS take care of the application platform and provide a range of tools to quicken the development cycle. Heroku Examples of established public PaaS products that provide a variety of developer stacks are and Engine Yard [5]. Software as a Service (SaaS) is the term for an application that is provided as a service to the service consumer. The only settings the service user must manage are users

and application-specific parameters. Infrastructure, application logic, and deployments are all handled by the service provider in relation to the supply of the goods or service. Customer relationship management (CRM), enterprise resource planning (ERP), payroll, and accounting software are some of the most popular SaaS apps. The deployment models are now here. In cloud computing, there are three different types of deployment models: public, private, and hybrid.

Public Cloud: company, academic, or governmental institution, or some mix of these, owns, manages, and operates the public can access public cloud infrastructure through the internet. It is present at the location of the cloud provider [5]. Only the amount used is charged to the consumer. Private Cloud: One enterprise with numerous users (such as business units) is the only one allowed to access the cloud infrastructure. It may be located on or off-site and owned, managed, and run by the company, a third party, or a mix of both [11]. Hybrid clouds provide both public and private cloud infrastructure at once. In other words, a hybrid cloud is one where different environments are used by different applications. Almost no one currently only uses the public cloud, thus hybrid. Consequently, hybrid cloud computing strategies are common. Cloud computing environments must be secure, which requires cryptography. As cloud computing becomes more popular, businesses and organizations are moving their critical data and apps there, which raises new security concerns. Fundamentally addressing these problems and safeguarding data from illegal access, theft, and modification is cryptography. The authors talk about the value of encryption in cloud computing and how it may be used to protect sensitive data. The study explores the use of hash functions, symmetric and asymmetric encryption, digital signatures, and other cryptographic methods to protect cloud computing environments.

[13]. The authors also stress the importance of using cryptography to safeguard the confidentiality, accuracy, and usability of data in a cloud computing context.

METHODOLOGY

The following actions are included in the creation of the suggested system to fully automate it while keeping the database integration strategy in mind.

1. The application offers user friendliness with a variety of controls.
2. Technology greatly simplifies and increases the flexibility of overall project management.
3. While project development is ongoing, there is zero chance of data mishandling at any level.

Algorithm: ECC uses a curve defined over a finite field, which has the form $y^2 = x^3 + ax + b$, where a and b are constants. The paper explains that ECC operates on points on the curve, which are represented by pairs (x, y) that satisfy the curve equation.

ECC involves two main operations: point addition and point doubling. Point addition involves adding two points on the curve, while point doubling involves doubling a single point on the curve [16]. These operations are used to perform the following steps in ECC encryption:

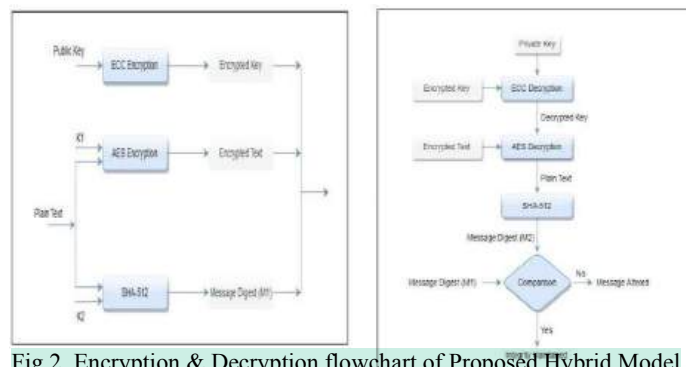


Fig.2. Encryption & Decryption flowchart of Proposed Hybrid Model

1 Key Generation: A private key is generated using random number, and a public key is created by multiplying the private key by a base point on the curve.

2 Encryption: The plain text is represented as a point on the curve, and a random number is generated as an encryption key. The encryption key is used to multiply the base point on the curve, and the resulting point is added to the plain text point to obtain the cipher text point.

3. Decryption: The private key is used to multiply the cipher text point, and the resulting point is subtracted from the encryption key multiplied by the base point to obtain the plaintext point.

Key Generation: Firstly, Receiver input ECC public key, then system will generate ECC public Key. Secondly, the Sender inputs the AES key. Key generation using AES-ECC hybrid cryptography involves using both AES and ECC algorithms to generate a secure key for encryption and decryption. The process of key generation involves the following steps:

1) Choose a random key: The AES algorithm's initial step is to select a random key that will be used for encryption and decryption. The length of the key will depend on the security requirements of the system.

2) Generate an ECC key pair: Making an ECC key pair, which comprises of a private key and a public key, is the next stage. While the public key can be freely transmitted, the private key should be kept a secret and should not be disclosed to anybody.

3) Encrypt the AES key: The AES key is encrypted using the public key generated in step 2, which is an ECC key. This process involves using ECC encryption to encrypt the AES key, resulting in a new encrypted key.

4) Share the encrypted AES key: The encrypted AES key can now be shared with the intended recipient of the message.

5) Decrypt the AES key: The recipient can use their private key, which is part of the ECC key pair generated in step 2, to decrypt the encrypted AES key. This process involves using ECC decryption to recover the original AES key.

6) Use the AES key for encryption and decryption: The recipient can now use the original AES key to encrypt and decrypt messages using the AES algorithm. Given that the AES key is symmetric, the same key is utilized for both encryption and decryption.

The use of both AES and ECC algorithms in key generation provides a high level of security for the system. The AES algorithm provides strong symmetric encryption, while the ECC algorithm provides secure key exchange and encryption of the AES key. The resulting key can be used to encrypt and decrypt messages with a high level of security.

Use of The AES for File Encryption: With its great security for data encryption, the Advanced Encryption Standard (AES) is a popular symmetric encryption method. With a configurable key size of 128, 192, or 256 bits, AES employs constant block sizes of 128 bits. The number of rounds the algorithm does depends on the key size and consists of numerous rounds of operations. The paper explains that the AES algorithm performs the following operations on each block of plain text:

1. **Sub Bytes:** A substitution process where every block's byte is changed for a corresponding one from a predetermined table.

2. **Shift Rows:** This permutation operation cycles through a set number of locations shifting the bytes of each row in the block.

3. **Mix Columns:** An action of transformation where each block's column is multiplied by a fixed matrix and then reduced modulo a polynomial.

4. **Add Round Key:** A bitwise XOR operation in which each byte of the block is combined with a corresponding byte from the key schedule.

Following are the procedures for implementing SHA-512 in an AES-ECC hybrid model: 256-bit AES keys should be chosen at random for encryption and decryption. Make a pair of ECC keys with a private key and a public key. To create a 512-bit fixed-length digest, compute the AES key's SHA-512 hash. To create a digital signature, sign the SHA-512 digest with the private key from the ECC key pair. The signature makes sure that the AES key's integrity is upheld during transmission. To create a cipher text, encrypt the AES key using the ECC key pair's public key. This makes sure that only the intended receiver may access the original plaintext message and decrypt the AES key. Deliver the intended receiver, cipher text and a digital signature are sent. To make sure the AES key hasn't been modified during transmission, the recipient validates the digital signature using the ECC key pair's public key. To get the original AES key, the recipient uses the private key from the ECC key pair to decipher the cipher text. The recipient uses the AES key to encrypt and decrypt the message using the AES algorithm. The hybrid model ensures that the integrity of the AES key is preserved during transmission by employing SHA-512 in this fashion and that only the intended recipient can decrypt the cipher text and access the original plaintext message.

SHA-512 can be used in an AES-ECC hybrid model for key generation and digital signature generation. Following the steps:

1. Hashed value generated from original file.

2. Hashed value generated from received file.

3. Comparing the hash value for integrity checking. The ECC key pair's private key is used to sign the AES key to create a digital signature after the AES key has been hashed using SHA-512 to create a fixed-length digest.

The AES key is then encrypted using the public key of the ECC key pair to obtain a cipher text, which is transmitted to the intended recipient along with the digital signature. The recipient verifies the

digital signature and decrypts the ciphertext using the private key of the ECC key pair to obtain the original AES key, which is used to encrypt and decrypt the message using the AES



1) Encryption Method

- 1st Layer ECC is used with AES encryption.
- Text file entry is encrypted using AES.
- 2nd Layer ECC application to the output text file
- Implementing Hashing and producing Cipher text.

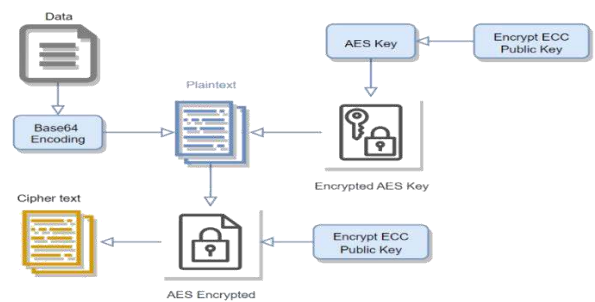


Fig. 3. Encryption Phase

2) Decryption Method

- Use of the ECC secret key in the decryption process.
- AES keys are used to retrieve plaintext.

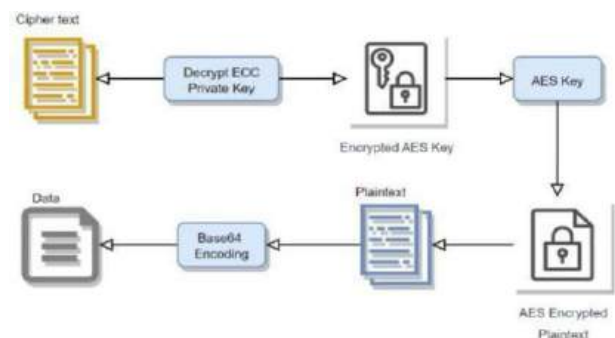


Fig. 4. Decryption Phase

3) Key Generation

1. Receiver input ECC public key, then system will generate ECC public Key

2. Sender input AES key

4) Data Recovery Method

1. File data is decompressed.

2. Reversing the base64 form to its initial state.

COMPARISON, DISCUSSION AND RESULTS

7

Obtained results:

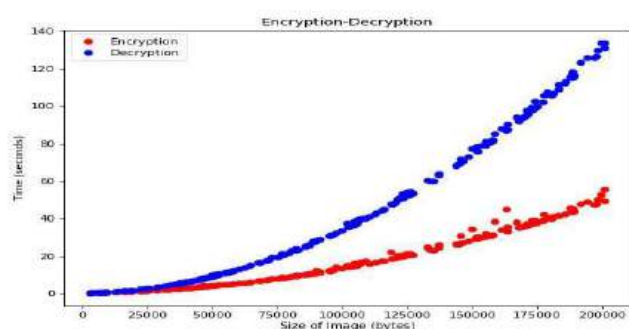


Fig. 5. Image Encryption and Decryption Time

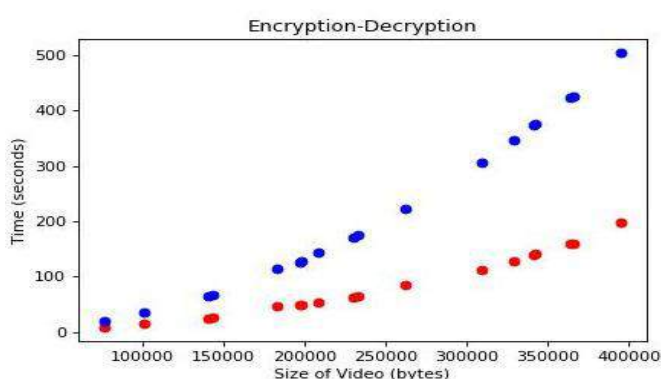


Fig. 6. Video Encryption and Decryption Time

Audio Encryption-Decryption time: The size of the encrypted audio files is represented by the graph with 48441 and 192510 bytes. The smallest audio file (48441 bytes) takes 3.894 seconds to encrypt. This demonstrates that the key and encryption technique

used for this file size are both reasonably effective and can transform the audio information into a safe and unreadable format as soon as possible. This file size's decryption time is 8.721 seconds, which is slower than the encryption time but still rather quick.

The encryption time for the larger audio file (192510 bytes) is 45.313 seconds. This indicates that it takes more time to convert a larger audio file into a safe and secure format because it is significantly longer than the encryption time for the smaller audio file. This file size has a decryption time of 120.068 seconds, which is considerably longer than the encryption time. The smaller video file (76239 bytes) took 8.236 seconds to encrypt. This shows that the key and encryption techniques employed for this file size are reasonably effective and can swiftly convert the video data into a safe and unreadable format. This file size's decryption time is 20.182 seconds, which is slower than the encryption time but still quite quick.

decryption time of 120.068 seconds, which is considerably longer than the decryption time. The smaller video file (76239 bytes) took 8.236 seconds to encrypt. This shows that the key and encryption techniques employed for this file size are reasonably effective and can swiftly convert the video data into a safe and unreadable format. This file size's decryption time is 20.182 seconds, which is slower than the encryption time but still faster one.

video Encryption-Decryption time: The encryption time for the larger video file (101244 bytes) is 14.607 seconds. This indicates that it takes more time to convert a larger video file into a safe and Unreadable format because it is longer than the encryption time for the smaller video file. This file size's decryption time is substantially longer, taking 34.712 seconds, which is a lot longer than the Time to decrypt the smaller file.

In conclusion, it can be seen from the graphs of encryption and decryption durations for image, audio, and video files that larger files often take longer to encrypt and decrypt than smaller files. This is because it takes longer to convert larger files into a safe and unreadable format because doing so requires more processing power and resources.

The size of the file, the encryption algorithm, and the key used all affect how long it takes to encrypt and decode data. Encryption and decryption times for smaller data are typically faster than for bigger files, especially during the decryption process.

Chiefly the operations of encryption and decryption are essential for protecting the security of digital data during transmission and storage.

Comparison Analysis

TABLE1

Existing algorithm Vs proposed algorithm.

Algorithm	Size (In Kb) Encryption + Decryption			Avg. Time
	32	64	128	
AES	6.5 sec	23.6 sec	112.8 sec	47.63 sec
DES	8.2 sec	30.36 sec	126.2 sec	54.92 sec
RSA	27.45 sec	72.3 sec	138.5 sec	79.41 sec
ECC	6.20 sec	26.04 sec	86.23 sec	53.03 sec
Hybrid	9.62 sec	31.70 sec	121.2 sec	54.17 sec

11

TABLE2
Other Hybrid Models vs proposed Model

Message Length (Kb)	Encoding-Decoding Time (seconds)		
	Hybrid Methods		
	AES-RSA	RSA-ECC	AES - ECC
32	12.27 sec	10.31 sec	9.62 sec
64	38.02 sec	35.68 sec	31.70 sec
128	137.12 sec	128.84 sec	121.2 sec

REFERENCES

- [1] R. Latif, L. Abbas, S. Assar and Q. Ali, "Cloud computing risk assessment: a systematic literature review", 2019. Available: <https://hal.archives-ouvertes.fr/hal02397600>. [Accessed 28 July 2022]
- [2] R. Adeeb and H. Mouratidis, "A Dynamic Four-Step Data Security Model for Data in Cloud Computing Based on Cryptography and Steganography", *Sensors*, vol. 22, no. 3, p. 1109, 2022. Available: 10.3390/s22031109.
- [3] S. Kumar, G. Karnani, M. Gaur and A. Mishra, "Cloud Security using Hybrid Cryptography Algorithms", 2021. Available: <https://www.researchgate.net/publication/352145362>.
- [4] Z. Mahmood, *Cloud computing concepts, technology & architecture*. London: Springer, 2013.
- [5] M. Kavis, *Architecting the Cloud: Design Decisions for Cloud Computing Service Models (SaaS, PaaS, and IaaS)*. Hoboken: John Wiley & Sons, Inc., 2014.
- [6] A. Ahmad, M. Abbas and A. Rehman, "Cryptography in Cloud Computing: A Basic Approach to Ensure Security in Cloud," 2015 International Conference on Cloud Computing and Secure Computing (ICCCSC), Dubai, 2015, pp. 269-273, doi: 10.1109/ICCCSC.2015.36.
- [7] M. Y. Ibrahim, M. A. Maarof and A. Z. A. Rahman, "Cloud Computing Service Composition: A Systematic Literature Review," 2016 6th International Conference on Information and Communication Technology for The Muslim World (ICT4M), Jakarta, 2016, pp. 1-6, doi: 10.1109/ICT4M.2016.7853285.
- [8] K. As'habi, A. Vafabakhsh and S. Borji, "DATA TRANSMISSION SECURITY IN CLOUD COMPUTING", *Indian Journal of Fundamental and Applied Life Sciences*, vol. 6, no. 22316345, 2016. Available: <http://www.cibtech.org/sp.ed/jls/2016/01/jls.htm>.
- [9] C. Ponnusamy, S. Padmavathi, R. Swathy and S. Rakesh, "Efficient Data Security Using Hybrid Cryptography on Cloud Computing", 2020. Available: <https://www.researchgate.net/publication/345438033>.
- [10] P. Bindlish, "Optimization of Cryptography Algorithms in Cloud Computing", *International Journal of Computer Trends and Technology*, vol. 46, no. 2, pp. 67-72, 2017. Available: 10.14445/22312803/ijctt-v46p114.
- [11] F. Thabit, A. Alhomdy, A. Al-Ahdal and P. Jagtap, "A new lightweight cryptographic algorithm for enhancing data security in cloud computing", *Global Transitions Proceedings*, vol. 2, no. 1, pp. 91-99, 2021. Available: 10.1016/j.gltp.2021.01.013.
- [12] Y. Li, K. Gai, L. Qiu, M. Qiu and H. Zhao, "Intelligent cryptography approach for secure distributed big data storage in cloud computing", *Information Sciences*, vol. 387, pp. 103-115, 2017. Available: 10.1016/j.ins.2016.09.005.
- [13] M. Sohal and S. Sharma, "BDNA-A DNA inspired symmetric key cryptographic technique to secure cloud computing", *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 1, pp. 1417-1425, 2022. Available: 10.1016/j.jksuci.2018.09.024.
- [14] K. Nicholas, C. Wilson and A. Muthoni, "Enhancing Confidentiality and Integrity in Cloud Computing using RSA Encryption Standard and MD5 Hashing Algorithm", *International Journal of Computer Applications*, vol. 181, no. 14, pp. 23-27, 2018. Available: 10.5120/ijca2018917788.
- [15] H. Abroshan, "A Hybrid Encryption Solution to Improve Cloud Computing Security using Symmetric and

CONCLUSION

By combining these three cryptographic techniques, the algorithm can provide a higher level of security to protect sensitive data. The development of this hybrid algorithm is expected to provide a more robust and reliable method for securing data against unauthorized access. The strength of the project lies in its use of multiple cryptographic techniques, handling different data formats than others. The proposed models focused on developing a hybrid cryptography algorithm using AES, ECC, and SHA to secure various types of data including text, doc, image, audio, and video. The algorithm is designed to be fast and efficient, providing a higher level of security than other hybrid algorithms. Although hybrid models using the AES, ECC, and SHA algorithms are widely deemed as being secure and reliable, the following issues should be considered.

Key management: How the keys are managed has a significant impact on the system's security. The entire system may be vulnerable if the keys are not generated. Using side-channel attacks, such as measuring power usage or electromagnetic emissions, attackers may be able to learn more about the encryption or decryption process & it might not be suitable for systems with limited resources or for encrypting massive volumes of data.

Asymmetric Cryptography Algorithms", International Journal of Advanced Computer Science and Applications, vol. 12, no. 6, 2021. Available: 10.14569/ijacsa.2021.0120604.

- [16] D. D. Doye and E. B. Samson. "Elliptic Curve Cryptography: A Review." International Journal of Computer Applications 181, no. 28 (2018): 13-19.

An Analysis of Bangla Tweets on Social Media Platform for Polarity Detection Using Machine Learning Algorithms

Habibur Rahman, Md. Munam Kazi, Md. Samiul Islam, Jannatul Maoya, Md. Monjurul Arif, Mahir Tagwar, Mahamudul Hasan, Md Sawkat Ali, Taskeed Javid, Maheen Islam

Department Computer Science and Engineering

East West University, Dhaka, Bangladesh

Email: { habibresearchmail, jannatulmaoya42, kazimdmunam0, adnansami54268, monjurularif, munna09bd}@gmail.com, {taskeed, alim, maheen}@ewubd.edu

Abstract— Nowadays, social media is a platform that is gaining popularity. Public and private thinking on a wide range of issues is continually expressed and transmitted via social media. So, the sentiment analysis of those tweets is a topic of the study. Bangla sentiment analysis is one of those research studies. In this regard, this paper proposes a supervised machine learning method for sentiment polarity detection in Bangla social media tweets, whether they are depressive or non-depressive. For analysis, we first preprocessed the data by removing stopwords (punctuation and less important words) from the corpus. For assigning the importance of words, we used TF-IDF feature extraction with N-Grams (Unigram, Bigram, and Trigram) concepts. Eight machine learning algorithms, including logistic regression, decision tree, random forest, K-nearest neighbor, multinomial naive bayes, linear SVM, radial basis function SVM, and stochastic gradient descent, are used to detect polarity in social media tweets. The confusion matrix calculates accuracy, precision, recall, and f1-score. Using Unigram TF-IDF, the highest score achieved by Random Forest (RF) was 99.65%; using Bigram TF-IDF, the highest score achieved by Logistic Regression (LR) was 98.72%; and using Trigram TF-IDF, the highest score achieved by Radial Basis Function SVM (RBFSVM) was 99.19%. This study might make a small addition to Bangla textual analysis's model in natural language processing (NLP).

Keywords: NLP, Sentiment Analysis, Bangla Social media Tweets, Stopwords, Supervised, N-gram, TF-IDF.

I. INTRODUCTION

The present world is an era of the internet. As time goes by, people are getting used to the internet. As technologies are upgraded with the internet, people are getting closer to several social media platforms. Nowadays, social media is the center for expressing the thoughts, opinions, emotions, feelings, etc. of people. Thus, the reliability of social media is increasing day by day. Different people have different emotions. They express their various emotions on social media platforms such as Facebook, Twitter, Instagram, and YouTube in various ways. We are easily influenced by those social media comments, posts, etc. Like people who underestimate others by replying to bad comments or posting different quotes. Sometimes it creates rumors in societies [1]. Thus, people are getting depressed by the influence of social media platforms. Sometimes it leads people to commit horrible acts like suicide, murder, etc. So, it has been a significant research topic. This research is done with the help of the Natural Language Processing (NLP) concept based on the textual expression expressed on social media [2]. NLP is a vast area that works on the analysis of sentiment classification of text, virtual assistants and chatbots, extracts of text, machine translation, summarization of text, market intelligence, and auto-

correction. NLP includes two sections: natural language understanding (NLU) and natural language generation (NLG) [3][4].

Natural Language Understanding (NLU) is a form of artificial intelligence that deciphers text or audio input using computer software. It understands human language and helps the computer to understand commands that do not follow a strict grammar[5]. Siri is an NLU program that seeks to enhance human-computer interaction and aid in the development of chatbots capable of interacting with humans autonomously [3]. Semantic analysis is a branch of NLP that analyzes the context, logical sentence structure, and grammatical responsibilities of a document to determine its meaning. There are two categories: lexical semantic analysis and compositional semantic analysis [6][4]. Sentiment analysis is a technique that focuses on emotions and polarity to determine the sentiment of a text. It employs rule-based, automatic, and hybrid natural language processing methods [7]. Types of sentiment analysis include sentiment analysis at a finer level, emotion detection, element-based sentiment analysis, and multilingual sentiment analysis.

Natural Language Generation (NLG) is a branch of AI that specializes in converting data into English. It is an intelligent system that executes rules-based operations using pre-existing templates and represents significant concepts in data in a more simple, relevant, and understandable manner. Tokenization is the process of separating a paragraph or sentence into its constituent words [3]. Text cleaning eliminates words and objects from a corpus of text data; and POS tagging designates each word in a phrase with its appropriate part of speech [8][3].

NLG helps businesses automate mundane processes and do communication analysis, boosting productivity and allowing human resources to be redirected to higher-value tasks. Stopwords are terms that do not add much to the document's meaning. NLTK offers stopwords in a variety of languages. Stopword elimination reduces the data set size and boosts performance. Lemmatization decreases the number of inflected words while also guaranteeing that the root word is a part of the language.

This research paper proposes a model to detect sentiment and emotion in Bangla social media platform tweets. It applies a model of TF-IDF, an N-gram featuring a concept, and eight machine learning models: Logistic Decision Tree (DT), Logistic Regression (LR), Random Forest (RF), Multinomial Naive Bayes (MNB), K-Nearest Neighbour (KNN), Linear SVM (LSVM), Radial Basis Function SVM (RBFSVM), and Stochastic Gradient Descent (SGD) to extract the emotion of

Bangla social media platform tweets, whether they are depressing or non-depressing [14] [15]. The objectives of the research are to propose a model that can take raw Bangla text (social media platform tweets) as input and detect the sentiment of that textual data using machine learning algorithms with higher accuracy. The contribution of the research is that different business organizations analyze the sentiment of their product reviews to develop their product quality, etc. Our research is all about designing a model for detecting sentiment from Bangla social media platform tweets, whether it is depressing or non-depressing.

II. LITERATURE REVIEW

Sentiment analysis Auto-detection is an important NLP research area. These methods usually use machine learning and deep learning. Since we integrated machine learning, there is a brief explanation. There are several machine-learning methods. These include supervised, semi-supervised, unsupervised, and reinforcement machine learning [17] [18].

Machine learning relies on supervised learning to train algorithms with labelled data. This method requires precise tagging of data and works well with supervised learning. Machine learning under supervised learning uses a limited dataset to comprehend the problem, solve it, and indicate the data to be dealt with. Finally, correlations relate variables to datasets to demonstrate cause and effect. Despite implementation, supervised machine learning can discover new patterns and connections.

The algorithm for unsupervised machine learning can process unlabelled data without human intervention. It can identify data connections, but without labels, it cannot create hidden structures. Despite predefining, the algorithm is able to resolve problem statements and change hidden structures. This approach permits post-deployment development, unlike supervised learning.

III. RELATED WORKS

Natural Language Processing (NLP) is a field of study concerned with sentiment analysis, text classification, chatbots and virtual assistants, text extraction, machine translation, text summarization, market intelligence, and text auto-correction. By analyzing the sentiment of their evaluations, businesses and other organizations are improving their services and quality.

Bengali is ranked among the top four languages in the world and has almost 250 million native speakers. A huge amount of research has been done on this topic with various types of language [11]. [4]. The Naive Bayes continuous learning framework has been used to analyze the sentiment of e-commerce product reviews [9]. Deep learning concepts are used in text analysis. K-Mean, TF-IDF vectorizer, and SVM concepts have been used to determine the emotion of Amazon food reviews [19].

The polarity detection method of sentiment on Bengali book reviews is such one. Machine learning models using TF-IDF and a count vectorizer have been used to detect the sentiment of Bangla newspapers. Six supervised ML classifiers (i.e., Multinomial Naive Bayes (MNB), K-Nearest Neighbor (K-NN), Random Forest (RF), Decision Tree (DT), Logistic Regression (LR), and Linear Support Vector Machine (LSVM)) are used to analyze the best classifier.

Naive Bayes is popular for the classification of sentiments from text data using the probability of each emotion class [13] [16]. As there is a lack of resources, there is little research on Bangla sentiment analysis. Therefore, this research paper proposes to detect sentiment and emotion from Bangla social media tweets.

IV. PROPOSED METHOD

A. Problem Study:

The proposed model aims to predict the sentiment of a given social media Bangla tweet. It takes Bangla tweets as input and pre-processes them to extract features. Classification models are trained, and the sentiment is detected using the selected testing model. For pre-processing, a stopwords list is created and stopwords are removed. For feature extraction, the N-gram feature and the TF-IDF concept are used.

After extracting the feature, the data is trained on eight machine learning classification algorithms: decision tree (DT), logistic regression (LR), random forest (RF), multinomial naive bayes (MNB), K-nearest neighbor (KNN), linear SVM (LSVM), radial basis function SVM (RBF SVM), and stochastic gradient descent (SGD). After being trained, the sentiment is detected, whether it is depressive or non-depressive. The entire process is illustrated in Fig. 1.

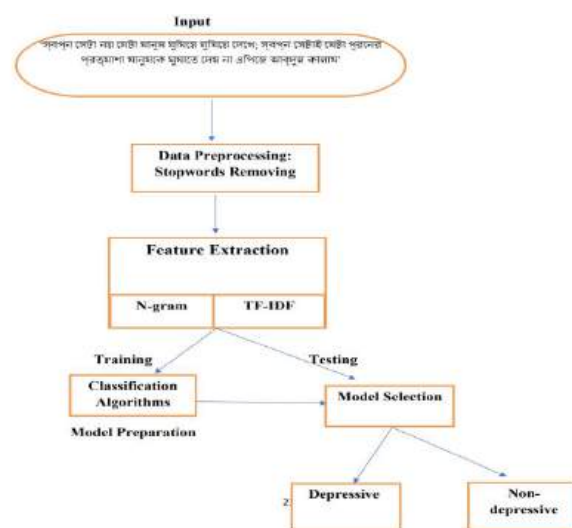


Fig. 1. System flow

B. Dimensionality Reduction:

Dimension reduction is an important part of data analysis, as it minimizes the dimensionality of a strong-dimensional data set by casting it into a subordinate-dimensional domain. However, too many features can make it difficult to propose a training set and then take action on it. This is divided into two parts: selection of features and feature extraction.

C. Feature Extraction:

The process of converting raw data into numerical system characteristics while preserving the details of the primary data set is known as feature extraction. Compared to directly using machine learning on raw data, it yields better results.

There are three types of feature extraction methods: 1) Filter Method: 2) Embedding Method 3) Wrapper Method. In

our model, the embedding method has been used for dimension reduction.

Decision Tree(DT):

The decision tree is a powerful and extensively utilized classification and forecasting tool. It begins with the leaf node and is computed by analyzing the attributes of each node and advancing the entire branch of the tree to the attribute values [20]. In a decision tree, subtrees and novel nodes are processed in the same way as before in order to acquire information [21].

Logistic Regression (LR):

Logistic regression is a classification technique. Instead of using a linear function, it employs a more complex cost function called the "sigmoid function" or "logistic function". It can quickly discover the most efficient categorization factors and forecast two maximum values, (0 or 1) [22]. Because the logistic regression (LR) hypothesis restricts the cost function to a value between 0 and 1, linear functions cannot describe it.

$$y: 0 \leq h_{\theta}(x) \leq 1 \quad \dots \dots (1)$$

Random Forest (RF):

Random Forest is a regression and classification expression algorithm. It is the merging of many decision trees based on a diverse data collection. To develop accuracy, it needs the highest number of decision trees. It takes data set as training and creates a decision tree with the selected data point. Finally, it predicts the new data point and puts it in all categories by majority of counting.

The most important details are that the algorithm takes data set as training, creates a decision tree with the selected data point, repeats this process for the number of decision trees needed, and predicts the new data point by majority of counting.

K-Nearest Neighbour (KNN):

The supervised learning approach is used by K-Nearest Neighbours, a machine learning algorithm. It makes an assumption of similarity between the latest and still remaining cases and puts the latest cases in the same category as existing categories. All existing data is maintained by this algorithm, and it categorizes the latest data points based on parallelism. The K-NN technique is utilized for regression and classification tasks, and it operates in a non-parametric manner, which means it disregards the underlying data.

K-NN works by taking K as the number of neighbours and finding the Euclidean distance of K. Take the K nearest neighbours for each Euclidean distance and add the number of data elements in each category. This new data is put into each category to maximize the Euclidean distance..

Multinomial Naive Bayes (MNB):

Spontaneous text classification is becoming increasingly important due to the growing use of electronic documents. A supervised learning technique assigns class labels from a fixed or predetermined class to each new document. This process differs from how we normally communicate with someone, where each message consists of a random handful of words [23]. Large documents typically contain a large number of words and are characterized by a high-dimensional feature space.

Linear SVM(LSVM):

The Linear Support Vector Machine (LSVM) is an algorithm for machine learning that solves classification and regression issues. Its purpose is to investigate the decision boundary for classifying n-dimensional space so that new data can be classified further. The optimal boundary is a simple hyperplane, which takes the nearest data point of a hyperplane. This is called a support vector, and the method is a linear support vector machine.

Radial Basis Function SVM(RBFSVM):

This algorithm calculates the distance between two points X_1, X_2 . It determines how close X_1 and X_2 are to one another. This algorithm is a kind of mathematical function as follows: where, 1) ' σ ' represents the variance and the hyperparameter. 2) $\|x_1 - x_2\|$ represents the Euclidean distance between x_1 and x_2 .

Dataset Summary Visualization:

This project aims to classify the sentiment of a review as depressive or non-depressive. A Bengali social media tweet corpus has been created with 5880 tweets, with 3016 reviews annotated as depressive and 2864 as non-depressive. There is no null value in the dataset, with depressive reviews annotated as 1 and non-depressive reviews labeled as 0.

```
Total Reviews: 5880
Total Depressive Reviews: 3016
Total Non-depressive Reviews: 2864

Class Name : Depressive
Number of Documents: 3016
Number of Words: 54832
Number of Unique Words: 5059

Class Name : Non-depressive
Number of Documents: 2864
Number of Words: 53108
Number of Unique Words: 4781
```

Fig.2. Data Summary

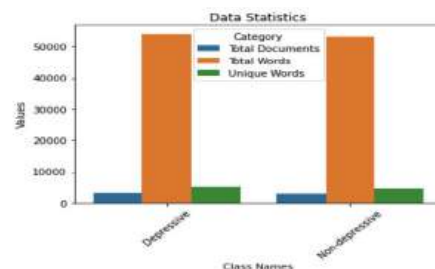


Fig.3. Data Statistics

V. IMPLEMENTATION DETAILS

Preprocessing of Data:

Data initialization is essential for data analysis, but Bangla text does not require text lowering. Stopwords, which do not have significant importance to form a sentence, have been removed to clean the corpus.

এই, করে, জন্য, একটি, আমার, একটা, এর, যে, তার, এবং, থেকে, ও, এ, কি, কোন, আমি, আর

Fig.4. Stopwords

We have also removed the punctuation marks such as (!, (), [], {}, "", '\', <, >, ., /, @, #, \$, %, ^, &, * _ ~ ' — ||" %%, etc.)

Original:
এই পৃথিবীর সবচাইতে ধৈর্যশীল মানুষ হচ্ছে আমাদের মা ! একমত ?
Cleaned:
পৃথিবীর সবচাইতে ধৈর্যশীল মানুষ হচ্ছে আমাদের মা একমত
Sentiment:-- Non-depressive

Fig.5. Pre-processed Sample

TF-IDF:

TF-IDF is a system of summing up a word's number in an ensemble of documents to indicate its connection to the corpus and document. It is utilized for information extraction and text analysis by dividing two metrics (number of occurrences of a word in a document and inverse word-document frequency).

TF-IDF Weight Updating:

For a term i in document j :

$$W_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right) \quad \dots \dots (2)$$

$tf_{i,j}$ = number of occurrences of i in j

df_i = number of documents containing i

N = total number of documents

Shape of TF-IDF Corpus =====> (5880, 7826)

Sample Review: যত ধৈর্য তত বেশি সফলতা। সমস্যা হলো আমাদের যৌথধারন ক্ষমতা কতটা বেশি
tfidf
গোছের 0.354605
যৌথধারন 0.354605
তুলো 0.354605
ধৈর্য 0.327355
তত 0.310510
সফলতা 0.310510
সমস্যা 0.268207
ক্ষমতা 0.258395
যত 0.258346
বেশি 0.212677
হলো 0.204005
আমাদের 0.185864

Fig.6. Data Sample with TF-IDF

N-gram technique is an adjacent sequence of 'N' items in each sequence of input text, which can be a character, word, or phrase. It has been used with TF-IDF in our model.

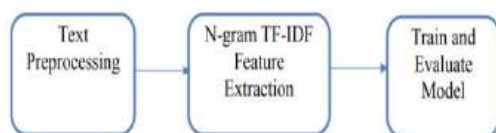


Fig.7. N-gram technique with TF-IDF.

Unigram Feature with TF-IDF:

If $N=1$, then it is referred to as a unigram where an adjacent sequence of 1 item in each sequence of input text.

```

(0, 4435) 0.17430715161542326
(0, 6165) 0.29531174774486896
(0, 7448) 0.08868336244648874
(0, 7429) 0.11107596129012143
(0, 5229) 0.14269155645277234
(0, 4800) 0.15384195423506883
(0, 3669) 0.10862386954343956
:
(5878, 6907) 0.2873335262136968
(5878, 1519) 0.17996822229847165
(5878, 7404) 0.22784371304835338
(5879, 3852) 0.19753731789211515
(5879, 4726) 0.19753731789211515
(5879, 862) 0.19753731789211515
(5879, 5197) 0.19753731789211515
  
```

Fig.8. Unigram TF-IDF Feature Vector

Bigram Feature with TF-IDF:

If $N=2$, then it is referred to as a bigram where an adjacent sequence of 2 items in each sequence of input text.

```

(0, 13273) 0.12551625123132854
(0, 15707) 0.12551625123132854
(0, 7116) 0.12551625123132854
(0, 15409) 0.12551625123132854
(0, 21021) 0.12551625123132854
(0, 25183) 0.12551625123132854
(0, 25016) 0.10990844948862505
:
(5879, 8989) 0.13823075572871493
(5879, 1299) 0.13823075572871493
(5879, 13652) 0.13823075572871493
(5879, 16191) 0.13823075572871493
(5879, 2784) 0.13823075572871493
(5879, 17662) 0.13823075572871493
(5879, 13712) 0.13823075572871493
  
```

Fig.9. Bigram TF-IDF Feature Vector

Dataset Split:

For learning and prediction, we have separated the dataset into test and training portions. The training set has 70% and the test set has 30% of the dataset.

Machine Learning Models:

Eight machine learning models, including Decision Tree, Logistic Regression, Random Forest, Multinomial Naive Bayes, K-Nearest Neighbor, Linear SVM, Radial Basis SVM, and Stochastic Gradient Descent, have been utilized to train sentiment analysis of Bangla tweets.

VI. RESULT ANALYSIS AND DISCUSSION

This research aims to identify the best supervised techniques for sentiment classification in natural language processing, using accuracy, F1-score, precision, and recall to interpret performance measurement.

Performance using Unigram TF-IDF:

	Accuracy	Precision	Recall	F1 Score
LR	96.83	96.75	96.75	96.75
DT	98.19	97.70	98.61	98.15
RF	98.70	97.72	99.65	98.68
MNB	97.62	98.23	96.86	97.54
KNN	95.86	94.67	96.98	95.81
Linear SVM	93.37	94.29	91.99	93.12
RBF SVM	95.41	96.21	94.31	95.25
SGD	98.30	98.26	98.26	98.26

Fig.10. Unigram TF-IDF Performance

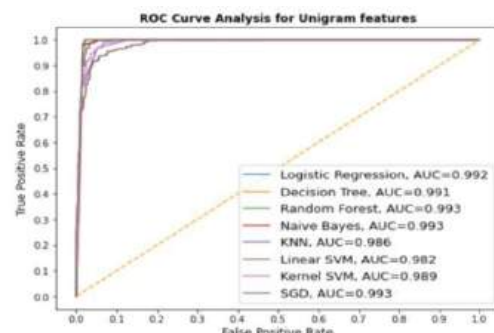
Receiver Operating Characteristic (ROC) Curve:

Fig.11. Unigram TF-IDF ROC Curve

Highest Accuracy with Unigram TF-IDF:

Highest Accuracy achieved by RF at = 98.7
 Highest F1-Score achieved by RF at = 98.68
 Highest Precision Score achieved by SGD at = 98.26
 Highest Recall Score achieved by RF at = 99.65

Fig.12. Unigram Best Performance

Performance using Bigram TF-IDF:

	Accuracy	Precision	Recall	F1 Score
LR	98.53	98.27	98.72	98.49
DT	97.90	97.69	98.03	97.86
RF	98.53	98.27	98.72	98.49
MNB	98.19	97.70	98.61	98.15
KNN	95.18	98.26	91.75	94.89
Linear SVM	98.07	97.80	98.26	98.03
RBF SVM	97.96	97.80	98.03	97.91
SGD	98.30	98.26	98.26	98.26

Fig.13. Bigram TF-IDF Performance

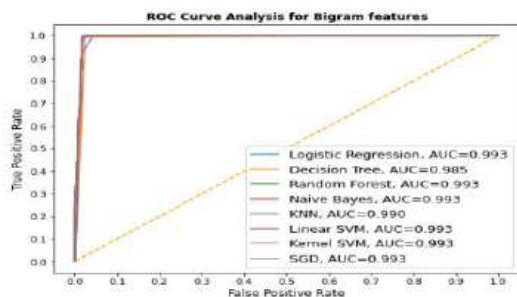
Receiver Operating Characteristic (ROC) Curve:

Fig.14. Bigram TF-IDF ROC Curve

Highest Accuracy with Bigram TF-IDF:

Highest Accuracy achieved by LR at = 98.53
 Highest F1-Score achieved by LR at = 98.49
 Highest Precision Score achieved by LR at = 98.27
 Highest Recall Score achieved by LR at = 98.72

Fig.15. Bigram Best Performance

Performance using Trigram TF-IDF:

	Accuracy	Precision	Recall	F1 Score
LR	98.30	98.26	98.26	98.26
DT	98.19	98.25	98.03	98.14
RF	98.53	98.27	98.72	98.49
MNB	98.19	97.70	98.61	98.15
KNN	94.67	98.24	90.71	94.32
Linear SVM	98.24	98.26	98.14	98.20
RBF SVM	98.75	98.27	99.19	98.73
SGD	98.64	98.27	98.95	98.61

Figure.16. Trigram TF-IDF Performance

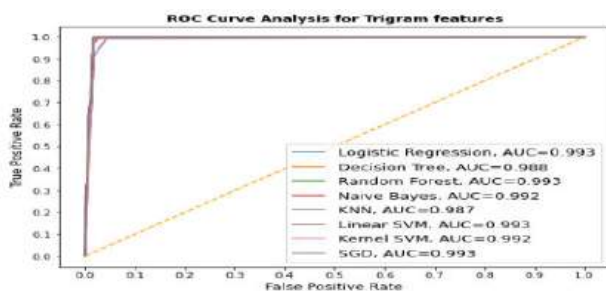
Receiver Operating Characteristic (ROC) Curve:

Figure.17. Trigram TF-IDF ROC Curve

Highest Accuracy with Trigram TF-IDF:

Highest Accuracy achieved by RBF SVM at = 98.75
 Highest F1-Score achieved by RBF SVM at = 98.72999999999999
 Highest Precision Score achieved by RF at = 98.27
 Highest Recall Score achieved by RBF SVM at = 99.19

Figure.18. Trigram Best Performance

Result Comparison:

	Unigram TF-IDF				Bigram TF-IDF				Trigram TF-IDF			
	Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score
LR	98.83	96.75	96.75	96.75	98.53	98.27	98.72	98.49	98.30	98.26	98.26	98.26
DT	98.19	97.70	98.61	98.1	97.90	97.69	98.03	97.86	98.19	98.25	98.03	98.14
RF	98.70	97.72	99.65	98.68	98.53	98.27	98.72	98.49	98.53	98.27	98.72	98.49
MNB	97.62	98.23	96.86	97.5	98.19	97.70	98.61	98.15	98.19	97.70	98.61	98.15
KNN	95.86	94.67	96.98	95.8	95.18	98.26	91.75	94.89	94.67	98.24	90.71	94.32
Linear SVM	93.37	94.29	91.99	93.1	98.07	97.80	98.26	98.03	98.24	98.26	98.14	98.20
RBF SVM	95.41	96.21	94.31	95.2	97.96	97.80	98.03	97.9	98.75	98.27	99.19	98.73
SGD	98.30	98.26	98.26	98.26	98.30	98.26	98.26	98.26	98.64	98.27	98.95	98.61

Table 1: Performance Table

Unigram TF-IDF:

Random Forest achieved the highest accuracy of 98.70% for unigram TF-IDF, while stochastic gradient descent had the highest precision of 98.26%. Random Forest had the highest recall of 99.65% and the highest F-1 score of 98.68%. Overall, Random Forest had the highest majority score, with a maximum score of 99.65%. Linear SVM had lower accuracy due to difficulty classifying hyperlines.

Bigram TF-IDF:

Logistic regression attained the highest accuracy, precision, recall, and F-1 score for Bigram TF-IDF with 98.53% accuracy, 98.27% precision, 98.72% recall, and 98.49% F-1 score. KNN performed poorly with a minimum recall score of 91.75% since two items in each sequence of input text had identical distances from their neighbor clusters.

Trigram TF-IDF:

Radial Basis Function SVM achieved the highest accuracy of 98.75% for unigram TF-IDF, while Random Forest achieved the highest precision of 98.27%. The Radial Basis Function SVM had the highest recall at 99.19% and the highest F-1 score at 98.73%. Overall, Radial Basis Function SVM had the highest majority score of 99.65%, while Trigram also achieved the minimum recall of 90.71%.

VII. CONCLUSION

This research focuses on the use of multiple sentiment analyzers in conjunction with machine learning algorithms to explore the approach with the highest rate of accuracy for gaining knowledge about sentiment detection. Eight machine learning models—Decision Tree (DT), Radial Basis Function SVM (RBFSVM), Linear SVM (LSVM), Multinomial Naive Bayes (MNB), Logistic Regression (LR), Random Forest (RF), and Stochastic Gradient Descent

(SGD)—have been proposed to learn a machine to predict the depressive and non-depressive sentiment from Bangla social media tweets. Primary preprocessing has been implemented by removing less meaningful words from the data corpus. For better textual analysis, N-gram TF-IDF feature extraction methods have been applied. For performance measurement accuracy, the f1-score, precision, and recall have been calculated from the confusion matrix. In terms of recall, we have the best performance for the Random Forest algorithm. Since sentiment detection leads to improvement in different sections of quality or services, Bangla sentiment analysis could add a new dimension to the artificial intelligence area.

REFERENCES

- [1] K. I. Islam, S. Kar, M. S. Islam, and M. R. Amin, "SentNoB: A Dataset for Analysing Sentiment on Noisy Bangla Texts," pp. 3265–3271, 2021, doi: 10.18653/v1/2021.findings-emnlp.278.
- [2] N. R. Bhowmik, M. Arifuzzaman, and M. R. H. Mondal, "Sentiment analysis on Bangla text using extended lexicon dictionary and deep learning algorithms," *Array*, vol. 13, p. 100123, 2022, doi: 10.1016/j.array.2021.100123.
- [3] C. Cardie, "Machine Learning for Natural Language Processing (and Vice Versa?)," no. May, pp. 2–2, 2005, doi: 10.1007/11564126_2.
- [4] S. J. Maisha, N. Nafisa, and A. K. Muhammad Masum, "Supervised Machine Learning Algorithms for Sentiment Analysis of Bangla Newspaper," *Int. J. Innov. Comput.*, vol. 11, no. 2, pp. 15–23, 2021, doi: 10.11113/ijic.v11n2.321.
- [5] S. Sharmin and D. Chakma, "Attention-based convolutional neural network for Bangla sentiment analysis," *AI Soc.*, vol. 36, no. 1, pp. 381–396, 2021, doi: 10.1007/s00146-020-01011-0.
- [6] L. W. Ku and C. W. Sun, "Calculating emotional score of words for user emotion detection in messenger logs," *Proc. 2012 IEEE 13th Int. Conf. Inf. Reuse Integr. IRI 2012*, pp. 138–143, 2012, doi: 10.1109/IRI.2012.6303002.
- [7] W. Li, L. Zhu, Y. Shi, K. Guo, and E. Cambria, "User reviews: Sentiment analysis using lexicon integrated two-channel CNN–LSTM family models," *Appl. Soft Comput. J.*, vol. 94, p. 106435, 2020, doi: 10.1016/j.asoc.2020.106435.
- [8] Rahul, S. Adhikar, and Monika, "NLP based Machine Learning Approaches for Text Summarization," *Proc. 4th Int. Conf. Comput. Methodol. Commun. ICCMC 2020*, no. Iccmc, pp. 535–538, 2020, doi: 10.1109/ICCMC48092.2020.ICCMC-00099.
- [9] F. Xu, Z. Pan, and R. Xia, "E-commerce product review sentiment classification based on a naïve Bayes continuous learning framework," *Inf. Process. Manag.*, vol. 57, no. 5, p. 102221, 2020, doi: 10.1016/j.ipm.2020.102221.
- [10] N. C. Dang, M. N. Moreno-García, and F. De la Prieta, "Sentiment analysis based on deep learning: A comparative study," *Electron.*, vol. 9, no. 3, 2020, doi: 10.3390/electronics9030483.
- [11] D. S. Sachan, M. Zaheer, and R. Salakhutdinov, "Revisiting LSTM networks for semi-supervised text classification via mixed objective function," *33rd AAAI Conf. Artif. Intell. AAAI 2019, 31st Innov. Appl. Artif. Intell. Conf. IAAI 2019 9th AAAI Symp. Educ. Adv. Artif. Intell. EAAI 2019*, pp. 6940–6948, 2019, doi: 10.1609/aaai.v33i01.33016940.
- [12] A. Onan, "Sentiment analysis on product reviews based on weighted word embeddings and deep neural networks," *Concurr. Comput. Pract. Exp.*, vol. 33, no. 23, pp. 1–12, 2021, doi: 10.1002/cpe.5909.
- [13] R. A. Tuhin, B. K. Paul, F. Nawrine, M. Akter, and A. K. Das, "An automated system of sentiment analysis from Bangla text using supervised learning techniques," *2019 IEEE 4th Int. Conf. Comput. Commun. Syst. ICCCS 2019*, pp. 360–364, 2019, doi: 10.1109/CCOMS.2019.8821658.
- [14] B. S. Rintyarna, R. Sarno, and C. Fatichah, "Enhancing the performance of sentiment analysis task on product reviews by handling both local and global context," *Int. J. Inf. Decis. Sci.*, vol. 12, no. 1, pp. 75–101, 2020, doi: 10.1504/IJIDS.2020.104992.
- [15] X. Liu, M. Niethammer, R. Kwitt, N. Singh, M. McCormick, and S. Aylward, "Low-Rank Atlas Image Analyses in the Presence of Pathologies," *IEEE Trans. Med. Imaging*, vol. 34, no. 12, pp. 2583–2591, 2015, doi: 10.1109/TMI.2015.2448556.
- [16] S. Robin, M. Sawan, J. F. Harvey, M. Abdel-Gawad, T. M. Abdel-Baky, and M. M. Elhilali, "New implantable microstimulator dedicated to selective stimulation of the bladder," *Annu. Int. Conf. IEEE Eng. Med. Biol. - Proc.*, vol. 4, no. 1, pp. 1792–1795, 1997, doi: 10.1109/iembs.1997.757075.
- [17] M. E. Mowlaei, M. Saniee Abadeh, and H. Keshavarz, "Aspect-based sentiment analysis using adaptive aspect-based lexicons," *Expert Syst. Appl.*, vol. 148, p. 113234, 2020, doi: 10.1016/j.eswa.2020.113234.
- [18] H. M. Ahmed, M. J. Awan, N. S. Khan, A. Yasin, H. Muhammad, and F. Shehzad, "Sentiment analysis of online food reviews using big data analytics," vol. 20, no. 2, pp. 827–836, 2021, doi: 10.17051/ilkonline.2021.02.93.
- [19] V. Bhati and J. Kher, "Survey for Amazon Fine Food Reviews," pp. 601–603, 2019.
- [20] B. R. Patel and K. K. Rana, "A Survey on Decision Tree Algorithm for Classification," *Ijedr*, vol. 2, no. 1, pp. 1–5, 2014.
- [21] D. Singh and S. Malhotra, "Intra News Category Classification using N-gram TF- IDF Features and Decision Tree Classifier," *Ijsart*, vol. 4, no. 3, pp. 508–514, 2018.
- [22] S. Nusinovic et al., "Logistic regression was as good as machine learning for predicting major chronic diseases," *J. Clin. Epidemiol.*, vol. 122, pp. 56–69, 2020, doi: 10.1016/j.jclinepi.2020.03.002.
- [23] A. M. Kibriya, E. Frank, B. Pfahringer, and G. Holmes, "Multinomial naive bayes for text categorization revisited," *Lect. Notes Artif. Intell. (Subseries Lect. Notes Comput. Sci.)*, vol. 3339, pp. 488–499, 2004, doi: 10.1007/978-3-540-30549-1_43.

Heart Disease Prediction Using Machine Learning Algorithms

1st KAZI MOSTAQ HRIDOY

*Computer Science and
Engineering*

East West University
Dhaka, Bangladesh

mostafahridoy778@gmail.com

2nd SAIFUL ISLAM AKASH

*Computer Science and
Engineering*

East West University
Dhaka, Bangladesh

saifulakash1122@gmail.com

3rd FAHMIDA AFROSE DIPTI

*Computer Science and
Engineering*

East West University
Dhaka, Bangladesh

fahmidadipti75@gmail.com

4th MAHAMUDUL HASAN

*Computer Science &
Engineering*

East West University,
Dhaka, Bangladesh

munna09bd@gmail.com

5th JESAN AHAMMED OVI

*Computer Science &
Engineering*

East West University,
Dhaka, Bangladesh

jesan@ewubd.edu

6th MD AL-IMRAN

*Computer Science &
Engineering*

East West University,
Dhaka, Bangladesh

al.imran@ewubd.edu

7th Dr. SARWAR JAHAN

*Computer Science &
Engineering*

East West University,
Dhaka, Bangladesh

sjahan@ewubd.edu

Abstract— The cardiovascular system plays a vital role in all living organisms, responsible for circulating blood throughout the body, delivering essential oxygen and nutrients to cells, and eliminating waste products. Predicting cardiovascular diseases holds significant importance in clinical data analysis. Early detection measures have proven valuable in making critical decisions for high-risk patients, leading to primary prevention efforts. Addressing this challenge necessitates the development of a disease prediction system. Given the abundance of medical information, the healthcare system relies on machine learning algorithms to make reliable decisions in cardiovascular prediction. These algorithms analyze the data to predict the occurrence of cardiac failure. To predict coronary illness, this study processes the data. It assesses the accuracy of a number of machine learning techniques, including Naive Bayes, Random Forests, Decision Trees, Logistic Regression, Support Vector Machines, K-Nearest Neighbors, and Gradient Boosting.

Index Terms— Label-Encoding, Standardization, Support Vector Machine, Logistic Regression, Precision, Recall.

I. INTRODUCTION

The importance of the cardiovascular system to human health cannot be overstated, yet problems develop when there is inadequate blood flow, which leads to cardiovascular failure. The intricacy of cardiovascular illnesses and the limits of current techniques make diagnosis difficult. Accurate sickness prediction algorithms are required to support existing diagnostic techniques to solve this.

Rising healthcare expenses and the burden of diseases throughout the world are caused by variables including genetics and changing lifestyles. One such example is the nearly 12

million deaths caused by coronary artery disease each year [1]. Researchers are using data, such as WHO statistics, to generate insights to counter this. To fully comprehend the cardiovascular system, however, collaboration between several medical professions is required due to its complex design. The treatment of cardiovascular health may be completely transformed by technological innovations like non-invasive imaging and wearable technologies paired with AI. The effect of cardiovascular illnesses on both people and societies can be lessened because of these breakthroughs, which hold out hope for improving disease prediction accuracy and individualized treatment approaches.

Our findings highlight Random Forest's remarkable potential to accurately forecast cardiovascular events. This ensemble learning approach combines many decision trees to improve accuracy. Its attractiveness is increased by how easy it is to use and how widely it is used [2]. To determine the best algorithm for predicting cardiovascular events, we examined seven others. Because it is effective at handling complicated data conversations and reducing overfitting, Random Forest excelled in this study. When processing complex cardiovascular data, this is essential. We gained important understandings into the patterns underlying these occurrences by studying a wide range of datasets.

The results of our testing, which included cross-validating, validated the reliability of Random Forest predictions. By identifying high-risk patients and adapting treatment options, this technique has the potential to revolutionize cardiovascular treatment and lessen the burden of cardiovascular diseases.

II. RELATED WORK

As the central organ of the cardiovascular system, the circulatory system in humans is in charge of distributing blood. For diagnosing, forecasting, and treating heart problems, regular cardiovascular examinations are crucial. The study's use of machine learning techniques has been crucial in this situation. Given the complexity of the cardiovascular system, managing it is essential to avert potentially fatal circumstances. Many researchers have used machine learning approaches to find cases of heart disease.

Patel et al. employed the WEKA method to assess Decision Tree categorization, enhancing performance [3]. They analyzed decision tree techniques using datasets from the Cleveland cardiovascular disease database in the UCI repository aiming to uncover trends and predict disease onset.

In the study "Prediction of heart disease using K-means and artificial neural network as a hybrid approach to improve accuracy" by Kamath, A., Kadam, and Malav (2017) [4], the focus is on enhancing accuracy. The project aims to develop an application that predicts heart issues more accurately, utilizing neural network backpropagation and the K-means algorithm for attribute organization.

To assess the accuracy of the K-mean clustering approach, Bala Sundar V et al. studied actual and synthetic datasets used to predict the diagnosis of cardiac disorders [5].

According to Sayali D. Jadhav et al., the problem should be viewed in terms of the relationship between an assortment of variables associated with features and an outcome variable of significance [6].

The "HDPS: Heart disease prediction system" [7], which was introduced in 2011 by Chen, A.H. et al., uses an artificial neural network algorithm to classify heart disease based on medical criteria. These clinical requirements are used by this algorithm to reliably determine if a person has cardiovascular difficulty.

Thomas, J. and Princy, R.T. (2016) carried out the "Human heart disease prediction system using data mining" [8] study, which investigates various classifying mechanisms to predict individual risk levels based on factors like age, gender, hypertension, fat, and heart rate, revealing that including additional variables enhances prediction accuracy.

In the study titled "Cardiovascular Disease Prediction Using Data Mining Techniques: A Survey" [9], Kaur and Arora examine a variety of classification methods, including Support Vector Machine, K-NN classifier, Naive Bayesian, Decision Tree, Blended Approach, and Artificial Neural Network, to shed light on their potential to improve cardiovascular disease prediction.

Krishnan et al. [10] created a multi-layer perceptron prototype employing CAD technology for the aim of

identifying cardiovascular diseases in people. The more individuals who use the prediction approach to foresee their conditions, the more people will become aware of the ailment, and the lower the death rate of cardiovascular patients will be.

The creation and application of Support Vector Machines (SVM) have been briefly addressed in the paper by Jan Ramon, Fabian Guiza, Maurice Brynooghe, and Geert Meyfroidt titled "Machine learning techniques to examine large patient databases" [11].

The Heart Disease Prediction System (HDPS) is presented in Dangare and Apte's article, "A data mining approach for prediction of heart disease using Neural Networks" [12]. SVMs effectively analyze large datasets. This technology uses neural networks to improve accuracy by considering a number of variables, such as blood pressure, cholesterol, sex, smoking, and obesity.

III. PROPOSED SYSTEM

The data set was obtained, and pre-processing procedures were carried out, including outlier detection, label encoding, and normalization. Methods like Gradient Boosting, Logistic Regression, Naive Bayes, Decision Tree, Random Forest, K-Nearest Neighbor, SVM were used to classify the processed data. The following performance evaluation standards were used: scores for accuracy, precision, recall, and f1 as well as the roc-auc score. Charts provide a visual representation of the algorithms' overall effectiveness.

Fig.1 illustrates the working procedure of the proposed system,

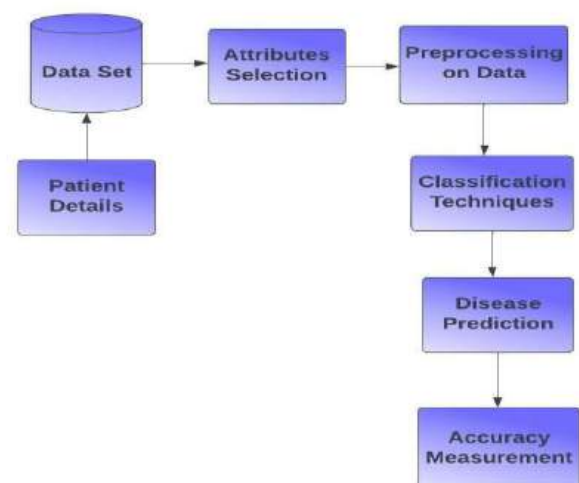


Fig. 1: Architecture of the proposed system

Below are the brief descriptions of the used algorithms:

A. Decision Tree

A decision tree is a hierarchical structure that separates the dataset into subgroups depending on the values of certain characteristics. To get a final forecast, it makes a sequence of judgments by posing queries at each node. In our scenario, a decision tree learned patterns like "If cholesterol level is high and age is above 50, then predict heart disease."

B. Random Forest

It is an ensemble learning technique that blends many decision trees to increase accuracy and reduce overfitting. Every tree in the forest makes a forecast, and the outcome is decided by a majority vote. Random Forest captured complicated correlations between characteristics for heart disease prediction and enhanced generalization by averaging over multiple trees.

C. K-Nearest Neighbor

It is a straightforward classification technique that determines a new data point's class based on the classes of its k closest neighbors in the training data. In our situation, KNN was able to determine if a patient's characteristics and those of known instances were comparable enough to predict the possibility of heart disease.

D. Support Vector Machine

It is an effective classifier that looks for the hyperplane that best divides several classes in the feature space. When classes cannot be separated linearly, SVM is useful. SVM might identify a decision boundary for heart disease prediction that successfully distinguishes between people with and without heart disease based on their attribute values.

E. Gradient Boosting

It is an ensemble strategy that gradually develops several weak learners, with each learner focused on the flaws of the one before it. It integrates their guesses to come up with a reliable conclusion. Gradient Boosting, which iteratively improves predictions based on prior iterations, might catch tiny patterns in the data when used to predict heart disease.

F. Logistic Regression

Heart disease can be predicted using the statistical model known as logistic regression, which was created for binary classification. Based on a person's characteristics, it determines the likelihood that they will develop heart disease and makes forecasts using a predetermined decision threshold.

G. Naive Bayes

By estimating the likelihood that a person has heart disease based on their medical characteristics, naive Bayes can help in the prediction of heart disease cases. Naive Bayes provides effective and understandable estimates of the chance of heart disease despite its assumption of feature independence, making it a useful tool for assessing early risk.

IV. DATA SET AND DESCRIPTION

A. Data source

There are databases with information regarding cardiac illness, one of the most severe complications throughout the planet, at the Cardiovascular Foundation and the UCI Machine Learning Repository [13]. Ten hospitals were tasked with collecting this information. Despite 920 data and 76 medical factors, we concentrated on 14 parameters relating to cardiovascular testing.

V. DATA PRE-PROCESSING

To assure accuracy and the highest potential performance of machine learning algorithms, pre-processing data is crucial for data analysis and research. To increase its value for the study, the dataset underwent a variety of data pre-processing procedures.

A. Data Cleaning

By correcting missing numbers, removing duplicates, and resolving conflicts, we made sure the data set was consistent. This procedure was necessary to provide a clean and reliable dataset that served as a base for correct analysis and predictable model results.

B. Downcasting

We came upon a collection of numbers, each with a distinct number of decimal places. To solve this, we used downcasting, which involves converting these numbers to a lower precision, often by reducing the number of decimal places.

C. Standardization

We performed standardization for a set of numerical values with various decimal places in order to enhance our analysis. Using this method, we scaled the numbers to have a mean of 0 and a standard deviation of 1, improving their comparability and making them better suited for statistical modeling.

D. Handling outliers

During data analysis and modeling, we had to control outliers in order to solve the problems brought on by unexpected data points. With this strategy, we were able to avoid having extreme data lead us to incorrect conclusions, improving the accuracy and dependability of our results.

E. Encoding

Using this method, we were able to help machine learning algorithms understand and accurately analyze the data, enabling exact computations and comparisons. We improved the usefulness of categorical variables in predictive models by using this strategy.

VI. RESULTS AND DISCUSSION

This section's major focus is on evaluating the outcomes and degree of efficiency produced by various classification

methods. These methods include a range of methods frequently used in categorization. Utilizing a variety of assessment criteria to gauge the efficacy of categorization algorithms is necessary. These metrics provide crucial details regarding the rate of success and accuracy of the employed classification approaches.

A. Accuracy Score

In the realm of machine learning, the accuracy score determines the ratio of precise projections to all forecasts in order to evaluate the model's competence [14].

When it comes to accuracy score, Random Forest performs better than other algorithms. Since Random Forest algorithms can manage complicated interactions and minimize overfitting, they frequently produce results with improved accuracy. Random Forest can identify a variety of patterns in the data and generate reliable predictions because of the ensemble aspect of the method, which mixes numerous decision trees. Furthermore, Random Forest uses ways like feature randomization and bootstrap sampling, which lessen overfitting and enhance generalization, resulting in improved overall accuracy.

Displayed in Fig. 2 below is the accuracy score of the utilized algorithms.

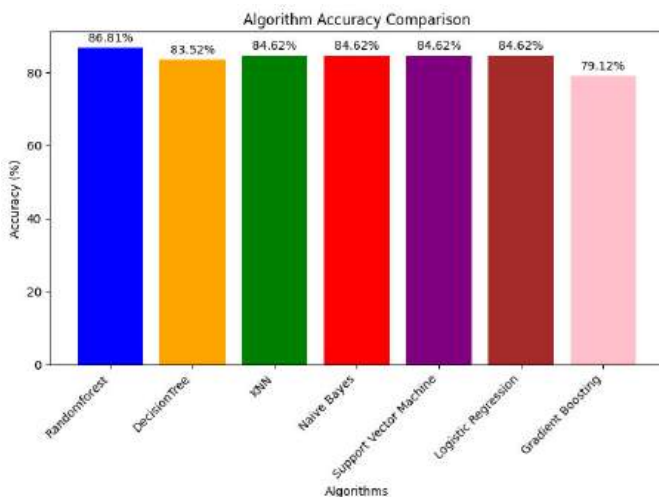


Fig. 2: Accuracy Scores of the classification models

B. Precision

The precision of a classifier, sometimes referred to as its positive predictive value, gauges how effectively it foresees the positive class.

It is obvious that the naive Bayes method provides a greater precision score. Naive Bayes classifiers usually yield higher accuracy scores given that they significantly assume

component interdependence. They can efficiently capture and take advantage of relationships between characteristics and the target variable thanks to this supposition. Naive Bayes may provide predictions that are more accurate by eliminating the influence of irrelevant or correlated information, which results in greater precision scores.

Fig. 3 below exhibits the precision scores of the employed algorithms

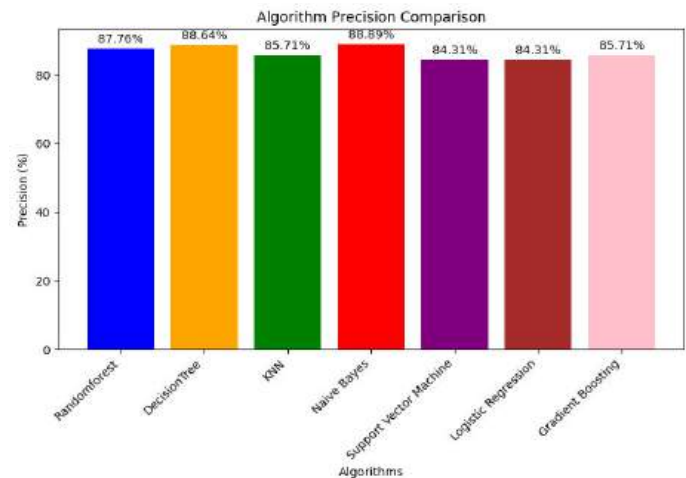


Fig. 3: Precision Scores of the classification models

C. Recall

Recall is the proportion of data samples from a class of interest, also known as the "positive class," that was properly identified by a machine learning model as corresponding to that specific group.

Comparatively speaking, it can be shown that logistic regression provides a greater recall score. In instances where accurately recognizing positive examples is important, logistic regression can offer greater recall scores since it directly optimizes for eliminating false negatives. Higher recall scores may be achieved by optimizing the amount of true positive predictions with logistic regression by utilizing the logit transformation and decision threshold.

In Fig. 4 below, the recall scores of the utilized algorithms are presented.

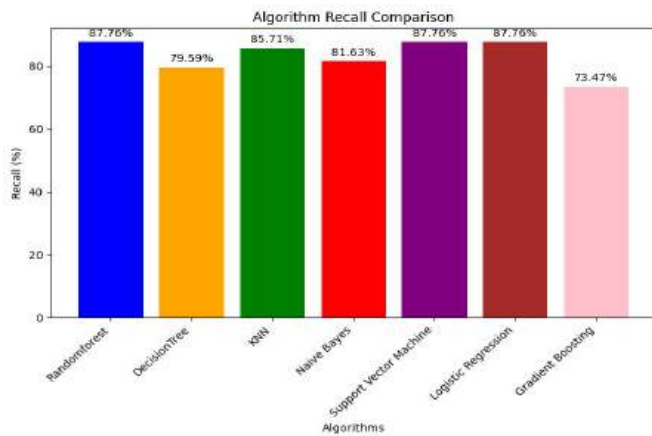


Fig. 4: Recall Scores of the classification models

D. F1-Score

Given the importance of both precision and recall, a single metric that combines them is necessary. This led to the formulation of the harmonious average of accuracy and recall, symbolized by the F1 score, as an approach to integrate these two metrics [15].

In terms of the f1-score, the random forest classifier outperforms the competition. Due to its capacity to recognize complicated interconnections and manage high-dimensional feature spaces, random forest models frequently provide better F1-scores. Higher F1-scores that balance both recall and precision are obtained by Random Forest due to its ensemble structure, which integrates several decision trees to handle a variety of data patterns and generate reliable predictions.

Fig. 4 below displays the F1-score of the employed algorithms.

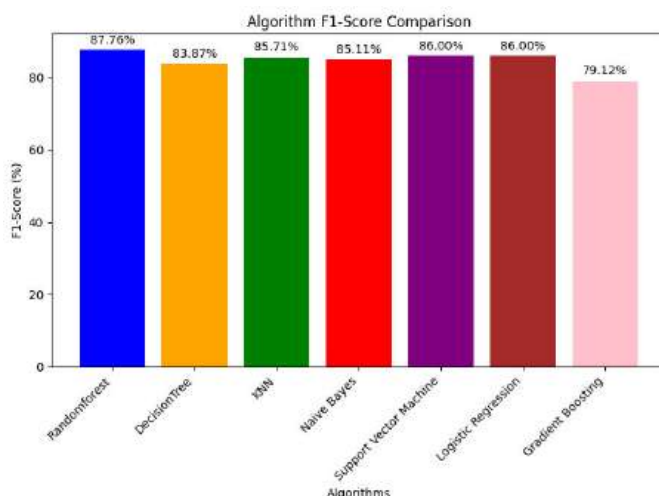


Fig. 4: F1-Scores of the classification models

E. ROC-AUC CURVE

This assessment indicator is crucial for assessing a classification model's efficacy. An improved score implies that the model performed more effectively, making it a crucial success indicator [16].

A higher AUC (area under the curve) score signifies better overall model performance. Notably, Random Forest demonstrates the highest AUC score. Thanks to its ability to capture complex nonlinear relationships, provide well-calibrated probabilities, and offer a natural ranking mechanism based on anticipated probabilities—crucial factors for AUC computation—Random Forest often yields superior AUC values. Its ensemble nature and inherent ability to handle intricate patterns contribute to its strong predictive power in binary classification problems.

In fig. 5 and 6, the ROC Curves for the Random Forest and Logistic Regression algorithms are presented, showcasing their performance in classification tasks.

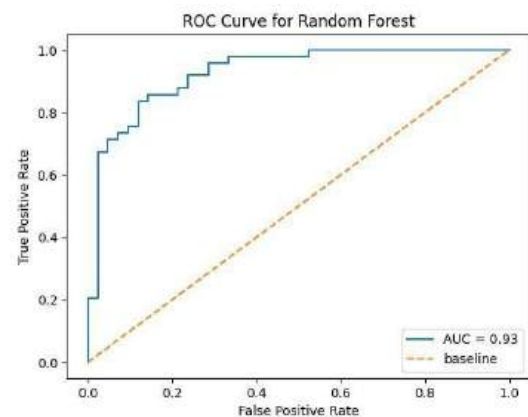


Fig. 5: ROC-AUC Curve of the random forest algorithm

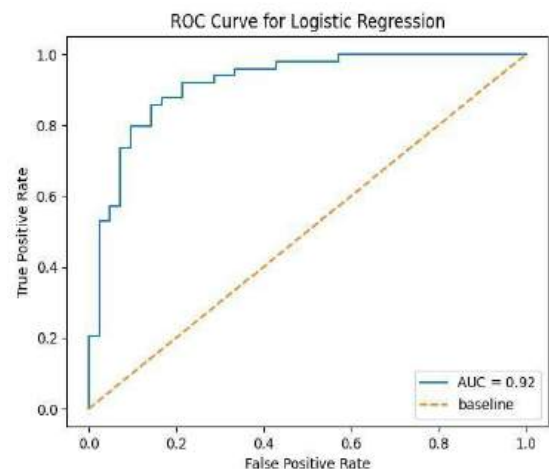


Fig. 6: ROC-AUC Curve of the logistic regression algorithm

VII. LIMITATIONS

Instead of investigating different data cleaning and pruning approaches to create an acceptable data set, this work has largely focused on the use of classification models for heart disease prediction. It is vital to bear in mind, however, that a data set that has been meticulously cleaned and trimmed generally outperforms a chaotic data set with missing values in terms of accuracy. More accurate analysis and prediction may be accomplished by using adequate categorization algorithms in conjunction with appropriate data-cleaning procedures.

VIII. CONCLUSION

The need for a rapid and precise prediction system has increased due to the rise in cardiovascular disease-related mortality. This study set out to identify the best machine-learning technique for identifying cardiovascular issues. The data set was used to assess the Support Vector Machine, Random Forest, Logistic Regression, Decision Tree, Gradient Boosting, Naive Bayes, and KNN algorithms. The study discovered that the performance of different approaches varied, with each having specific advantages and disadvantages for detecting cardiovascular disease. Future research may examine more machine learning techniques, bigger datasets, and combine Explainable AI with Federated Learning. A web application might improve usability and make clinical settings easier to adopt.

IX. REFERENCES

- [1]. Organization, W.H. (2023) Cardiovascular diseases, World Health Organization. Available at: <https://www.who.int/health-topics/cardiovascular-diseases> (Accessed: 23 May 2023).
- [2]. Quinlan, J.R. (1986) 'Induction of Decision Trees', Machine Learning, 1(1), pp. 81–106. doi:10.1007/bf00116251.
- [3]. S. Patel, J. Patel and S. Tejalupadhyay, "Cardiovascular Disease prediction using Machine learning and Data Mining Technique", Journal - IJCSC, vol. 7, 2022. Available: 10.090592/IJCSC.2016.018.
- [4]. Malav, A., Kadam, K. and Kamat, P. prediction of heart disease using K-means and artificial neural network Available at: https://www.researchgate.net/publication/319486202_Prediction_Of_Heart_Disease_Using_K-Means_and_Artificial_Neural_Network_as_Hybrid_Approach_to_Improve_Accuracy (Accessed: 22 May 2023).
- [5]. Sundar V,B.,Thirupathi, D.and Saravanan, N. (2012) Development of a data clustering algorithm for predicting heart disease. Available at: https://www.researchgate.net/publication/258651551_Development_of_a_Data_Clustering_Algorithm_for_Predicting_Heart_Disease (Accessed: 22 May 2023).
- [6]. Jadhav, S.D. and Channe, H.P. (2013) Comparative study of k-nn, naive Bayes and decision tree ... - IJSR. Available at: <https://www.ijsr.net/archive/v5i1/NOV153131.pdf> (Accessed: 22 May 2023).
- [7]. Chen, A.H. et al. (2011) HDPS: Heart disease prediction system.Avalableat:<https://ieeexplore.ieee.org/abstract/document/6164626> (Accessed: 22 May 2023).
- [8]. Thomas, J., and Princy, R.T. (2016) Human heart disease prediction system using data mining. - IEEE Xplore. Available at: <https://ieeexplore.ieee.org/abstract/document/7530265> (Accessed: 22 May 2023).
- [9]. A. Kaur and J. Arora, "CARDIOVASCULAR DISEASE PREDICTION USING DATA MINING TECHNIQUES: A SURVEY", International Journal of Advanced Research in Computer Science, vol. 9, no. 2, pp.569572,2022. Available: IJARCS 2015-2019.
- [10]. S. Krishnan J and K. J, "Prediction of Cardiovascular Disease Using Machine Learning Algorithms", 2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT), 2022. Available: 10.1109/ICIICT1.2019.8741465.
- [11]. Geert Meyfroidt, Fabian Guiza, Jan Ramon, Maurice Brynooghe" Machine learning techniques to examine large patient databases"-Best Practice & Research Clinical Anesthesiology, Elsevier Volume 23 (1) – Mar 1, 2009.
- [12]. Dangare, C. and Apte, S. (2012) A data mining approach for prediction of heart disease using Neural Networks, SSRN. Available at: <https://papers.ssrn.com/sol3/papers.cfm?abstractid=2175569> (Accessed: 22 May 2023).
- [13]. Welcome to the UC Irvine Machine Learning Repository (no date) UCI Machine Learning Repository. Available at: <https://archive.ics.uci.edu/> (Accessed: 19 June 2023).
- [14]. Chicco, D. and Jurman, G. (2020) The advantages of the Matthews correlation coefficient over F1 score and accuracy in binary classification evaluation - BMC Genomics, BioMed Central. Available at: https://bmcbgenomics.biomedcentral.com/articles/10.1186/s12864-019-6413-7?trk=public_post_comment-text (Accessed: 05 July 2023).
- [15]. Sklearn.metrics.f1_score scikit. Available at: http://scikitlearn.org/stable/modules/Generated/sklearn.metrics.f1_score.html (Accessed: 10 August 2023).
- [16]. Sklearn.metrics.roc_auc_score scikit. Available at: https://scikitlearn.org/stable/modules/generated/sklearn.metrics.roc_auc_score.html (Accessed: 15 July 2023).

DeepTestDroid A Platform for Automated Application Testing Using Deep Learning

1st Md. Huzaifadept. Computer Science and
Engineering

East West University

Dhaka, Bangladesh

saadhuzaiifa2497@gmail.com

2nd Fabiha Bushra Fabindept. Computer Science and
Engineering

East West University

Dhaka, Bangladesh

fabihabushra200@gmail.com

3rd Nazia Tabassum Natashadept. Computer Science and
Engineering

East West University

Dhaka, Bangladesh

natasha.nazia@gmail.com

4th Imran Fakirdept. Computer Science and
Engineering

East West University

Dhaka, Bangladesh

imranahsan017@gmail.com

5th Sabiha Afsana Falgunidept. Computer Science and
Engineering

East West University

Dhaka, Bangladesh

afsanaafalguni19@gmail.com

6th Mahamudul Hasandept. Computer Science and
Engineering, Senior Lecturer

East West University

Dhaka, Bangladesh

munna09bd@gmail.com

7th Mohammad Rifat Ahmmad Rashiddept. Computer Science and
Engineering, Assistant Professor

East West University

Dhaka, Bangladesh

rifat.abir@gmail.com

Abstract—Black and white box testing is used to measure an application. Testing can be complex and costly. Various testing techniques can measure multiple parts of an application. DeepTestDroid performs UI performance testing, which is black box testing. It can predict the density of an application's page from its wireframe picture. Higher density means a longer load time. UI performance testing can only be performed with human interaction. However, DeepTestDroid makes it automatic. To detect density, DeepTestDroid uses mobilenetV3Large. MobileNetV3Large was selected after an experimental analysis alongside other image processing models. This project can reduce the cost and step of UI performance testing. It will also make it easier to complete the testing phase of SDLC.

Keywords— android testing, mobile net, resnet50, efficient net, Rico

I. INTRODUCTION

Black box testing is the process of evaluating a system without knowledge of its source code. A tester provides input to the software and observes the system's output or behavior. Testing identifies how the system responds to expected and unexpected user actions, response time, and reliability. Black-box testing is highly efficient as it thoroughly evaluates a system from initiation to completion. Similarly, as end-users prioritize the system's functionality over its coding or structure and anticipate appropriate responses to their requests, testers can simulate user actions to assess whether the system meets its intended objectives. Employing a black box testing methodology, all relevant subsystems, including the user interface (UI)/user experience (UX), web or application, and integrated systems, are thoroughly examined throughout the workflow.

By leveraging deep learning, app testing tools can provide more efficiency for development teams in the creation and analysis phases of testing through both black-box and white-

box testing. Black box testing involves testing system usability and reliability issues. By leveraging deep learning, app testing tools can provide more efficiency for development teams in the creation and analysis phases through black-box testing. Black box testing involves testing system usability and reliability issues. The tester is aware of the application's internal structure and implementation. Due to the rapid release cycle and limited human resources, creating test cases manually and promptly can take time and effort. Consequently, many automated test input generators have been developed. Tools and software already enable automated testing of apps, including Monkey, Dyno Droid, Espresso, etc. The success of an automated test relies on picking the appropriate interaction for a given UI and allowing access to new relevant UI states. However, it can be challenging for a machine to interpret the GUI and select the appropriate button to click or scroll; because of this, most automated test generators use different types of GUI elements instead of random selection or brute force to choose which one to test. Human testers are better at this than automated test input generators because they can quickly determine which parts of the GUI need to be tested. [1]

II. RELATED WORKS

Deep Xplore, introduced in a research publication [2], revolutionizes the systematic testing of real-world deep learning systems. This system efficiently discovers improper corner case behavior in state-of-the-art deep learning models with thousands of neurons trained on popular datasets. Deep Xplore generates the wrong test inputs in one second on a laptop. Unlike differential testing approaches, deep Xplore detects errors even when profound neural network outputs are consistent. Most deep neural networks are designed and trained independently, making it unlikely that all of them will commit the same error in practice.

One research study [3] focuses on developing a mobile app-based artificial intelligence prototype using deep learning to predict screen taps and actions. Convolutional neural networks encode visuals and screen elements, while long-short-term memory remembers touch positions. Deconvolutional and linear neural networks indicate tap location and action type. The research demonstrates the potential of deep learning for automating mobile app testing through model construction and training. Further tests are planned to improve outcomes, including longer training times, larger datasets, and transfer learning. The final goal is to train the prototype and use it as an automatic input generator for testing.

Another work [4] proposes new testing methods for system faults. The proposed method uses a web scraper, ML, and Selenium. Machine learning and Selenium are used to improve web-based error detection and discovery. The researchers plan to develop a web testing tool that delivers accurate URL test results. They want to provide specialized testing methods for distinct web elements and expand test cases to support diverse testing tactics.

Another study [5] automates and scales Android app security and robustness testing. The paper introduces an Android-specific software analysis technique that creates several fuzz test scenarios. They also offer a cloud-based test infrastructure that runs these test cases on several simulated Android devices. This method underlines the need for smartphone-specific guided fuzzing strategies rather than brute-force fuzzing. The white-box strategy tries to identify vulnerabilities efficiently, decreasing the time and computational resources needed for real-world fuzz testing.

In another study [6], the authors propose a new Android app testing automation method. Machine learning and frequent test situations improve testing. Empirical analysis shows that their improved testing tool outperforms conventional methods in natural settings. Activity classifications and preset features limit the researchers' approach. They are working with the Test Project R&D team to add activity categories and reduce the testing algorithm's reliance on hard-coded test cases.

Another research study [7] introduces Crash Scope, an automated Android developer tool that creates crash reports and test scripts. It gives developers thorough error reports and detects a similar number of crashes as other advanced tools. The crash report contains screenshots, steps to reproduce the crash, stack traces of exceptions, and a script that can consistently recreate the issue. The researchers hope to use model-based GUI testing and static analysis to streamline bug reports and improve their systematic exploration method.

Humanoid is a new method for automated black-box testing of humanoid Android apps using deep learning and a deep neural network model [8]. The input generator is tested on open-source and commercial programs to verify dependability and applicability. However, Humanoid has limitations, including system broadcasts, sensor events, and text input. Researchers propose enhancing the interaction model or using alternative text input generation methods to address these constraints.

In a research study [9], the researchers have proposed automating Android app testing to identify GUI problems, identifying existing and undetected system weaknesses. However, there are limitations to this approach. They suggest using state machines for I/O and concurrency primitive invocations and Java static analysis tools using model-based verification. This approach helps identify pattern and state-machine violations during development, minimizing dependency on automated testing and improving system dependability.

A grey literature review on using artificial intelligence (AI) to improve test automation is presented in another study. [10] AI is being used to create and enhance test code, according to 136 papers found in 1,200 sources. AI-enabled problems and solutions are categorized in the report. Code and automated test generation are examples of common issues and solutions. The report also suggests adding literature sources and using developer surveys and interviews to corroborate the findings on AI in test automation. Traditional and AI-based methods can be compared to assess AI's benefits in testing.

Software automation and machine learning (ML) frameworks were extensively explored in the [11] paper. Testing tools were rated on test performance, accuracy, scope, time, expertise, and manual labor. To improve software quality and ML frameworks for automation software. Future studies will examine and organize software testing and machine learning research to help researchers and engineers develop guidelines for using ML approaches to software testing.

This case study [12] presents a novel automated testing framework that blends user interaction characteristics, historical bug data, and an interest point detector and descriptor to find user interaction concerns effectively. This methodology detects user interaction problems empirically. The researchers want to create a customized exploration environment to influence target application event sequences during testing. Each application's scope will be expanded to assess its usefulness and resilience further.

The researchers in [13] created a mechanism to generate tests for Android resource leak problems automatically. They propose a resource-efficient, neutral GUI event flow. Their resource leak testing method surpasses a non-automated approach from earlier research. The study shows that static control flow analysis can automatically develop comprehensive Android GUI tests for repeating behaviors. Static or dynamic analysis-based test execution, monitoring, and prioritizing methodologies need improvement.

AM-TaaS, a cloud-based automated mobile app testing system, is described in [14] paper. The framework gives users on-demand testing services. Results reveal that framework-designed automated test cases tested all emulated devices. Future research will optimize and extend the cloud infrastructure, automate more test cases based on specified criteria, build new evaluation methods for test cases, and expand the framework to accommodate Windows Phone and iOS.

In the paper [15], the researchers address the performance degradation of mobile devices due to the simultaneous use of multiple mobile applications. They have developed a performance testing tool (PTT) with a user-friendly interface

and accurate performance testing results. The plan for PTT includes enhancing performance indicators such as electricity and GPU.

The research paper [16] examines model-based testing of Android user interfaces, notably the BBC News widget case study. Case study models and tools are open-source. The researchers emphasize that although the technology (TEMA) employed still needs to provide automatic tools for reverse engineering models, no technical limitations hinder this capacity. The device could include a scripting language with keywords or a Robot Framework library.

The research paper [17] presents an innovative method for automating the functional testing of mobile software. This method combines machine learning techniques with standard test scenarios for effective and practical testing. They demonstrate that their system can perform a significant portion of the tasks typically done by human testers, potentially reducing manual testing efforts.

The paper [18] examines a sophisticated Android remote UI testing infrastructure. Developers and researchers can test scripts on several Android versions to find bugs across platforms quickly. The platform cannot imitate real-world use; therefore, it cannot test applications primarily relying on speech, gesture, or movement inputs. Multitouch events and device orientation testing are planned to improve the platform.

The paper [19] discusses automated software interface testing, particularly behavior-driven testing, a key component of BDD. Behavior-based testing is used for Android app user interface automation. Behavior-driven testing involves constant reworking to adapt to environmental changes; therefore, upgrading the instrument to enable Cucumber may not be viable for existing products with instrumented tests.

A privacy framework in the paper [20] limits undesired permissions and prevents programs from collecting user data. The proposed technique uses data mining to identify superfluous permissions and modify permission information for privacy and application operation. The framework may not work for sealed, protected applications in sectors like finance, which require additional protection. The researchers want to strengthen the framework for such applications' security and protection procedures.

Finally, researchers in the paper [21] use Deep Reinforcement Learning (RL) to automate Android app exploration. ARES and FATE, their model-based Android testing tools, highlight their approach's efficacy. Coverage and problem discovery are better with configuration structures. The process may degrade, and system-level events that require rooted devices cannot be performed. Android app fault categories and iOS adaptations are planned. The papers propose novel methods and frameworks for automating Android application development, addressing constraints, and suggesting further study.

A. Problem Analysis

Android application UI testing is a type of "black box" testing that focuses on ensuring the user interface of an application is functioning correctly. UI testing is used here to provide the

application's intuitive and flexible user interface. Design specifications better accompany the performance of the load. To carry out UI testing, the issues mentioned below arise, making the testing a bit stiff to obtain the quality assurance of the Android application's interface.

- In some states, UI testing tools might be unable to adapt to changes in the segment of the application's graphical user interface, which leads to false negatives or false positives, which means a layout can be sorted as medium density. Still, the updated version can be categorized as having a high density.
- As applications update over time, automated UI tests must be updated and make sure they are still relevant and precise.
- Manual UI testing requires maintaining and updating regularly; that can be lengthy and waste time.
- UI tools have limitations in their ability to collaborate with specific parts or manage particular types of inputs that can affect the accuracy and integrity of the testing. For example, a UI testing tool might not mesh with the dynamically generated properties or be hidden, such as pop-ups; it can also be overlays that appear only on particular conditions.
- During model prediction, test data manually sorted for the model will have a considerable impact. If the sorted data is not classified correctly, it can have substantial negative implications.

B. Research Questions

- Which type of black box testing is addressed in the paper?
 - What data will be used for testing and training the model?
 - How will the data set be categorized?
 - How will the testing be automated using a deep learning approach?
- How are the testing approaches built?
 - How will the real techniques leverage, as well as the amount of input information that is required to perform testing?
 - How will process detailed information on the design and implementation of test approaches?
- How can the measure effectiveness proposal automated testing approach?
 - Which quality attributes, metrics, datasets, and procedures for the literature can be used for measuring the effectiveness of the approach?

C. Objectives

To build a system that predicts test outcomes using deep learning. This paper integrates black-and-white-box testing with deep understanding, a novel approach. The goal is to automate six black-box and white-box tests and generate test data for a deep learning algorithm. The objective is to achieve the most accurate prediction using the deep learning algorithm, ensuring Android applications' durability, vulnerability, and performance. Automated UI testing requires the creation of five categories based on the number of components in the UI layout. The data needs to be sorted according to these categories. Using the sorted data, a model can be trained. Finally, a platform will be developed for UI

testing of Android applications, classifying the layouts into their respective categories.

D. Design and implementation

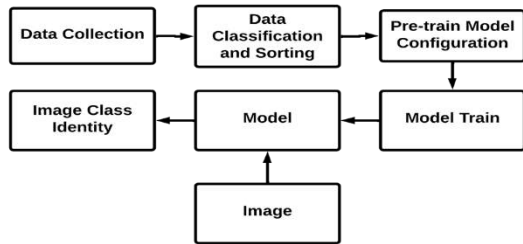


Fig. 1. Workflow chart.

- Data collection: Data was collected from the Rico dataset, which includes UI screenshots and hierarchies with semantic annotations.
- Data classification and sorting: Five categories were created based on the number of UI components in the layout (very low, low, medium, high, and very high), and the data were sorted accordingly.
- Pre-train model configuration: The model was configured category-wise, adding layers to the existing model.
- Model training: Six models were trained, and their losses and accuracy were compared to determine the best-performing model.
- Model selection: The model with the best performance was selected.
- Image class identification: The image layout category was identified using the selected model.

E. Data Collection

The dataset used in this study is called Rico. Mining Android applications created Rico while it was running, using a combination of programmed and manual investigation. The infrastructure used for app mining in Rico does not require access to or modification of an app's source code. Crowd workers accessed the applications through a web interface after downloading them from the Google Play Store. The dataset includes photographs and hierarchies with semantic annotations for Android applications' user interface (UI).

Rico dataset consists of over 66,000 UI screens and hierarchies, with semantic annotations that describe the meaning and usage of elements on the screen. These annotations provide insights into what different UI components, buttons, and icons signify. Semantic screenshots in Rico utilize a distinctive color scheme, where each class is assigned a unique color. The mapping between semantic concepts and their corresponding colors is documented in three files known as "component_legend.json," which correspond to specific component categories.[22]

For this study, approximately 60,600 datasets were available. The datasets were sorted according to the categorized UI screenshots of the apps. The sorting technique employed is known as "crowdsourcing."

Below are user interface displays from the dataset and their respective semantic annotations.

A few examples of UI screens and their semantic annotations are shown below.



Fig. 2. Screenshot of RICO dataset [22]

A UI was designed to facilitate sorting datasets based on categories. Here are the screenshots of the UI:



Fig. 3. Tools for data sorting

JSON files of the layouts were utilized to generate data, which was then classified into five density categories based on the number of components present in the UI.

Table I. Density Class Example

Component range	Density	Screenshots
1-6	Very low	 [22]
7-11	Low	 [22]
12-18	Medium	 [22]
19-30	High	 [22]
31-60	Very high	 [22]

Multiple methodologies were employed to sort the data, but the final sorting method was chosen based on its reliability compared to other methods.

F. Experimental Analysis

The proposed "DeepTestDroid" model was designed to address the mentioned problems and perform UI testing using a deep learning approach. The implementation of the model utilized MobileNetV3, ResNet50, EfficientNet_b0,

EfficientNet_b1, and EfficientNetV2_b0 neural networks. Python was used for the performance, and specific libraries were employed to handle the app layouts based on the created categories.

MobileNetV3 is a convolutional neural network architecture widely adopted in various mobile applications, including TensorFlow. It offered the capability to run on embedded systems and was implemented in small and large versions. We used **MobileNetV3small** and **MobileNetV3large**. [23]

ResNet50 is a state-of-the-art convolutional neural network model for image classification. It was trained on ImageNet, a large-scale classification dataset. [24]

EfficientNet_b0 is another convolutional neural network architecture trained on a vast dataset from the ImageNet database. It can classify images into one thousand object categories, including various objects and animals. [25]

EfficientNet_b1 is a scalable convolutional neural network architecture that uniformly scales dimensions using a compound coefficient. [26]

EfficientNetV2_b0 is a convolutional neural network architecture that performs highly on image classification tasks while maintaining parameters and computational cost efficiency. [27]

The models were trained with different configurations, classes, and datasets, and for the final analysis, a consistent design, types, and datasets were used to evaluate all models effectively. Out of the 60,600 semantic annotations available, 10,000 data points were used for testing the model. Split folders, a Python library, were utilized to divide the data into test, train, and validation sets. A Python script was also developed to determine the data distribution per class.

G. Result Analysis

Table II. Results of the Models

Name	Test		Validation Accuracy	Train Accuracy
	Loss	Accuracy		
MobileNetV3 Large	0.89850693 94111633	0.74601596 59385681	0.73486745 35751343	0.7784764 766693115
MobileNetV3 Small	0.98004519 93942261	0.68725097 17941284	0.68084043 26438904	0.6609975 695610046
EfficientNet B0	0.92321687 93678284	0.71215140 81954956	0.71185594 7971344	0.6885808 110237122
EfficientNet B1	1.00359773 63586426	0.62749004 36401367	0.62831413 74588013	0.6112619 638442993
EfficientNet V2B0	0.94237416 98265076	0.68824702 501297	0.71535766 12472534	0.6418464 779853821
ResNET50	1.08122932 9109192	0.62350594 99740601	0.59579789 63851929	0.5839645 266532898

Multiple deep-learning models were experimented with, including variations that were significantly different from each other. All models were trained using the same configuration, classes, and data. The results from the experiments can be observed in the provided table or diagram.

The highest test loss was observed in the ResNet50 model, while the lowest test loss was achieved by the

MobileNetV3Large model. The ResNet50 model had the most insufficient test accuracy, whereas the MobileNetV3Large model reached the highest. Among the tested models, ResNet50 performed poorly on the dataset, while MobileNetV3Large emerged as the best-performing model. Although all models were trained on the ImageNet dataset, variations in architecture led to differences in data outcomes.

Summary

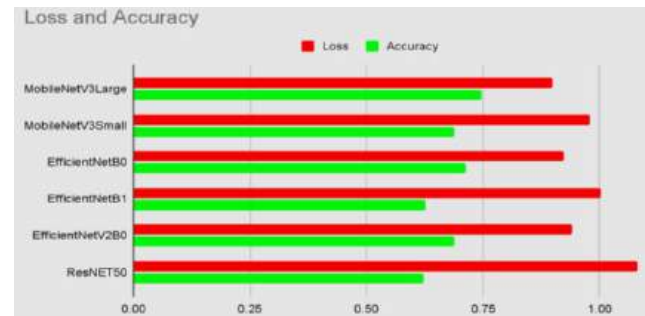


Fig. 4. Results

III. TECHNOLOGIES

Python is used as the programming language for writing code. The deep learning models are implemented using the **TensorFlow** library: **Kaggle Notebook** and **Jupyter Notebook** as the IDEs for the work.

IV. SOCIAL AND ENVIRONMENTAL IMPACT OF ENGINEERING

Putting "black box" testing into action using deep learning can simplify GUI testing by replacing manual testing with AI-powered automation. This approach is particularly beneficial for testing mobile applications that require specialized testing strategies. GUI testing can be complex and costly, especially as GUIs evolve and change over time. Regression testing, often used for GUI testing, can be time-consuming and expensive. Test case prioritization can help overcome these constraints by selecting a subset of test cases while maintaining the criteria for testing.

This study categorized two types of screenshots from various apps, and a dataset of 66,000 data points was organized. A model was developed to recognize UI behaviors and classify applications as having excellent or poor UI. Manual high-level tests, such as acceptance and GUI-based system tests, are typically expensive, tiresome, and error-prone. Test automation is suggested as a solution, but it may have limitations related to GUI layout changes or dependencies on specific implementations or frameworks. [6]

TensorFlow tools can replicate user actions on GUI-driven systems, regardless of the implementation language, operating system, or platform. TensorFlow provides a versatile solution for mimicking end-user interactions with the GUI using test scripts and image recognition techniques. However, there are limitations and drawbacks to using TensorFlow and other automated test approaches that must be further understood and assessed in industry applications. The paper discusses an empirical investigation at two Swedish organizations, focusing on transitioning from

manual system test cases to automated VGT tests. The study successfully automated test suites and detected defects in complex systems, demonstrating the feasibility and effectiveness of the robotic approach. [28]

In this case, challenges were encountered while sorting and training the datasets using TensorFlow. The study concludes that despite its challenges, GUI testing remains valid, adaptable, and cost-effective. The proposed model offers decision assistance for GUI development and usage and the potential for incorporating additional automated testing methods. [3]

V. CONCLUSION

Any kind of testing is costly. Here we are automating UI testing to reduce test costs. We used the Rico dataset to create our image processing model. Moreover, to find the best model for our problem through an experimental analysis. This same method can be applied to other UI-based testing. There are still many improvement points in this paper. Example: model accuracy, recent UI screenshot, tools.

REFERENCES

- [1] I. Banerjee, B. Nguyen, V. Garousi, and A. Memon, "Graphical user interface (GUI) testing: Systematic mapping and repository," *Information and Software Technology*, vol. 55, no. 10, pp. 1679–1694, Oct. 2013. DOI: 10.1016/j.infsof.2013.03.004.
- [2] K. Pei, Y. Cao, J. Yang, and S. Jana, "DeepXplore: Automated Whitebox Testing of Deep Learning Systems," in *SOSP 2017 - Proceedings of the 26th ACM Symposium on Operating Systems Principles*, Oct. 2017, pp. 1–18. DOI: 10.1145/3132747.3132785.
- [3] E. Alégroth, R. Feldt, and L. Ryrholm, "Visual GUI testing in practice: challenges, problems and limitations," *Empir Softw Eng*, vol. 20, no. 3, pp. 694–744, Jun. 2015, doi: 10.1007/s10664-013-9293-5.
- [4] R. Mahmood, N. Esfahani, T. Kacem, N. Mirzaei, S. Malek and A. Stavrou, "A whitebox approach for automated security testing of Android applications on the cloud," *2012 7th International Workshop on Automation of Software Test (AST)*, Zurich, Switzerland, 2012, pp. 22–28, doi: 10.1109/IWAST.2012.6228986.
- [5] Y. Li, Z. Yang, Y. Guo and X. Chen, "Humanoid: A Deep Learning-Based Approach to Automated Black-box Android App Testing," 2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE), San Diego, CA, USA, 2019, pp. 1070–1073, doi: 10.1109/ASE.2019.00104.
- [6] D. Li, G. Advisors, Rubén, T. Liesa, and X. Gù Ardía Latorre, "A Deep Learning Based Approach to Automated App Testing."
- [7] A. Vilhunen, "User interface test automation for an Android application," *User interface test automation for an Android application*, May 16, 2022.
- [8] C. Hu and I. Neamtiu, "Automating GUI testing for Android applications," in *Proceedings of the 6th International Workshop on Automation of Software Test (AST '11)*, New York, NY, USA, 2011, pp. 77–83, DOI: 10.1145/1982595.1982612.
- [9] A. Rosenfeld, O. Kardashov, and O. Zang, "Automation of Android Applications Functional Testing Using Machine Learning Activities Classification," *arXiv.org*, Sep. 04, 2017.
- [10] N. Paul and R. Tommy, "An Approach of Automated Testing on Web Based Platform Using Machine Learning and Selenium," *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*, Coimbatore, India, 2018, pp. 851–856, DOI: 10.1109/ICIRCA.2018.8597297.
- [11] F. Ricca, A. Marchetto and A. Stocco, "AI-based Test Automation: A Grey Literature Analysis," *2021 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, Porto de Galinhas, Brazil, 2021, pp. 263–270, DOI: 10.1109/ICSTW52544.2021.00051.
- [12] A. Méndez-Porras, J. Alfaro-Velasco, and A. Martínez, "Evaluation of an Automated Testing Framework: A Case Study," *Evaluation of an Automated Testing Framework: A Case Study*.
- [13] K. Moran, M. Linares-Vasquez, C. Bernal-Cardenas, C. Vendome and D. Poshyvanyk, "CrashScope: A Practical Tool for Automated Testing of Android Applications," *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*, Buenos Aires, Argentina, 2017, pp. 15–18, doi: 10.1109/ICSE-C.2017.16.
- [14] I. K. Villains, E. A. B. Costa, and A. C. Dias-Neto, "Automated Mobile Testing as a Service (AM-TaaS)," in *Proceedings - 2015 IEEE World Congress on Services, SERVICES 2015*, Aug. 2015, pp. 79–86. DOI: 10.1109/SERVICES.2015.20.
- [15] B. Mishra *et al.*, "Privacy Protection Framework for Android," *IEEE Access*, vol. 10, pp. 7973–7988, 2022, DOI: 10.1109/ACCESS.2022.3142345.
- [16] S. Khan, Z. Jiangbin, and A. Wahab, "Design and Development of Android Performance Testing Tool," in *2020 IEEE Conference on Big Data and Analytics, ICBDA 2020*, Nov. 2020, pp. 57–60. DOI: 10.1109/ICBDA50157.2020.9289714.
- [17] T. Takala, M. Katara, and J. Harty, "Experiences of system-level model-based GUI testing of an android application," in *Proceedings - 4th IEEE International Conference on Software Testing, Verification, and Validation, ICST 2011*, 2011, pp. 377–386. DOI: 10.1109/ICST.2011.11.
- [18] A. Rosenfeld, O. Kardashov, and O. Zang, "Automation of Android Applications Functional Testing Using Machine Learning Activities Classification," in *Proceedings - International Conference on Software Engineering*, May 2018, pp. 122–132. DOI: 10.1145/3197231.3197241.
- [19] S. Fatima, B. Mansoor, L. Ovais, S. A. Sadruddin, and S. A. Hashmi, "Automated Testing with Machine Learning Frameworks: A Critical Analysis," *MDPI*, Jul. 28, 2022.
- [20] J. Kaasila, D. Ferreira, V. Kostakos, and T. Ojala, "Testdroid: Automated remote UI testing on android," in *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia, MUM 2012*, 2012. DOI: 10.1145/2406367.2406402.
- [21] H. Zhang, H. Wu, and A. Rountev, "Automated test generation for detection of leaks in Android applications," in *Proceedings - 11th International Workshop on Automation of Software Test, AST 2016*, May 2016, pp. 64–70. DOI: 10.1145/2896921.2896932.
- [22] B. Deka, Z. Huang, C. Franzen, J. Hibschan, D. Afergan, Y. Li, J. Nichols, R. S. Kumar, "Rico: A Mobile App Dataset for Building Data-Driven Design Applications," in *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, 2017, pp. 4237–4249.
- [23] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, Hartwig Adam; *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 1314–1324
- [24] Mo, N., Yan, L., Zhu, R., & Xie, H. (2019, January 30). Class-Specific Anchor Based and Context-Guided Multi-Class Object Detection in High-Resolution Remote Sensing Imagery with a Convolutional Neural Network. *Remote Sensing*, 11(3), 272, doi: 10.3390/rs11030272
- [25] Mekanapura, N., Sujatha, C., Patil, P. R., & Desai, P. (2022, January 1). Classification of plant seedlings using deep convolutional neural network architectures. *Journal of Physics: Conference Series*, 2161(1), 012006, DOI: 10.1088/1742-6596/2161/1/012006
- [26] Jie, Y., Ji, X., Yue, A., Chen, J., Deng, Y., Chen, J., & Zhang, Y. (2020, December 21). Combined Multi-Layer Feature Fusion and Edge Detection Method for Distributed Photovoltaic Power Station Identification. *Energies*, 13(24), 6742.
- [27] V. -T. Hoang and K. -H. Jo, "Practical Analysis on Architecture of EfficientNet," 2021 14th International Conference on Human System Interaction (HSI), Gdańsk, Poland, 2021, pp. 1–4, DOI: 10.1109/HSI52170.2021.9538782.
- [28] RVS College of Engineering & Technology and Institute of Electrical and Electronics Engineers, *Proceedings of the International Conference on Inventive Research in Computing Applications (ICIRCA 2018)*

A Machine Learning and Deep Learning Integrated Model to Detect Criminal Activities

Kanij Fatema Arpa, Tanni Mittra, Tasnim Ferdous, Nusrat Jahan, Riffat Ara Khan Tayna, Mahamudul Hasan, Mohammad Rifat Ahmmad Rashid, and Md Sawkat Ali

Department of Computer Science and Engineering East West University
Dhaka, Bangladesh

Email: kanijfatemaarpa@gmail.com, tanni@ewubd.edu, tasnimferdous216@gmail.com, nusratjahanmily99@gmail.com, riffatarakhantayna@gmail.com, munna09bd@gmail.com, rifat.abir@gmail.com, alim@ewubd.edu

Abstract—Crime is considered an intended execution of an act that is harmful to society and punishable under criminal law. Bangladesh as a developing country has experienced severe crime-related issues in recent years due to the constantly growing criminal activities. The law enforcement sector of Bangladesh is dedicated to implementing the law and reducing fear of offense to make Bangladesh a secure place to live. Despite this, the traditional ways of detecting crimes in Bangladesh are currently slow and less effective because they are not getting enough technological support. As a result, most of the time the crimes get reported after the occurrence. So to attain efficiency in policing activities it has become significant to develop such a system that can notify the police department before the occurrence of a crime. In our proposed model, we develop a system that can prevent crime by detecting and notifying real-time criminal activities beforehand. To carry out this concept, we build a central system consisting of three modules i.e. criminal activity identification module, weapon detection module, and criminal detection module. Each of the modules is centrally connected and can give an alert when any type of abnormality occurs. We use Deep Learning and Machine Learning algorithms to perform criminal activity detection and face recognition of criminals. For each of the modules, we collect and prepare our own data set. We perform various experiments to investigate the effectiveness of the method. The evaluation result shows that our proposed method performs impressively in identifying criminal activities and criminals.

Index Terms—Criminal Activities, Law Enforcement, Personal-ity Analysis, Face Recognition, Deep Learning, Machine Learning

I. INTRODUCTION

Bangladesh evolved independently on 16 December 1971 and is now considered a developing country. Crime has been a crucial problem from the beginning of the country's independence [1]. Different economic factors like violence, unemployment, and injustice cause crime here. At present, the crime rate of Bangladesh is around 69.94%, which is very threatening¹. However, most of the time, the crimes remain unreported or get reported after the occurrence, which induces risks and challenges for the police to attain efficiency in their policing activities. To overcome the risks and facilitate the work of the law enforcement sectors of Bangladesh, it has become significant to develop a system that can notify them

whenever a crime occurs. It would be quite helpful if a smart system could detect real-time crime by itself and alert the police immediately about the crime scene and the probable suspects.

To our best knowledge, no such system exists in Bangladesh to detect or forecast crime in real-time and warn high authorities to prevent further damage. However, recent advances in machine Learning (ML) and deep learning (DL) algorithms would be excellent support for this because these algorithms provide effective results in real-time by analyzing and processing several parameters. So, in our proposal, we develop a core system that can identify real-time criminals and detect criminal activities to prevent crime in Bangladesh. We use both approaches of ML and DL to analyze data sets and detect crime in real-time. It will also assist law enforcement authorities in receiving early warning of illegal activities and allow them to take appropriate action on time.

In this paper, we propose three significant modules of a pipeline i.e. criminal activity detection, weapon detection, and criminal prediction. Each of the modules works independently and sends notifications to the central system. For the criminal activity detection module, we analyze the UCF-Crimes dataset to detect crime through video surveillance cameras. We use the Long-term Recurrent Convolutional Network (LRCN) [2], a deep learning model to detect criminal activities like explosion, stealing, robbery, assault, fighting, road accidents, and burglary. For the weapon detection module, we use the Pashtaw dataset to detect weapons through video surveillance cameras. We use a real-time object detector model YOLOv7 [3] to detect weapons. We create a criminal database by ourselves for the criminal forecasting module with their wanted status. For criminal detection, we use dlib's state-of-the-art deep learning model Face Recognition² to detect the face of a criminal from the database along with basic information. Then we examine the social media status of the suspected person to forecast his personality. To analyze social media accounts, we use TextBlob [4]. In the result analysis section, we perform different experiments to investigate the effectiveness of the System.

¹https://www.numbeo.com/crime/country/_result.jsp?country=Bangladesh

²https://github.com/ageitgey/face_recognition

The rest of the paper is organized as follows. In Section II, related work on this topic has been discussed along with some background data. Section III represents details of the proposed model. The analysis and outcomes of the experiment are shown in Section IV. The conclusionV contains shortcomings and future work of the proposed model.

II. RELATED WORK

In recent years, machine learning and deep learning algorithms have been widely used in the crime detection field. In [5], the authors demonstrated the effectiveness of ML algorithms in predicting crime patterns and other applications, such as identifying criminal hotspots, developing criminal profiles, and identifying criminal trends using WEKA. Kim et.al [6] used ML predictive models K-Nearest Neighbour (KNN) and boosted decision tree to predict crime. A total of 560,000 crime datasets between 2003 and 2018 were analyzed with an accuracy of 39% to 44%. In [7], the prediction of crime-related statistics in Philadelphia and Pennsylvania was discussed using ML techniques like logistic regression, KNN, ordinal regression, and tree techniques with 69% accuracy.

In [8], the authors examined a dataset made up of several crimes and made predictions about the kinds of crimes that would happen soon using models including KNN classification, logistic regression, decision trees, random forests, support vector machine (SVM), and Bayesian approaches with an accuracy of roughly 78%. Prithi et.al [9] showed how to anticipate crime rates utilizing a graphical user interface and ML algorithms. The authors of [10] and [11] employed KDE and social media data to predict crime. In [12] they used a deep neural network (DNN)-based feature-level data fusion method to forecast the occurrence of crime with accuracies of 67.01% and 66.33%, respectively. In [13] random forest, decision tree, and k-nearest neighbor (KNN) algorithms were used to forecast criminal conduct with 99.16% accuracy.

The decision tree (J48) algorithm was taken into consideration in [14] when creating a crime prediction model. In [15] a brand-new method was applied to detect crime scene video surveillance systems in real-time violence detection using the spatiotemporal (ST) technique with a classification method based on Deep Reinforcement Neural Networks (DRNN). In [16] random forest, naïve Bayes, and SVM were used to forecast the frequency of crimes and hotspots with an accuracy of 75.6% and 65.3%. In [17] future crime patterns in Bangladesh are predicted using a linear regression model. The work cited in [18] listed a number of the analysis and forecasting of various crime reports using data from the Bangladesh Police Dataset and the ACLED Dataset. Obuandike et al. [19] attempted to employ the ZeroR algorithm and a decision tree in their study but were unable to get an accuracy of more than 60%.

III. PROPOSED WORK

The main target of our research is to develop a core system that can alert crime fighters immediately after detecting real-time criminal activities or criminals through video surveillance

cameras. For this purpose, we propose a pipeline integrated with three modules i.e. criminal activity detection, weapon detection, and criminal prediction module. Each of the modules connects centrally and give alert to the high authorities to prevent crime. The following subsections contain detailed descriptions of the proposed three modules.

A. Criminal Activity Detection

To detect criminal activities, we collect UCF-Crimes³, a large-scale dataset that consists of 1900 real-world surveillance videos of 13 abnormalities. After analyzing the videos we collected 1000 videos of 7 different abnormalities, which are more common in Bangladesh, and built a new dataset. To obtain a more accurate and clean cut of the crime scene, we perform different data pre-processing such as video resizing using KAPWING⁴, a modern video editing tool. We split the data into train and test groups using a 75:25 ratio. After collecting video data, we build an LRCN model using Tensorflow [20]. The LRCN model combines two prominent deep learning algorithms CNN [21] and LSTM [21] to provide more accuracy. The process flow of the criminal activity detection module is depicted in Figure 1.

B. Weapon Detection

For our weapon detection module, we use the dataset Pashtaw⁵, which contains 3,000 images of labeled pistols. We pre-process the dataset for the best fit of our model. We have cloned our dataset with YOLOv7, a real-time object detector with computer vision techniques to detect weapons in real time. We split the data into train and test groups using an 80:20 ratio. Figure 2 represents the workflow of the weapon detection technique.

C. Criminal Prediction

In this module, we detect the face of a criminal through video surveillance cameras along with the basic information from a criminal database. If the criminal is wanted, our system gives alerts immediately. If no information is available, then we analyze the social media status of the detected face to analyze the personality. We divide this module into two subsections i.e. criminal detection, and sentiment analysis.

1) *Criminal Detection*: As we can not collect any central database of criminals, we create our dataset, which contains images and their basic information like name, address, wanted status, ETC. For face recognition in real-time, we use the face recognition library from dlib's state-of-the-art model with deep learning algorithms. From the model, we use the Histogram of Oriented Gradients (HOG)+Linear SVM face detector technique⁶. After detecting any criminal, we check it on the criminal database to collect information about the criminals. Figure 3 represents the workflow of the face detection technique.

³<https://www.kaggle.com/datasets/odins0n/ucf-crime-dataset>

⁴<https://www.kapwing.com/video-editor>

⁵<https://github.com/topics/weapons-detection>

⁶shorturl.at/ePRS9

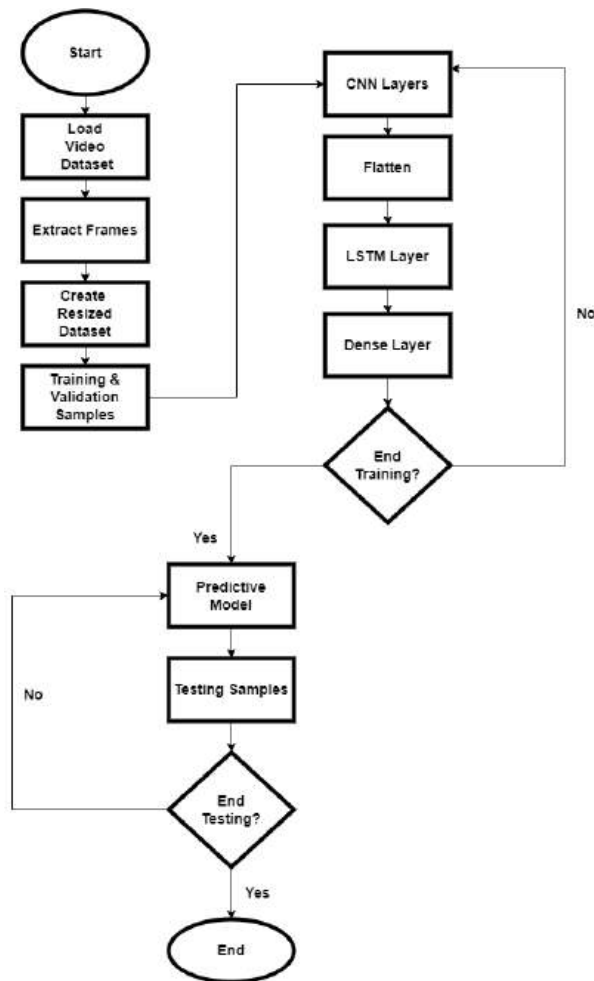


Fig. 1. Criminal Activity Detection Method

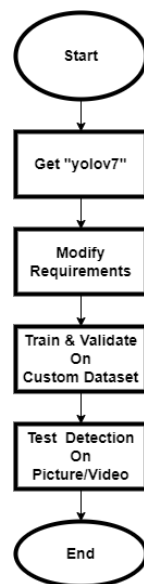


Fig. 2. A Systematic Approach for Weapon Detection

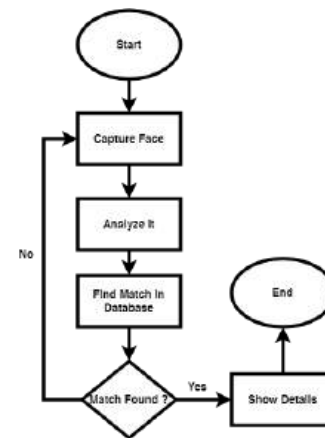


Fig. 3. A Systematic Approach for Face Detection

2) *Sentiment Analysis*: To add more clarity to our work, we analyze and predict sentiment based on the social media status of detected persons. We also create a dataset by ourselves. We use Tweepy [22] to access the Twitter API and collect tweets of selected users. As we collect text data, this may contain different noises. We perform some preprocessing, such as removing usernames, patterns, @, hashtags, and short words, splitting the white space, replacing digits with spaces, and removing punctuation, numbers, and special characters.

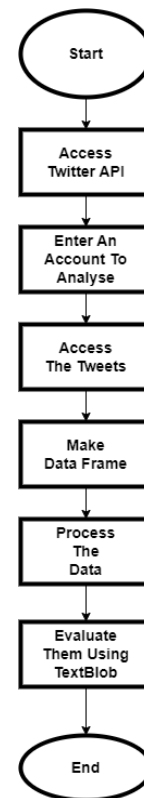


Fig. 4. A Systematic Approach for Sentiment Analysis

We use TextBlob to dive into the natural language process-

ing (NLP)⁷ task for sentimental analysis. We evaluate the data using a Scatter Plot of Subjectivity-Polarity, a Bar Chart of Sentiment, and a World Cloud view for the most used words in all Tweets to anticipate any crime that may be committed in the near future. Figure 4 represents the workflow of the sentiment analysis technique.

IV. EXPERIMENTAL RESULT AND ANALYSIS

The following subsections contain a detailed description of the results obtained from different experiments.

A. Criminal Activity Detection

As mentioned, we collect a dataset consisting of 1,000 videos of 7 abnormalities from the UCF-Crimes dataset. Among them, 174 videos are assault-related, 120 are explosion-related, 196 are stealing-related, 198 are robberies, 184 are road accidents, and 144 are burglaries videos. We train the model using algorithms like CNN and LSTM. Figure 5 represents video samples of criminal activities captured by security cameras from the collected dataset.



Fig. 5. Criminal Activity Detection Test Cases

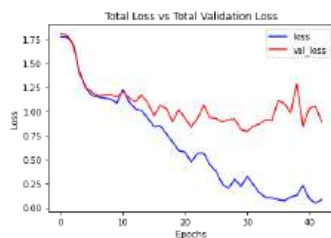


Fig. 6. Total Loss VS Total Validation Loss

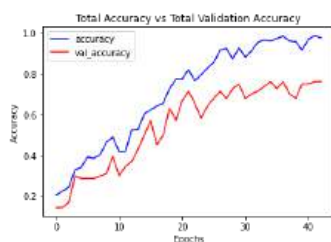


Fig. 7. Total Accuracy VS Total Validation Accuracy

Here, Figure 6 represents the training and validation loss of the model. Figure 7 represents the model accuracy curve where training accuracy is slightly greater than validation accuracy. We achieved an accuracy of 84% for the criminal activity detection task, which is quite impressive.

⁷<http://dl.acm.org/citation.cfm?id=2078183.2078186>

B. Weapon Detection

As mentioned earlier, for weapon detection, we use the Pashtaw dataset. Among the 3000 images of labeled pistols, we collected 1120 images for this task. Figure 8 represents a sample of a weapon detection dataset where a short gun has been detected using YOLOv7.



Fig. 8. Weapon Detection Test Case

To evaluate this module, we calculate the F1-Curve, PR-Curve, and Confusion Matrix, which are shown in Figure 9, 10, and 11, respectively. The confidence rating that maximizes precision and recall based on Figure 9 is 0.360. The F1 value for this model is about 0.8 and is similar to the distance from the maximum value of 0.8. We choose a confidence level of 0.78 for Figure 10 as a suitable point.

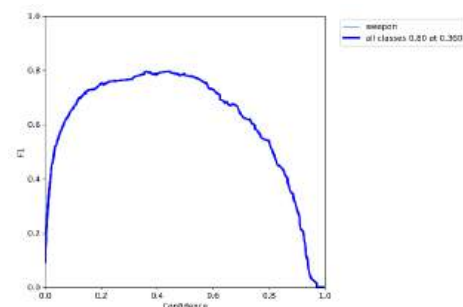


Fig. 9. F1 Score Curve for a Custom YOLOv7 Weapon Detection

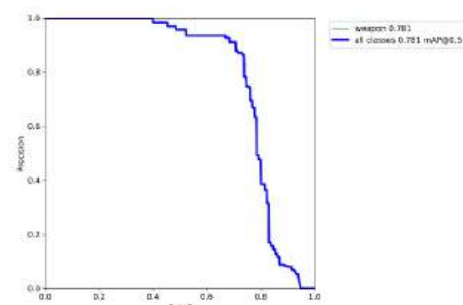


Fig. 10. PR-Curve for Weapon Detection

From these figures, we attain precision value 1, which is the best value, recall 0.8, and F1-Score 0.88, which are pretty

	Actual: weapon	Actual: background FN
Predicted: weapon	0.75	0.08
Predicted: background FN	0.25	0.92



A bar chart titled 'Sentiment' on the x-axis. The y-axis is labeled 'Counts' and ranges from 0 to 100 in increments of 20. There are three blue bars: 'Neutral' with a count of approximately 105, 'Positive' with a count of approximately 75, and 'Negative' with a count of approximately 25.

Sentiment	Counts
Neutral	105
Positive	75
Negative	25

```
Criminal information
Name - test1
Phone - 012434
Crime - Stealing
Wanted Info - Yes
```



for the most used words in All Tweets is also shown in Figure 16.

Here, in Figure 14, we can see that the majority of the dots are on the positive or neutral (Polarity = 0) side of the line, indicating that the user's social media posts are safe. On the other hand, the negativity line is around 20 in Figure 15, indicating that there is a low likelihood of the user committing a crime shortly. Lastly, in Figure 16, no harmful or negative words are seen.

V. CONCLUSION AND FUTURE WORK

Artificial intelligence (AI) is gradually but steadily becoming a capable tool for punishing criminals and preventing criminal conduct. It is no longer just a speculative notion. AI is now being used by law enforcement agencies to improve the productivity of their officers. Our proposed model can be used for surveillance, such as monitoring a crowd for anomalies, analyzing video footage for criminal activity, and applying facial recognition or weapon recognition to better advantage. This policing pipeline is expected to result in changes in security and assurance for society. The combination of our pipeline with current AI programs may prove to be quite beneficial given the rising crime rate. To a greater extent, this technology can aid in the prediction and detection of crime as well as criminals.

In our proposed model, we propose a core system consisting of three modules i.e. criminal activity detection, weapon detection, and criminal forecasting to prevent crime. To achieve high accuracy, we mix deep learning and machine learning techniques. For the sake of our research, we have gathered and examined real-world datasets that include surveillance videos. We employ LSTM and CNN algorithms based on ARNN to detect criminal activities, and for weapon detection, we use YOLOv7. We use Dlib's HOG and Linear SVM to identify criminals by recognizing their faces. In addition to identifying the criminal, we look at their social media status to look forward to potential criminal activities. To analyze the effectiveness of our suggested model, we run considerable examinations of our datasets and evaluate the performance of each module. The experimental study and findings demonstrate that our strategy produces good outcomes in each part.

Even though we have accomplished our objectives, there is still much room for improvement in our core system modules. In the future, by accessing different social media platforms, we can extend our social media analysis using images, videos, check-ins, comments, etc. Moreover, we can use geolocation to predict criminal hotspots and study criminals' body language before committing a crime to prevent crime.

REFERENCES

- [1] M. K. Hasan, M. Khatun *et al.*, "The investigation of the rising crime and the reason for people to indulge in the crime in bangladesh," *Advances in Applied Sociology*, vol. 10, no. 10, p. 421, 2020.
- [2] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *CVPR*, 2015.
- [3] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," *arXiv preprint arXiv:2207.02696*, 2022.
- [4] S. Loria, "textblob documentation," *Release 0.15*, vol. 2, 2018.
- [5] L. McClendon and N. Meghanathan, "Using machine learning algorithms to analyze crime data," *Machine Learning and Applications: An International Journal*, vol. 2, pp. 1–12, 03 2015.
- [6] S. Kim, P. Joshi, P. Kalsi, and P. Taheri, "Crime analysis through machine learning," 11 2018, pp. 415–420.
- [7] C. Tabedzki, A. Thirumalaiswamy, P. van Vliet, S. Agarwal, and S. Sun, "Yo home to bel-air: predicting crime on the streets of philadelphia," *University of Pennsylvania, CIS*, vol. 520, 2018.
- [8] P. Kumari, A. Gahalot, S. Uprant, Dhiman, and L. Chouhan, "Crime prediction and analysis," 02 2020, pp. 1–6.
- [9] S. Aravindan, E. Anusuya, and M. Ashok Kumar, "Gui based prediction of crime rate using machine learning approach," 2020.
- [10] M. S. Gerber, "Predicting crime using twitter and kernel density estimation," *Decision Support Systems*, vol. 61, pp. 115–125, 2014. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0167923614000268>
- [11] X. Chen, Y. Cho, and S. Y. Jang, "Crime prediction using twitter sentiment and weather," in *2015 Systems and Information Engineering Design Symposium*, 2015, pp. 63–68.
- [12] H.-W. Kang and H.-B. Kang, "Prediction of crime occurrence from multi-modal data using deep learning," *PloS one*, vol. 12, no. 4, p. e0176244, 2017.
- [13] S. Hossain, A. Abtahee, I. Kashem, M. Hoque, and I. Sarker, *Crime Prediction Using Spatio-Temporal Data*, 07 2020, pp. 277–289.
- [14] E. Ahishakiye, D. Taremwaa, E. Opiyo, and I. Niyonzima, "Crime prediction using decision tree (j48) classification algorithm," *International Journal of Computer and Information Technology (ISSN: 2279 – 0764)*, 05 2017.
- [15] K. B. Sahay, B. Balachander, B. Jagadeesh, G. Anand Kumar, R. Kumar, and L. Rama Parvathy, "A real time crime scene intelligent video surveillance systems in violence detection framework using deep learning techniques," *Computers and Electrical Engineering*, vol. 103, p. 108319, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0045790622005419>
- [16] P. Stalidis, T. Semertzidis, and P. Daras, "Examining deep learning architectures for crime classification and prediction," *Forecasting*, vol. 3, pp. 741–762, 10 2021.
- [17] M. Awal, J. Rabbi, S. Hossain, and M. Hashem, "Using linear regression to forecast future trends in crime of bangladesh," 05 2016.
- [18] P. Rahman, A. I. Hoque, M. F. Ahmed, I. Kashem, A. Alam, and N. Hossain, "Bangladesh crime reports analysis and prediction," 08 2021, pp. 453–458.
- [19] O. G. N., A. Isah, and J. Alhasan, "Analytical study of some selected classification algorithms in weka using real crime data," *International Journal of Advanced Research in Artificial Intelligence*, vol. 4, no. 12, 2015. [Online]. Available: <http://dx.doi.org/10.14569/IJARAI.2015.041207>
- [20] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: <https://www.tensorflow.org/>
- [21] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," *CoRR*, vol. abs/1411.4389, 2014. [Online]. Available: <http://arxiv.org/abs/1411.4389>
- [22] J. Roesslein, "Tweepy: Twitter for python!" URL: <https://github.com/tweepy/tweepy>, 2020.

DeepCONVSVM: A Comprehensive Model for Detecting Disease in Mango Leaves

Md. Shafayat Hossain, Sarder Iftekhar Ahmed, Md. Nadim, Md. Mizanur Rahman, Maria Mehjabin Shenjuti, Mahamudul Hasan, Taskeed Jabid, Maheen Islam, Md Sawkat Ali

Department of Computer Science and Engineering East West University
Dhaka Bangladesh

Email: shafayat1777@gmail.com, iftekhar.sarder@gmail.com, nadimewu.cse@gmail.com, mizanurewu@gmail.com, mariamehjabin10@gmail.com, munna09bd@gmail.com, taskeed@ewubd.edu, maheen@ewubd.edu, alim@ewubd.edu

Abstract—Mango is the fruit of the world which has huge socioeconomic implications. Various diseases and pests destroy mango leaves, causing socioeconomic loss. There has been research done on diseases of mango leaves in the past, but it does not provide a complete solution for identifying common mango leaf diseases. For this study, the major objective is to find a solution to the problem of correctly diagnosing a wide range of frequent mango leaf illnesses. DeepCONVSVM, a hybrid of CNN and SVM, has been proposed as a convolutional neural network based architecture to accomplish this. CNN and ResNet50 have been built to see if the proposed model is efficient. According to the study's experimental data, the proposed model outscored the other two models with 91% accuracy, 91% precision, and 90% recall.

Keywords—Convolutional Neural Network, Image classification, Machine Learning, Deep Learning, Support Vector Machine

I. INTRODUCTION

According to the United Nations, there are 7.6 billion people living in the world today, and by 2050, that number is expected to increase to 9.8 billion [1]. According to a report released by the Food and Agriculture Organization (FAO), food output must increase by 70% by 2050 [2].

In the agricultural sector to produce an extra 70% of food, there are two main challenges: the first one is Diseases and the second one is Disorder. Diseases are usually caused by insects, bacteria, fungus, and disorders are natural disasters like temperature, rainfall, moisture, nutrition soil, etc. Most of the time farmers are trying to manage disease in the field with naked eyes and the use of various pesticides as treatment. It is extraordinarily time-consuming and often inaccurate. Mango is a major source of fruits and is essential for the preservation of biodiversity. However, many mango leaf ailments have a significant impact on mango productivity. Diseases like Dieback, Powdery Mildew, Red Rust, Bacterial Canker, Sooty Mold, and Anthracnose, among others, can affect mango leaves. Anthracnose affects about 39% of mango trees, whereas powdery mildew can kill up to 23% of untreated trees [3]. A fatal condition called bacterial canker can reduce mango production by 10% to 100% [3]. Due to these diseases, fruit quality and agricultural production are severely disrupted. As a result of misdiagnosis by various botanists and agricultural officials, wrong medicine is applied to the leaves. Hence, diseased

leaves do not heal and confusion is created among the farmers [4]. If the diseases affected by mango leaves are properly identified, the yield will be more upward. Initially, machine learning was used to diagnose diseases in less time, but its techniques have some limitations. When working with large datasets, accurate results are not available. The problem can be solved using deep learning, a subset of machine learning. In this paper, our aim is to identify mango leaf diseases using Deep learning so that appropriate medication can be applied to the infected leaf. We have worked on about 4000 pieces of data in this project. Here, we have worked on all types of common mango diseases. However, Deep learning is constantly improving the agricultural sector [2].

II. RELATED WORK

Singh et al. in [5] describes a multilayer convolutional neural network that can categorize mango leaves that have the fungus Anthracnose on them. In this work, the dataset includes 2200 photos of both healthy and diseased leaves. They have classified images whether the leaf is diseased (Anthracnose) or healthy using MCNN inspired by AlexNet architecture.

In order to recognize the Anthracnose disease in mango leaves, Kumar et al. in [6] developed a CNN architecture based on VGG-16. Real-time Anthracnose-infected mango leaves and images of other plants taken from PlantVillage were used in this study. They have worked with only Anthracnose mango disease and have classified the mango leaves affected by the Anthracnose and healthy. They have also classified other plant leaves collected from PlantVillage whether it is healthy or diseased.

Merchant et al. in [7] have proposed an unsupervised machine learning model i.e. clustering technique to detect various nutrient deficiencies of mango leaves. Using digital image processing, they created the data set by extracting the different features of mango leaves.

Using a DCNN model to identify different forms of rice leaf diseases has been published by Lu et al. in [8]. They have worked with 10 healthy and diseased classes of 500 images and Real-time data is used in the validation process. The 10-cross-fold validation approach is used to improve classification accuracy.

In [9], Barbedo et al. outline the various aspects and difficulties that have an impact on how well a network functions. Using plant photos from DigiPath that contain twelve distinct diseases, a CNN model is used to demonstrate the factors and difficulties of a network. Researchers in [10] classified healthy and unhealthy data utilizing VGG-16, ResNet, DenseNet, and Inception as four different CNNs models. The PlantVillage data set, which included 38 diseased classes and 14 healthy classes, was utilized to validate the classification. The authors came to the conclusion that DenseNet was better at classifying data and took less time to run than other architectures.

Golhani et al. in [11] have shown various neural network studies to classify and identify the plant images. With context to hyperspectral images, various imaging concepts are presented.

Iqbal et al. in [12] have shown several studies to classify and identify the citrus plant leaves diseases. The authors have investigated numerous methods for identifying the condition and have also covered the benefits and drawbacks of the study.

Picon et al. in [13] presented a DCNN model that was used for the classification of three types of fungal diseases relating to wheat plants. The data for this project were gathered from two real-world places.

In [14], Jain et al. described a CNN architecture for real-time disease classification of firecracker and pomegranate plant leaves. This work is done in a cloud-based environment. The images were collected from a real-time environment.

Gandhi et al. in [15] employed a mobile application to identify plant leaf disease using CNN and Generative Adversarial Networks (GANs). They worked with multiple plant images collected from PlantVillage. This work achieved 92% accuracy.

Multiple diseases of cucumber plants can be identified using DCNN developed by Juncheng Ma et al. in [16]. In this study, they tested four cucumber diseases and used real-time photos to verify the results. They suggest the use of thermal infrared imaging to enhance their research.

Zhangn et al. presented another two deep CNN models which are GoogLeNet and Cifar-10 in [17] used for maize or corn diseases detection. They have used Plant-Village dataset and pictures gathered from Google. The authors are quite enthusiastic about incorporating the model in mobile devices and also adding different types of plant diseases. These two models had outperformed other transfer networks in terms of accuracy.

Durmucs et al. applied AlexNet and then SqueezeNet deep learning networks to classify plant leaf diseases in [18]. The image datasets are from the plant village database, and they are of tomato plant leaves in ten different classes.

Convolutional neural networks and linear SVMs for image classification in [19] have been proposed by Fred M. Agarap. They have worked with 10 classes of 70000 images and the validation is done on MNIST dataset.

A deep learning algorithm based on linear support vector

machines has been suggested by Yichuan Tang in [20]. His L2-SVM model was tested on the different types of datasets like MNIST, CIFAR-10, and the ICML.

III. METHODOLOGIES

A. Data

The whole dataset for this work has only self-captured data of Mango leaves. This data was collected in 2021. The dataset consists of 4000 pictures, each measuring 240 by 320 pixels and are stored in PNG files. The following diseases: Powdery Mildew, Sooty Mold, Anthracnose, Cutting Weevil, Die Back, Bacterial Canker, Gall Midge, and Healthy images are the eight classes of images. The sample dataset, which consists of four photos of mango leaves, is shown in Figure 1.

B. Dataset Collection

The images of mango leaves were taken at several mango orchards in Bangladesh. They were gathered from the mango gardens at Sher-e-Bangla Agricultural University, Jahangirnagar University, Rajbari District's Udaipur Village, and in Nilphamari District's Itakhola Village. For this work, only RGB(Red, Green, Blue) images were used. The dataset images were gathered using a mobile phone camera. In the beginning, the number of images captured was roughly 6000. After conducting preprocessing to the dataset, the number of images was decreased to 4000 images which are categorized among eight classes. Table I displays the total number of images for each category.

TABLE I
DETAILS OF IMAGE CATEGORIES

Class Name	Number of Images
Anthracnose	500
Bacterial Canker	500
Cutting Weevil	500
Die Back	500
Gall Midge	500
Healthy	500
Powdery Mildew	500
Sooty Mold	500

C. Data Pre-Processing

According to the proposed methodology, after data collection image preprocessing and augmentation were done. A series of actions must be taken to make the dataset useful. Unwanted images have to be excluded from the data. Images containing multiple diseases were filtered at the time of cleaning. Only 4000 images remained after the dataset was cleaned up. It was required to crop some of the images in order to remove unnecessary or redundant boundaries from the images. Some samples of the processed images are depicted in figure 2. For image preprocessing TensorFlow

image data generator was used which includes the following operation:

- Scaled down by factor 255.
- Shared by 10%.
- Zoom range was taken 10%.
- Horizontal flip was set true for mirror reflection.



Fig. 1. Sample Images



Fig. 2. Processed Images

D. ResNet50

Resnet50 is a type of CNN that was first proposed by He et al. in the year of 2016 [24]. The full form of ResNet50 is residual neural networks which include a MaxPool layer and Average Pool layer, and 48 Convolution layers. ResNet50 is similar to other networks in that it comprises numerous layers, which includes fully-connected, convolutional, and pooling layers. Every five segments of the ResNet50 model has a convolution and identity block. Identity and convolution both blocks have three convolution layers. Over 23 million parameters can be taught in this model. There are various issues with vanishing gradient and deterioration in a convolutional neural network [21]. According to the idea of vanishing gradient, and depending on how far it is from the last layer, the backpropagation error signal inside a feedforward network either drops or climbs exponentially. In order to move from one layer to the next and arrive at the final forecast, ResNet calculates the delta. By allowing

gradients to flow along an additional shortcut path, ResNet is able to overcome the issue of vanishing gradients [22]. Figure 3 shows the architecture of Resnet50.

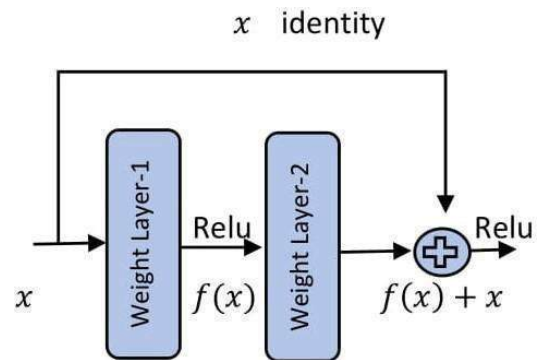


Fig. 3. Residual Neural Network

E. SVM

In the 1990s Vapnik developed the support vector machine (SVM). It falls under supervised learning models. For binary classification linear SVM was articulated. The SVM performs data analysis for regression and classification. In actuality, the SVM moves the input vector to a higher dimension feature space. Increasing the margin, the best hyperplane between two classes is discovered. It is also called a support vector network (SVN). Finding the best hyperplane to divide two classes in a given dataset is the goal of SVM. It is possible to express any hyperplane as $w \cdot x + b = 0$. Here, b is an offset and w is a vector normal to the hyperplane. By solving an optimization problem (Eq. 1) SVM learns the parameters w .

$$\text{minimize } \frac{1}{2} \|\omega\|^2 + C \sum_{n=1}^M \epsilon_i \quad (1)$$

SVM that utilize the linear sum of the slack variables are known as L1-SVMs, whereas those that use the square sum are known as L2-SVMs. It is simply a difference in the objective function used to optimize the sum. With the standard hinge loss, Eq. 1 is known as L1-SVM. L2-SVM (Eq. 2), its differentiable equivalent, produces more consistent results [20].

$$\text{minimize } \frac{1}{2} \|\omega\|^2 + \frac{C}{2} \sum_{n=1}^M \epsilon_i^2 \quad (2)$$

In the above equations ω represents the weight vector that determines the orientation of the decision boundary in the feature space, ϵ is a slack variable that accounts for instances that fall within the margin or on the wrong side of the decision boundary, M is the number of training examples or data points in the dataset and C is a regularization parameter that controls the trade-off between maximizing the margin and minimizing the classification error.

F. CNN

Convolutional Neural Networks (CNN) is an artificial network. It is a multilayer neural network that functions similarly to human brain neurons. Each neuron communicates with another, exchanges information, and then makes a choice. CNN is one of the most widely used

artificial networks of the present day. It's best when working with huge amounts of data, and it's often utilized in processing of both image and videos, as well as pattern reorganization. Because neurons in CNN architecture are linked to a tiny section of other layers instead of being entirely linked to every neuron, the overall number of neurons is reduced [6]. Low-level properties are extracted by the first convolution layer, such as edges, lines, and corners, whereas higher levels extract higher-level features [23]. Pooling layers reduces the resolution of the feature. In pooling layers, the window size basically is selected 2×2 or 3×3 and for each window it picks the maximum value. Pooling may be done in two ways, maximum pooling and average pooling [23]. If the maximum value is taken from the window, it is called maximum pooling and if it is taken average value from the window it is called average pooling. The fully-connected layer is the final layer of the DNN.

G. Proposed Work: DeepCONVSVM

The SVM is commonly considered as a classification model, but it can also be used to solve regression problems. In this work DeepCONVSVM, a combination of CNN based architecture and SVM, has been proposed. DeepCONVSVM architecture is depicted in Figure 4. The algorithm is depicted in Table II. With five convolutional layers, a ReLU activation function, three max-pooling, two dense or FC layers as an output layer where SVM is used, this model combines CNN and SVM. The images are converted to a 1D array by using a flattened hidden layer. Convolutional and max-pooling layers are respectively 3×3 and 2×2 . Input size and feature map can however vary as seen in Figure 4.

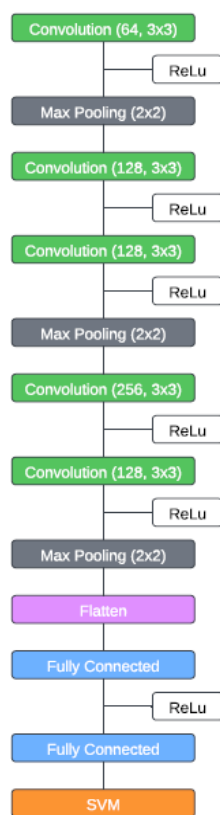


Fig. 4. Architecture of the proposed work

The proposed DeepCONVSVM architecture is described in detail. Steps involved:

- This model transforms a standard-sized image into a feature-set for additional processing using a number of layers.
- The 1st layer comprises 64 filters of convolutional layers and activation function of ReLU.
- The following layer is the max pooling, which reduces the size of the convoluted image to a (2, 2).
- The following two layers are also convolutional, with an activation function of ReLU and 128 filters.
- There are two strides in the max-pooling layer, and has a pool size of (2, 2).
- The model is then constructed using activation function ReLU, and a convolutional layer of 256 filters.
- The following convolutional layer, which has 128 filters, employs ReLU.
- A max-pooling layer of (2,2) and one stride follows that.
- An additional layer is added to flatten the output of the earlier CNN model.
- The outcome of flattened technique is the feature set of each image.
- 2 fully connected layers are included which will be used to classify images using the produced featureset.
- There are 512 hidden neurons in this dense layer, which is used for the artificial neural network's hidden layer.
- In the output layer, SVM is implemented. There's a parameter called "kernel regularizer" and inside this regularizer, we have used the l2 norm and passed the softmax activation function.

TABLE II
PROPOSED ALGORITHM

Algorithm: DeepCONVSVM

INPUT $224 \times 224 \times 3$
 CONV2D: 3×3 , 64 filters, 2 strides
 ReLU
 MaxPool2D: 2×2 size, 2 strides
 CONV2D: 3×3 , 128 filters, 2 strides
 ReLU
 CONV2D: 3×3 , 128 filters, 2 strides
 ReLU
 MaxPool2D: 2×2 size, 2 strides
 CONV2D: 3×3 , 256 filters, 2 strides
 ReLU
 CONV2D: 3×3 , 128 filters, 1 stride
 ReLU
 MaxPool2D: 2×2 size, 1 stride
 Flatten
 FC: 512 hidden neurons
 ReLU
 FC: 8 output classes, l2 regularizers

H. Optimizers, Loss Function, Batch Size, and Evaluation Metrics

This study's classification objective necessitated a multi-class classification. So for the loss function, categorical cross-entropy was applied in both ResNet50 and CNN, while squared hinge loss was worked with for CNN-SVM. The batch size is another crucial parameter that affects a model's performance. Generally speaking, models with bigger batch sizes outperform those with smaller batch sizes. During the process, the three models were trained with 128 batches. A total of 100 training epochs were used

for each model. The dataset was split in 80-10-10 percent for training, testing and validation purposes.

Adam optimizer, is used in this suggested work because this is a method of stochastic optimization for calculating parameter learning rates. The trained models were validated with test images when the training was completed. Accuracy, f1-score, precision and recall were utilized to assess the performance matrix of the models. By dividing the entire number of data points, precision is calculated. categorized positively by the model by the number of true positives.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (3)$$

The true positive rate, commonly referred to as recall, is computed by dividing the entire number of positive data points by the number of true positive data points.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (4)$$

The F-score is a means of combining the precision and recall of the model and is defined as:

$$F1_Score = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5)$$

Accuracy is defined as the proportion of accurately predicted observations to total observations. It is appropriate for symmetric data sets, which have almost identical false positive and false negative values.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \quad (6)$$

I. Result

The model was created in Google Colab and executed on a GPU. To achieve the required classification accuracy, a total of 100 epochs were used. The results of the DeepCONVSVM model obtained after training are depicted in Figures 5, 6, and 7. Various image preprocessing was done prior to training. Comparison of training and validation loss graphs are shown in Figure 6. As the quantity of epochs rose, training and validation losses decreased dramatically. Figure 5 depicts the comparison of training and validation accuracy curves. The training accuracy was initially about 15%, but as the number of epochs increased, it increased to 98.78% on the 100th epoch. Likewise, the accuracy of validation was 12% for the first epoch and 90.50% for the hundredth epoch.

TABLE III
CLASSIFICATION REPORT OF DEEPCONVSVM

Class Name	Precision	Recall	F1_Score
Anthraxnose	0.95	0.99	0.97
Bacterial Canker	0.95	0.71	0.82
Cutting Weevil	1.00	0.98	0.99
Die Back	0.97	0.96	0.96
Gall Midge	0.85	0.95	0.90
Healthy	0.83	0.93	0.88

Powdery Mildew	0.82	0.99	0.90
Sooty Mold	0.91	0.73	0.81

Accuracy is calculated for the model as a whole but precision and recall are calculated for an individual class. We evaluated the model's precision, recall, and f1-score with Mango leaf diseases as the positive class to see if it consistently performed well on certain classes. The performance is shown in Table III. A confusion matrix was initially devised to accomplish this. The confusion matrix's rows represent the ground truth classes, while its columns represent DeepCONVSVM's projected classes. Using this information, the accuracy and recall were computed. The confusion matrix for the proposed DeepCONVSVM is displayed in Figure 7. The model has a macro average precision of 90% and macro average recall of 91%. Whereas both weighted average of precision and weighted average of recall have found 91%.

TABLE IV
PERFORMANCE COMPARISON OF DEEPCONVSVM WITH OTHER MODELS

Model	Precision	Recall	F1_Score
ResNet50	79%	75%	77%
CNN	87%	85%	86%
DeepCONVSVM	91%	90%	90.5%

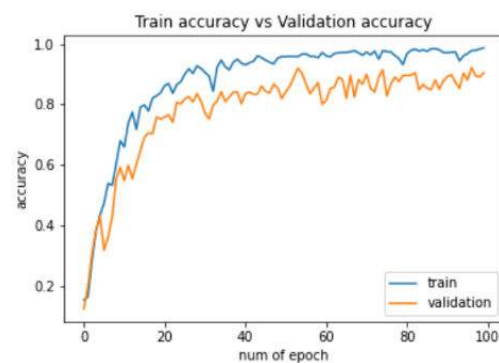


Fig. 5. Comparison of Training accuracy vs. validation accuracy

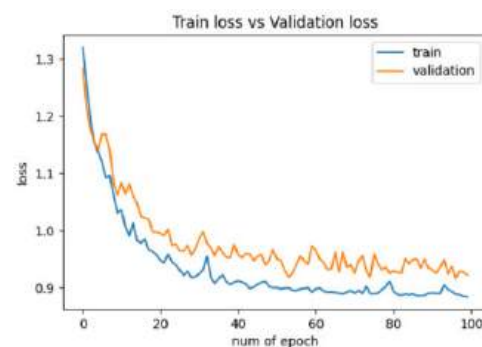


Fig. 6. Comparison of Training loss vs. validation loss

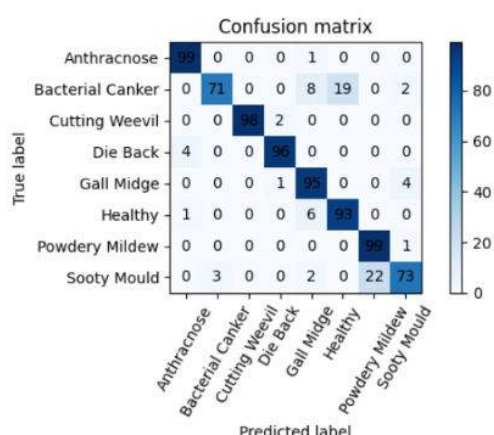


Fig. 7. Confusion matrix of DeepCONVSVM

DeepCONVSVM outperformed the other two approaches employed in this study in terms of performance. In Table IV, the performance of DeepCONVSVM has been compared with CNN and ResNet50. The proposed architecture outperformed the previous two models with an accuracy of 91%, precision of 91%, recall of 90%, and finally F1_score of 90.5%.

IV. CONCLUSION

Diseases need to be identified in the early state to increase mango production. To increase mango production, it is essential to identify diseases in their early state. Therefore, in this study, we have proposed a comparative model that identifies almost all kinds of common diseases in mango leaves with higher accuracy. For that, we have made our dataset that contains eight classes and then implemented various models which are resnet50, CNN, and finally, CNN with SVM. Among these models we have found that DeepCONVSVM performs much better with an accuracy of 91%. For future works, more diseases can be added to the classification for the model to have better detection for mango leaves. This model can also be utilized to create real-time mobile applications that quickly and effectively identify diseases. Therefore, this will halt the spread of illnesses, enhance fruit quality, and boost productivity.

REFERENCES

- [1] M. Merchant, V. Paradkar, M. Khanna, and S. Gokhale, "Mango Leaf deficiency detection using digital image processing and machine learning," in 2018 3rd International Conference for Convergence in Technology (I2CT), pp. 1–3, 2018.
- [2] S. Arya and R. Singh, "A comparative study of cnn and alexnet for detection of disease in potato and mango leaf," pp. 1–6, 09 2019.
- [3] M. Mia, S. Roy, S. Das, and M. Rahman, "Mango leaf disease recognition using neural network and support vector machine," Iran Journal Of Computer Science, vol. 3, 09 2020.
- [4] P. Kumar, S. Ashtekar, S. S. Jayakrishna, K. P. Bharath, P. T. Vanathi, and M. Rajesh Kumar, "Classification of mango leaves infected by fungal disease anthracnose using deep learning," in 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), pp. 1723–1729, 2021.
- [5] U. P. Singh, S. S. Chouhan, S. Jain, and S. Jain, "Multilayer convolutional neural network for the classification of mango leaves infected by anthracnose disease," IEEE Access, vol. 7, pp. 43721–43729, 2019.
- [6] P. Kumar, S. Ashtekar, S. Jayakrishna, K. Bharath, P. Vanathi, and M. R. Kumar, "Classification of mango leaves infected by fungal disease anthracnose using deep learning," in 2021 5th International

- Conference On Computing Methodologies and Communication (ICCMC), pp. 1723–1729, IEEE, 2021.
- [7] M. Merchant, V. Paradkar, M. Khanna, and S. Gokhale, "Mango Leaf deficiency detection using digital image processing and machine learning," in 2018 3rd International Conference for Convergence in Technology (I2CT), pp. 1–3, IEEE, 2018.
- [8] Y. Lu, S. Yi, N. Zeng, Y. Liu, and Y. Zhang, "Identification of rice diseases using deep convolutional neural networks," Neurocomputing, vol. 267, pp. 378–384, 2017.
- [9] J. G. Barbedo, "Factors influencing the use of deep learning for plant disease recognition," Biosystems engineering, vol. 172, pp. 84–91, 2018.
- [10] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, "A comparative study of fine-tuning deep learning models for plant disease identification," Computers and Electronics in Agriculture, vol. 161, pp. 272–279, 2019.
- [11] K. Golhani, S. K. Balasundram, G. Vadmalai, and B. Pradhan, "A Review of neural networks in plant disease detection using hyperspectral data," Information Processing in Agriculture, vol. 5, no. 3, pp. 354–371, 2018.
- [12] Z. Iqbal, M. A. Khan, M. Sharif, J. H. Shah, M. H. ur Rehman, and K. Javed, "An automated detection and classification of citrus plant diseases using image processing techniques: A review," Computers and electronics in agriculture, vol. 153, pp. 12–32, 2018.
- [13] A. Picon, A. Alvarez-Gila, M. Seitz, A. Ortiz-Barredo, J. Echazarra, and A. Johannes, "Deep convolutional neural networks for mobile capture device-based crop disease classification in the wild," Computers and Electronics in Agriculture, vol. 161, pp. 280–290, 2019.
- [14] L. Jain, H. Vardhan, M. Nishanth, and S. Shylaja, "Cloud-based system for supervised classification of plant diseases using convolutional neural networks," in 2017 IEEE International Conference on Cloud Computing In Emerging Markets (CCEM), pp. 63–68, IEEE, 2017.
- [15] L. Jain, H. Vardhan, M. Nishanth, and S. Shylaja, "Cloud-based system for supervised classification of plant diseases using convolutional neural networks," in 2017 IEEE International Conference on Cloud Computing In Emerging Markets (CCEM), pp. 63–68, IEEE, 2017.
- [16] J. Ma, K. Du, F. Zheng, L. Zhang, Z. Gong, and Z. Sun, "A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network," Computers and electronics in agriculture, vol. 154, pp. 18–24, 2018.
- [17] X. Zhang, Y. Qiao, F. Meng, C. Fan, and M. Zhang, "Identification of maize leaf diseases using improved deep convolutional neural networks," IEEE Access, vol. 6, pp. 30370–30377, 2018.
- [18] X. Zhang, Y. Qiao, F. Meng, C. Fan, and M. Zhang, "Identification of maize leaf diseases using improved deep convolutional neural networks," IEEE Access, vol. 6, pp. 30370–30377, 2018.
- [19] A. F. Agarap, "An architecture combining convolutional neural network (CNN) and support vector machine (SVM) for image classification," CoRR, vol. abs/1712.03541, 2017.
- [20] Y. Tang, "Deep learning using linear support vector machines," arXiv preprint arXiv:1306.0239, 2013.
- [21] I. Z. Mukti and D. Biswas, "Transfer learning based plant disease detection using resnet50," in 2019 4th International Conference on Electrical Information and Communication Technology (EICT), pp. 1–6, IEEE, 2019.
- [22] D. Theckedath and R. Sedamkar, "Detecting affect states using vgg16, resnet50 and se-resnet50 networks," SN Computer Science, vol. 1, no. 2, pp. 1–7, 2020.
- [23] S. Hijazi, R. Kumar, C. Rowen, et al., "Using convolutional neural networks for image recognition," Cadence Design Systems Inc.: San Jose, CA, USA, pp. 1–12, 2015.
- [24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778, 2016.

Unmasking Ovary Tumors: Real-Time Detection with YOLOv5

Nishat Vasker *

Department of Computer Science
and Engineering
East West University
Dhaka, Bangladesh
nishatvasker@gmail.com

Ab. Rahim Ahmed Sowrov

Department of Computer Science
and Engineering
East West University
Dhaka, Bangladesh
sajimahmed3030@gmail.com

Mahamudul Hasan

Department of Computer Science
and Engineering
East West University
Dhaka, Bangladesh
munna09bd@gmail.com

Md Sawkat Ali

Department of Computer Science
and Engineering
East West University
Dhaka, Bangladesh
alim@ewubd.edu

Mohammad Rifat Ahmmad Rashid

Department of Computer Science
and Engineering
East West University
Dhaka, Bangladesh
rifat.abir@gmail.com

Mohammad Manzurul Islam

Department of Computer Science
and Engineering
East West University
Dhaka, Bangladesh
mohammad.islam@ewubd.edu

Abstract—Ovary tumor is an unexpected growth of a women's ovary. For successful treatment planning and identification, ovary tumor detection is critical. The present medical method to detect ovary tumors is dependent on ultrasonic imaging, which can be constrained by its high cost, time-intensive nature, and dependence on specialized equipment and proficient personnel. With an advanced object detection algorithm, our system proposes an approach for the real-time diagnosis of ovarian tumors by utilizing YOLOv5. The proposed methodology obviates the need for ultrasound imaging by leveraging a machine learning model trained on a dataset comprising images of both healthy and malignant ovaries. Our system is able to precisely identify ovarian tumors through the analysis of real-time photos or video, using it as an alternative to ultrasound imaging. Our system can be easily used in a variety of medical applications. Consistent with the contemporary practice of remote patient care, our state-of-the-art technology for detecting ovarian tumors in real-time enables medical practitioners to diagnose such tumors from a distance, thereby facilitating prompt intervention and treatment. In addition, the integration of our approach into the increasing prevalence of robotic-assisted surgery may facilitate the autonomous detection of ovarian tumors during surgical procedures, thereby improving surgical accuracy and patient outcomes. In addition, the proposed methodology offers a cost-effective means of identifying ovarian neoplasms, particularly advantageous for socioeconomically disadvantaged populations that encounter challenges in obtaining costly ultrasound apparatus and remote medical evaluations. Our approach decreases patients' financial burden by removing the need for costly ultrasound scans and permitting remote medical consultations. Our study proposes a unique approach for detecting ovarian tumors in real time using YOLOv5. The system's capacity to detect ovarian tumor using real-time photos or video, as well as its application in remote medical settings and cost, making it a promising diagnostic tool for ovarian tumors. This study paves the path for better ovarian tumor identification, accessibility, and treatment results.

Index Terms—Real-time ovary tumor detection, YOLOv5, ultrasound alternative, remote ovary tumor detection, robotic

surgery, affordable ovary tumor detection, marginalized populations, cost-effective diagnosis.

I. INTRODUCTION

Ovarian tumor is a prevalent disease that causes significant morbidity and mortality globally. Oftentimes, early-stage malignancies lack discernible symptoms, resulting in delayed diagnoses and inferior treatment outcomes. Existing detection methods, such as pelvic examinations and transvaginal ultrasound, have cost, accessibility, and precision limitations. Moreover, the efficacy of the biomarkers currently employed for ovarian tumor detection varies. Thus, innovative approaches are required to enhance ovarian tumor early detection and prognosis evaluation. This research project intends to address the difficulties associated with ovarian tumor detection by proposing a novel method employing real-time imaging and sophisticated object detection algorithms. Specifically, we employ YOLOv5, a cutting-edge object detection algorithm, to detect ovarian malignancies in real time without ultrasound imaging. We can accomplish accurate and efficient detection of ovarian tumors from real-time images or recordings by training the model with a dataset containing both normal and tumor ovaries.

There are two objectives of this research. First, we intend to create a system that can detect ovarian lesions in real time and provides an alternative to current ultrasound-based techniques. The second goal is to show how flexible and potentially influential our approach is in many medical contexts. Our method can help with remote ovarian tumor identification in particular, enabling doctors to locate tumors and deliver medication at an appropriate period. Our suggested method provides an identification of ovarian tumors in practical means. In addition, our method has remarkable

significance and probable effects on ovarian tumor detection. With the help of this method, the use of pricey ultrasound scans can be reduced. In robotic-assisted operations, we can easily include this method. We can also use this method to improve surgical accuracy and patient outcomes. This could be especially beneficial for those who have limited access to pricey medical technologies, impoverished populations, and long-distance consultations. This technology can provide an advanced AI-based medical service that can aid a doctor to detect tumors more accurately and remotely diagnose a patient at the lowest possible cost. Moreover, the real-time detection feature enables our technology to early diagnoses, which can greatly improve patient survival rates and enhance treatment outcomes. This study aims to revolutionize the identification of ovarian tumors through the use of a real-time detection approach based on YOLO (You only look once) version 5. Thus, the research has great potential to make a positive impact on patient outcomes and contribute to advancements in ovarian tumor diagnosis.

II. RELATED WORKS

Several studies have been conducted in the field of identification and classification of objects based on deep learning. Oksuz et al. (2020) utilized the YOLOv2 model to localize brain tumors with encouraging results. Ahmed (2021) devised a convolutional neural network (CNN) based on deep learning for detecting real-time crevices. Hossain et al. (2022) presented a deep learning model based on YOLOv5 for classifying and detecting brain abnormalities in a portable microwave-based imaging system. Xie et al. (2022) proposed a lightweight YOLOv5-based framework fused with a Coordinate Attention mechanism for SAR ship detection in the object detection domain. Combining FIRST radio and SDSS optical image data, Zhang et al. (2022) developed a deep-learning process model that can automatically find out and categorize the images of a radio galaxy. Li et al. (2022) concentrated on an object detection system that can sense low-altitude UAV remote images by utilizing a modified YOLOv5 model. Using drones, they addressed the challenge of real-time detection.

Comparing the efficacy of YOLOv5, YOLOv6, and YOLOv7, Yung et al. (2022) examined safety helmet detection using deep learning. Li et al. (2022) proposed a system to count the seedling maize based on YOLOv5 and an algorithm monitoring the Kalman filter for the agricultural domain. They were able to accurately detect and count maize plants in complex environments. Li et al. (2023) investigated the detection of gray mold spores on cucumber using microscopic images and deep learning, and they introduced the MG-YOLO detection algorithm. In addition, Yang et al. (2022) investigated a refuse classification system based on deep learning that integrates image classification and object detection. Their efforts centered on obtaining precise classification and detection of various categories of trash.

Ananda-Rajah (2019) implemented a fungal Ai-based technology for antifungal stewardship in hematological-oncology patients with the intention of enhancing surveil-

lance and decision-support practices. Chen (2019) developed a computer-aided detection system for colorectal adenoma during colonoscopic examinations using deep learning.

Whitney (2019) examined the effect of visual sequential dependence on clinical visual search and proposed methods to reduce perceptual errors. Limerutti (2020) investigated the use of deep learning for COVID classification of chest X-ray images. Benitez (2021) sought to validate the performance of an AI tool for automatic chest X-ray classification. Based on multimodal data integration, Jin (2021) developed an AI-based procedure that can predict the effects of the drug and interpret digital pathological diagnoses. Liao (2022) developed an algorithm weakly supervised for identifying and locating traumatic splenic injuries with the help of three-dimensional deep learning technology. He conducted a study on the prognosis and antiplatelet strategies for percutaneous coronary intervention (PCI) patients with a high risk of hemorrhage. Jin (2022) sought to create an AI system for tumor risk assessment, diagnosis, and treatment result prediction via multimodal data integration. Jia (2020) worked on a clinical trial using imaging genomics for molecular characterization and risk prediction of breast tumors using machine learning.

These clinical studies show how artificial intelligence, especially deep learning, is being used in a range of healthcare disciplines, including infectious illnesses, imaging, pathology, and treatment outcomes.

III. METHODOLOGY

This section describes the study methods, including the usage of the YOLOv5 model, the dataset utilized, and the performance measures used for assessment.

A. YOLOv5

YOLOv5 is one of the most popular computer vision model in the world that represents open-source research into future vision AI methods. YOLOv5 is also well known for high-performance object identification and track record in object detection. Figure 1 depicts the algorithm's three key steps.

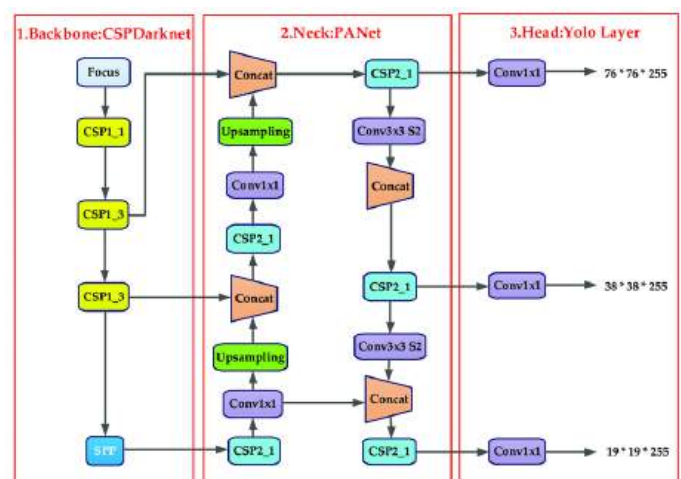


Fig. 1. YOLOv5 Architecture

The first stage comprises feature extraction, with the CSP network serving as the underlying architecture. By including gradient changes into the feature map, the CSPNet tackles the issue of duplicate gradient information in large-scale backbones. The number of model parameters is reduced while extraction speed and accuracy are maintained. The Path Aggregation Network (PANet) is employed in the subsequent stage of improving the global information flow. A feature pyramid network architecture and dynamic feature pooling employs by PANet to enable the exchange of relevant information across many levels of an object's hierarchy. This improves object localization and estimate accuracy. In the third and final phase, a prediction model based on the YOLO head from YOLOv3 is used. The YOLO provides us with a layer of feature maps that can varying size, allowing for multiscale object recognition and easier identification of items of diverse sizes. To improve early detection, accessibility, and affordability of ovarian tumor diagnostics, our suggested solution makes use of powerful object detection algorithms.

B. Dataset

The study made use of the Ovary Tumor Dataset, which had 5,000 photos divided into normal ovary (1,000 images) and ovary tumor (4,000 images). All the data was collected from Google. Those data are publicly available on the internet. This data set ensures a diversified representation within each class. Figure 2 shows an example from the normal ovarian class, whereas Figure 3 shows the images of an ovary tumor.

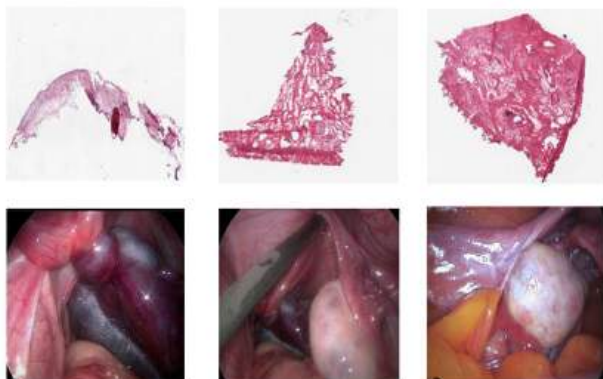


Fig. 2. Normal Ovary Image Dataset Sample

To improve accuracy, we used both diseased and healthy ovaries data. Finally, the model was improved in a way that it can accurately detect ovarian tumors in real-time photos and videos

C. Performance Metrics

Several performance measuring metrics were used to assess the success of the object detection algorithm including Precision (Prec), Average Precision (AP), mean Average Precision (mAP), Recall, and F1 Score. These measurements help us to

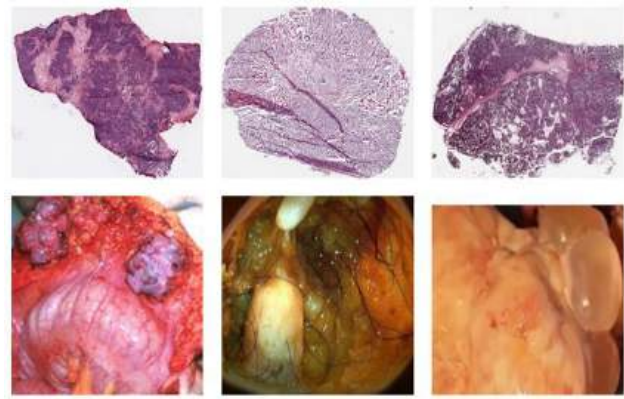


Fig. 3. Ovary Tumor Image Dataset Sample

understand how the object detection models are performing. The formulae [5, 6] are used to calculate these metrics:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad (1)$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad (2)$$

$$\text{F1 Score} = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}} \quad (3)$$

$$\text{AP} = \frac{\sum(\text{Precision} \times \text{Recall})}{\text{Recall}} \quad (4)$$

$$\text{MAP} = \frac{\sum(\text{AP})}{\text{Number of recalls}} \quad (5)$$

The performance of this model is supervised regularly to improve accuracy. The effectiveness of the model has been calculated in numerous scenarios to measure its precision, speed, and resilience

D. Integration of Real-Time Ovary Tumor Detection:

A responsive website was created to allow for real-time evaluation of ovarian cancers. The website combines the learned YOLOv5 object identification model, allowing users to upload photographs and obtain quick estimates about the existence of ovarian tumors. To give a user-friendly interface, the website employs JavaScript, HTML, CSS, and Bootstrap. When a picture is submitted, it is processed by the YOLOv5 model using the implemented API key. The model detects objects in the picture and provides bounding boxes around any detected ovarian tumor. The predictions are then displayed on the website, highlighting the identified areas of concern.

This real-time ovary tumor detection functionality enhances the practicality and accessibility of the research, enabling users to quickly evaluate the presence of ovary tumors in images without the need for complex software setups. It can be a useful tool for healthcare providers, researchers, and those worried about ovarian tumor.

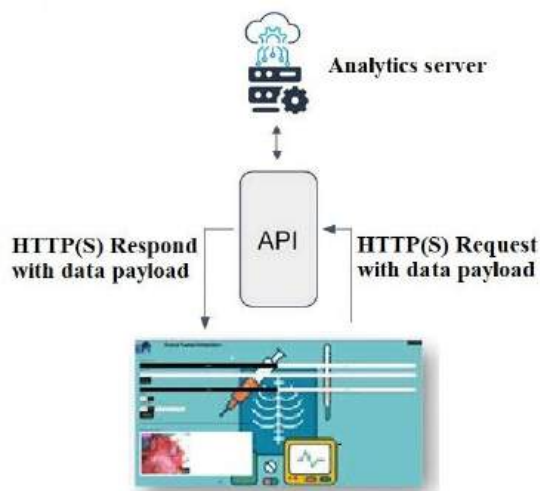


Fig. 4. Architecture of Real-Time Ovary Tumor Detection with Responsive Website

IV. EXPERIMENTAL RESULTS

Our endeavor was carried out on a solid experimental platform running a 64-bit version of Windows 10. High-performance system specs were an NVIDIA GeForce 950M graphics card, an Intel i7-7500U CPU running at 2.7 GHz, and 16 gigabytes of random access memory (RAM). These strong components meant that our object identification tests had the best possible computing capabilities. We used transfer learning, a commonly used approach, to improve the performance of our object identification models. The experiment employed YOLOv5 algorithms (s, m, l, and x) that were trained on the extensive 80-class MSCOCO dataset. By fine-tuning these models on our two-class dataset of normal ovary and ovarian tumor photos, we effectively translated information from the larger dataset to our specialized goal. This method dramatically increased the detection accuracy of our models. To effectively train and assess our models, we carefully separated our dataset into three independent subsets: training, validation, and testing. Approximately 70% of the dataset, which included 1000 normal ovary photos and 4000 ovarian tumor images, was dedicated to training the models. This enabled them to learn from a wide range of instances. A further 20% of the dataset was committed to the validation subset, allowing us to track the models' performance throughout training and fine-tune hyperparameters as needed.

TABLE I
PARAMETERS OF MODELS

Variable	Value
Image resolution	640x640
Learning Rate	0.01
Weight Reduce	0.0005
Momentum	0.937
Batch capacity	2
Repetition Number	20

Finally, the remaining 10% of the dataset was put aside for testing the final models on previously undisclosed data, allowing for a thorough assessment of their effectiveness. Each YOLOv5 model version was rigorously designed with a unique set of settings. Table I lists the settings that were carefully chosen to achieve the best balance of detection accuracy and processing efficiency. By tailoring the parameters for each model variant, we aimed to maximize their accuracy and resource utilization.

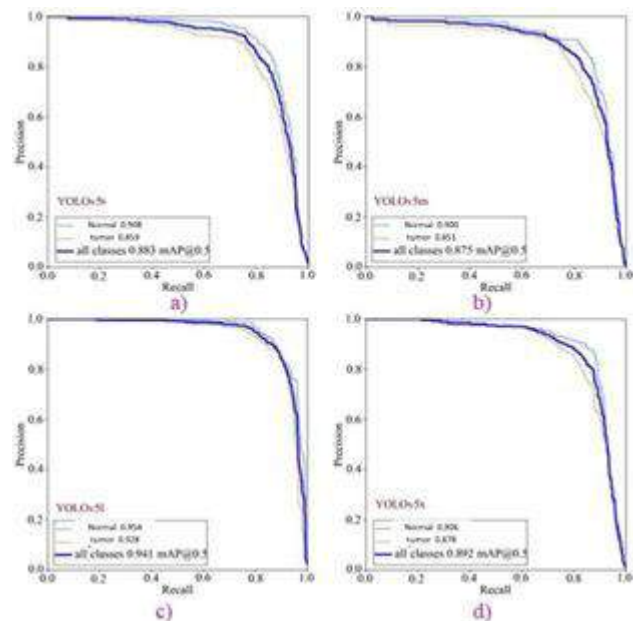


Fig. 5. Precision-Recall Curves of YOLOv5s, m, l, and x.

Precision-Recall curves had been plotted to analyze the effectiveness of the models, as depicted in Figure 5. All training and testing circumstances were the same for all four models. The mean Average Precision (mAP) for the YOLOv5s model was determined to be 0.883. According to the results, YOLOv5x, the extra-large version, had a higher mAP (mean average precision) of 0.892 than earlier versions. In a similar vein, YOLOv5m had a mAP of 0.875. The mAP score of 0.941 for the YOLOv5l variant shows that it offers the best performance overall. Results for each model's AP, mAP, Precision, Recall, and F1 Score are summarized in Table II.

TABLE II
PERFORMANCE VALUES OF DIFFERENT MODELS

Model	Normal	Tumor	mAP	Precision	Recall	F1-Score
YOLOv5s	0.908	0.859	0.883	0.900	1.00	0.94
YOLOv5m	0.900	0.851	0.875	0.872	0.99	0.92
YOLOv5l	0.954	0.928	0.941	0.872	1.00	0.93
YOLOv5x	0.906	0.878	0.892	0.893	1.00	0.94

Furthermore, the study conducted a comparative analysis of various models in terms of their time efficiency. Table III presents a comparative analysis of the inference times of the YOLOv5s and YOLOv5x models. The research revealed that the YOLOv5s model exhibited the most efficient infer-

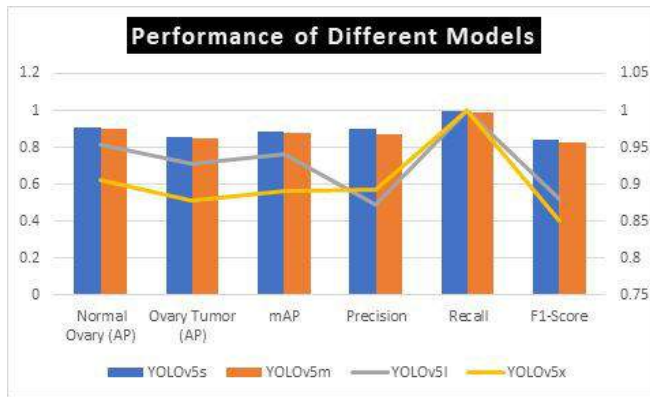


Fig. 6. Graph of Performance Values of Different Models

ence time, while the YOLOv5x model demonstrated the least efficient performance. The YOLOv5l model demonstrated superior performance with regards to detection, as depicted in Figure 6. The model's aptitude in object classification and localization is evidenced by its capacity to differentiate between a benign ovary and a malignant one in a sequence of images.

This study also examined the temporal efficiency of various models. Table III presents a comparative analysis of the inference times of the YOLOv5s and YOLOv5x models. The study conducted measurements of inference times and determined that the YOLOv5s model exhibited the shortest duration, while the YOLOv5x model demonstrated the longest duration.

TABLE III
VALUES OF TIME-PERFORMANCE FOR VARIOUS MODELS

Model	Test Time (5000 images)
YOLOv5s	32.548 s
YOLOv5m	47.123 s
YOLOv5l	78.215 s
YOLOv5x	123.987 s

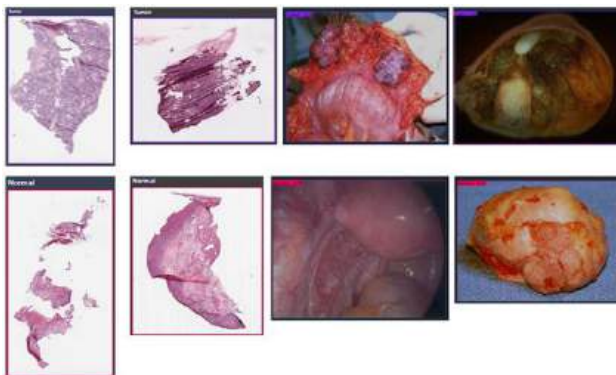


Fig. 7. Detection Results

Figure 7 illustrates the detection results achieved by the YOLOv5l model, which demonstrated superior performance. The model accurately classified and localized objects, as

demonstrated by the correctly identified normal ovary and ovary tumor in the images.

V. CONCLUSION

This research effort showed that a real-time ovarian tumor detection system employing the YOLOv5 object identification algorithm is feasible and impactful. We detected ovarian tumor from real-time pictures or video by training the algorithm using a large dataset of normal and malignant ovaries. The suggested technology has many benefits over ultrasound-based methods. It eliminates the need for expensive and time-consuming ultrasound imaging, making it cost-effective for underprivileged communities without access to expensive medical equipment and distant consultations. Our technology decreases patient costs and improves diagnosis by making ovarian tumor detection cheaper. Our system's versatility expands its medicinal applications. Our real-time ovarian tumor detection technology lets doctors remotely detect and treat tumors. This is especially useful when medical facilities are few. Our method automatically detects ovarian tumor during robotic-assisted operations, improving accuracy and patient outcomes. The YOLOv5 algorithm was trained on a broad dataset of normal and malignant ovaries to recognize ovarian lesions in real-time pictures and videos. The system's performance evaluation showed precision, speed, and resilience across scenarios. This research effort intends to revolutionize ovarian tumor diagnosis with a real-time YOLOv5-based detection system. Ovarian tumor identification is enhanced by the system's real-time imaging, remote medical flexibility, and cost. This technology might improve early detection, quick diagnosis, treatment results, and patient survival rates. This research effort improves ovarian tumor diagnosis, early detection, accessibility, and treatment results. Ovarian tumor detection might change with more study and development.

VI. REFERENCES

- [1] Oksuz, C., Gullu, M. K. (2020). Brain Tumor Localization Using YOLO V2. *28th Signal Processing and Communications Applications*, 2020.
- [2] Ahmed, K. R. (2021). Smart Pothole Detection Using Deep Learning Based on Dilated Convolution. *Sensors (Basel, Switzerland)*, 2021. (IF: 3)
- [3] Hossain, A., Islam, M. T., Almutairi, A. F. (2022). A Deep Learning Model to Classify and Detect Brain Abnormalities in Portable Microwave Based Imaging System. *Scientific Reports*, 2022.
- [4] Xie, F., Lin, B., Liu, Y. (2022). Research on The Coordinate Attention Mechanism Fuse in A YOLOv5 Deep Learning Detector for The SAR Ship Detection Task. *Sensors (Basel, Switzerland)*, 2022. (IF: 3)
- [5] Zhang, Z., Jiang, B., Zhang, Y. (2022). Automatic Detection and Classification of Radio Galaxy Images By Deep Learning. *Publications of the Astronomical Society of the Pacific*, 2022.
- [6] Li, Z., Namiki, A., Suzuki, S., Wang, Q., Zhang, T., Wang, W. (2022). Application of Low-Altitude UAV Remote

Sensing Image Object Detection Based on Improved YOLOv5. *Applied Sciences*, 2022.

[7] Yung, N. D. T., Wong, W., Juwono, F. H., Sim, Z. A. (2022). Safety Helmet Detection Using Deep Learning: Implementation and Comparative Study Using YOLOv5, YOLOv6, and YOLOv7. *2022 International Conference on Green Energy, Computing*, 2022.

[8] Li, Y., Bao, Z., Qi, J. (2022). Seedling Maize Counting Method in Complex Backgrounds Based on YOLOV5 and Kalman Filter Tracking Algorithm. *Frontiers in Plant Science*, 2022.

[9] Yang, Z., Bao, Y., Liu, Y., Zhao, Q., Zheng, H., Bao, Y. (2022). Research on Deep Learning Garbage Classification System Based on Fusion of Image Classification and Object Detection Classification. *Mathematical Biosciences and Engineering: MBE*, 2022.

[10] Li, K., Zhu, X., Qiao, C., Zhang, L., Gao, W., Wang, Y. (2023). The Gray Mold Spore Detection of Cucumber Based on Microscopic Image and Deep Learning. *Plant Phenomics (Washington, D.C.)*, 2023.

[11] Ananda-Rajah, M. D. (2019). Innovative Use of FungalAi for Antifungal Stewardship in Haematology-oncology Patients. 2019. (IF: 5)

[12] Chen, P. J. (2019). Computer-aided Detection With Deep Learning for Colorectal Adenoma During Colonoscopic Examination. 2019. (IF: 5)

[13] Whitney, D. (2019). Isolating and Mitigating Sequentially Dependent Perceptual Errors in Clinical Visual Search. 2019. (IF: 5)

[14] Limerutti, G. (2020). Covid Radiographic Images Dataset for A.I. 2020. (IF: 6)

[15] Jia, G. Y. (2020). Clinical Study of Imaging Genomics Based on Machine Learning for Breast Cancer Molecular Typing and Risk Prediction (BCIG). 2020. (IF: 5)

[16] Benitez, S. E. (2021). Multicenter Validation Study of An Artificial Intelligence Tool for Automatic Classification of Chest X-rays. 2021. (IF: 3)

[17] Jin, Y. (2021). Research and Development of An Artificial Intelligence Technology System for Digital Pathological Diagnosis and Therapeutic Effect Prediction Based on Multimodal Data Fusion of Common Tumors and Major Infectious Diseases in The Respiratory System Using Deep Learning Technology. 2021. (IF: 5)

[18] Liao, C. H. (2022). The Three-dimensional Weakly Supervised Deep Learning Algorithm for Traumatic Splenic Injury Detection and Sequential Localization. 2022. (IF: 4)

[19] He, Y. (2022). Prognosis and Antiplatelet Strategies for Patients With PCI and High Bleeding Risk: A Study Protocol. 2022. (IF: 5)

[20] Jin, Y. (2022). Development of An Artificial Intelligence System for Assessment of Tumor Risk and Diagnosis and Treatment Based on Multimodal Data Fusion Using Deep Learning Technology. 2022. (IF: 6)

Performance Evaluation of Intrusion Detection System Using Machine Learning and Deep Learning Algorithms

Md. Sabbir Hossain, Dipayan Ghose, All Masror Partho, Minhaz Ahmed, Md. Tanvir Chowdhury, Mahamudul Hasan,

Md Sawkat Ali, Taskeed Javid, Maheen Islam,

Department of Computer Science and Engineering,

East West University, Dhaka, Bangladesh

Email: {sabbirhossain1338, ghosedipayan, Masror.partho, minhazahmed39, mdtanvirchowdhury015, munna09bd}@gmail.com,

{alim, taskeed, maheen}@ewubd.edu

Abstract- Now that Internet access is so widely used, our society has a greater number of networked technologies. Data travels between them because of their daily activities. Due to the server's weaknesses, hackers may get access to the system through difficult-to-identify network breaches. One of the most well-known defense mechanisms against these attacks on networked devices is the Intrusion Detection System (IDS), which is built into the system. IDS has previously received extensive training in the classification of threats using traditional machine learning-based models and pre-assembled datasets. In this research, we presented two deep learning-based models, the Multilayer Perceptron Model (MLP) and Long-Short Term Memory (LSTM), along with five machine learning-based models, including Naive Bayes (NB), Decision Tree (DT), K-Nearest Neighbor (KNN), Random Forest (RF), and Support Vector Machine (SVM). The NSL-KDD dataset has been used to achieve 89.6% accuracy with normalization and 89.2% without normalization, 97.77% with LSTM and 96.89% with MLP. Each record in the data collection has 43 features, including two labels and 41 features that are related to traffic input.

Index Terms- Intrusion Detection System (IDS), KNN, LSTM, MLP, Classification, Accuracy

I. INTRODUCTION

Researchers looked into how machine learning (ML) and deep learning (DL) techniques could be used to create an intrusion detection system (IDS) that could satisfy contemporary network security needs. The advancement of technology and the shift towards online transactions have led to an increase in network and endpoint attacks, posing risks to data integrity, confidentiality, and availability. The researchers emphasize that while traditional security measures such as access control, password protection, and firewalls are important, they are not sufficient to protect against sophisticated intrusions. Hence, IDS is employed as a real-time monitoring system that can identify suspicious activities and send warnings when unauthorized access or malicious attacks occur.

Machine learning and deep learning are subfields of artificial intelligence (AI) that are well suited for analyzing massive volumes of data and extracting meaningful information. These techniques enable the IDS to accurately predict both typical and deviant actions based on learned patterns from network traffic. The study covers various aspects of network intrusion risks, including conventional

and rule-based procedures, as well as innovative machine learning and deep learning techniques. By leveraging these advanced approaches, the IDS can enhance the detection and prevention capabilities, thereby strengthening the overall cybersecurity posture. In conclusion, the researchers suggest using deep learning and machine learning approaches to create an efficient intrusion detection system. This approach leverages the power of AI to analyze network traffic, detect anomalies, and provide timely warnings, thereby improving the security of online transactions and protecting against unauthorized access and attacks.

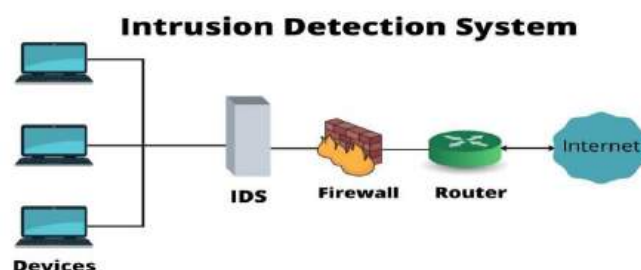


Fig.1. Intrusion Detection System

II. BACKGROUND AND RELATED WORK

A. Machine Learning Algorithm

Since the beginning of civilization, people have employed a variety of tools to complete a range of tasks in ways that are more practical and less complicated to meet their needs. Thanks to the human mind's capacity for creation, a wide range of tools and objects have been made. By enabling people to meet a range of needs, such as those related to travel, industry, and computing, these devices facilitate the ease of human living. Machine learning is the most extreme and stands out nowadays [3]. These gadgets made it easier for people to live their lives by helping to address a variety of needs, including those pertaining to travel, business, and computing. The most notable of these is machine learning [3]. Making it feasible for computers to learn without being explicitly instructed is the aim of the area of computer science known as machine learning, according to Arthur Samuel [4]. The focus of machine learning techniques is function approximation issues, where the aim is represented by a function, and the learning problem is to improve the accuracy of that function using experience from a sample of input-output pairs that are well-known for the function [5]. As a result, the primary challenge of supervised learning is a lack of enough labeled data. Unsupervised learning, on the other hand, draws out relevant feature information from unlabeled data, greatly increasing the accessibility of training material. In terms of detection, supervised learning

methods frequently beat unsupervised learning methods [10].

Supervised Learning Algorithm:

A collection of data is used as input in supervised learning, and a machine learning model is used to identify a connection between the feed and the outcome. The two categories are regression and classification.

1. **Support Vector Machine (SVM):** In SVMs, the objective is to locate a max-margin separation hyperplane in an n-dimensional feature space. SVMs can produce good results even with limited training sets because the separation hyperplane only requires a small number of support vectors to be set. SVMs, however, are noise-sensitive close to the hyperplane. SVMs excel at handling linear issues. Kernel functions are usually used for data that is not linear [10]. The kernel technique translates inputs implicitly to high-dimensional feature spaces. It essentially defines the boundaries between the classes. The margins are designed to maximize the distance between the margin and the classes, hence decreasing classification error. Because SVMs and other machine learning methods frequently use kernel techniques [3].
2. **Naïve Bayes (NB):** The Naive Bayes method is a classification algorithm based on attribute independence and conditional probability. The example is put in the outcome class with the highest probability [10]. Because it simplifies the assumptions about the qualities, it is referred to as "naive". A probabilistic algorithm is what the Naive Bayes algorithm is known as, and its formula is.

$$\frac{c_j}{p(x_j)} = \frac{p(x_j/c_j)p(c_j)}{p(x_j)}$$

3. **Decision tree (DT):** One of the main methods for supervised machine learning, DT applies a set of decisions (rules) to classify and predict data using both regression and classification. A typical tree structure with nodes, branches, and leaves is included in the model [6]. The simplest classifier is the decision tree. The extreme gradient boosting (XGBoost), which consists of several decision trees with parent and root nodes, and random forest are examples of advanced techniques [12].
4. **Random Forest (RF):** Random Forest is a supervised learning technique that generates training sample sets using the Bagging (Bootstrap aggregation) technique and creates multiple decision trees when a new set of data is input. When this new set of samples is supplied, each decision tree in the forest makes a prediction on it separately, and then the predictions of all the trees are combined to provide a result [12]. Most of the time, even without the use of a hyperparameter, it is possible to obtain acceptable results. It is one of the most favored techniques due to the speed and

accuracy with which it produces results, even for mixed, incomplete, and noisy datasets [13].

Unsupervised Learning Algorithm:

Unsupervised learning models identify hidden patterns without human assistance but may require human involvement.

1. **K-Nearest Neighbor (KNN):** The manifold theory serves as the foundation for KNN's main idea. If most of its neighbors also fall into that class, there is a substantial likelihood that the sample will as well. As a result, only the k closest neighbors are associated to the classification outcome. The parameter k has a big impact on how well KNN models perform. The likelihood of overfitting increases as k decreases because the model becomes more complex. On the other hand, as the work grows, the model gets simpler and loses its capacity to fit data.

B. Deep Learning

Artificial neural networks are used in deep learning, a branch of machine learning, to simulate human thought and learning. The analysis of Big Data, picture classification, language translation, and speech recognition all use it nowadays. Data scientists who gather, examine, and understand vast volumes of data can also benefit from it. A deep learning artificial neural network applies signals to nodes using weights to produce outputs. It can identify data using binary true or false queries, but it also needs robust hardware and data sets. Due to its capacity to grow and learn through time, it has recently become more and more pertinent. Over time, the facial recognition program will successfully recognize faces in this scenario [15]. While first proposed in the 1980s, deep learning has only recently gained popularity for two reasons.

1. Deep learning needs lots of data that has been classified. For instance, millions of images and many hours of video are needed to create self-driving cars.
2. For deep learning, a lot of processing power is required. High-performance GPUs are ideal for deep learning, allowing systems to group data and provide precise predictions by taking cues from the human brain. Deep learning algorithms conduct logical analyses of data to draw conclusions that are comparable to those of humans [16].

C. Deep Neural Network

An artificial neural network is the foundation of the advanced machine learning technology known as deep learning. It needs a lot of data to learn on, and it labels the information utilized during training. Only once a deep learning model has been trained and achieved an acceptable level of accuracy can it interpret unstructured data. Neural networks are made up of interconnected nodes, called neurons, which are based on our brain's organic neurons. Weights are used to create connections between neurons, and each node has a weight and threshold associated with it. Artificial neural networks can be used to swiftly classify and cluster data, and training data is utilized to learn and improve accuracy. One of the most well-known neural networks is the one that powers Google's search engine.

1. **Perceptron:** Perceptron is a single neuron that processes input values and transfers them to an activation function to generate binary output.
2. **Feedforward neural networks (MLP):** The neurons and hidden layers make up Feed Forward (FF) neural networks, which move data forward without backpropagation. Flow control begins at the input level and goes to the output level, allowing for customization of weights and improved learning FF neural networks are used in classification, speech recognition, face recognition, pattern recognition.
3. **Multi-layer perceptron's (MLPs):** Multi-layer perceptron's, which can be used for multi-class and binary classification, are bi-directional, with inputs propagating forward and weight changes propagating backward.
4. **Recurrent neural networks (RNN):** parallel neuronal systems (RNNs) are deep learning techniques used in popular applications such as Siri, voice search, and Google Translate. RNNs use a Hidden Layer to remember specific information about a sequence. They use the same parameters for each input, reducing the complexity of the parameters. The primary drawback of RNN is the Vanishing Gradient problem, which makes it impossible to remember the weights of earlier layers.
5. **Long Short-Term Memory Networks (LSTM):** Long Short-Term Memory Networks (LSTMs) are a variation of recurrent neural networks that can be used to tackle the Vanishing Gradient problem. LSTMs can identify long-term dependencies and use gates to decide which outputs should be used and which should be ignored. The input gate determines which data should be kept in memory, while the output gate regulates the data transferred to the following layer. LSTMs are used in various applications such as: gesture recognition, speech recognition, text prediction.

III. METHODOLOGIES

A. Research Analysis

This section of the proposed approach discusses the chosen approaches for supporting or analyzing a set of data or a desired case. In order to predict results, the effective model collects a dataset, analyzes it, and then applies machine learning and deep learning algorithms to it. In this research proposed approach, we will investigate an ABC dataset utilizing some of the most popular machine learning methods as well as some proven deep learning techniques. The algorithms used were Support Vector Machine, Naive Bayes, Decision Tree, Random Forest Classifier, MLP, and LSTM. The suggested model's schematic process diagram is shown in Fig. 2. Following the preprocessing of the acquired dataset, observations were produced, paving the way for the feature selection technique's identification of key features. The dataset is divided into train and test sets after the class imbalance issue has been addressed. In order to train the models, classifiers are fed the training dataset. Making predictions using test instances is the next step in evaluating

Identify applicable funding agency here. If none, delete this text box.

the trained models' performance.

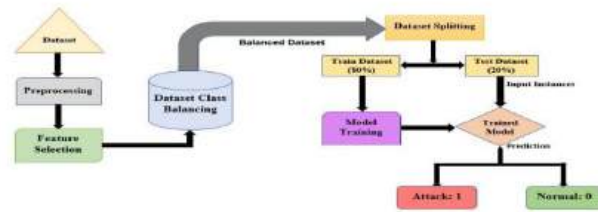


Fig.2. Workflow Diagram

B. Data Overview

The NSL-KDD dataset enhances KDD'99 for testing intrusion detection methods. It has training and testing sets: KDDTrain+, KDDTest+, and KDDTest-21. They include attack-type labels in CSV. DoS, U2R, R2L, probing, and normal instances are covered. These attacks fall into surveillance, DoS, and probing categories. U2R is unauthorized local superuser access, R2L is unauthorized remote access. Our study evaluated methods using DoS attacks. NSL-KDD is recommended over KDD Cup'99 for addressing issues. It's vital for building intrusion detection systems and studying cybersecurity. Industry also uses datasets like ADFA-ID, ISCX-UNB. Let's explore NSL-KDD's improvements.

Data Preprocessing

Because noisy and contradictory data can result in a fatal mistake [27], it is preprocessed using the techniques described in it goes through preprocessing using the methods outlined in Fig.2.

Due to extraction or input problems, a portion of the dataset comprises some noisy data, duplicate values, missing values, infinite values, and so on. As a result, we begin by preprocessing the data. The following is the basic fundamental.



Fig.3. Different preprocessing phases

IV. EVALUATION OF MODELS

A. Evaluation Metrics

To assess our model's performance, we utilize accuracy. We also discuss the false positive rate and the detection rate. The number of records that are appropriately rejected and identified as anomalies is referred to as True Positive (TP). True Negative (TN) signifies the opposite. True Negative (TN) indicates normal records, whereas False Negative (FN)

indicates the opposite. The following measures are used to assess the effectiveness of self-taught learning:

1) **Accuracy:** It is defined as the proportion of properly categorized records among all records.

2) **Precision:** The proportion of records with true positives (TP) to all records with true positives and false positives (FP) is calculated.

3) **Recall:** The ratio of true positive records to all true positives and false negative (FN) category records, expressed as a percentage.

4) **F-measure:** Precision and recall's harmonic mean is characterized as a balance between the two.

B. Machine Learning Model Performance without Normalization

Five different machine learning algorithms—Naive Bayes (NB), Decision Tree (DT), K-Nearest Neighbors (KNN), Random Forest (RF), and Support Vector Machine (SVM)—have been used in this study. We have determined the accuracy, precision, recall, and f1-score of each algorithm. The following are the machine learning algorithms results shown in Table 1:

TABLE I. Without Normalization results

Algorithms	Accuracy	Precision	Recall	F1-Score
	(%)	(%)	(%)	(%)
Naive Bayes (NB)	80.3	97	58	72
Decision Tree(DT)	88.6	97	88	92
K-Nearest Neighbors (KNN)	73.5	90	73	80
Random Forest(RF)	89.2	96	86	91
Support Vector Machine (SVM)	88.4	89	88	88

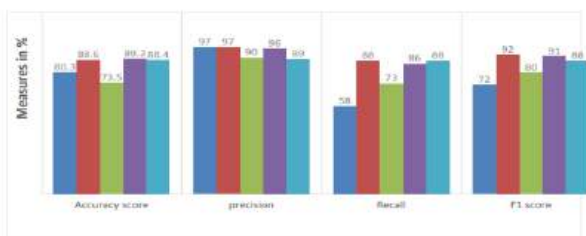


Fig.4. Plotting of Machine Learning Algorithms Results(Without Normalization)

So, from our machine learning of all algorithms, we can see that the Random Forest (RF) method gave us the most accuracy, 89.2%, and the KNN algorithm gave us the lowest accuracy, 73.5%. Now, if we look at the F1 score for machine learning algorithms, we can find that the DT method has the best score (92%), and the Naive Bayes (NB) algorithm has the lowest score (72%). From our table and charts, we can also observe the precision and recall values for the machine learning methods.

C. Machine Learning Model Performance with Normalization

TABLE II. With Normalization Results

Algorithms	Accuracy	Precision	Recall	F1-Score
	(%)	(%)	(%)	(%)
Naive Bayes (NB)	75.9	98	53	69
Decision Tree(DT)	88.2	97	87	92
K-Nearest Neighbors (KNN)	87.0	96	85	90
Random Forest(RF)	89.6	97	87	92
Support Vector Machine (SVM)	87.6	87	89	88

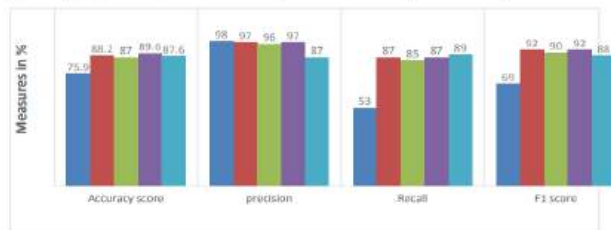


Fig.5. Plotting of Machine Learning Algorithms Results (With Normalization)

So, from our machine learning of all algorithms, we can see that the Random Forest (RF) method gave us the best accuracy, which was 89.6%, while the NB algorithm gave us the lowest accuracy, which was 75.9%. Now, if we take F1 score into account for machine learning algorithms. We can see that the Naive Bayes (NB) algorithm gave us the lowest score, 69%, while both the DT and RF algorithms gave us the greatest score, 92%. From our table and charts, we can also observe the precision and recall values for the machine learning methods.

D. Deep Learning Model Performance

We have employed the Long-Short Term Memory (LSTM) and the Feed Forward Neural Network (MLP) as two separate deep learning techniques. All these algorithms' training accuracy, testing accuracy, precision, recall, and f1-score have been calculated. The following is the deep learning algorithms results shown in the Table 3:

TABLE III. Deep Learning Algorithms Results

Algorithms	Accuracy	Precision	Recall	F1-Score
	(%)	(%)	(%)	(%)
Multilayer perceptron Model (MLP)	96.89	97.22	96.91	97.06
Long-Short Term Memory (LSTM)	97.77	97.23	98.62	97.92

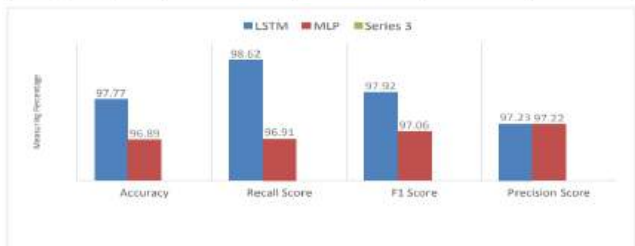


Fig.6. Plotting of Deep Learning Algorithms Results

Now, if we consider When using deep learning techniques, we can find that the algorithm's accuracy was 97.77%, while the MLP algorithm's training accuracy was 96.89%. Additionally, we achieved a top f1 score of 97.23% using

the LSTM algorithm. From our table and charts, we can also examine the precision and recall numbers for deep learning algorithms.

E. Comparison Between ML (Normalization) & DL Algorithms Result

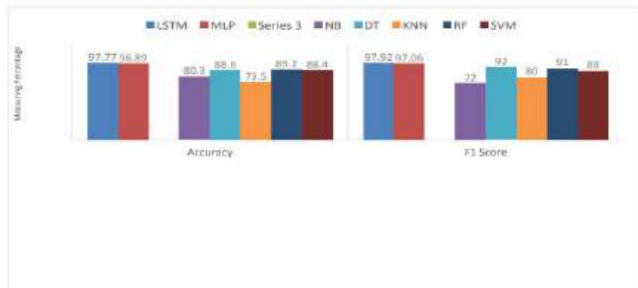


Fig.7. ML & DL Algorithms Classification

We can see from the plot above that there are two deep learning algorithms and five machine learning algorithms. The deep learning algorithms are LSTM and MLP, whereas the machine learning methods are RF, DT, SVM, NB, and KNN. We can see that RF, a machine learning algorithm, gave us the best accuracy out of all the methods, albeit deep learning algorithms also produced better results. From the KNN, a machine learning algorithm, we obtained the lowest accuracy.

F. Deep Learning Model Accuracy and Loss Curve (LSTM)

We can now observe the accuracy vs. epoch plotting curve for deep learning algorithms as well as the loss vs. epoch charting curve for train and test datasets. The accuracy of the LSTM algorithm for the train and test datasets is plotted as a function of epoch in the chart below.

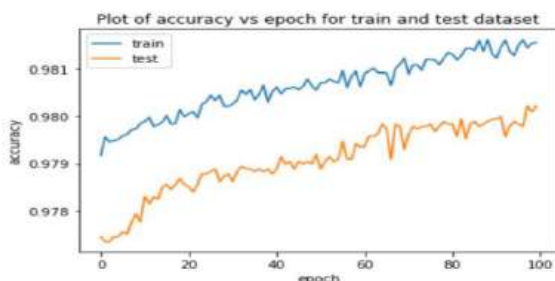


Fig.8. Plot of LSTM Algorithm of Accuracy Vs Epoch for Train and Test Dataset

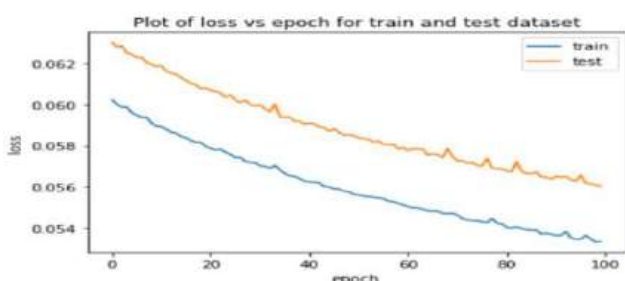


Fig.9. Plot of LSTM Algorithm of Loss Vs Epoch for Train and Test Dataset

G. Deep Learning Model Accuracy and Loss Curve (MLP)

Now, for deep learning algorithms, we shall examine the accuracy vs. epoch plotting curve for the train and test

datasets as well as the loss vs. epoch plotting curve. The accuracy vs. epoch plotting curve for the MLP algorithm for the train and test dataset is shown below:

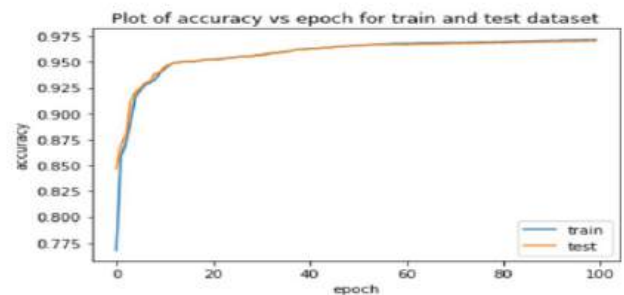


Fig.10. Plot of MLP Algorithm of Accuracy Vs Epochs for Train and Test Dataset

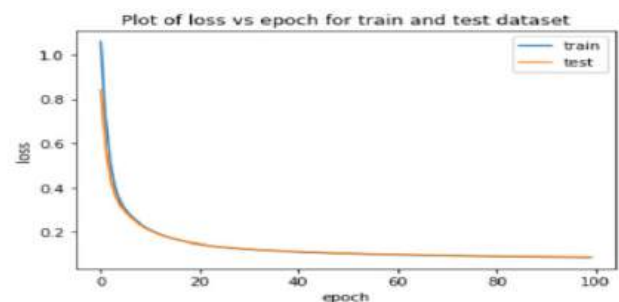


Fig.11. Plot of MLP Algorithm of Loss Vs Epochs for Train and Test Dataset

V. CONCLUSION

The intrusion detection system is evaluated in this paper using machine learning and deep learning techniques. It demonstrates that, except for random forests and decision trees, the model using neural networks achieves greater accuracy than typical machine learning models. The model may enhance both the capability to identify the type of intrusion and the accuracy of intrusion detection. In future, it is recommended to lower the average accuracy while increasing system efficiency by decreasing the imbalance ratio. The NSL KDD dataset has no duplicate data, which enables us to identify the models' maximum accuracy.

REFERENCES

- [1] The History and Evolution of Intrusion Detection from Global Information Assurance Certification Paper <https://www.giac.org/paper/gsec/1294/history-evolution-intrusion-detection/10057>
- [2] Dataset History- A Deeper Dive into the NSL-KDD Data Set <https://towardsdatascience.com/a-deeper-dive-into-the-nsL-kdd-data-set-15c753364657#:~:text=The%20data%20set%20contains%2043,of%20the%20traffic%20input%20itself.>
- [3] Mahesh, Batta. "Machine learning algorithms-a review." International Journal of Science and Research (IJSR).[Internet] 9 (2020): 381-386.
- [4] Qifang Bi, Katherine E Goodman, Joshua Kaminsky, Justin Lessler, What is Machine

- Learning? A Primer for the Epidemiologist, AMERICAN JOURNAL OF EPIDEMIOLOGY, Volume 188, Issue 12, December 2019, Pages 2222–2239, <https://doi.org/10.1093/aje/kwz189>
- [5] Jordan, Michael I., and Tom M. Mitchell. "Machine learning: Trends, perspectives, and prospects." *Science* 349.6245 (2015): 255-260.
- [6] H. Wang, C. Ma and L. Zhou, "A Brief Review of Machine Learning and Its Application," 2009 International Conference on Information Engineering and Computer Science, 2009, pp. 1-4, doi: 10.1109/ICIECS.2009.5362936.
- [7] O. M. Surakhi, A. M. García, M. Jamoos and M. Y. Alkhanafseh, "A Comprehensive Survey for Machine Learning and Deep Learning Applications for Detecting Intrusion Detection," 2021 22nd International Arab Conference on Information Technology (ACIT), 2021, pp. 1-13, doi: 10.1109/ACIT53391.2021.9677375.
- [8] Ma, Y.; Liu, K.; Guan, Z.; Xu, X.; Qian, X.; Bao, H. Background Augmentation Generative Adversarial Networks (BAGANs): Effective Data Generation Based on GAN-Augmented 3D Synthesizing. SYMMETRY 2018, 10, 734. <https://doi.org/10.3390/sym10120734>
- [9] Regression for Machine Learning by Jason Brownlee <https://machinelearningmastery.com/logistic-regression-for-machine-learning/> 10
- [10] A Machine Learning Approach to Network Intrusion Detection System Using K Nearest Neighbor and Random Forest <https://yourpastquestions.com/product/a-machine-learning-approach-to-network-intrusion-detection-system/>
- [11] Intrusion Detection Systems using Machine Learning and Deep Learning Techniques <https://rke.abertay.ac.uk/en/studentTheses/intrusion-detection-systems-using-machine-learning-and-deep-learn>
- [12] Intrusion detection using machine learning algorithms <https://thescholarship.ecu.edu/handle/10342/7650>
- [13] INTRUSION DETECTION SYSTEM USING MACHINE LEARNING TECHNIQUES IN CLOUD COMPUTING. https://efaidnbmnnnibpcajpcgclclefindmkaj/https://s3-ap-southeast-1.amazonaws.com/gtusitecirculares/uploads/Synopsis-Patel%20Pinal-129990907010_446069.pdf
- [14] Neural Networks| IBM Cloud Education. <https://www.ibm.com/cloud/learn/neural-networks>
- [15] Facial recognition. <https://link.springer.com/article/10.1007/s13198-022-01844-6>
- [16] Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions <https://link.springer.com/article/10.1007/s42979-021-00815-1>
- [17] Anomaly-Based Network Intrusion Detection Using Machine Learning <https://ieeexplore.ieee.org/document/8456129> Vinayakumar, R., Soman, K. and Prabakaran, P. (2020) Evaluation of Recurrent Neural Network and Its Variants for Intrusion Detection System (IDS). <https://doi.org/10.4018/978-1-7998-0414-7.ch018> Amarasinghe, K. and Manic, M. (2018) Improving User Trust on Deep Neural Networks Based Intrusion Detection Systems. IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society, Washington DC, 21-23 October 2018, 3262-3268. <https://doi.org/10.1109/IECON.2018.8591322>
- [18] Zarai, R. , Kachout, M. , Hazber, M. and Mahdi, M. (2020) Recurrent Neural Networks & Deep Neural Networks Based on Intrusion Detection System. Open Access Library Journal, 7, 1- 11. doi: 10.4236/oalib.1106151.
- [19] How to choose cross-entropy loss function in Keras? <https://androidkt.com/choose-cross-entropy-loss-function-in-keras/>
- [20] Activation functions and its types by Vinodkumar Baskaran. <https://medium.com/@vinodhb95/activation-functions-and-its-types-8750f1287464>
- [21] IDS 2018 Intrusion CSVs (CSE-CIC-IDS2018) Dataset. IDS 2018 Intrusion CSVs (CSE-CIC-IDS2018) | Kaggle
- [22] García, S., Luengo, J., Francisco, H.: Data Preprocessing in Data Mining, vol. 72. Springer (2015)
- [23] M. M. E. M. S. a. A. M. H. Nasr, "Building Sentiment analysis Model using Graphlab," IJSER, 2017
- [24] Srimani PK, Patil MM (2016) Mining data streams with concept drift in massive online analysis framework. Wseas Trans Comput 15
- [25] Mishra P, Varadharajan V, Tupakula U, Pilli ES (2018) A detailed investigation and analysis of using machine learning technique for intrusion detection, IEEE Commun Surv Tutorials
- [26] What Is Deep Learning? <https://www.mathworks.com/discovery/deep-learning.html>
- [27] Intrusion detection model using machine learning algorithm on Big Data environment <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-018-0145-4>

Improving EEG-Based Epileptic Seizures Detection Using Self-Normalizing Neural Network

Sakorn Mekruksavanich¹, Ponnipa Jantawong¹ and Anuchit Jitpattanakul^{2,*}

¹Department of Computer Engineering, School of Information and Communication Technology
University of Phayao, Phayao, Thailand

sakorn.me@up.ac.th and ponnipa.jantawong@gmail.com

²Intelligent and Nonlinear Dynamic Innovations Research Center, Department of Mathematics
Faculty of Applied Science, King Mongkut's University of Technology North Bangkok, Bangkok, Thailand
anuchit.j@sci.kmutnb.ac.th

Abstract—Epilepsy, a neurological disorder, is typically identified by analyzing electroencephalogram (EEG) signals. However, manually inspecting epileptic seizures is a time-consuming and labor-intensive task. Previous automatic detection algorithms utilizing traditional methods have demonstrated good accuracy in specific EEG classification scenarios but have shown poor performance in others. To tackle this challenge, our study introduces a new deep learning model called the self-normalizing neural network. This model leverages convolutional neural networks and recurrent neural networks to enhance the detection of epileptic seizures using EEG signals. To evaluate our approach, we designed and assessed the proposed detection model using a publicly available Epileptic Seizures Recognition Dataset (ESRD). The experimental findings indicate that the self-normalizing neural network achieves a performance improvement of up to 1.03% accuracy and 1.78% F1-score in epileptic seizure detection, surpassing traditional convolutional neural network approaches.

Keywords—epileptic seizure detection, deep learning, electroencephalography, pyramidal network, classification

I. INTRODUCTION

Epileptic seizures are a prevalent neurological condition that impacts a significant number of elderly individuals worldwide. It is considered the second most prevalent brain disorder, following stroke [1]. Electroencephalogram (EEG) is an electrophysiological technique employed to monitor and record the brain's electrical activity. The EEG signal has been extensively utilized for epilepsy detection [2], [3]. However, manually examining EEG data is a demanding and time-consuming task. Therefore, automated analysis of EEG signals is crucial for efficient clinical screening and accurate epilepsy diagnosis.

Recently, a considerable amount of research has been conducted on the automated detection of epileptic seizures using EEG signals. These studies have been evaluated using the University of Bonn EEG dataset, a widely recognized benchmark database for seizure recognition [4]. The published research in EEG-based epileptic seizure detection can be broadly categorized into two main groups: traditional methods and deep learning methods. Traditional approaches typically involve extracting features from raw EEG data and applying

classification algorithms. The performance of these traditional techniques heavily relies on handcrafted feature extraction methods and the choice of classifiers. Various machine learning algorithms have been developed, utilizing statistical, time-based, frequency-based, time-frequency domain, and nonlinear parameters to detect epileptic seizures [5]. However, in conventional machine learning techniques, the selection of features and classifiers often involves a trial-and-error process [6].

Developing an accurate model requires a strong understanding of signal processing and data mining techniques [7]–[11]. Furthermore, these models tend to perform well only with limited data [12], [13]. To address these limitations, numerous approaches grounded in deep learning technology have been extensively explored across various domains of research [14]–[20], including the realm of epilepsy detection [21].

Unlike traditional machine learning methods, deep learning models require essential data during training. This is because these models possess numerous feature spaces, and when there is an insufficient amount of data, they are susceptible to the issue of overfitting.

In order to tackle the challenge mentioned above in deep learning, this study introduces a new and innovative deep learning model called the self-normalizing neural network. This model aims to enhance the detection of epileptic seizures using EEG signals by incorporating both convolutional neural networks (CNNs) and pyramidal residual modules. To assess the effectiveness of our approach, the proposed detection model was specifically designed and evaluated using a publicly available dataset known as the Epileptic Seizures Recognition Dataset (ESRD).

The structure of the paper is outlined as follows: Section II provides a comprehensive review of the relevant literature. Section III presents the proposed deep learning-based framework for detecting epileptic seizures. The results are presented and analyzed in Section IV. Finally, Section V concludes the paper and discusses potential avenues for future research.

II. RELATED WORKS

This section highlights several researchers who have utilized deep learning techniques to classify epileptic and non-

6

epileptic activities. For instance, Petrosian et al. [22] employed recurrent neural networks with signal wavelet decomposition to capture temporal patterns to detect epileptic seizures. Lin et al. [23] introduced a deep learning framework based on stacked sparse autoencoders, enabling the learning of sparse and high-level representations from EEG signals. Hussein et al. [2] devised an optimized deep neural network utilizing long short-term memory (LSTM) to capture temporal dependencies in EEG data, facilitating robust detection of epileptic seizures.

CNNs have gained significant attention for epilepsy detection as well. Researchers such as Acharya et al. [24] implemented a 13-layer deep CNN algorithm to classify normal, preictal, and seizure classes. Similarly, Liu et al. [25] developed their models using CNNs, which exhibited good accuracy for two-class classification but needed support to achieve satisfactory performance for three-class classification. Zhao et al. [26] proposed a CNN-based model comprising three convolutional blocks and three fully-connected (FC) layers. Each convolutional block consisted of five different layer types. Their experimental results showcased favorable outcomes for both two-class and three-class classifications.

Moreover, several approaches have emerged that integrate CNN with traditional technologies for seizure detection. For instance, Ullah et al. [27] developed an ensemble system that combined pyramidal one-dimensional CNN models with a majority vote mechanism. San-Segundo et al. [28] introduced a model that combined Fourier transform with CNN. Gao et al. [29] proposed a method incorporating approximate entropy, recurrence quantification analysis, and CNN. Turk and Ozerdem [30] generated 2D frequency-time scalograms of EEG records using continuous wavelet transform and subsequently utilized CNN for classification.

To our knowledge, there has not been a CNN model for epilepsy detection using EEG signals consistently demonstrating strong performance across all binary classification tasks within the ESRD dataset. Therefore, we aimed to develop a new CNN-based model to tackle this challenge. Unlike the previously mentioned CNN models, our proposed model incorporates two CNNs to extract time-invariant features from EEG signals. The experimental results reveal that our proposed model performs excellently across all binary classification problems within the dataset.

III. METHODOLOGY

This section provides an overview of the methodology employed in our study to assess the significance of EEG in detecting epileptic seizures. The framework outlines the key steps involved in our methodology for seizure detection. The proposed methodology encompasses four main steps: (i) data acquisition, (ii) data pre-processing, (iii) data generation, and (iv) classification. These steps are elaborated upon below, and the framework is visually presented in Fig. 1.

A. Epileptic Seizures Recognition Dataset

This study utilized a publicly accessible dataset known as the Epileptic Seizures Recognition Dataset (ESRD) [31].

This dataset is widely recognized and widely employed to recognize epileptic seizures and is well-organized. The ESRD dataset comprises 500 subjects, and a total of 11,500 time-series EEG signal data samples, which have been specifically curated to investigate EEG signal changes during seizures. Prior to its online availability, the original dataset underwent pre-processing by UCI. Each sample consists of 4,097 data points, which are further divided into 23 segments. Each segment encompasses 178 data points, corresponding to a one-second interval. The 23 segments are subsequently randomly rearranged, resulting in a collection of 11,500 time-series EEG signal samples derived from the 500 subjects. The UCI epileptic seizure recognition dataset encompasses five distinct health conditions, including one associated with epileptic seizures and four normal conditions where the subjects do not exhibit symptoms of epilepsy. A comprehensive description of all five classes and their respective samples can be seen in Table I.

Although the UCI ESRD dataset contains five different classes, it is common for researchers to focus on binary classification tasks, particularly distinguishing class 1 (representing epileptic seizures) from the remaining classes. Similarly, our objective is to differentiate between individuals who are healthy and those who experience epileptic seizures.

B. Data Pre-processing

The recorded EEG data is susceptible to noise, which can obscure weak EEG signals. Therefore, pre-processing the EEG data is crucial to eliminate distortions caused by various artifacts, including sweating, blinking, electric supply interference, tongue movements, and muscle movements. In the epileptic seizure detection model proposed by [32], the EEG signals are subjected to filtering using a Butterworth bandpass filter. The filter's lower and upper cut-off frequencies are set at 0.35 Hz and 30.5 Hz, respectively, to eliminate undesired signals. Additionally, a notch filter with a frequency of 50 Hz is employed to mitigate the impact of electrical supply bias in the EEG signal.

C. The Proposed Self-Normalizing Neural Network Architecture

The self-normalizing neural network (SNN) is an adapted version of the CNN in which the conventional activation function of CNN is substituted with a scaled exponential linear unit (SELU). The SELU function is provided below:

$$\text{selu}(x) = \lambda \begin{cases} x, & \text{if } x > 0 \\ \alpha e^x - \alpha, & \text{if } x \leq 0 \end{cases}$$

The SELU function has the capability to align the mean and variance of activation values from the preceding layer to those of the subsequent layer. It accomplishes this by iteratively mapping the activation values to a standard normal distribution.

The performance of a deep neural network with numerous hidden layers does not necessarily surpass that of a shallow network with fewer hidden layers. When the available dataset

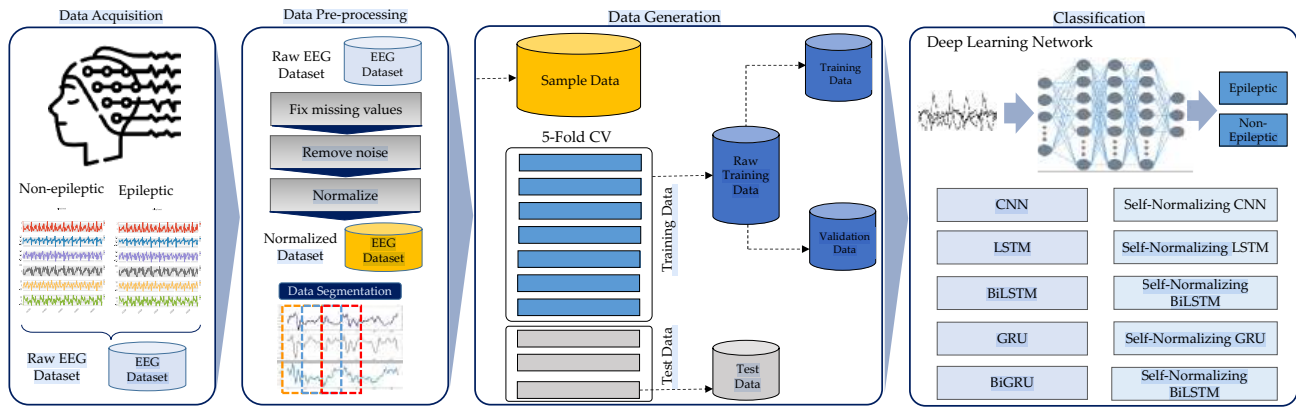


Fig. 1. The proposed EEG-based epileptic seizures detection framework.

TABLE I
DETAILS OF ALL FIVE CLASSES IN ESRD DATASET

Class	No. of Samples	ESR Classes	Description
1	2,300	ictal	Signal recorded during seizures
2	2,300	Pre-ictal	Signal recorded before the occurrence of one-site seizure
3	2,300	Inter-ictal	Signal recorded before the occurrence of consecutive seizure
4	2,300	Healthy (close eye)	A healthy subject having closed eyes
5	2,300	Healthy (open eye)	A healthy subject having open eyes

is limited, employing a deep neural network with many hidden layers can lead to overfitting issues. After conducting several trial runs, a SNN structure, depicted in Fig. 2, was chosen for epileptic seizure detection in this study.

The network architecture comprises two convolutional layers (Conv1 and Conv2), one pooling layer (Maxpool), three SELU activation layers, one fully connected layer (FC), and one Softmax regression layer. The parameter settings for the hidden layers before network training are as follows:

- 1) Convolution layer: Conv1 consists of 16 kernels, and Conv2 consists of 8 kernels. The window size for all kernels is fixed at 1×3 .
- 2) Pooling layer: Max pooling is employed, and the pooling window size remains constant at 2.
- 3) Fully connected layer: It comprises 128 neurons.
- 4) Dropout layer: To mitigate overfitting, nodes within the neural network are randomly deactivated during forward propagation based on a 50% probability rule.
- 5) Softmax layer: This layer facilitates multi-classification tasks, mapping the output of multiple neurons to a probability distribution.

IV. EXPERIMENTS AND RESULTS

In this section, we outline the experimental setup and present the results obtained from evaluating the deep learning models for epileptic seizure recognition.

A. Experiments

This study conducted all experiments on the Google Colab Pro platform with Tesla V100. The implementation was carried

out using Python programming language, specifically Python. The deep learning models were built using TensorFlow, Keras, Scikit-Learn, Numpy, and Pandas libraries.

We implemented a traditional CNN and the proposed SNN using the ESRD dataset to compare the recognition performances. The experiments considered two different scenarios, where different data were used for training and testing the deep learning models. The details of these scenarios can be found in Table II.

B. Results

For each experiment, the ESRD dataset was employed to train the deep learning models, and their performance was evaluated using the 5-fold cross-validation method. The study examined the recognition performance of both a traditional CNN and the proposed SNN models in the two scenarios described in Table III and Table IV.

Table III presents the results of training and testing the deep learning models using EEG data from Scenario I. The experimental findings indicate that the proposed SNN model outperformed the other models, achieving an average accuracy of 96.59% and an average F1-score of 96.58%. These results demonstrate that the incorporation of the self-normalizing mechanism in the SNN model led to notable improvements, increasing the accuracy by 8.37% and the F1-score by 11.7%.

Table IV displays the outcomes of training and testing the deep learning models using EEG data from Scenario II. The experimental findings reveal that the proposed SNN model exhibited superior performance compared to other models,

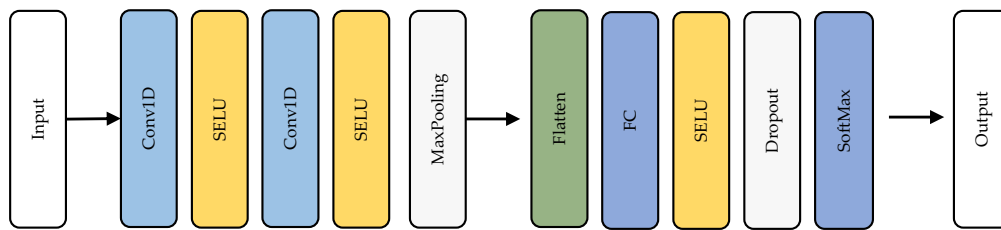


Fig. 2. The architecture of the proposed SNN.

TABLE II
TWO DIFFERENT SCENARIOS USED IN THIS WORK

Scenario	Description
I	Only subjects in class 1 have epileptic seizure and only subjects in class 3 have been defined as non-epileptic
II	Only subjects in class 1 have epileptic seizure And subjects in other class have been defined as non-epileptic

TABLE III
PERFORMANCE METRICS OF THE ENSEMBLE DEEP LEARNING NETWORKS
TRAINED AND EVALUATED USING EEG DATA FROM SCENARIO I

Model	Identification Effectiveness		
	Accuracy	Loss	F1-Score
CNN	88.22%(±19.11%)	0.24(±0.23)	84.88%(±25.78%)
SNN	96.59%(±0.27%)	0.14(±0.03)	96.58%(±0.27%)

TABLE IV
PERFORMANCE METRICS OF THE ENSEMBLE DEEP LEARNING NETWORKS
TRAINED AND EVALUATED USING EEG DATA FROM SCENARIO II

Model	Identification Effectiveness		
	Accuracy	Loss	F1-Score
CNN	96.31%(±0.44%)	0.21(±0.06)	93.94%(±0.74%)
SNN	97.34%(±0.75%)	0.14(±0.03)	95.72%(±1.32%)

achieving an average accuracy of 97.34% and an average F1-score of 95.72%. These results demonstrate that the inclusion of the self-normalizing mechanism in the SNN model led to improvements, increasing the accuracy by 1.03% and the F1-score by 1.78%.

V. CONCLUSION AND FUTURE WORKS

Presently, the examination of epileptic activity in EEG recordings involves the utilization of various traditional and emerging technologies. Automating this process offers numerous advantages, including quicker diagnosis, continuous monitoring, and reduced medical treatment costs. This study conducted experiments to compare the performances of a conventional CNN and the proposed self-normalizing CNN. The comparative analysis presented in the study demonstrated that the SNN improved the detection performance by up to 8.37% in accuracy and 11.7% in F1-score.

For future research, there is potential to enhance the accuracy of the model by employing more complex architectures and applying it to individual patients' data. Personalized deep learning models have the potential to be highly effective in

detecting epileptic seizures across various age groups and geographical locations. The future holds promise, and an efficient deep learning model is expected to be deployed as an application to assist medical professionals.

ACKNOWLEDGMENT

This research project was supported by Thailand Science Research and Innovation Fund; University of Phayao under Grant No. FF66-UoE001; National Science, Research and Innovation Fund (NSRF); and King Mongkut's University of Technology North Bangkok with Contract no. KMUTNB-FF-67-B-10.

REFERENCES

- [1] J. Corsini, L. Shoker, S. Sanei, and G. Alarcon, "Epileptic seizure predictability from scalp eeg incorporating constrained blind source separation," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 5, pp. 790–799, 2006.
- [2] R. Hussein, H. Palangi, R. K. Ward, and Z. J. Wang, "Optimized deep neural network architecture for robust detection of epileptic seizures using eeg signals," *Clinical Neurophysiology*, vol. 130, no. 1, pp. 25–37, 2019.
- [3] S. Mekruksavanich and A. Jitpattanakul, "Deep learning approaches for epileptic seizures recognition based on eeg signal," in *2023 46th International Conference on Telecommunications and Signal Processing (TSP)*, 2023, pp. 33–36.
- [4] S. Wong, A. Simmons, J. Rivera-Villicana, S. Barnett, S. Sivathamboo, P. Perucca, Z. Ge, P. Kwan, L. Kuhlmann, R. Vasa, K. Mouzakis, and T. J. O'Brien, "Eeg datasets for seizure detection and prediction—a review," *Epilepsia Open*, vol. 8, no. 2, pp. 252–267, 2023.
- [5] K. Wang, J. He, and L. Zhang, "Attention-based convolutional neural network for weakly labeled human activities' recognition with wearable sensors," *IEEE Sensors Journal*, vol. 19, no. 17, pp. 7598–7604, 2019.
- [6] A. Murad and J.-Y. Pyun, "Deep recurrent neural networks for human activity recognition," *Sensors*, vol. 17, no. 11, 2017.
- [7] S. Mekruksavanich and A. Jitpattanakul, "Rnn-based deep learning for physical activity recognition using smartwatch sensors: A case study of simple and complex activity recognition," *Mathematical Biosciences and Engineering*, vol. 19, no. 6, pp. 5671–5698, 2022.
- [8] U. Satija, B. Ramkumar, and M. S. Manikandan, "A review of signal processing techniques for electrocardiogram signal quality assessment," *IEEE Reviews in Biomedical Engineering*, vol. 11, pp. 36–52, 2018.
- [9] S. Mekruksavanich and A. Jitpattanakul, "Hybrid convolution neural network with channel attention mechanism for sensor-based human activity recognition," *Scientific Reports*, vol. 13, no. 1, p. 12067, 2023.

- [10] P. Rojanavas, A. Jitpattanakul, and S. Mekruksavanich, "Comparative analysis of lstm-based deep learning models for har using smartphone sensor," in *2021 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunication Engineering*, 2021, pp. 269–272.
- [11] S. Mekruksavanich and A. Jitpattanakul, "A deep learning network with aggregation residual transformation for human activity recognition using inertial and stretch sensors," *Computers*, vol. 12, no. 7, 2023.
- [12] N. Hnoohom, A. Jitpattanakul, I. You, and S. Mekruksavanich, "Deep learning approach for complex activity recognition using heterogeneous sensors from wearable device," in *2021 Research, Invention, and Innovation Congress: Innovation Electricals and Electronics (RI2C)*, 2021, pp. 60–65.
- [13] N. Hnoohom, A. Jitpattanakul, and S. Mekruksavanich, "Real-life human activity recognition with tri-axial accelerometer data from smartphone using hybrid long short-term memory networks," in *2020 15th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP)*, 2020, pp. 1–6.
- [14] N. Hnoohom, S. Mekruksavanich, and A. Jitpattanakul, "An efficient resnetse architecture for smoking activity recognition from smartwatch," *Intelligent Automation & Soft Computing*, vol. 35, no. 1, pp. 1245–1259, 2023.
- [15] S. Mekruksavanich and A. Jitpattanakul, "Sport-related activity recognition from wearable sensors using bidirectional gru network," *Intelligent Automation & Soft Computing*, vol. 34, no. 3, pp. 1907–1925, 2022.
- [16] S. Mekruksavanich, C. Promsakon, and A. Jitpattanakul, "Location-based daily human activity recognition using hybrid deep learning network," in *2021 18th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, 2021, pp. 1–5.
- [17] S. Mekruksavanich, P. Jantawong, A. Charoenphol, and A. Jitpattanakul, "Fall detection from smart wearable sensors using deep convolutional neural network with squeeze-and-excitation module," in *2021 25th International Computer Science and Engineering Conference (ICSEC)*, 2021, pp. 448–453.
- [18] S. Mekruksavanich and A. Jitpattanakul, "Fallnext: A deep residual model based on multi-branch aggregation for sensor-based fall detection," *ECTI Transactions on Computer and Information Technology (ECTI-CIT)*, vol. 16, no. 4, p. 352–364, Sep. 2022.
- [19] W. Zhong, N. Yu, and C. Ai, "Applying big data based deep learning system to intrusion detection," *Big Data Mining and Analytics*, vol. 3, no. 3, pp. 181–195, 2020.
- [20] S. Mekruksavanich and A. Jitpattanakul, "Automatic recognition of construction worker activities using deep learning approaches and wearable inertial sensors," *Intelligent Automation & Soft Computing*, vol. 36, no. 2, pp. 2111–2128, 2023.
- [21] J. Prasanna, M. S. P. Subathra, M. A. Mohammed, R. Damaševičius, N. J. Sairamy, and S. T. George, "Automated epileptic seizure detection in pediatric subjects of chb-mit eeg database—a survey," *Journal of Personalized Medicine*, vol. 11, no. 10, 2021.
- [22] A. Petrosian, D. Prokhorov, R. Homan, R. Dasheiff, and D. Wunsch, "Recurrent neural network based prediction of epileptic seizures in intra- and extracranial eeg," *Neurocomputing*, vol. 30, no. 1, pp. 201–218, 2000.
- [23] Q. Lin, S.-q. Ye, X.-m. Huang, S.-y. Li, M.-z. Zhang, Y. Xue, and W.-S. Chen, "Classification of epileptic eeg signals with stacked sparse autoencoder based on deep learning," in *Intelligent Computing Methodologies*. Cham: Springer International Publishing, 2016, pp. 802–810.
- [24] U. R. Acharya, S. L. Oh, Y. Hagiwara, J. H. Tan, and H. Adeli, "Deep convolutional neural network for the automated detection and diagnosis of seizure using eeg signals," *Computers in Biology and Medicine*, vol. 100, pp. 270–278, 2018.
- [25] J. Liu and B. Woodson, "Deep learning classification for epilepsy detection using a single channel electroencephalography (eeg)," in *Proceedings of the 2019 3rd International Conference on Deep Learning Technologies*, ser. ICDLT '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 23–26.
- [26] W. Zhao, W. Zhao, W. Wang, X. Jiang, X. Zhang, Y. Peng, B. Zhang, and G. Zhang, "A novel deep neural network for robust detection of seizures using eeg signals," *Computational and Mathematical Methods in Medicine*, vol. 2020, pp. 1–9, 04 2020.
- [27] I. Ullah, M. Hussain, E. ul Haq Qazi, and H. Aboalsamh, "An automated system for epilepsy detection using eeg brain signals based on deep learning approach," *Expert Systems with Applications*, vol. 107, pp. 61–71, 2018.
- [28] R. San-Segundo, M. Gil-Martín, L. F. D'Haro-Enríquez, and J. M. Pardo, "Classification of epileptic eeg recordings using signal transforms and convolutional neural networks," *Computers in Biology and Medicine*, vol. 109, pp. 148–158, 2019.
- [29] X. Gao, X. Yan, P. Gao, X. Gao, and S. Zhang, "Automatic detection of epileptic seizure based on approximate entropy, recurrence quantification analysis and convolutional neural networks," *Artificial Intelligence in Medicine*, vol. 102, p. 101711, 2020.
- [30] O. Turk and M. S. Ozerdem, "Epilepsy detection by using scalogram based convolutional neural network from eeg signals," *Brain Sciences*, vol. 9, no. 5, 2019.
- [31] A. Goldberger, L. Amaral, L. Glass, S. Havlin, J. Hausdorg, P. Ivanov, R. Mark, J. Mietus, G. Moody, C.-K. Peng, H. Stanley, and P. PhysioBank, "Components of a new research resource for complex physiologic signals," *PhysioNet*, vol. 101, no. 23, pp. E215–20, 2000.
- [32] S. Shah, V. Sasane, S. Vardam, and V. Bharate, "Eeg based epileptic seizure detection," *International Journal of Engineering Trends and Technology*, vol. 33, no. 4, pp. 191–195, 2016.

Hybrid Firefly Algorithm with Rao Algorithm for Optimization Problems

Sakkayaphop Pravesjit, Krittika Kantawong,
School of Information and Communication Technology
University of Phayao
Phayao, Thailand
sakkayaphop.pr@up.ac.th, krittika.ka@up.ac.th

Duangjai Jitkongchuen
Government Big Data Institute
Digital Economy Promotion Agency
Bangkok, Thailand
duangjai.ji@depa.or.th

Abstract— This paper proposes an improvement of firefly algorithm for optimization problems. In this study, the rao algorithm was applied in crossover and mutation operators instead of procedures of moving firefly i towards j . The algorithm was tested on six benchmark problems and compared with differential evolution (DE), JDE self-adaptive algorithm, and intersection mutation differential evolution (IMDE) algorithm. The computation results illustrated that the proposed algorithm can produce optimal solutions for two of six functions. The proposed algorithm has provided the best results compared to the other three algorithms. The findings prove that the algorithm should be improved in this direction and show that the algorithm produces several solutions obtained by the previously published methods, especially for the continuous step function, the multimodal function and the discontinuous step function.

Keywords— optimization function, genetic algorithm, Rao algorithm, crossover operator; mutation operator

I. INTRODUCTION

In recent years, natural-inspired optimization algorithms (NIOAs) have been brought to solve the optimization problems. Many researchers have proposed various categories of natural-inspired optimization algorithms (NIOAs) consist of evolutionary algorithm (EAs) and swarm-intelligence (SI) based algorithms. These algorithms are applied to solve many problems such as classification, prediction, forecasting, optimization and much more. Optimization is a process for finding the optimal solution. It's divided into two main categories depending on the type of variables including discrete and continuous. The main objective of optimization problem is finding the maximum or minimum a real function by systematically choosing input values from within an allowed set and computing the value of the function. Examples of such algorithms for solving optimization problem, Bee colony Optimization [1], Particle Swarm Optimization (PSO) [2], Differential Evolution (DE) [3], Simulated Annealing (SA) [4] and Ant Colony Optimization [5], Self-adaptive differential evolution algorithm (Evo-DE) [6], Genetic Algorithm (GA) [7], and firefly mating algorithm [8].

The Firefly Algorithm (FA) was presented by Yang [9] in 2009. The main idea was to use flashing characteristics of fireflies to develop a firefly-inspired algorithm. Description the original firefly algorithm based on three rules:

(i) All fireflies are unisex so that one firefly is attracted to all other fireflies.

(ii) Attractiveness is proportional to its brightness, thus for any flashing fireflies, the less brighter one will move towards the brighter ones. However, if there are no fireflies brighter than a given firefly, that firefly will move randomly.

(iii) The distance will affect the brightness of the fireflies. Since light is absorbed as it passes through the medium, the brightness of the firefly j seen by the firefly i is given in (1).

$$I_j(r) = I_j(0)e^{-\gamma r^2} \quad (1)$$

where γ is a light absorption coefficient of the medium.

$I_j(0)$ is the brightness of the firefly j at r .

$$r = \|x_i - x_j\| = \sqrt{\sum_{k=1}^d (x_i^k - x_j^k)^2} \quad (2)$$

x_i and x_j are the positions of the fireflies i and j .

If a firefly located at x_j is brighter than another firefly located at x_i , the firefly located at x_i will move to x_j according to the following equation (3).

$$x'_i = x_i + I_j(r)(x_j - x_i) + \alpha(rand) \quad (3)$$

From equation (3) α is a random parameter and $rand$ is a uniform random value in the range $[-0.5, 0.5]$.

Rao algorithm was developed by Rao [10] in 2020. The main idea is the adoption of the three-simple metaphor-less optimization algorithms for solving the optimization problems. The equation of the three-simple metaphor-less show by Equation 4, 5, and 6.

$$x'_{j,k,i} = x_{j,k,i} + r_{1,j,i}(x_{j,best,i} - x_{j,worst,i}) \quad (4)$$

$$x'_{j,k,i} = x_{j,k,i} + r_{1,j,i}(x_{best,i} - x_{worst,i}) + r_{2,j,i}(|x_{j,k,i} \text{ or } x_{j,l,i}| - |x_{j,l,i} \text{ or } x_{j,k,i}|) \quad (5)$$

$$x'_{j,k,i} = x_{j,k,i} + r_{1,j,i}(x_{best,i} - |x_{worst,i}|) + r_{2,j,i}(|x_{j,k,i} \text{ or } x_{j,l,i}| - (x_{j,l,i} \text{ or } x_{j,k,i})) \quad (6)$$

where $x_{best,i}$: is the best population in the i^{th} iteration.

$x_{worst,i}$: is the worst population in the i^{th} iteration.

$x'_{j,k,i}$: is the update value of $x_{j,k,i}$.

$r_{1,j,i}$ and $r_{2,j,i}$: are the random value in the range [0,1].

$x_{j,k,i}$: is the solution k.

$x_{j,l,i}$: is the randomly solution l.

In Equation (5) and (6), in term $x_{j,k,i}$ or $x_{j,l,i}$ if the fitness value of $x_{j,k,i}$ is better than $x_{j,l,i}$, $x_{j,k,i}$ can be used, or use $x_{j,l,i}$; and in term $x_{j,l,i}$ or $x_{j,k,i}$ if the fitness value of $x_{j,l,i}$ is better than $x_{j,k,i}$, $x_{j,l,i}$ is used, or use $x_{j,k,i}$. The performance of Rao algorithm is good and quite competitive.

This paper presents a hybrid firefly algorithm with Rao algorithm to solve optimization problems. The performance of the proposed algorithm was compared with differential evolution (DE), JDE self-adaptive algorithm, and intersection mutation differential evolution (IMDE) algorithm.

Following this introduction, Section 2 presents a proposed algorithm. In Section 3, the experimental results are presented and discussed. Finally, Section 4 is the conclusion of the study.

II. THE PROPOSED ALGORITHM

This paper aims to hybrid Rao algorithm in the step of move firefly i towards j of firefly algorithm for continuous optimization problems. After initial populations and evaluation of their fitness values, the rao algorithm is applied to improve procedures of moving firefly i towards j and include mutation method in this step. The concept of model for the proposed as shown in Fig.1. The processes are explained in details as follows:

(i) After initial population and evaluation of fitness all of population, selection best and worst solution in the populations.

(ii) The moving step if random value more than 0.5 and number of firefly located at x_j brighter than another firefly located at x_i more than 1, then used original moving firefly algorithm else used the Rao algorithm and mutation method.

(iii) Selection operator selects initial population and children from the two operators are evaluated for their fitness value. Then, the new generations are selected by their ranking value.

The new population is then improved all processes until the stopping condition is met.

TABLE I. THE PARAMETERS USED IN THE EXPERIMENTS

Parameters	Values
Population size, NB	25
Dimension, (ND)	30
Mutation parameters, Pm	0.8

III. EXPERIMENTAL RESULTS

The proposed algorithm was tested on 6 different benchmark functions from Yao et al. [11]. The tested functions are presented in Table 2, including Sphere function, Schwefel function, Rosenbrock function, Step function, Quartic function, and Ackleg function.

The computational results shown in Table 3 present a comparison of the proposed algorithm with differential evolution (DE), JDE self-adaptive algorithm, and intersection mutation differential evolution (IMDE) algorithm. It indicates that the proposed algorithm has offered better results than the other three algorithms, namely Step function, Quartic function, and Ackleg function. Regarding Schwefel function, JDE algorithm has produced better results than the proposed algorithm. In addition, JDE algorithm, IMDE 1st process, and IMDE 2nd process give the results of the Rosenbrock function better than the proposed algorithm.

IV. CONCLUSIONS

This paper has proposed hybrid firefly algorithm with Rao algorithm for optimization problems. In order to hybrid the performance of firefly algorithm, the Rao algorithm is applied to improve procedures of moving firefly i towards j and include mutation method in this step. The testing on six various functions has demonstrated that the proposed algorithm provides the optimal value for three of six functions. The experimental results has confirmed that the proposed algorithm can solve the continuous functions, the discontinuous functions, and the multimodal function efficiently.

ACKNOWLEDGMENT

The authors would like to acknowledge School of Information and Communication Technology, University of Phayao, Thailand for all resources and financial support.

REFERENCES

- [1] D. Karaboga, B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm", Journal of Global Optimization 39 (2007) 459–471
- [2] Kennedy, J.; Eberhart, R. (1995). "Particle Swarm Optimization". Proceedings of IEEE International Conference on Neural Networks. IV. pp. 1942–1948
- [3] P. Storn, R. and Price, K. (1997), „Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces“, Journal of Global Optimization, 11, pp. 341–359.
- [4] David Bookstaber, "Simulated Annealing for Traveling Salesman Problem", Spring, 1997
- [5] M. Dorigo, T. Stutzle, "Ant Colony optimization", A Bradford book, MIT Press Cambridge, Massachusetts london, England (2004) .
- [6] Duangjai Jitkongchuen and Arit Thammano, (2014), A self-adaptive differential evolution algorithm for continuous optimization problems, Artificial Life and Robotics, 19, pp.201-208.
- [7] THolland, J.H. "Adaptation in Natural and Artificial Systems". MIT Press, 1992
- [8] Ritthipakdee, A., Thammano, A., Premasathian, N., & Jitkongchuen, D. (2017). Firefly mating algorithm for continuous optimization problems. Computational Intelligence and Neuroscience, 2017.
- [9] Yang, X. S. (2009, October). Firefly algorithms for multimodal optimization. In International symposium on stochastic algorithms (pp. 169-178). Berlin, Heidelberg: Springer Berlin Heidelberg.
- [10] Rao, R. (2020). Rao algorithms: Three metaphor-less simple algorithms for solving optimization problems. International Journal of Industrial Engineering Computations, 11(1), 107-130.
- [11] Yao, X., Liu, Y., & Lin, G. (1999). Evolutionary programming made faster. IEEE Transactions on Evolutionary computation, 3(2), 82-102.

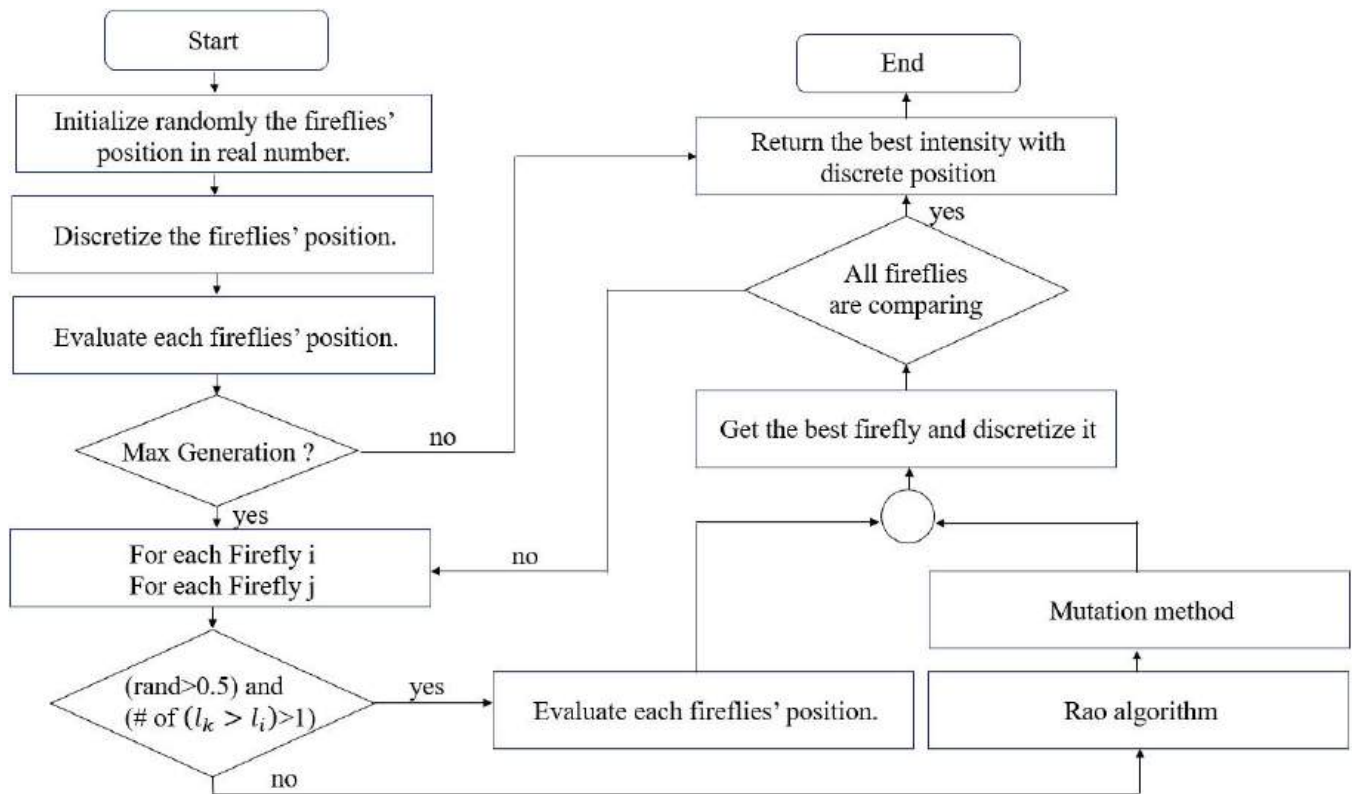


Fig. 1. The concept of research.

TABLE II. THE BENCHMARK FUNCTIONS

Function	Iteration	Dimension (D)	Search space	fmin
$f_1(x) = \sum_{i=1}^D x_i^2$	1,500	30	[-100, 100]	0
$f_2(x) = \max_i\{ x_i , 1 \leq i \leq D\}$	5,000	30	[-100, 100]	0
$f_3(x) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2) + (x_i - 1)^2]$	20,000	30	[-30, 30]	0
$f_4(x) = \sum_{i=1}^D ([x_i + 0.5])^2$	1,500	30	[-100, 100]	0
$f_5(x) = \sum_{i=1}^D ix_i^4 + rand[0,1)$	3,000	30	[-1.28, 1.28]	0
$f_6(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{30} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{30} \sum_{i=1}^D \cos 2 \pi x_i\right) + 20 + e$	1,500	30	[-32, 32]	0

TABLE III. THE EXPERIMENTAL REUSLTS

Functions	DE	JDE	IMDE 1 st process	IMDE 2 nd process	Proposed algorithm
F1	1.58E - 3	1.1E - 28	2.5E - 32	2.1E - 35	2.69E - 25
F2	1.9E-3	0	0.2E-3	3.4E-24	1.42E -20
F3	8.35E-27	0	0	0	1.36E - 12
F4	0	0	0	0	0
F5	2.63E-3	3.15E-3	2.4E-4	3.4E-4	0
F6	1.5017	7.7E-15	4.9E-15	4.6E-15	3.54E-15

A Hybrid Deep Learning Neural Network for Recognizing Exercise Activity Using Inertial Sensor and Motion Capture System

Sakorn Mekruksavanich¹, Ponnipa Jantawong¹ and Anuchit Jitpattanakul^{2,*}

¹*Department of Computer Engineering, School of Information and Communication Technology
University of Phayao, Phayao, Thailand*

sakorn.me@up.ac.th and ponnipa.jantawong@gmail.com

²*Intelligent and Nonlinear Dynamic Innovations Research Center, Department of Mathematics
Faculty of Applied Science, King Mongkut's University of Technology North Bangkok, Bangkok, Thailand*
anuchit.j@sci.kmutnb.ac.th

Abstract—The recognition of human activities holds great significance in the fields of AI and human-computer interaction studies. Human activity recognition (HAR) finds extensive application, including in behavioral assessment, gesture identification, fall detection, and gait evaluation. Recent strides in wearable technology and AI have showcased the potential to accurately classify movement events for HAR, achieved by analyzing time-series data collected from wearable sensors. This study focuses on deep learning networks, particularly CNN and LSTM, to extract spatial and temporal features from wearable sensor data for efficient exercise activity classification. A comprehensive assessment compared two sensor data types: 3-axis linear acceleration from IMUs and 3D acceleration from MOCAP markers. Findings highlight comparable accuracy between IMUs and MOCAP in exercise identification. The hybrid CNN-LSTM demonstrated high efficacy, achieving remarkable F1-scores of 98.85% for IMU and 98.23% for MOCAP data.

Keywords—human activity recognition, deep learning, wearable sensor, inertial measurement unit, motion capture

I. INTRODUCTION

Human Activity Recognition (HAR) involves categorizing people's actions within a specific time-frame by using various measurements such as acceleration, rotation velocity, and geographical location collected from personal electronic devices [1]. In recent years, the use of intelligent wearable technology has become increasingly widespread, enabling the monitoring and detection of user behaviors. These devices can identify activities through data from accelerometers and gyroscopes, which is presented as time-series data [2]. Activity detection holds the potential to assess an individual's health status, as shown in [3]–[5], [5]–[7]. HAR technology finds applications even in sports like volleyball and badminton [8], [9]. The accuracy and precision of activity recognition are crucial for enhancing service delivery in various applications.

Sports activity identification falls within the realm of HAR. This technology serves two main purposes: first, it automates the recording of workouts, relieving trainers from the

cumbersome task of manual data entry; second, it provides prompt access to unbiased records for instructors, coaches, and medical professionals [10]. In our present study, we specifically concentrate on activities demanding physical exertion, particularly those involving vigorous effort. The scope narrows to encompass various activities contributing to the enhancement or maintenance of an individual's overall well-being and fitness.

The challenges related to the implementation of sport tracking techniques have gained significant recognition. Vision-based methods, while promising, bring up concerns about privacy and occlusion, especially in environments with multiple users engaged in fitness activities [11]. On-body wearables often result in user discomfort during exercise and require bulky equipment to ensure accurate tracking of diverse workouts [12]. Wireless sensing approaches tend to fall short in scenarios involving multiple users [13]. Moreover, there exists a noticeable gap in discussions concerning the feasibility of detecting precise and nuanced errors, along with the subsequent implications for delivering prompt and effective feedback. Each of these strategies presents its own set of advantages and disadvantages. For the purpose of this study, we narrow our focus to wearable sensors, exploring their potential within these contexts.

Lately, there have been suggestions to leverage machine learning (ML) techniques for the purpose of accurately classifying sporting events by making use of wearable sensors [14]. While substantial advancements have been made in applying deep learning (DL) models to analyze multi-channel time series data for identifying routine human activities, there remains a lack of research that explicitly concentrates on tracking free-weight exercises.

This study contributes to the advancement of exercise activity recognition (EAR) research by delving into deep learning networks, specifically focusing on convolutional neural networks (CNN) and long short-term memory neural networks (LSTM). The primary goal is to automatically extract spatial and temporal characteristics from data collected by

wearable sensors, enabling precise classification of physical activities. Our investigation aims to evaluate the effectiveness of two foundational deep learning networks—the CNN and LSTM—as well as the proposed CNN-LSTM hybrid model. We assess their performance in predicting three distinct actions: walking, sit-to-stand transitions, and squatting. To achieve this, we analyze data obtained from EAR measurements. The experimental dataset encompasses 3D acceleration data derived from two separate sources: a cost-effective inertial measurement system comprising two inertial measurement units (IMUs), and an optoelectronic motion capture (MOCAP) device.

II. RELATED WORKS

A. Activity Recognition Using IMU

In today's era, wearable sensors are experiencing continuous enhancements, incorporating a variety of intricate electronic components. To harness the potential of the data generated by these components, ML and DL techniques are applied. The objective is to enhance the functionality of these electronic devices across a wide range of applications, including the recognition of human activities [15].

Research in HAR has been conducted with a specific emphasis on EAR across several application domains, including gymnastics, weight workouts, aerobic and sports [16]. In the field of literature, IMUs have emerged as the most frequently employed data stream [17]. The process of identifying several discrete labels based on sensor data streams is often referred to as EAR. Frequently, identification methods include a manual feature extraction process, which is afterwards coupled with a classification technique.

While DL techniques, notably CNN and LSTM, have gained widespread recognition as advanced approaches for HAR [18], [19], their application in EAR remains relatively unexplored [20]. As an illustration, researchers in [21] employ a recurrent design to identify shoulder rehabilitation exercises using data streams from wrist-worn IMU devices, achieving an accuracy of 88.9%. However, it's important to note that their dataset is not publicly available, and their methods may not seamlessly translate to other exercise domains due to the absence of sensors capable of capturing motion beyond the wrist.

In contrast, the publicly accessible MEx dataset [20] has recently presented insights into the effectiveness of a single sensor in recognizing physiotherapy exercises. This dataset incorporates four sensors: two accelerometers on the wrist and thigh, a pressure mat, and a depth camera. The resulting F1-score for these sensors are 63.35%, 90.15%, 74.08%, and 87.20% respectively. These outcomes underscore the insufficiency of relying solely on a single sensor to accurately identify a diverse range of physical activities. Consequently, it becomes evident that there is a need for leveraging multimodal learning approaches to enhance the accuracy and efficiency of recognition tasks [22]–[24]. This requirement prompts the integration of various sensor inputs through structural and methodological means, including attention mechanisms, to effectively capture the nuances of diverse physical exercises.

B. Activity Recognition Using Mo-cap Data

Numerous studies have focused on HAR by employing MOCAP data as the primary dataset. Picard et al. [25] conducted a study utilizing MOCAP data to precisely identify diverse culinary tasks, achieving an impressive 95% accuracy in cross-validation on the training dataset. In the present research, participants were visually represented as stick figures using MOCAP data. To capture temporal information, a Hidden Markov model was incorporated during the post-processing stage to refine the results. It's worth noting that the dataset underwent meticulous labeling, including the correction of a limited number of incorrect labels. Furthermore, the data was randomized to prevent any preexisting order before training, which significantly contributed to achieving a noteworthy level of precision. For the classification task, two distinct classifiers were employed to handle different classes, and their outcomes were subsequently combined to provide a comprehensive assessment.

III. EXERCISE ACTIVITY RECOGNITION FRAMEWORK

The methodology used in this study for EAR comprises five core steps: data collection, data pre-processing, data segmentation, model development, and model refinement, as depicted in Fig. 1.

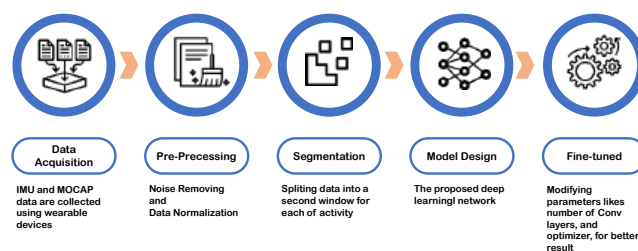


Fig. 1. The EAR workflow based on IMU sensors and MOCAP used in this work.

A. IMU_MOCAP Dataset

To evaluate the effectiveness of the proposed approach in recognizing human behaviors, this study focused on three common everyday movements: walking, sit-to-stand transitions, and squatting. The assessment involved a group of 10 individuals in good physical condition. Experimental data was gathered at the Lisbon Biomechanics Laboratory, utilizing two distinct setups: an IMU and a sophisticated optoelectronic system. The optoelectronic configuration comprised 14 infrared ProReflex 1000 cameras, and data sampling occurred at a frequency of 100 Hz. This comprehensive setup allowed for a thorough analysis of the chosen motions and their recognition using the suggested methodology.

B. Data Pre-processing

The raw sensor data underwent manipulation during the data pre-processing stage, which included noise reduction and data normalization techniques. In this study, noise reduction was achieved through the implementation of an average smoothing

filter. This filter was applied to both the IMU and MO-CAP data, accounting for all three dimensions. Following the denoising step, the sensor data underwent normalization, a method employed to address the challenge of model learning by standardizing data outputs to a comparable range. This standardization enables gradient descents to potentially achieve faster convergence.

The process of normalization involved dividing the data using fixed-width sliding windows, each containing 100 samples and having non-overlapping proportions. This windowing approach facilitated the effective handling of the data, contributing to the subsequent analysis.

C. DL Models

This study undertook a comparative analysis of various DL models, with a specific emphasis on CNN and LSTM architectures. The main objective was to evaluate their performance in recognizing exercise activities. The employed CNN model adheres to the traditional architecture, comprising N convolutional layers, a Rectified Linear Unit (ReLU) activation function, and a hidden dense layer. For a visual representation of the CNN's structure, refer to Fig. 2.

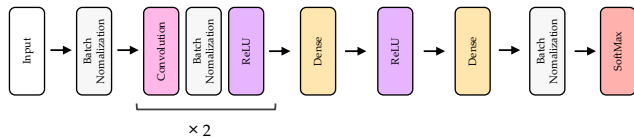


Fig. 2. The CNN architecture used in this work.

In CNNs, hyperparameters include the number of convolutional layers, the number of filters within each convolutional layer, and the number of neurons within the hidden dense layer. In this study, the omission of pooling layers was justified by the rationale that reducing the spatial size of the sequence is often unnecessary when an ample number of convolutional layers are in place.

LSTM network stands out as a commonly utilized variant within the recurrent neural network (RNN) framework. Proposed by [26], the LSTM model was devised to tackle the issue of vanishing gradients that arise when learning extended dependencies. To effectively capture distant relationships within temporal sequences, the LSTM architecture employs memory cells featuring three gates and their corresponding parameters. These gates play a crucial role in governing the timing of status updates and the erasure of previous states, thereby coordinating the overall functionality of the memory cells. A visual representation of the LSTM structure is provided in Fig. 3 for clarity.

In this study, we introduce an innovative hybrid deep learning network known as CNN-LSTM, designed to enhance recognition effectiveness in the current investigation. The CNN-LSTM model is composed of two convolutional layers and one LSTM layer, as illustrated in Fig. 4. Specifically, the CNN-LSTM architecture incorporates two 1D-CNN layers with 3×1 kernels, strategically harnessed to extract intricate

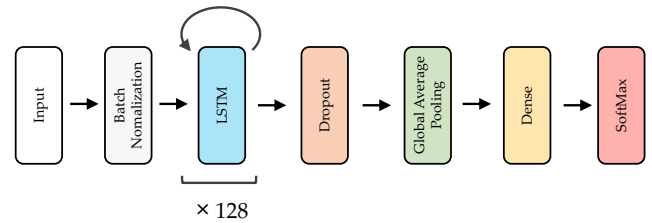


Fig. 3. The LSTM architecture used in this work.

spatial features from sensor data collected by smartwatches. The initial 1D-CNN layer is configured with 128 filters, while the subsequent layer employs 64 filters. The recommended activation function for this model is the rectified linear function.

Upon processing the data through the second one-dimensional CNN (1D-CNN) layer, a dropout strategy is introduced. Subsequently, a LSTM layer with 128 units is employed to capture temporal information, followed by the application of a dropout layer. In the final phase, a fully connected dense layer is employed to process the LSTM's output at the final time step, incorporating a SoftMax function for appropriate computation. This integrated approach aims to significantly enhance the recognition capabilities of the proposed CNN-LSTM model.

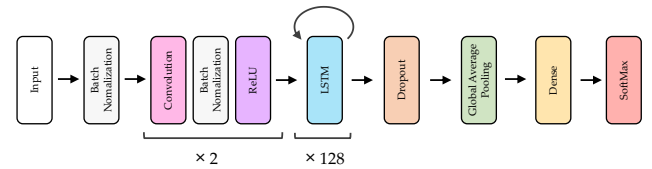


Fig. 4. The CNN-LSTM architecture proposed in this work.

IV. EXPERIMENTS AND RESULTS

This section offers an overview of the experimental setup and presents the results derived from evaluating three distinct DL models: CNN, LSTM, and the proposed CNN-LSTM model.

Table I showcases the efficiency findings for DL models using IMU data for identification purposes. The outcomes of this study underscore the superior performance of the presented CNN-LSTM model, achieving an impressive F1-score of 98.85%, thus solidifying its heightened effectiveness.

TABLE I
PERFORMANCE METRICS OF DL MODELS USING IMU DATA

Model	Identification Effectiveness		
	Accuracy	Loss	F1-score
CNN	98.27%(±0.81%)	0.05(±0.02)	98.25%(±0.82%)
LSTM	98.54%(±1.17%)	0.05(±0.04)	98.53%(±1.20%)
CNN-LSTM	98.85%(±0.40%)	0.03(±0.01)	98.85%(±0.40%)

Another study employed data from MOCAP to explore this further. The results of this study reveal the CNN-LSTM

network's strong capability for accurately classifying exercise activities, with F1-score values surpassing 98.23%, as detailed in Table II.

TABLE II
PERFORMANCE METRICS OF DL MODELS USING MOCAP DATA

Model	Identification Effectiveness		
	Accuracy	Loss	F1-score
CNN	98.08%(±0.40%)	0.07(±0.01)	98.20%(±0.36%)
LSTM	98.08%(±0.55%)	0.06(±0.02)	98.20%(±0.51%)
CNN-LSTM	98.19%(±1.07%)	0.06(±0.04)	98.23%(±1.07%)

V. CONCLUSION AND FUTURE WORKS

Through the application of DL techniques, researchers have effectively accomplished the recognition of various exercise behaviors, such as walking, sit-to-stand, and squatting. This achievement was made possible by employing just two IMUs and MOCAP systems, strategically positioned at identical locations on the individual's body. DL networks were harnessed to interpret raw data as time series, involving the use of a sliding window approach to extract relevant features from segmented sequences. Leveraging sensor data from IMUs yielded precise outcomes, which were further validated by utilizing high-quality MOCAP data. The integration of DL networks, including CNN, LSTM, and the innovative CNN-LSTM architecture, culminated in the achievement of an impressive F1-score exceeding 98%. Notably, the hybrid CNN-LSTM model emerged as a standout performer, delivering highly accurate predictions with a remarkable F1-score of 98.85% and 98.23% when employing IMU and MOCAP data, respectively.

In our upcoming pursuits, we plan to expand our research on multimodal exercise activity identification by integrating additional DL networks, such as InceptionTime and Temporal Transformer. Moreover, we recognize the potential of data augmentation as a viable strategy for boosting model performance, especially when faced with imbalanced datasets. This approach holds promise in addressing the specific challenge at hand.

ACKNOWLEDGMENT

This research project was supported by Thailand Science Research and Innovation Fund; University of Phayao under Grant No. FF66-UoE001; National Science, Research and Innovation Fund (NSRF); and King Mongkut's University of Technology North Bangkok with Contract no. KMUTNB-FF-67-B-10.

REFERENCES

- [1] M. Strackiewicz, P. James, and J.-P. Onnela, "A systematic review of smartphone-based human activity recognition for health research," *npi Digital Medicine*, vol. 4, no. 148, 2021.
- [2] S. Mekruksavanich and A. Jitpattanakul, "Recognition of real-life activities with smartphone sensors using deep learning approaches," in *2021 IEEE 12th International Conference on Software Engineering and Service Science (ICSESS)*, 2021, pp. 243–246.
- [3] S. Mekruksavanich and A. Jitpattanakul, "Cnn-based deep learning network for human activity recognition during physical exercise from accelerometer and photoplethysmographic sensors," in *Computer Networks, Big Data and IoT*. Singapore: Springer Nature Singapore, 2022, pp. 531–542.
- [4] N. Hnoohom, A. Jitpattanakul, I. You, and S. Mekruksavanich, "Deep learning approach for complex activity recognition using heterogeneous sensors from wearable device," in *2021 Research, Invention, and Innovation Congress: Innovation Electricals and Electronics (RI2C)*, 2021, pp. 60–65.
- [5] S. Mekruksavanich, P. Jantawong, A. Charoenphol, and A. Jitpattanakul, "Fall detection from smart wearable sensors using deep convolutional neural network with squeeze-and-excitation module," in *2021 25th International Computer Science and Engineering Conference (ICSEC)*, 2021, pp. 448–453.
- [6] N. Hnoohom, A. Jitpattanakul, and S. Mekruksavanich, "Real-life human activity recognition with tri-axial accelerometer data from smartphone using hybrid long short-term memory networks," in *2020 15th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP)*, 2020, pp. 1–6.
- [7] S. Mekruksavanich, C. Promsakon, and A. Jitpattanakul, "Location-based daily human activity recognition using hybrid deep learning network," in *2021 18th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, 2021, pp. 1–5.
- [8] F. Haider, F. A. Salim, D. B. Postma, R. van Delden, D. Reidsma, B.-J. van Beijnum, and S. Luz, "A super-bagging method for volleyball action recognition using wearable sensors," *Multimodal Technologies and Interaction*, vol. 4, no. 2, 2020.
- [9] S. Mekruksavanich, P. Jantawong, N. Hnoohom, and A. Jitpattanakul, "Badminton activity recognition and player assessment based on motion signals using deep residual network," in *2022 IEEE 13th International Conference on Software Engineering and Service Science (ICSESS)*, 2022, pp. 80–83.
- [10] A. Hussain, K. Zafar, A. R. Baig, R. Almakki, L. AlSuwaidan, and S. Khan, "Sensor-based gym physical exercise recognition: Data acquisition and experiments," *Sensors*, vol. 22, no. 7, 2022.
- [11] S. Milanko and S. Jain, "Demo abstract: Liftright: Quantifying training performance using a wearable sensor," in *2019 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*, 2019, pp. 1–2.
- [12] B. Zhou, M. Sundholm, J. Cheng, H. Cruz, and P. Lukowicz, "Never skip leg day: A novel wearable approach to monitoring gym leg exercises," in *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2016, pp. 1–9.
- [13] F. Xiao, J. Chen, X. Xie, L. Gui, L. Sun, and R. Wang, "Seare: A system for exercise activity recognition and quality evaluation based on green sensing," *IEEE Transactions on Emerging Topics in Computing*, vol. 8, no. 3, pp. 752–761, 2020.
- [14] Y. Mahajan, J. Pinnamraju, J. L. Burns, J. W. Gichoya, and S. Purkayastha, "Using machine learning approaches to identify exercise activities from a triple-synchronous biomedical sensor," in *Intelligent Systems Design and Applications*. Cham: Springer International Publishing, 2022, pp. 1216–1226.
- [15] S. Mekruksavanich and A. Jitpattanakul, "Automatic recognition of construction worker activities using deep learning approaches and wearable inertial sensors," *Intelligent Automation & Soft Computing*, vol. 36, no. 2, pp. 2111–2128, 2023.
- [16] S. Mekruksavanich and A. Jitpattanakul, "Sport-related activity recognition from wearable sensors using bidirectional gru network," *Intelligent Automation & Soft Computing*, vol. 34, no. 3, pp. 1907–1925, 2022.
- [17] S. Mekruksavanich, P. Jantawong, and A. Jitpattanakul, "Deep learning-based action recognition for pedestrian indoor localization using smartphone inertial sensors," in *2022 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT & NCON)*, 2022, pp. 346–349.
- [18] F. J. Ordóñez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, 2016.
- [19] S. Mekruksavanich and A. Jitpattanakul, "Rnn-based deep learning for physical activity recognition using smartwatch sensors: A case study of simple and complex activity recognition," *Mathematical Biosciences and Engineering*, vol. 19, no. 6, pp. 5671–5698, 2022.

- [20] A. Wijekoon, N. Wiratunga, and K. Cooper, "Mex: Multi-modal exercises dataset for human activity recognition," 2019.
- [21] D. M. Burns, N. Leung, M. Hardisty, C. M. Whyne, P. Henry, and S. McLachlin, "Shoulder physiotherapy exercise recognition: machine learning the inertial signals from a smartwatch," *Physiological Measurement*, vol. 39, no. 7, p. 075007, 2018.
- [22] S. Mekruksavanich and A. Jitpattanakul, "Hybrid convolution neural network with channel attention mechanism for sensor-based human activity recognition," *Scientific Reports*, vol. 13, no. 1, p. 12067, 2023.
- [23] S. Mekruksavanich and A. Jitpattanakul, "A deep learning network with aggregation residual transformation for human activity recognition using inertial and stretch sensors," *Computers*, vol. 12, no. 7, 2023.
- [24] S. Mekruksavanich and A. Jitpattanakul, "Sensor-based complex human activity recognition from smartwatch data using hybrid deep learning network," in *2021 36th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*, 2021, pp. 1–4.
- [25] C. Picard, V. Janko, N. Reščič, M. Gjoreski, and M. Luštrek, *Identification of Cooking Preparation Using Motion Capture Data: A Submission to the Cooking Activity Recognition Challenge*. Singapore: Springer Singapore, 2021, pp. 103–113.
- [26] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 11 1997.

Beans Disease Detection Using Convolutional Neural Network

Sumaya Akter Suma

*Department of Computer Science and Engineering
East West University
Dhaka, Bangladesh
Sumasumayaakter@gmail.com*

Akibul Haque

*Department of Computer Science and Engineering
East West University
Dhaka, Bangladesh
akibul.haque1999@gmail.com*

Nishat Vasker

*Department of Computer Science and Engineering
East West University
Dhaka, Bangladesh
nishatvasker@gmail.com*

Mahamudul Hasan

*Department of Computer Science and Engineering
East West University
Dhaka, Bangladesh
munna09bd@gmail.com*

Jesan Ahammed Ovi

*Department of Computer Science and Engineering
East West University
Dhaka, Bangladesh
jesan@ewubd.edu*

Maheen Islam

*Department of Computer Science and Engineering
East West University
Dhaka, Bangladesh
maheen@ewubd.edu*

ABSTRACT

Beans are a crucial protein source, but disease-infected crops like angular spot leaf disease and bean rust can cause financial and physical problems. A detection system is essential to prevent crop loss and ensure farmers can take appropriate actions. A convolutional neural network algorithm called AlexNet has been used to detect if a bean crop is infected or not. In this research paper, we trained AlexNet to classify if the beans are healthy or infected from the bean leaf images dataset with one class being healthy and the other class being infected. The AlexNet model achieved 99.7% accuracy on the training dataset and 96.8% accuracy on the test set, including diseased bean leaves for verification, instruction, and evaluation.

KEYWORDS

Deep Learning, Convolutional Neural Network, AlexNet, Bean Disease Detection.

I. INTRODUCTION

Infected bean leaves are a sign that all the bean crops are at risk of being infected, and proper measures need to be taken as soon as possible to avoid losing all the bean crops, which can be devastating for both the farmer and the consumers if there is a lack of bean crop production. However, nowadays, with machine learning, we can develop systems that can easily verify and detect anything, which helps us take proper measures. And convolutional neural networks are one of the best options for making systems that can classify and detect objects from images. Deep learning technology has enabled detection systems for crop diseases, enabling testing

and training models. Our AlexNet model, trained on a dataset of infected and non-infected bean leaves, differentiates between infected and non-infected leaves. In our research paper, the AlexNet model achieved 96.8% accuracy on the test dataset and 99.7% on the training dataset that contained the images of the infected bean leaves, and healthy bean leaves for validation, training, and testing the model.

II. BACKGROUND AND RELATED WORKS

There have been several research papers and deep neural network models that have been based on detecting and differentiating different crop diseases. Such as a convolutional neural network model that was trained and tested on four kinds of corn leaf images to detect corn leaf diseases so that proper prevention can be taken, which was made by Kshyanaprava Panda Panigrahi, Abhaya Kumar Sahoo and Himanshu Das and the model achieved 98.78% of average detection [1].

Another plant leaf disease detection model was developed by Jay Trivedi, Yash Shamnani and Ruchi Gajjar, their convolutional neural network architecture achieved 95.81% accuracy after being trained on Plant Village dataset which contains 54,305 images of 38 different plant classes [3].

Another instance of a deep learning model was developed in order to detect bean leaf diseases automatically was made by Sudad H. Abed, Alaa S. Al-Waisy, Hussain J. Mohammed and Shumoos Al-Fahdawi, which was evaluated on 1295 images and their best performance in binary classification task was achieved using Densenet121 model with a CAR of 98.31%, Precision of 98.45% and F1 score of 98.74% [2].

Another model made by E. Elfatimi, R. Eryigit and L. Elfatimi was trained using MobileNet architecture on a public

dataset containing two infected classes and one healthy class of bean leaves and the model achieved 97% on the training dataset and 92% on the test dataset [4].

S. Ghosal and K. Sarkar used transfer learning to develop a VGG16-based model and trained and tested the model using a dataset collected from different rice fields and the images available on the internet. Their model achieved an accuracy of 92.46% [5].

Prajwala Tm, Alla Pranathi, Kandiraju SaiAshritha, Nagaratna B. Chittaragi, Shashidhar G. Koolagudi developed a small variation of the convolutional neural network model called LeNet for the detection and identification of tomato leaves diseases and their model was able to achieve an average accuracy of 94% – 95% even under the harsh conditions [6].

S. V. Militante, B. D. Gerardo and R. P. Medina developed a model using deep learning techniques and computer vision to detect sugarcane diseases and help the sugarcane farmers with their model which is based on deep learning algorithms and the model was able to achieve accuracy of 95% after it was trained and tested on a dataset which had 13,842 sugarcane image containing two classes, which were infected and healthy leaves [7].

Another deep learning-based design was developed by S. V. Militante, B. D. Gerardo and N.V. Dionisio where the designed system was able to detect multiple diseases in various plants the dataset that was used for training the model consisted 35,000 images of healthy and infected plant leaves and the deep learning model was able to detect and recognize plant diseases and the absence these of diseases. Their trained model achieved an accuracy rate of 96.5% and the designed system was able to register up to 100% accuracy in detecting and recognizing the plant variety and the type of diseases the plants were infected with [8].

Y. Xu, B. Zhao, Y. Zhai, Q. Chen and Y. Zhou developed a deep learning model based on an AlexNet model, a convolutional layer and new Inception module were added in order to increase the ability of the AlexNet features extraction and to avoid over-fitting because of too many parameters, that is why they used the global pooling layer in order to replace the original fully-connected layer and besides that, they also used transfer learning method so that they can avoid over-fitting due to the lack of sample data. Their improved model was able to reduce over-fitting and epochs to increase the performance of their deep learning model compared with convolutional neural network models VGGNet-16, DenseNet, ResNet-50 and AlexNet in recognition accuracy for maize disease recognition [9].

Overall, our model performs the best and achieves the highest accuracy out of all the models analyzed in the previous study.

III. METHODOLOGY

In our research, we used deep convolutional neural networks (CNN's) called AlexNet, which have two sections: a convolutional section for finding the most important characteristics and a fully connected section for collecting relevant data. Hyperparameters and trainable parameters are the two types of

parameters we implemented. Hyperparameters are parametric variables that must be defined before using a learning method, whereas trainable parameters are non-parametric variables that are automatically calculated during training. Model hyperparameters, such as hidden units, layers, filters, and convolutional blocks, are involved in the construction of the model. Optimizer hyperparameters are relevant to the optimization and training processes.

A. MODEL VISUALIZATION

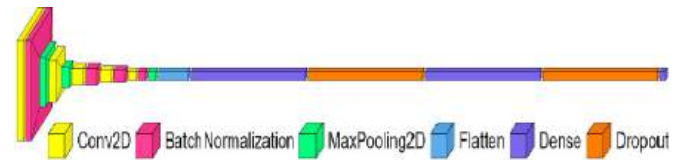


Fig. 1. Visualization of the model using visulkeras

A visual representation of the model has been shown in Figure 1 using visulkeras, where we can see the layers of the model and understand how the model works.

B. FLOW CHART

The system of the proposed deep learning model is represented in the following flowchart in Figure 2.

C. STEPS OF THE SYSTEM

- Step-1: The images need to be preprocessed and for this model since there are only 1035 trainable data, so using ImageDataGenerator functions where the zoom_range = 0.2, rescale = 1/255, shear_range = 0.2, horizontal flip has been set to True. With this data augmentation the overfitting issue was lowered for this model.
- Step-2: First we provide an image which is 227 x 227 x 3 where 227 is the height and width and 3 is the RGB channel and then we have a convolutional layer with a filter size 11 x 11 and the stride is set to 4 and we are using 96 kernels. This convolutional layer turns the image into a 55 x 55 x 96 size.
- Step-3: Then the max pooling layer with a stride of 2 and 3 x 3 of size turn the image into 27 x 27 x 96 where the kernel size remains the same.
- Step-4: After that we have another convolutional layer of 5 x 5 with padding = 2 with 256 kernels and the image becomes 27 x 27 x 256 which is an increase of kernels at this stage of the process.
- Step-5: We are doing max pooling after the previous layer and this time in max pooling it is again 3 x 3 with the stride = 2 and the image becomes 13 x 13 x 256 at this stage.
- Step-6: At this stage we use another convolutional layer with 3 x 3 with the padding of 1 and kernel size = 384 which turns the image into 13 x 13 x 384.
- Step-7: We use another convolutional layer again with 3 x 3 padding = 1 kernel size 384 and then pass it through another convolutional layer with 3 x 3 with same padding as before and this time we use 286 kernels.

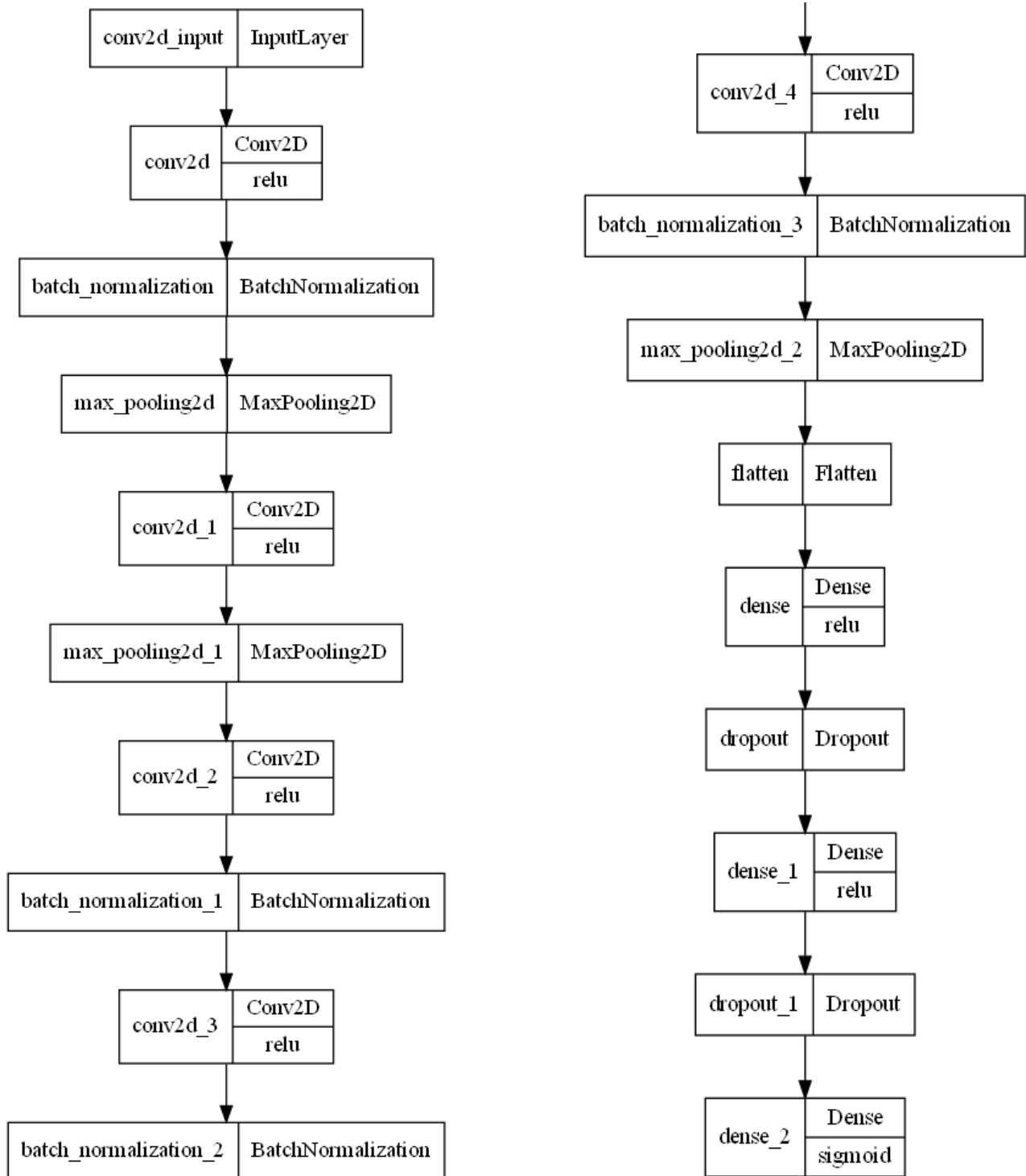


Fig. 2. Flow chart of the system using convolutional neural network.

- Step-8: Then we used another max pooling layer of 3 x 3 and stride = 2 which turns the image into 6 x 6 x 256 and finally get a fully connected layer with 4096 nodes which are the dense nodes. Where the image input will go through these dense nodes and through another layer of dense nodes. And another layer before the final layer which has a dropout rate of 0.2.
- Step-9: Finally, the output layer consists of 2 nodes since it is binary classification and the output will be either

towards 0 or 1.

IV. EXPERIMENTAL SETUP

A. DATA SET

The dataset we used to train the proposed deep learning model were images of the infected and healthy bean leaf images that were taken in the various districts of Uganda by the Makerere AI lab which is in collaboration with the National

Crops Resources Research Institute, which is the national body in charge of research in agriculture in Uganda [10].

There are 990 images in total in the dataset. The dataset is divided into three classes: healthy, bean rust, and angular leaf spot. The training, testing, and validation sets of the dataset were already separated. Figure 3 shows the image of the dataset.

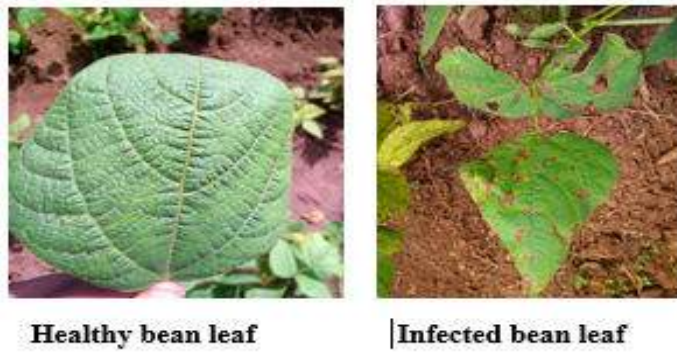


Fig. 3. Image of the Dataset

B. METRICS

To determine the performance of the proposed deep learning model, precision, recall and F1-score were calculated by evaluating the proposed model with a test image set using Keras precision and recall metrics. F1-score was calculated from the values of recall and precision after evaluation of the proposed model. Equation 1 shows that the recall of a deep learning model measures the model's ability to detect positive samples. The recall was calculated by dividing the true positives by the sum of the true positives and false negatives.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad (1)$$

Equation 2 shows the precision of a deep learning model and indicates how well the model is performing. Precision is calculated by dividing the true positives by the sum of true positives and false positives.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad (2)$$

The F1 score was calculated using the following formula:

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)$$

For our AlexNet model, the Keras built-in metrics precision and recall were used inside the compile of the model. Equation 3 shows that the F1-score was calculated from the precision and recall values after the evaluation of the model on another test dataset.

V. RESULT AND DISCUSSION

In Table 1, the results of the f1-score, precision, and recall are shown after evaluating the model on a test dataset, and the plots of accuracy and loss against epochs are shown in plots to show the progress of the model throughout the training duration. From the figure 4 plot, we can see that as we train the model more, the accuracy keeps increasing, even though there was a drop in accuracy around 80 epochs. After that, we can see a steady growth in accuracy. For the figure-5 loss vs. epochs plot, the more we trained the model, the loss kept decreasing except around 80 epochs, but after that, there was a steady decrease in loss in the model.

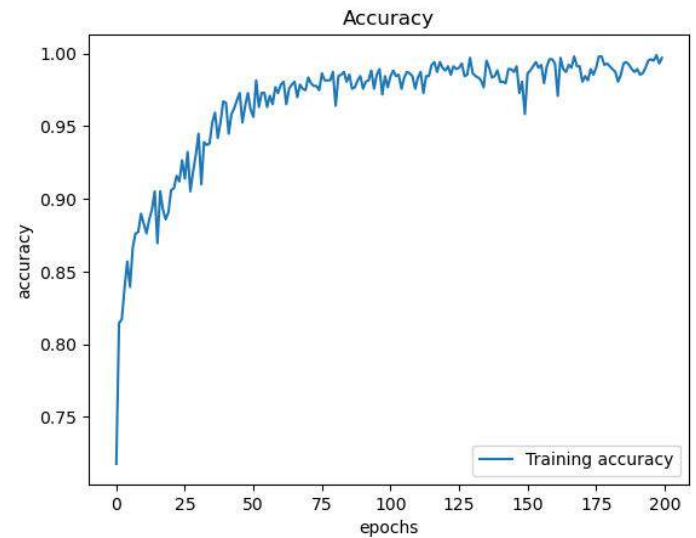


Fig. 4. Accuracy Vs Epochs.

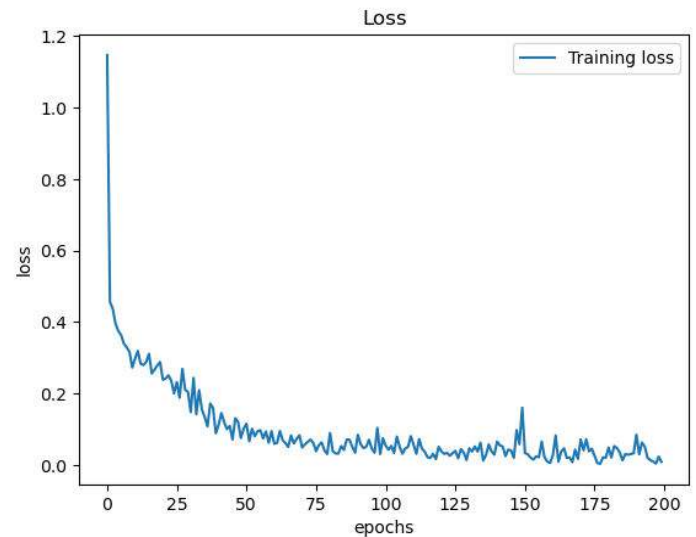


Fig. 5. Loss Vs Epochs.

VI. CONCLUSION

We implemented the AlexNet technique in our research paper. It is a model of a convolutional neural network. For bean

TABLE I
RESULT OF PRECISION, RECALL AND F1 SCORE

Parameter	Result
Recall	84%
Precision	83.25%
F1 Score	83.25%

leaves, a model has been proposed for finding bean disease. In this model, it is simple to tell whether a bean leaf is infected or not simply by looking at it. This significantly helps farmers and plant pathologists determine whether a bean leaf is healthy or not without wasting time and resources on costly laboratory tests. simply by inputting images from any smart device. Anyone can quickly determine if a bean leaf is healthy or infected using this approach. The model's limitations include a lack of explanation, optimism in its results, generalization, false positives, data bias, robustness to variability, and visual patterns. Adaptive data sources, transfer learning, semi-supervised learning, active learning, and collaborative platforms can all be used to increase accuracy and resilience. These methods improve user input, strengthen the model's outputs, and adjust it for similar plant species or diseases.

REFERENCES

- 1) K. P. Panigrahi, A. K. Sahoo and H. Das, "A CNN Approach for Corn Leaves Disease Detection to support Digital Agricultural System," 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184), Tirunelveli, India, 2020, pp. 678-683, doi: 10.1109/ICOEI48184.2020.9142871.
- 2) Abed, S.H., Al-Waisy, A.S., Mohammed, H.J. et al. A modern deep learning framework in robot vision for automated bean leaves diseases detection. *Int J Intell Robot Appl* 5, 235–251 (2021). <https://doi.org/10.1007/s41315-021-00174-3>
- 3) Trivedi, J., Shamnani, Y., Gajjar, R. (2020). Plant Leaf Disease Detection Using Machine Learning. In: Gupta, S., Sarvaiya, J. (eds) *Emerging Technology Trends in Electronics, Communication and Networking*. ET2ECN 2020. Communications in Computer and Information Science, vol 1214. Springer, Singapore. https://doi.org/10.1007/978-981-15-7219-7_23
- 4) E. Elfatimi, R. Eryigit and L. Elfatimi, "Beans Leaf Diseases Classification Using MobileNet Models," in *IEEE Access*, vol. 10, pp. 9471-9482, 2022, doi: 10.1109/ACCESS.2022.3142817.
- 5) S. Ghosal and K. Sarkar, "Rice Leaf Diseases Classification Using CNN With Transfer Learning," 2020 IEEE Calcutta Conference (CALCON), Kolkata, India, 2020, pp. 230-236, doi: 10.1109/CALCON49167.2020.9106423.
- 6) P. Tm, A. Pranathi, K. SaiAshritha, N. B. Chittaragi and S. G. Koolagudi, "Tomato Leaf Disease Detection Using Convolutional Neural Networks," 2018 Eleventh International Conference on Contemporary Computing (IC3), Noida, India, 2018, pp. 1-5, doi: 10.1109/IC3.2018.8530532.
- 7) S. V. Militante, B. D. Gerardo and R. P. Medina, "Sugarcane Disease Recognition using Deep Learning," 2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE), Yunlin, Taiwan, 2019, pp. 575-578, doi: 10.1109/ECICE47484.2019.8942690.
- 8) S. V. Militante, B. D. Gerardo and N. V. Dionisio, "Plant Leaf Detection and Disease Recognition using Deep Learning," 2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE), Yunlin, Taiwan, 2019, pp. 579-582, doi: 10.1109/ECICE47484.2019.8942686.
- 9) Y. Xu, B. Zhao, Y. Zhai, Q. Chen and Y. Zhou, "Maize Diseases Identification Method Based on Multi-Scale Convolutional Global Pooling Neural Network," in *IEEE Access*, vol. 9, pp. 27959-27970, 2021, doi: 10.1109/ACCESS.2021.3058267.
- 10) Makerere Artificial Intelligence Lab & National Crops Resources Research Institute. Bean Disease Dataset. Retrieved from <https://www.kaggle.com/datasets/therealose/bean-disease-dataset>.

Image Recognition of Different Tea Quality Based on Convolutional Neural Network

1st Sarmin Sultana Snigdha

Department of Computer Science and Engineering
East West University
Dhaka, Bangladesh
imsnigdha06@gmail.com

2nd Reyad Bin Satter

Department of Computer Science and Engineering
East West University
Dhaka, Bangladesh
rbsreyad@gmail.com

3rd Waliul Islam

Department of Computer Science and Engineering
East West University
Dhaka, Bangladesh
waliul153cse@gmail.com

4th Shafiur Rahman Chowdhury Tonay

Department of Computer Science and Engineering
East West University
Dhaka, Bangladesh
shafiurrah765@gmail.com

5th Mahamudul Hasan

Department of Computer Science and Engineering
East West University
Dhaka, Bangladesh
munna09bd@gmail.com

6th Maheen Islam

Department of Computer Science and Engineering
East West University
Dhaka, Bangladesh
maheen@ewubd.edu

7th Md Sawkat Ali

Department of Computer Science and Engineering
East West University
Dhaka, Bangladesh
alim@ewubd.edu

8th Taskeed Jabid

Department of Computer Science and Engineering
East West University
Dhaka, Bangladesh
taskeed@ewubd.edu

Abstract—Tea is one of the most famous beverages taken by mankind around the world. The quality of tea processing plays a fundamental role in determining its market value. In Bangladesh, cultivation, and tea production plays a vital role, and it is an integral part of the economy also. Due to the lack of an advanced quality monitoring technique tea customers have been facing difficulties in consuming good quality tea. The tea producers seek to modernize their production techniques to satisfy customer demands for products with greater consistency, better flavor, and product differentiation. This has promoted the exploration of the utilization of techniques to enhance quality through observance throughout the production process. To solve this problem image processing is one of the best options to detect the quality. The main goal of this research is to develop an image-processing system that can identify and classify the quality of tea. We applied CNN, VGG16, AlexNet, InceptionV3, and DenseNet121 algorithms in our study. Among the five models in this study, the InceptionV3 model has the highest training accuracy of 99.84 percent.

Index Terms—Tea, Tea quality, Image preprocessing, CNN, Image recognition,

I. INTRODUCTION

Customer satisfaction is the first priority in the tea sector. An enduring consumer is priceless. There is already competition among the corporations because there are numerous tea

producers. Customer happiness is the most crucial element in making a business stand out from its rivals. Consumption by the customer is influenced by quality. If a business offers high-quality tea, both the client and the business will be happy. The business will make the right amount of money, and the customers will receive high-quality goods. Quality has an impact on consumer satisfaction. Companies can make the required efforts to ensure that the consumer receives their best product and is satisfied if they can somehow maintain the target level of customer satisfaction. They will be able to provide clients with a better product as they receive more customer feedback. The three most crucial physical characteristics of tea are color, grain form, and size. When determining a product's quality, processed food's color, size, and form are all important considerations. Color and size are crucial components in the food processing industry. Since it affects the final quality of the tea, the fermentation process is essential to the production of tea. The quantity of material makes it difficult to frequently monitor the quality of the tea grains throughout fermentation, though. The way tea grains look, including their size and color, can tell you a lot about the tea's quality. CNN is a particular type of deep feed-forward ANN. Its local connection, weight sharing, and pooling functions

3

enable efficient network complexity reduction. It can make the model somewhat invariant to distortions, scaling, and other operations, which significantly increases its robustness and fault tolerance. As a result, training and enhancing the network structure are straightforward. CNN is therefore becoming increasingly valuable for the field of diagnosing crop diseases. Although there are images of tea leaves on CNN, there is no research on the diagnosis of leaf disease. This paper examines CNN's use of photos of tea leaf sickness. By changing the learning rate, iteration periods, and dropout, the network can converge to a stable value. Finally, the accuracy of CNN is raised and tested by comparing its results with those of other machine learning methods. The identification effectiveness of the disease image has a specified reference value for the automatic and rapid detection of tea leaf diseases.

II. RELATED WORKS

In the study [1], they classify tea quality levels using near-infrared spectroscopy based on CLPSO-SVM. The accuracy of their classification model was 99.17%. In this study [2], they compared the traditional machine learning algorithms, such as Support Vector Machine (SVM) and Random Forest, with TeaNet, TeaResnet, and TeaMobilenet in a thorough experimental investigation. The experimental results show that TeaNet, TeaResnet, and TeaMobilenet have the ability to surpass the accuracy limit of conventional machine learning algorithms and obtain 100% accuracy. Another study [3] discovered the optimal mixture of NIRS and CVS properties. From 99.29%, 96.67%, and 98.57% accuracy, the calibration, validation, and test sets' accuracy rose to 100.00%, 99.29%, and 100.00%. This study shows that combining NIRS and CVS features can be a useful way to categorize different types of black tea samples.

According to a study [4], the total J48 DT classification accuracy for black and green teas was 92.917% and 95%, respectively, and for REP trees it was 90.83% and 97.08%. Although J48 and REP trees performed nearly equally well, the REP tree structure was chosen because of its more straightforward design in order to identify the membership functions and rules of the fuzzy system. It was found that the DT-based fuzzy logic system can be used well to categorize several quality categories of tea based on image extraction attributes.

Jiewen Zhao, Quansheng Chen, Jianrong Cai, and Qin Ouyang developed the classification model using a support vector machine (SVM) in their study. The training and prediction sets had classification rates of 98% and 95%, respectively. The SVM algorithm exhibits excellent classification results when compared to other pattern recognition classifiers. Overall, the findings imply that green tea can be successfully classified using the hyperspectral imaging approach and an SVM classifier [5]. In their research study, Guangxin Ren, Ying Liu, Jingming Ning, and Zhengzhu Zhang showed that the CARS-linear kernel SVM model generated the most promising outcomes, with an accurate identification rate of 91.85% during the validation stage. According to the results, Vis-NIR spectroscopy can be used as an alternate

instrument for determining the quality of black tea that is quick, affordable, efficient, and reliable [6]. The findings of a study by Kaihua Wei, Bojian Chen, Zejian Li, Dongmei Chen, Guangyu Liu, Hongze Lin, and Baihua Zhang show that the performance of the VGG16 model is superior to the ResNet-34 model in their particular situation, and the classification model's accuracy is better when based on fluorescence images than when based on white-light illumination images. The accuracy of classifying fluorescence images reached 97.5%, indicating that the approach to LED-induced fluorescence imaging is promising for usage in daily life [7]. In a study by Qiang Cui, Baohua Yang, Biyun Liu, Yunlong Li, and Jingming Ning, tea recognition results utilizing the L-CNN-SVM technique outperformed MobileNet v2+RF, MobileNet v2+KNN, MobileNet v2+AdaBoost, AlexNet, and MobileNet v2. The three teas' combined accuracy rate using wavelet component reconstruction was 98.7% [8]. The MSC-RF-RFE-SVM is the best model for the classification of oolong tea, with training and test set accuracy of 100% and 98.73%, respectively, according to Yan Hu, Lijia Xu, Peng Huang, Xiong Luo, Peng Wang, and Zhiliang Kang's research article. It is feasible to infer that deep learning and fluorescence hyperspectral technologies can be utilized to categorize oolong tea [9].

III. METHODOLOGY

We went over how we processed our dataset, a full discussion of our model architecture, and the implementation process. To show how the model works, we'll start at the beginning, then move on to how the model takes the data and processes it, and finally how we calculate the results. In image processing approaches, normal identification, on the other hand, is not very accurate. As a result, we'll need to train the processor on the benefits of tea. To observe that the tea is trained using an image processing method. In image processing approaches, normal identification, on the other hand, is not very accurate. As a result, we'll need to train the processor on the benefits of tea. To observe that the tea is trained using an image processing method. Deep learning, machine learning, and AI applications have seen significant growth as a result of the improvement of efficient devices such as graphics processing units (GPU's). CNN is a deep learning model in which the conversational layer extracts information from the input and output. Image classification, pattern recognition, speech recognition, object identification, and other applications. A convoluted neural network is a deep learning model for solving classification challenges involving sophisticated pattern recognition and big databases. The uniqueness of architecture rests in its adaptability to various configurations based on executive outcomes. Many CNN models are available, including AlexNet, VGG, GoogleNet, ResNet, and others. To tackle pattern recognition difficulties, several customizable parameters such as dropout rate, categorization, and learning rate are used in complicated processing.

A. The architecture of CNN

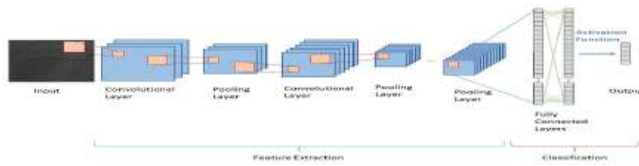


Fig. 1. Architecture of CNN

Convolutional neural networks (CNN's) are a type of deep neural network that was developed in the 1980s to recognize handwritten digits.

B. The architecture of VGG16

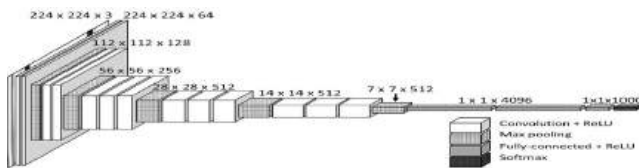


Fig. 2. Architecture of VGG16

Each of the 4090 nodes has two dense or fully connected layers. A dense or output layer with 6 nodes of the size that classifies image nets into 6 classes.

C. The architecture of AlexNet

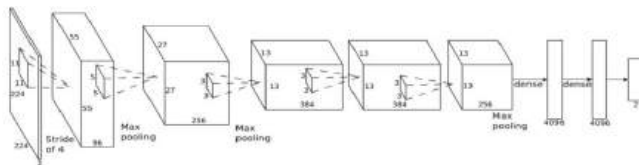


Fig. 3. Architecture of AlexNet

A 256x256 RGB image can be entered into AlexNet. It means that each image in the training set and each image in the test set must have a resolution of 256x256 pixels. The second layer that is entirely connected feeds six class labels on a softmax classifier.

D. The Architecture of Inception

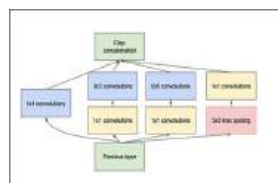


Fig. 4. Architecture of InceptionV3

InceptionV3 is the first game in the series. It's essentially a 22-layer deep convolutional neural network (CNN). Resized images of four scales with shorter dimensions of 256, 288,

320, and 352 were used in Inception V3. Inception v3 enhanced the number of units in each layer by adding parallel filters termed inception modules in each convolution layer with sizes of 1x1, 3x3, and 5x5. They also expanded the number of layers to 22. 1x1 before introducing another layer, which is mostly utilized for dimensionality reduction.

E. The Architecture of DenseNet121

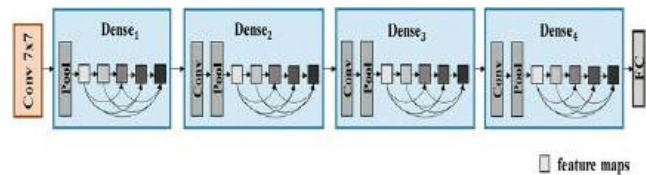


Fig. 5. Architecture of DenseNet121

A specific sort of convolutional neural network known as a "Dense Net" uses Dense Blocks to directly connect all layers to one another, forming dense connections between them. Each layer takes extra inputs from all lower layers for the purpose to preserve the feed-forward nature, and it transmits its feature maps to all upper layers in order to retain the feed-forward character.

IV. EXPERIMENTAL SETUP

We generated deep learning models with the Keras library on top of TensorFlow. All machine learning models have been trained on a computer with an AMD Ryzen 5 3600 processor, 16GB of RAM, and operating system Windows 10. And to get results easily, we use google collaboratory.

A. Data Set

At first, we bought personally 6 types of tea powder from Srimangal, Sylhet. Those are Green, Kaiyachera, WPF, YOF, YGBOP, Special. We create our data set by using smartphones to take pictures. Then we analyze to see if the dataset contains any duplicate data. The duplicate image data is then removed. The dataset was split into two parts: training and validation. There are 3370 images in the training set and 390 in the validation set, divided into six classes.



Fig. 6. Image of the Dataset

3

B. Metrics

Our performance is assessed using accuracy, recall, F1 score, precision, and precision. Let's start with the precision mathematical formula.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad (1)$$

When analyzing if False Positive costs are excessive, precision is a valuable criterion. The term "precision" describes how exact or precise our model is in terms of how many of the predicted positives really materialize as positive. Precision and recall are two statistics on opposite ends of the scale. If one falls, the other will rise. Let's look at the equation:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad (2)$$

We use recall as our metric whenever the expense of having False Negatives is too excessive. The F1 score is calculated as follows:

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)$$

Because we want the F1 score to have a low value when either precision or recall is 0, we utilize the harmonic mean rather than the typical arithmetic mean. The F1 score is a single statistic that combines precision and recall.

Accuracy is defined as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

Here TP = True Positives, TN = True Negatives, FP = False Positives, FN = False Negatives.

V. RESULT AND DISCUSSION

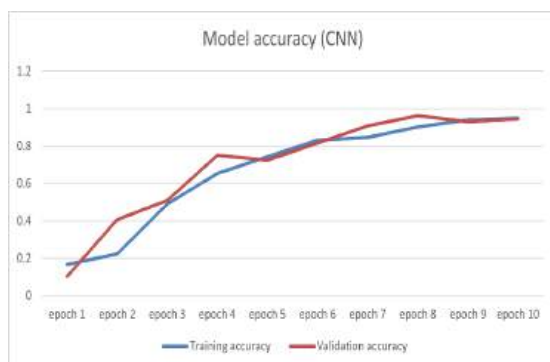


Fig. 7. Training and validation accuracy of CNN

Here the training and validation accuracy of CNN has been shown in the Fig. 7.

Here the training and validation loss of CNN has been shown in the Fig. 8.

Here the training and validation accuracy of InceptionV3 has been shown in the Fig. 9.

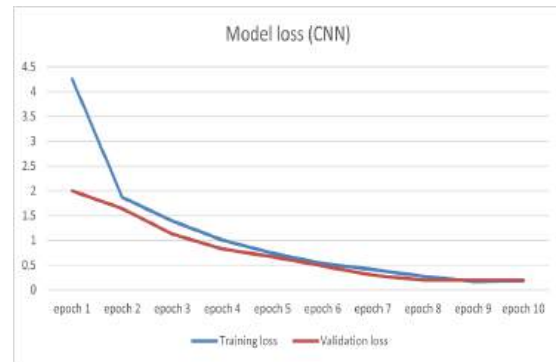


Fig. 8. Training and validation loss of CNN

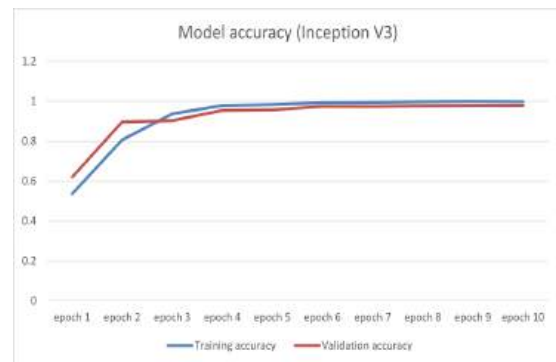


Fig. 9. Training and validation accuracy of InceptionV3

Here the training and validation loss of InceptionV3 has been shown in the Fig. 10.

Here the training and validation accuracy of DenseNet121 has been shown in the Fig. 11.

Here the training and validation loss of DenseNet121 has been shown in the Fig. 12.

Here the training and validation accuracy of VGG16 has been shown in the Fig. 13.

Here the training and validation loss of VGG16 has been shown in the Fig. 14.

Here the training and validation accuracy of AlexNet has been shown in the Fig. 15.

Here the training and validation accuracy of AlexNet has been shown in the Fig. 16.

Here in the Fig. 17, we can see all the accuracy of the algorithms. In the graph, we can see the validation accuracy is practically linearly increasing. InceptionV3 gives the best accuracy. The second-best accuracy gives VGG16 and then gradually DenseNet, AlexNet, and at last CNN.

Here in the Fig. 18, we can see all the losses of the algorithms. In the graph, we can see the validation loss is practically linearly decreasing. VGG16 gives the lowest loss. The second-lowest loss gives DenseNet and then gradually InceptionV3, CNN, and at last Alexnet.

We used CNN, VGG16, AlexNet, InceptionV3, and DenseNet121 in our study. On CNN, VGG16, AlexNet, InceptionV3, DenseNet121, we received test accuracies of 92.63%,

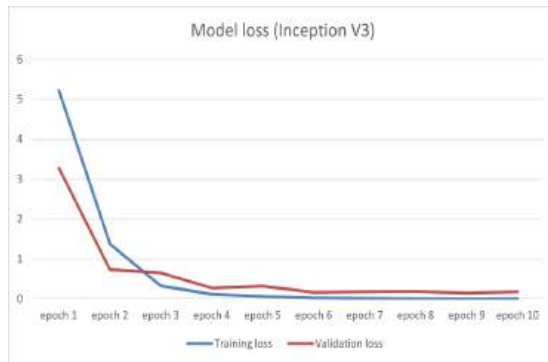


Fig. 10. Training and validation loss of InceptionV3

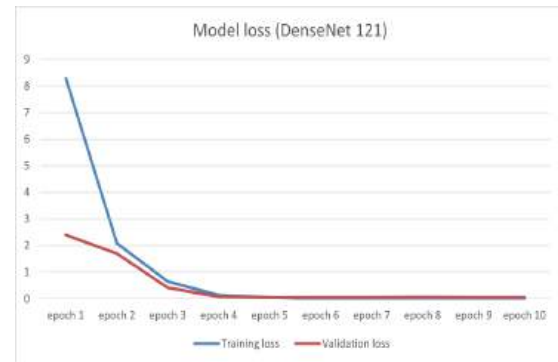


Fig. 12. Training and validation loss of DenseNet121

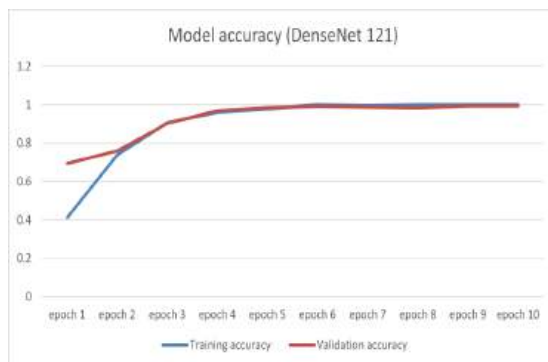


Fig. 11. Training and validation accuracy of DenseNet121

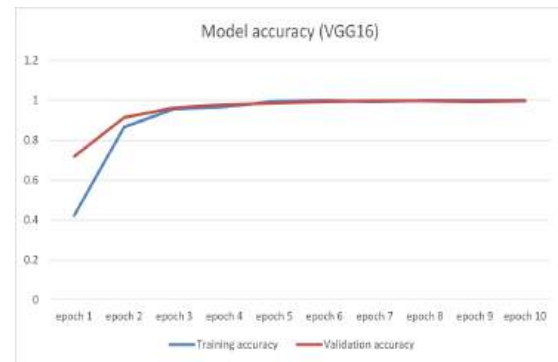


Fig. 13. Training and validation accuracy of VGG16

99.77%, 82.21%, 97.47%, and 99.08%, respectively. In terms of test accuracies, we were able to achieve low performance. However, VGG16 is the best in the low accuracy values. The predicted class is in the column, and the true class is in the row in the confusion matrix. CNN, VGG16, AlexNet, InceptionV3, DenseNet121 have precision values of 93%, 100%, 83%, 98%, and 99% respectively. CNN, VGG16, AlexNet, InceptionV3, DenseNet121 have recall values of 93%, 100%, 83%, 98%, and 99%, respectively. CNN, VGG16, AlexNet, InceptionV3, DenseNet121 had F1 scores of 92%, 100%, 83%, 98%, and 99% respectively. Because the F1 score is based on recall and

VI. CONCLUSION

In this paper, we proposed a method to identify tea quality. The developed system achieved an accuracy of 98% in classifying 6 different categories of tea. Our method only classifies 6 kinds of tea, but there are many kinds of tea available nowadays. The images taken by mobile phone cameras may be different in size, focus, quality and also the distance between the tea powder and the mobile phone can vary from person to person. The dataset used in our experiment contains only 3760 samples. By generating an extensive database with a bust number of different tea powders, better accuracy can be achieved. We hope to implement other methodology by proposing Deep Neural Networks and comparing them with our models. Working with other products with economic importance and identifying the quality of the products.

REFERENCES

- 1) Ding, Y.; Yan, Y.; Li, J.; Chen, X.; Jiang, H. Classification of Tea Quality Levels Using Near-Infrared Spectroscopy Based on CLPSO-SVM. *Foods* 2022, 11, 1658. <https://doi.org/10.3390/foods11111658>.
- 2) Jingru Yang, Jin Wang, Guodong Lu, Shaomei Fei, Ting Yan, Cheng Zhang, Xiaohui Lu, Zhiyong Yu, Wencui Li, Xiaolin Tang, TeaNet: Deep learning on Near-Infrared Spectroscopy (NIR) data for the assurance of tea quality, *Computers and Electronics in Agriculture*, Volume 190, 2021, 106431, ISSN 0168-1699, <https://doi.org/10.1016/j.compag.2021.106431>.
- 3) Yan Song, Xiaozhong Wang, Hanlei Xie, Luqing Li, Jingming Ning, Zhengzhu Zhang, Quality evaluation of Keemun black tea by fusing data obtained from near-infrared reflectance spectroscopy and computer vision sensors, *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, Volume 252, 2021, 119522, ISSN 1386 1425, <https://doi.org/10.1016/j.saa.2021.119522>.
- 4) Bakhshipour, A., Zareiforoush, H. & Bagheri, I. Application of decision trees and fuzzy inference system for quality classification and modeling of black and green tea based on visual features. *Food Measure* 14, 1402–1416 (2020). <https://doi.org/10.1007/s11694-020-00390-8>.
- 5) Jiewen Zhao, Quansheng Chen, Jianrong Cai, and Qin

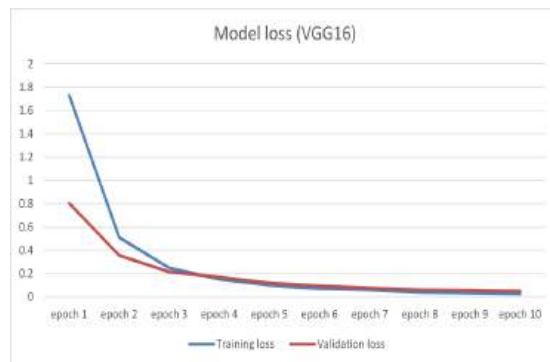


Fig. 14. Training and validation loss of VGG16

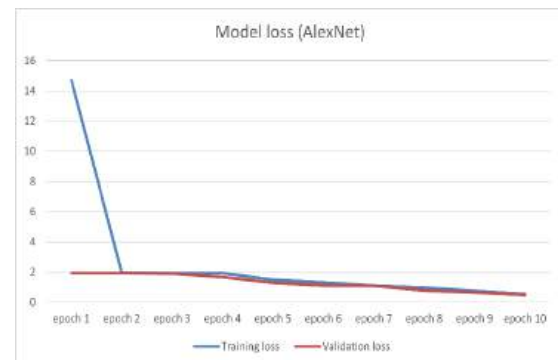


Fig. 16. Training and validation loss of AlexNet

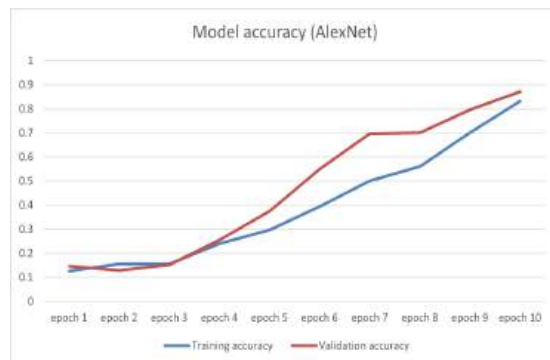


Fig. 15. Training and validation accuracy of AlexNet

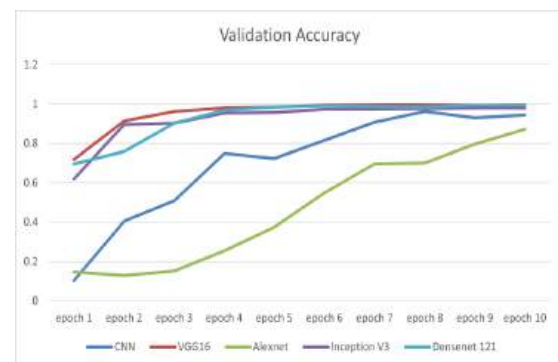


Fig. 17. Comparison of architectures based on validation accuracy

Ouyang, "Automated tea quality classification by hyper-spectral imaging," *Appl. Opt.* 48, 3557-3564 (2009).

- 6) Guangxin Ren, Ying Liu, Jingming Ning, Zhengzhu Zhang, Assessing black tea quality based on visible-near infrared spectra and kernel-based methods, *Journal of Food Composition and Analysis*, Volume 98, 2021, 103810, ISSN 0889-1575, <https://doi.org/10.1016/j.jfca.2021.103810>.
- 7) Wei, K.; Chen, B.; Li, Z.; Chen, D.; Liu, G.; Lin, H.; Zhang, B. Classification of Tea Leaves Based on Fluorescence Imaging and Convolutional Neural Networks. *Sensors* 2022, 22, 7764. <https://doi.org/10.3390/s22207764>.
- 8) Cui, Q.; Yang, B.; Liu, B.; Li, Y.; Ning, J. Tea Category Identification Using Wavelet Signal Reconstruction of Hyperspectral Imagery and Machine Learning. *Agriculture* 2022, 12, 1085. <https://doi.org/10.3390/agriculture12081085>.
- 9) Hu, Y.; Xu, L.; Huang, P.; Luo, X.; Wang, P.; Kang, Z. Reliable Identification of Oolong Tea Species: Nondestructive Testing Classification Based on Fluorescence Hyperspectral Technology and Machine Learning. *Agriculture* 2021, 11, 1106. <https://doi.org/10.3390/agriculture11111106>.

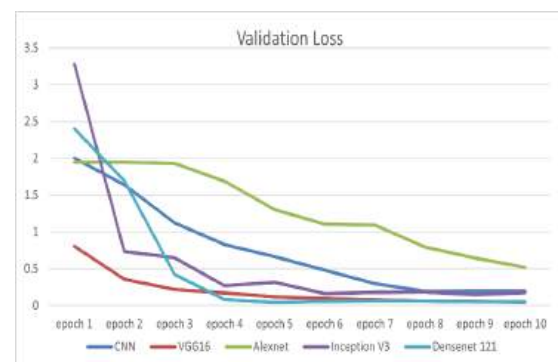


Fig. 18. Comparison of architectures based on validation loss

TABLE I
COMPARISON OF THIS STUDY WITH DIFFERENT ALGORITHMS
BASED ON ACCURACY

Model	Training accuracy	Validation accuracy
CNN	94.84%	94.54%
VGG16	100.00%	99.71%
AlexNet	83.16%	87.19%
InceptionV3	99.84%	97.99%
DenseNet121	100.00%	99.43%

Deep Learning-assisted Fracture Diagnosis: Real-time Femur Fracture Diagnosis and Categorization

Nishat Vasker *

*Department of Computer Science and Engineering
East West University
Dhaka, Bangladesh
nishatvasker@gmail.com*

Syed Nawazish Haider

*Department of Computer Science and Engineering
East West University
Dhaka, Bangladesh
nawazishhaider31@gmail.com*

Mahamudul Hasan

*Department of Computer Science and Engineering
East West University
Dhaka, Bangladesh
munna09bd@gmail.com*

Mohammad Salah Uddin

*Department of Computer Science and Engineering
East West University
Dhaka, Bangladesh
uddin@ewubd.edu*

Abstract—Fractures necessitate accurate and timely diagnosis to enable proper medical treatment, and are a frequent occurrence in the human body. X-ray image-based fracture detection through manual means is prone to human error and is a time-intensive process. This study proposes a new method for fracture diagnosis using AI-assisted techniques, particularly deep learning. The goal is to allay worries regarding fracture detection. A DNN model was created in order to quickly identify and classify femur fractures. For improving model accuracy and preventing overfitting when working with tiny datasets, data augmentation techniques are essential. Three trials were used in the study to assess the model's effectiveness utilizing the Adam optimizer and softmax activation. Fivefold cross-validation was used to assess the proposed model's capability to distinguish between healthy and fractured femurs. A 92.44% accuracy percentage was attained. On numerous subsets, the data accuracy was examined, and it was discovered to be greater than 95% on 10% of the data and 93% on 20% of the data. The model is more effective than currently used methods. The study examines how quickly and precisely artificial intelligence (AI) and deep learning can diagnose femur fractures. The study suggests an AI system that might revolutionize the identification of femur fractures. To improve speed and detection accuracy, the technique incorporates real-time analysis and classification of femur fractures.

Index Terms—AI-assisted fracture diagnosis, real-time detection, classification, bone fractures, deep learning, deep neural network (DNN), data augmentation, accuracy, X-ray images, medical intervention, overfitting, softmax activation, Adam optimizer, revolutionizing fracture detection.

I. INTRODUCTION

Femur fractures are common in the human body and are commonly triggered by traumatic events like slips and falls or vehicle accidents. A decrease in bone density brought on by aging raises the risk of femoral fractures in the elderly population. Giving patients the right medical care requires prompt and precise identification of femur fractures. The use of X-

ray and MRI scanners by skilled medical personnel is often required for the diagnosis and assessment of fractured femurs in the area of medicine. Manual methods for diagnosing femur fractures are known to take more time and have a greater rate of mistakes than automated methods[1]. Increasing the accuracy requires the creation of a computerized system. The creation of a computerized system is essential to improving the accuracy of femur fracture recognition, speeding up the diagnostic process, and lowering the possibility of mistakes. To increase the reliability of femur fracture detection, accelerate the diagnostic procedure, and reduce the risk of errors, a computerized system must be developed.

Machine learning technology developments have had significant impacts on several industries, including power electronics and medical imaging. The purpose of this study is to determine if computer-assisted diagnostic systems can successfully detect femur fractures using X-ray and MRI imaging techniques. Preprocessing procedures must be used correctly in order to improve the quality of femur scans by eliminating any unwanted noise. An overview of feature extraction methods used to analyze femur pictures is given in the paper, along with information on how frequently machine learning algorithms are utilized to accomplish classification objectives.

Previous research looked into several methods for identifying and categorizing femur fractures. This work suggests a unique approach way for detecting femur fractures that rely on deep learning and AI-assisted techniques. The aim of this research is to create a deep neural network (DNN) model that can rapidly and reliably identify and categorize femur fractures. A total of 100 distinct images of human femurs taken in various postures made up the dataset utilized in this investigation. For the purpose of addressing overfitting within the small dataset, a variety of picture-enhancing techniques

were used.

The accuracy of the model categorization was assessed in this work using a variety of qualitative analysis techniques, such as softmax activation and Adam's optimizer. The model, using a fivefold cross-validation procedure, was shown to have an accuracy rate of 92.44 percent in differentiating between intact and broken femurs. The study's results demonstrated excellent levels of accuracy, with an analysis of a randomly chosen 10% of the entire dataset yielding an absolute success rate of over 95%.

The methods will be used, as shown in the current study. We are pleased to introduce the proposed technique and give details on the dataset in Section 2. For clarity and structure, this material has been divided into four sections. The experiment's findings are provided in the third section of the publication. The study's fourth portion draws to a close by discussing the results' ramifications and outlining prospective future research trajectories in this area.

II. RELATED WORKS

"Bone Fracture Detection and Classification Using Deep Learning Approach," by Yadav and Rathor (2020), was presented. Their research used deep learning to identify and categorize bone fractures, demonstrating the promise of machine learning methods for automating fracture diagnosis [1]. In a comprehensive survey, Joshi and Singh (2020) provided insights into "A Survey of Fracture Detection Techniques in Bone X-ray Images." They explored various techniques for detecting fractures in bone X-ray images, offering a comprehensive overview of the state-of-the-art methods in the field [2]. Yang and Ding (2020) proposed a novel approach titled "Computer-Aided Fracture Diagnosis Based on Integrated Learning." Their work demonstrated the integration of multiple learning methods for improved fracture diagnosis, highlighting the benefits of combining different approaches [3]. Beyaz et al. (2020) introduced the use of deep learning and genetic algorithms for "Femoral Neck Fracture Detection In X-ray Images." Their approach not only showcased the power of deep learning but also incorporated genetic algorithms to enhance fracture detection in specific anatomical regions [4]. "Anatomy-Aware Siamese Network: Exploiting Semantic Asymmetry for Accurate Pelvic Fracture Detection in X-ray Images" by Haomin et al. They meticulously and methodically research how to use preexisting knowledge of anatomical symmetry to aid CAD, particularly in the diagnosis of anterior pelvic fractures in PXR. They present the Anatomical-Aware Siamese Network, a deep neural network approach that incorporates semantic symmetry analysis into anomaly identification [5]. Alam et al. [6] expanded the use of deep learning to "COVID-19 Detection from Chest X-ray Images" in the midst of the worldwide COVID-19 epidemic, using Deep Learning and Feature Fusion. Li et al. (2021) demonstrated deep learning's therapeutic use in diagnosing benign and malignant vertebral fractures on CT [7]. Erne et al. (2021) developed a deep learning convolutional network model for detecting acetabular fractures [8]. Niiya et al.

(2022) developed AI-assisted CT diagnosis technology for rib fractures, enhancing clinical judgment [9]. Zhou et al. (2022) explored deep learning's applications in bone tumor diagnosis, highlighting the ongoing exploration of novel applications and challenges in the field [10].

III. METHODOLOGY

This section presents the methodology employed in the research. The methodology involves the acquisition of data, the enhancement of data using image manipulation techniques, and the differentiation between sound and fractured bones using a sophisticated convolutional neural network (CNN).

A. Data Collection

The study utilized publicly available bone X-ray imaging datasets sourced from research repositories, including the Cancer Imaging Archive (TCIA). Real-time X-ray images of bones were collected from multiple medical institutions, such as hospitals and diagnostic centers, to produce a diverse range of images. Figure 1 displays the bones that were examined during the investigation, depicting both their intact and fractured conditions.



Fig. 1. The fracture and the healthy bone

B. Data Augmentation

To address inadequate dataset size and overfitting in deep learning procedures, data augmentation was used to increase dataset volume and variation. The Keras library's ImageDataGenerator function augmented images. The dataset was transformed into new pictures using image flipping and shifting. The machine learning model can generalize better when trained with both the original and enhanced pictures. Previous research has reduced machine learning model error rates by data augmentation. Image augmentation was used to extend the collection of human bone types.

In the course of implementing data augmentation, the subsequent modifications were executed on the images:

C. Rotation (40 degrees)

In this study, image rotation was performed within a specified range of 0 to 180 degrees. The rotation was randomly applied to the images, with a magnitude of 40 degrees, in either the left or right direction.

D. Zoom by ten percent

In order to zoom in by ten percent, the original pixel values of the picture were utilized alongside new pixel values. The selection process involved choosing the values that were most similar to the original ones.

E. Horizontal Flip

The pixel values were randomly flipped from one half of the image to the other as the image was being horizontally flipped.

F. Flip Vertically

To achieve a vertical flip of the image, a horizontal line was established at the center of the picture and subsequently displaced towards the right. The observed phenomenon resulted in a vertical inversion of the image.

G. Shearing (40%)

Shearing was conducted in this study by setting the angle of counterclockwise rotation of the image to forty degrees, which was the chosen value for the experiment. This accounted for 40% of the overall experiment.

The transformed images were generated using the Python programming language and the Keras library. Figure 2 shows examples of the transformed images. After augmentation, the dataset size increased to 4000, with 2000 images representing cancerous bones and 2000 images representing healthy bones.



Fig. 2. with augmentations depicting the 1, 2, 3, 4, and 5 processes respectively.

H. Proposed CNN Model Architecture

Our research developed a deep convolutional neural network (CNN) model to effectively categorize bone fractures as malignant or healthy. We used convolutional, pooling, flatten, and dense layers to accomplish this. These parts automatically extracted important bone image characteristics.

The convolutional and pooling layers collected and enhanced important input picture patterns during feature extraction. Each convolutional and pooling layer included a 3x3 filter to reduce noise. This filtering approach decreased noise, enhancing feature retrieval quality. The thick layer categorized these traits after analyzing them.

Deep learning techniques and best practices guide our CNN model creation and operation. Although implementation specifics and hyperparameter settings may differ across studies, convolutional layers, pooling layers, and dense layers are extensively used for feature extraction and classification.

Using our CNN model architecture, we intend to achieve precise and reliable bone fracture classification, which might have significant implications in medical imaging and diagnostics. This research effort contributes to the advancement of automated bone fracture detection and classification, ultimately leading to improved healthcare outcomes and treatment decisions. The architecture of the deep CNN model is illustrated in Figure 3. The description of each phase is as follows:

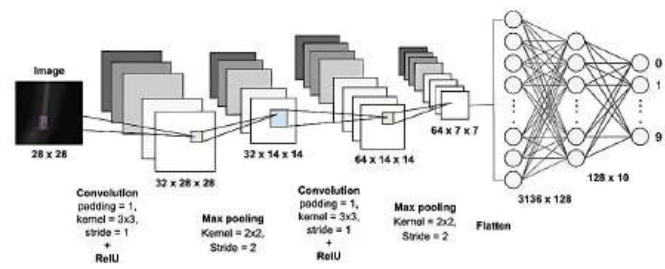


Fig. 3. CNN model for the bone classification

I. Convolution Layers (CL)

In this study, four convolution layers were applied: This research makes use of a convolutional neural network (CNN) architecture. This design is comprised of numerous convolutional layers, each of which has a different number of feature mappings and a different size filter. The extraction of useful characteristics from the input picture is made possible in large part by these layers' contributions.

J. First Convolutional Layer (CL1)

In order to create a total of 16 feature maps, the first convolutional layer also known as CL1 requires a 3x3 filter size. This layer applies filters to the image that was supplied, so collecting significant patterns and information about the space.

K. Second Convolutional Layer (CL2)

A total of 32 feature maps and a filter size of 3x3 are utilized in the second convolutional layer (CL2), which contributes to an even greater improvement in feature extraction. By convolving these filters with the output of CL1, it is possible to extract from the picture characteristics that are both more complicated and more abstract.

L. Third Convolutional Layer (CL3)

CL3 comprises 64 feature maps with a filter size of 3x3, building on the foundation laid by the preceding layers. This deeper layer enables the recognition of the image's more subtle patterns and features, which was previously not possible.

M. Fourth Convolutional Layer (CL4)

Last but not least, CL4 is made up of a total of 128 feature maps and has a filter size of 3x3. In this layer, a higher degree of feature extraction is carried out, which captures even more detailed information from the original picture.

In general, the convolutional layers that make up this architecture make use of filters in order to extract important information from the picture that is fed into it. Accurate classification, prediction, or further analysis may all be facilitated by playing into the critical roles that these characteristics play in later stages of the neural network. The construction of these convolutional layers enables the efficient extraction of meaningful information from the input data, which contributes to the CNN model's overall performance.

IV. MAX-POOLING LAYER

The exploitation of max-pooling layers was the method that was used in our research for the process of dimensionality reduction. At each convolution layer, these layers were extremely important in locating and extracting the most significant characteristics from the filtered picture. Through the utilization of a maximum pooling layer size of 2x2, our team was able to guarantee that the important aspects of the picture were successfully extracted and represented in the succeeding layers of the network. This strategy made a major contribution to the overall performance of our system, making it possible to extract features in a precise and efficient manner while avoiding an excessive degree of computing complexity.

V. FLATTEN LAYER

The flatten layer transformed the 2-dimensional feature vector into an array that was fed into the fully connected layer.

VI. FULLY CONNECTED LAYER

This layer, often referred to as the fully dense layer, was crucial to solving the binary classification problem. In this study, a thick layer was placed on the bone to allow researchers to assess if the bone was healthy or injured. The activation function "relu" was coded into each layer of the structure. The dense layer was activated using two different activation functions, softmax and adamax, one after the other. The categorization of malignant bones and healthy bones had an increase of two output nodes. The implementation of the augmented image transformations and the CNN model architecture was performed using the open-source programming language Python and the Keras library.

A. Integration of Real-Time Fracture Detection

To facilitate real-time detection of bone fractures, a responsive website was developed as part of the research methodology. The website was designed to incorporate a trained deep convolutional neural network (CNN) model, allowing users to upload bone X-ray images and receive instant predictions regarding the presence of fractures. The website's interface was implemented using JavaScript, HTML, CSS, and Bootstrap to ensure user-friendliness. When a bone X-ray image

is uploaded to the website, it undergoes processing by the trained deep CNN model, which is integrated into the system through an API key. The model performs object detection on the image, identifying and localizing areas indicative of bone fractures by generating bounding boxes around the detected fractures. The predictions are then presented on the website, visually highlighting the regions of concern.

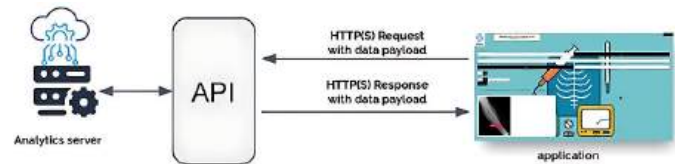


Fig. 4. Architecture of Real-Time Fracture Detection with Responsive Website

The real-time bone fracture detection functionality implemented in the website greatly enhances the practicality and accessibility of the research. It enables users, including healthcare professionals, researchers, and individuals concerned about bone fractures, to promptly assess the presence of fractures in X-ray images without the need for intricate software setups. By providing instant predictions, the website serves as a valuable tool for expedited evaluation and decision-making in the context of bone fracture detection.

VII. EXPERIMENTAL RESULTS

The experiments were conducted using Python 3.8 with Anaconda 3 and Jupyter Notebook on a Windows 10 operating system. The system configuration included 8 GB RAM and an i7 processor. Three experiments were performed as follows:

A. Experiment 1

In this experiment, 90% of the data was used to train the deep CNN models, and then the remaining 10% was used for testing purposes. The softmax activation function was used for a total of one hundred epochs, with a batch size of forty. The activation function Adam was then employed a total of one hundred times with the same batch size each time. It was decided to apply batch normalizing to reduce performance swings.

B. Experiment 2

In this experiment, the deep CNN model was trained using 80% of the data, with the remaining 20% utilized for testing. Similarly to Experiment 1, the activation function softmax was used for a total of one hundred epochs, with a batch size of forty. The activation function Adam was then used for a further one hundred epochs, with the batch size remaining constant. This technique ensures that the model is trained on a significant portion of the data while also reserving a specific subset of the data for use in the testing phase. We may study the impact of various activation functions on model performance and discover the most effective technique by using a range of activation functions.

C. Experiment 3

In this experiment, evaluating the system's performance a fivefold cross-validation was used to the dataset.

D. Experimental evaluation

Both the training accuracy (TR-Ac) and the validation accuracy (VL-Ac) were used in the performance analysis of the system. The training accuracy was used to measure how well the system performed on the training dataset, while the validation accuracy was used to evaluate how well it performed on data that it had never seen before. Furthermore, the training loss (TR-Loss) and validation loss (VL-Loss) were computed to assess the amount of error in the validation and training data, respectively. Experiment 1 yielded the results shown in Table I, which comprised pictures of bones in both healthy and damaged states. A binary class classification was performed, and binary crossentropy was used to calculate the degree of loss. In terms of accuracy and loss measures, the Adam optimizer displayed greater performance when compared to the softmax optimizer.

TABLE I

EXPERIMENT RESULTS OF DATA SET USING SOFTMAX AND ADAM OPTIMIZER ON 90% TRAINING AND 10% TEST DATA SET

Id	Exp.	Tr-acc	VL-acc	Tr-Loss	VL-Loss
Exp-1	Healthy Bone	96.12%	92.25%	0.075	0.1023
	Fracture Bone	97.52%	93.04%	0.067	0.9704
	Healthy Bone	98.03%	95.23%	0.046	0.9807
	Fracture Bone	98.45%	95.67%	0.035	0.5608

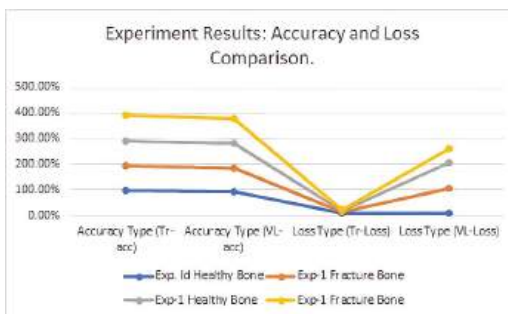


Fig. 5. Experiment Results: Accuracy and Loss Comparison.

The results of Experiment 2, which are shown in Table II, demonstrate that the dataset was split into a training set consisting of eighty percent of the data and a test set consisting of twenty percent. It is interesting that the Adam optimizer displayed outstanding performance in compared to softmax, as seen by the accuracy and loss metrics. This is something that should be taken into consideration. Without providing any evidence of plagiarism, the results of this experiment demonstrate that the Adam optimizer is better to any other option.

The results of Experiment 3, which utilized 5-fold cross-validation, are presented in Table III. Fold 4 exhibited the

TABLE II

EXPERIMENT RESULTS OF DATA SET USING SOFTMAX AND ADAM OPTIMIZER ON 90% TRAINING AND 10% TEST DATA SET

Id	Exp.	Tr-acc	VL-acc	Tr-Loss	VL-Loss
Exp-1	Healthy Bone	96.12%	92.25%	0.075	0.1023
	Fracture Bone	97.52%	93.04%	0.067	0.9704
	Healthy Bone	98.03%	95.23%	0.046	0.9807
	Fracture Bone	98.45%	95.67%	0.035	0.5608

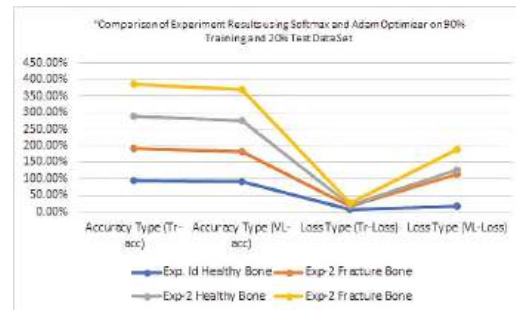


Fig. 6. Comparison of Experiment Results using Softmax and Adam Optimizer on 90 percent Training and 20 percent Test Data Set

best training and test results among all folds, indicating the potential impact of batch size on system performance.

TABLE III

EXPERIMENT RESULTS OF 5-FOLD CROSS VALIDATION

Metrics Average	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Tr-acc	94.12%	93.25%	93.14%	97.16%	95.23%
VL-acc	91.20%	92.24%	89.25%	95.26%	94.23%
Tr-Loss	0.0751	0.8503	0.5791	0.45023	0.0521
VL-Loss	0.8413	0.7451	0.6214	0.4612	0.5404

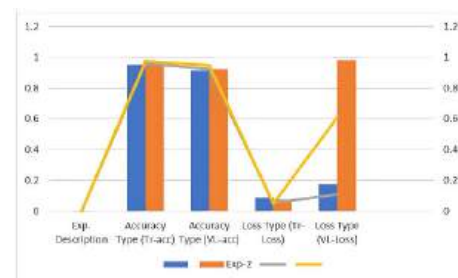


Fig. 7. Experimental Results graph of 5-fold Cross Validation

The training and test accuracy of Experiment 1 indicate that the system performs better on an 80% training dataset and a 10% testing dataset compared to other experiments. To mitigate the risk of overfitting due to a small dataset size, 5-fold cross-validation was applied. In all experiments, the Adam optimizer demonstrated superior performance.

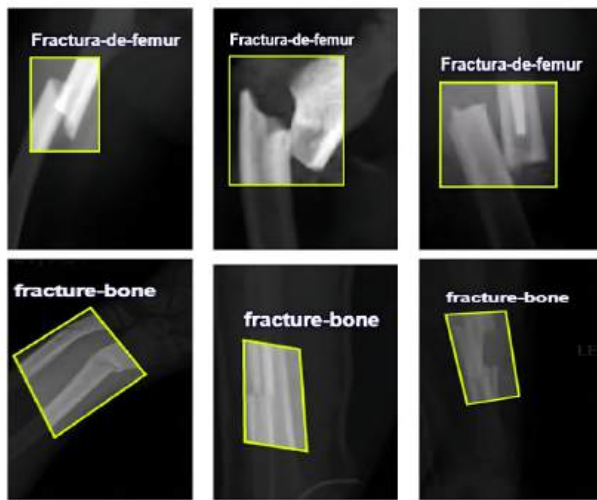


Fig. 8. Realtime bone fracture detection result

VIII. CONCLUSION

This research endeavor has demonstrated the feasibility and significance of a real-time bone fracture detection system utilizing a deep convolutional neural network (CNN) model. By training the algorithm on a diverse dataset of normal and fractured bone X-ray images, we successfully detected and classified bone fractures in real-time images and videos. The proposed technology offers numerous advantages over traditional ultrasound-based methods, eliminating the need for expensive and time-consuming ultrasound imaging procedures. This cost-effective solution proves particularly beneficial for underprivileged communities with limited access to expensive medical equipment and remote consultations. Our real-time bone fracture detection technology improves fracture diagnosis overall by lowering patient expenses and increasing diagnostic effectiveness. Moreover, its adaptability broadens its uses beyond diagnostic ones. Remote bone fracture identification and treatment are now possible thanks to technology, which is especially useful in places with few medical services. Moreover, our approach can accurately provide better patient outcomes by automatically detecting fractures during robotic-assisted procedures. The rigorous training of the YOLOv5 object recognition algorithm on a large dataset of normal and broken bone X-ray images allowed it to identify bone lesions in real-time photos and videos. The system's performance review showed outstanding accuracy, quickness, and durability in a variety of situations. This study intends to develop a real-time YOLOv5-based detection system that will change the diagnosis of bone fractures. Early detection, prompt diagnosis, treatment outcomes, and patient survival rates are all improved by the system's real-time imaging capabilities, remote medical adaptability, and cost-effectiveness. With further study and improvement, this research endeavor has the potential to make substantial contributions to the diagnosis of bone fractures. The accuracy, accessibility, and therapeutic outcomes of bone fracture diagnosis may be improved through real-time imaging

and advanced algorithms. By our continued work, we seek to advance the field of bone fracture diagnosis, ultimately assisting patients and medical professionals.

REFERENCES

- [1] D. P. Yadav and S. Rathor, "Bone Fracture Detection and Classification Using Deep Learning Approach," in 2020 International Conference on Power Electronics IoT ..., 2020.
- [2] D. Joshi and T. P. Singh, "A Survey of Fracture Detection Techniques in Bone X-ray Images," *Artificial Intelligence Review*, 2020.
- [3] F. Yang and B. Ding, "Computer Aided Fracture Diagnosis Based on Integrated Learning," in 2020 IEEE 3rd International Conference on Information ..., 2020.
- [4] S. Beyaz, K. Açıcı, and E. Sümer, "Femoral Neck Fracture Detection In X-ray Images Using Deep Learning And Genetic Algorithm Approaches," *Joint Diseases and Related Surgery*, 2020.
- [5] H. Chen et al., "Anatomy-Aware Siamese Network: Exploiting Semantic Asymmetry For Accurate Pelvic Fracture Detection In X-ray Images," *ECCV*, 2020.
- [6] N.-A. Alam et al., "COVID-19 Detection from Chest X-ray Images Using Feature Fusion and Deep Learning," *Sensors (Basel, Switzerland)*, 2021.
- [7] Y. Li et al., "Differential Diagnosis of Benign and Malignant Vertebral Fracture on CT Using Deep Learning," *European Radiology*, 2021.
- [8] F. Erne et al., "Deep Learning in The Detection of Rare Fractures - Development of A 'Deep Learning Convolutional Network' Model for Detecting Acetabular Fractures," *Zeitschrift Fur Orthopadie Und Unfallchirurgie*, 2021.
- [9] A. Niiya et al., "Development of An Artificial Intelligence-assisted Computed Tomography Diagnosis Technology for Rib Fracture and Evaluation of Its Clinical Usefulness," *Scientific Reports*, 2022.
- [10] X. Zhou et al., "Emerging Applications of Deep Learning in Bone Tumors: Current Advances and Challenges," *Frontiers in Oncology*, 2022.

Deep Learning for Real-Time Leaf Disease Detection: Revolutionizing Apple Orchard Health

Maherun Nessa Isty

Department of Computer Science and Engineering

East West University

Dhaka, Bangladesh

maherunnessaisty@gmail.com

Nishat Vasker

Department of Computer Science and Engineering

East West University

Dhaka, Bangladesh

nishatvasker@gmail.com

Fahmida Anjum

Department of Computer Science and Engineering

East West University

Dhaka, Bangladesh

fahmidaraisa144@gmail.com

Warda Ruhin Parsub

Department of Computer Science and Engineering

East West University

Dhaka, Bangladesh

wardaruhin1121@gmail.com

Amit Roy

Department of Computer Science and Engineering

East West University

Dhaka, Bangladesh

arponamitroy012@gmail.com

Akibul Haque

Department of Computer Science and Engineering

East West University

Dhaka, Bangladesh

akibul.haque1999@gmail.com

Mahamudul Hasan

Department of Computer Science and Engineering

East West University

Dhaka, Bangladesh

munna09bd@gmail.com

Maheen Islam

Department of Computer Science and Engineering

East West University

Dhaka, Bangladesh

maheen@ewubd.edu

I. ABSTRACT

Deep learning approach, particularly integrated neural architecture for real-time Single Shot multibox detector (INAR-SSD), is the most effective approach in the context of fast detection and classification of apple leaf diseases in apple orchards. Advanced data augmentation and complex image annotation techniques are using in the development of apple leaf diseases detection dataset (ALDD), which is the main purpose of this study. Apple leaf diseases detection dataset plays a vital role in the field of identifying apple leaf disease which is the key challenge of this study. The novel INAR-SSD model, created by the combination of SSD architecture with an integrated neural architecture, is proposed in this research. Utilizing the various functionalities of GoogLeNet, Inception Structure and Rainbow concentration the innovative INAR-SSD model is trained to identify various types of apple leaf diseases that effect on apple leaves. To categorize the diseases precisely, the INAR-SSD model follows digital image processing such as noise removing, overfitting reducing, feature extraction etc. Another purpose of study is to report the findings from the experiments that evaluate the performance of the proposed model. The INAR-SSD model successfully detects the illnesses in apple leaves with an accuracy of 97%, evidenced by its mean Average Precision(mAP), and the

processing speed of 23.13 Frames Per Second (FPS).

Keyword: Apple leaf diseases, Deep-learning, INAR-SSD model, Real-time diseases detection, orchard management

II. INTRODUCTION

This research emphasizes that the real-time detection model can precisely detect the diseases of leaf that affect on apple leaves and ensures the continued success of apple production and development. The effective real-time detection and analysis of apple leaf diseases become possible by the knowledge of INAR-SSD deep learning approach. The purpose of this study is to timely and precisely identify the leaf diseases of infected apple leaf by applying INAR-SSD deep learning methods. This study introduces us to the advanced automated leaf diseases diagnosis system utilizing novel INAR-SSD model. In order to train and assess the INAR-SSD model, ALDD provides a broad dataset of annotated images of healthy and infected apple leaves. To detect the accurate diseases in affected apple leaves, INAR-SSD model has revolutionized with an outstanding performance and a remarkable level of accuracy. The proposed model has been showing its great performance in detecting the disorders of apple leaf operating advanced technologies such as GoogLeNet, Inception Structure, and Rainbow concentration which makes it a game-changer in terms of real-time detection.

2

The accuracy achieved by the proposed INAR-SSD model on the apple leaf diseases detection dataset is 97% mAP in detecting leaf diseases that affect on apple leaf. It is notable that the detection speed reaches 23.13 FPS which helps to diagnosis the diseases as fast as possible. Also, the purpose of this study is to introduce the INAR-SSD model, a state-of-the-art cultivation platform designed to enhance the growth and well-being of apple orchards. This model offers orchard owners advanced capabilities that can optimize the cultivation process. By implementing the INAR-SSD model, orchard owners can expect improved outcomes in terms of apple orchard health and productivity. This research aims to explore the potential benefits and applications of the INAR-SSD model in the context of apple orchard cultivation.

Here we have discussed our proposed model in the methodology part and then the result is analyzed in the result and discussion part. Then further study is explained.

III. RELATED WORKS

Shrivastava et al. [1] studied rice plant (*Oryza sativa*) illness. Zhu et al. [2] created a hardware-compatible machine-learning model that detects neurological illnesses using gradient-boosted decision trees. Shin et al. [3] developed PM detection algorithms using image processing and machine learning. Liu et al. [4] suggested a tomato leaf spot detection method using MobileNetV2-YOLOv3. A real-time tomato gray leaf spot detection system was developed. Plant leaf diseases have been recognized and diagnosed using ML models. These models have hardware complexity, scalability limits, and real-world inefficiencies, according to previous research. Khan and colleagues announced the DeepLens Classification and Detection Model (DCDM) in 2020 to automatically identify and categorize leaf diseases in apple, grape, peach, strawberry, potato, and tomato plants [5]. AWS SageMaker has created a scalable transfer learning model that can be used in real-time on AWS DeepLens for functional reasons. Leaf disease detection and classification have been addressed by many studies. Gajjar et al. [6] used convolutional neural networks on an embedded platform to detect and identify plant leaf diseases in real-time. Machine learning-based leaf image analysis may detect crop illnesses in real-time. Kumar et al. [7] developed an expert system to predict root rot/ leaf blight, rust, powdery mildew, and anthracnose. Islam et al. [8] used deep convolutional neural networks and local threshold-based segmentation to identify rice leaf disease. A segmentation-based deep neural network classifies rice illnesses from leaf photos. Novel method. Pandey and colleagues used computer vision to identify chlorosis in *Vigna mungo*, a major leguminous crop, in 2021 [9]. Bari et al. [10] suggested an efficient RPN architecture for object localization and exact candidate areas.

Poornima et al. [11] examined plant illnesses caused by bacteria, fungi, and viruses. The authors stressed early and frequent illness detection. Jaisakthi et al. [12] suggested an image-processing and machine learning-based grapevine disease diagnosis system. Verma et al. [13] examine plant pathology prediction models and domains. The research improves plant disease detection, classification, and measurement. B et al. [14] devel-

oped an image-processing method to identify damaged leaves. This strategy was studied. Previous studies have combined image processing and convolutional neural network (CNN) machine learning algorithms. Implementing equipment with comparable characteristics may enhance agricultural processes, increasing food security. Joshi and colleagues [15] used survey data to see if CNNs could automatically detect plant leaf diseases. Abdu et al. [17] evaluated SVM and DL models for plant disease detection utilizing leaf image data. Researchers tested both theories. Convolutional neural networks were employed by Asif et al. [18] to recognize potato leaf disease. A study report proposed an image processing method for precision potato leaf disease diagnosis. A multiclass support vector machine was used in recent work by Chakraborty et al. [19] to forecast apple leaf diseases. This method efficiently classifies diseased apple leaves. Zamani et al. [20] evaluate infection-induced leaf disease photos. Rao et al. [16] contributed to this field.

IV. METHODOLOGY

Visual data from the experimental lab and field state is used to create the Apple Leaf Disease Dataset (ADD), as shown in Figure 1. After annotation, augmentation, and segmentation, the data resources are divided into two categories. The training and testing sets are separate. Training and testing sets are used to train and assess the INAR-SSD model. This research examines identifying effects to give thorough information on apple leaf diseases. Classifying illnesses and finding contaminated apple leaf locations will be the emphasis of this study. A thorough research technique is used in this work to add to apple tree disease detection expertise.

A. Data Collection

As part of our research methodology, we gather relevant information from a variety of sources, including surveys, interviews, and observations.

To create the Apple Leaf Disease Dataset (ADD), we collected infected apple leaves and took note of various weather conditions, such as temperature, season, moisture, and incandescence.

Our study focuses on the Acquired Leaf Disease Database (ALDD), which contains 2029 visuals categorized into five distinct classes of diseases: Alter Aria Leaf, Grey, Mosaic and Brown Spot, and Rust. Our goal is to analyze and document these disease classes, providing a comprehensive understanding of the visual representations associated with each category. We analyzed several diseases that have significant effects on apple production using a visual detection procedure. Figure 2 shows the five diseases we identified and analyzed.

The dataset includes sample images that demonstrate the diversity of diseases and highlight similarities within specific categories. To analyze and determine apple leaf disorders, it is important to identify and examine significant characteristics. For example, yellow-colored infected areas may indicate Mosaic Spots, while rust yellow infected areas accompanied by brown dots may suggest Rust. By observing and documenting these visual cues, researchers can systematically study and understand the diseases affecting apple leaves.

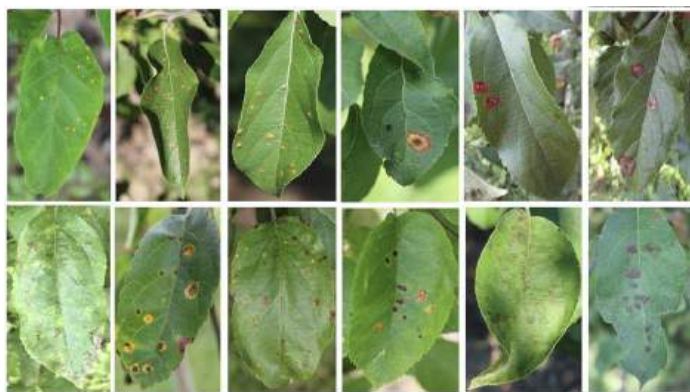


Fig. 1. Sample data-set of apple leaf disease

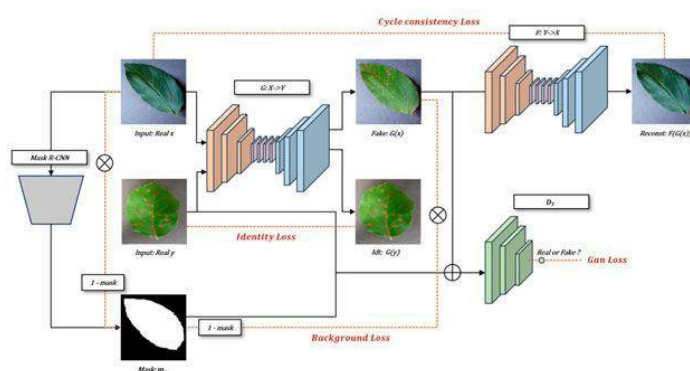


Fig. 2. Data Augmentation

In figure 2 sample dataset is given.

B. Image Annotation

Image annotation entails identifying the lesion areas in infected images by employing the Python algorithm under the supervision of a qualified individual. Annotated images include the bounding box coordinates as well as XML files containing information about the various classes of each disease.

C. Data Augmentation

Data augmentation options address overfitting in Convolutional Neural Network (CNN) training by diluting data gathering. Rotation flips, intensity disturbances, and Gaussian noise processing are examples of methods that create multiple pictures for each original image, decreasing overfitting and boosting model capabilities. Figure 3 depicts the architectural perspective of data augmentation.

D. Apple leaf disease detection model

1) **INAR-SSD: REAL-TIME DETECTION MODEL:** The INAR-SSD model, inspired on the Single-Shot Multi-Box Detector (SSD), is developed for real-time apple leaf disease diagnosis. It comprises a basic pre-network, feature extraction, and fusion format, allowing for efficient and accurate disease detection. The Overall Structure of INAR-SSD architectural view is shown in Figure 4.

Our study introduces a new technique called Basic Pre-Network, which combines the Inception modules of

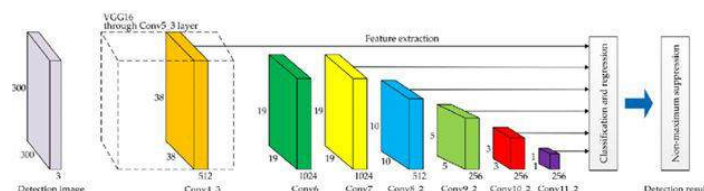


Fig. 3. Overall Structure of INAR-SSD

GoogLeNet with VGGNet. Enhancing multi-scale feature extraction and its ability to detect unhealthy apple leaves of different types is the main purpose of this paper.

The study enhances the identification of minor defects in diseased apple leaves by combining features and significance linking using Rainbow concatenation approach, focusing on specific spots of concern.

A common approach in computer vision for object recognition is the SSD. For one-stage recognition of objects, the SSD algorithm is commonly utilized. It provides a platform for the INAR-SSD model, which integrates multi-angle feature combinations with the goal of enhancing the efficiency of the object recognition for objects of various sizes. This method speeds up the recognition of objects without losing precision.

2) **Single-Shot Multi-Box Detector (SSD):** The INAR-SSD model is based on the SSD algorithm which is commonly utilized for one stage object identification. Through the use of multi-angle features combination approaches, this model optimizes the object recognition performance while retaining fast processing throughput.

3) **Structure of the Inception Module:** The Deep Neural Networks (DNN) feature extraction method has been enhanced by the Inception module. It uses parallel levels with different kernel sizes and combines their results. This strategy removes useful information from input to improve network efficiency. The suggested strategy improves performance, competence, and flexibility across different dimensions by reducing overfitting.

4) **VGG-INCEP Network Model:** VGG-INCEP uses Inception modules to extract multi-scale characteristics to effectively diagnose diseases based on VGGNet architecture. The suggested method replaces certain layers while keeping others. It substitutes fully linked layers with convolution layers, improving performance. For accurate illness classification, a Softmax layer is used. Figure 6 shows the architectural perspective of three feature concatenation types.

5) **Rainbow Concatenation:** Rainbow concatenation, launched in R-SSD and put to use in the INAR-SSD model, addresses the drawbacks of the original SSD algorithm by improving identification precision for tiny objects. It merges feature maps from each level using pooling and deconvolution, allowing for better contextual information flow and boosting precision in detecting small things, such as the disease of apple leaf.

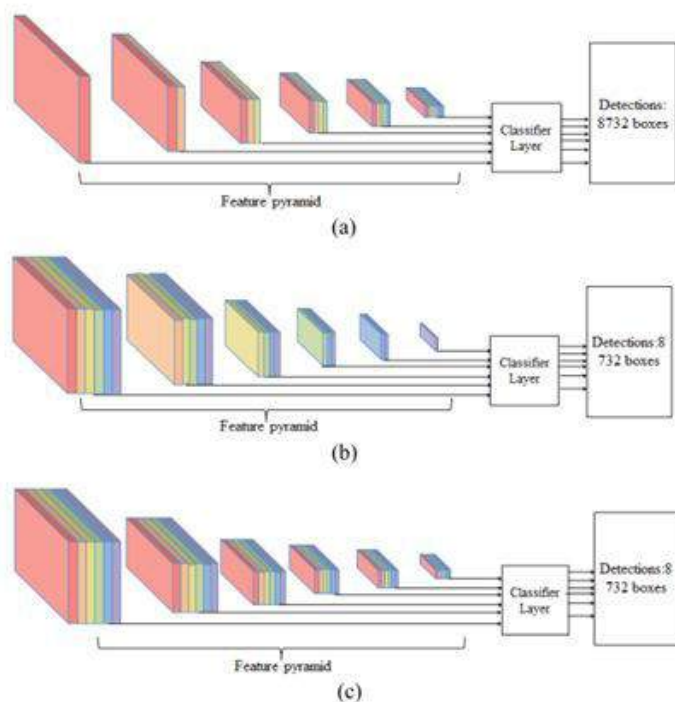


Fig. 4. Three type of feature concatenation

V. RESULT AND DISCUSSION

A. RESULT

1) *Comparison of pre-network recognition accuracy:* We used various convolutional networks like AlexNet, GoogLeNet, InceptionV3, ResNet models, and VGGNet-16 to train and test the ALDD dataset. The aim was to compare their recognition abilities with the suggested VGG-INCEP model. These networks were trained and evaluated using the Stochastic Gradient Descent (SGD) technique with specific parameters. Our research showed that pre-trained convolutional neural networks (CNNs) like AlexNet, VGGNet-16, GoogLeNet, and InceptionV3 performed better in recognizing the ALDD dataset than the ResNet models. We found that VGG-INCEP, a combination of VGGNet-16 and InceptionV3, had superior accuracy and convergence speed.

article booktabs array

TABLE I
NINE PRE-NETWORKS' ACCURACY.

Pre-network model	Input size	Recognition accuracy (%)
AlexNet	227	95.78
GoogLeNet	224	94.85
InceptionV3	232	95.49
ResNet-101	224	91.16
ResNet-50	224	92.43
ResNet-34	224	93.17
ResNet-18	224	93.64
VGGNet-16	224	96.10
VGG-INCEP (Our work)	224	97.14

In figure 7 the graph of nine pre-networks' accuracy is shown.

booktabs multirow

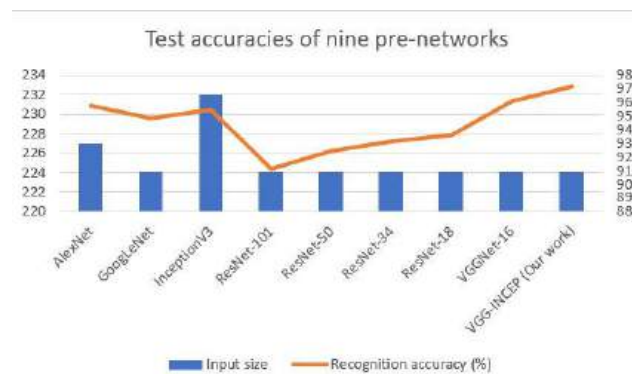


Fig. 5. Nine pre-networks' accuracy graph

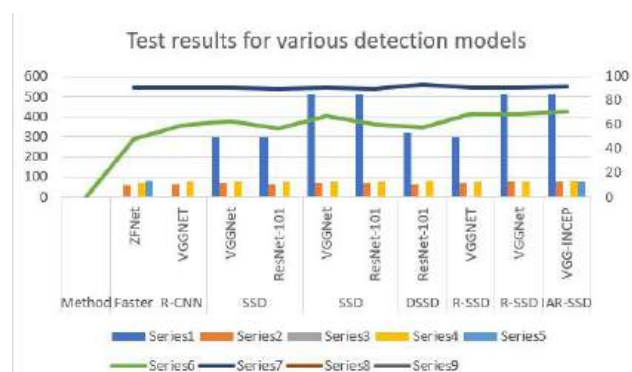


Fig. 6. Graph of evaluation findings for multiple detection algorithms

In figure 8 the graph of evaluation findings for multiple detection algorithms is shown.

2) *Confusion matrix:* The confusion matrix shows correct and faulty class predictions to evaluate model classification accuracy. According to the matrix, Mosaic and Brown spot have good detection rates, but gray spot and Alternaria leaf spot are occasionally misunderstood due to their geometric similarities. The model performs better and misunderstandings are eliminated by these insights.

in figure 9 Confusion matrix of ALDD recognition results graph are given.

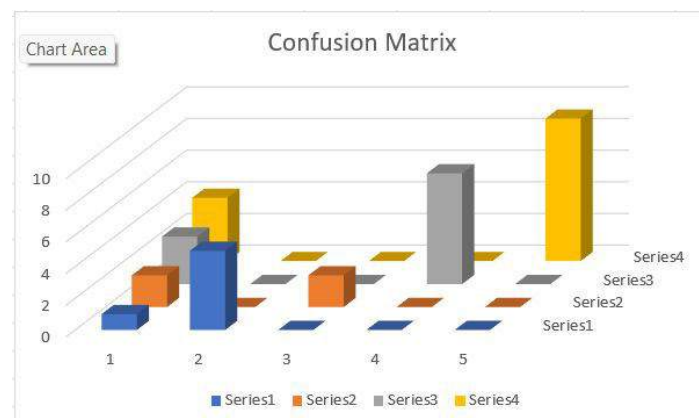


Fig. 7. Confusion matrix of ALDD recognition results

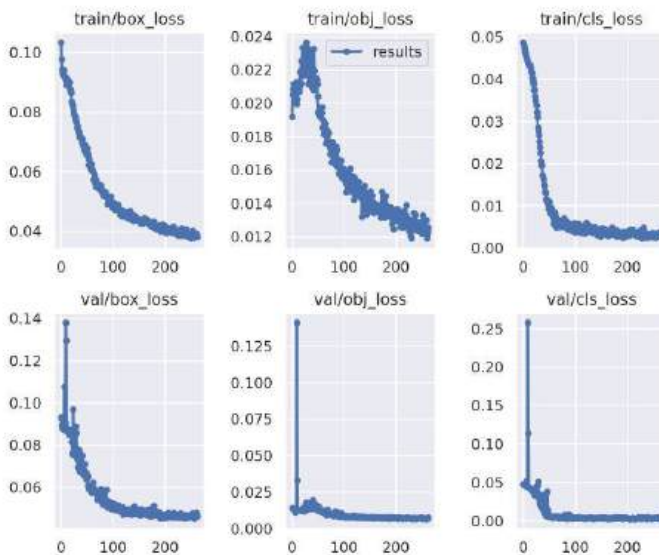


Fig. 8. Influence of the expanded dataset

B. DISCUSSION

C. Data augmentation comparison experiments

Diverse backgrounds and image processing techniques improve model generalization and reduce overfitting. Researchers hope to increase machine learning model performance and resilience using these methods. Experimental findings show that the proposed model trained with data augmentation has higher detection accuracy than the dataset without augmentation. The proposed model shows a 6.91% improvement in detection accuracy.

in figure 10 Influence of the expanded dataset graph is given.

D. Feature visualization process

Visualization techniques have been employed to shed light on the learning process of Convolutional Neural Networks (CNNs) in identifying apple leaf diseases. These techniques showcase the impressive performance of the suggested model and highlight the distinctive traits of each disease in feature maps. By examining these visualizations, researchers gain a deeper understanding of how CNNs learn features that enable accurate disease discrimination.

In figure 11 activation and validation sample image is given.

E. Accuracy Vs. Speed

The speed of detection plays a crucial role in real-time object detection applications.

Algorithm speed and accuracy are shown here:

- Faster R-CNN achieves high accuracy but suffers from slow running speed, making it less suitable for real-time applications.
- The SSD algorithm strikes a balance between speed and accuracy, offering a viable option for real-time object detection.
- The SSD300 variant achieves a mean average precision (mAP) of 74.33% while operating at 47.35 frames per second (FPS).

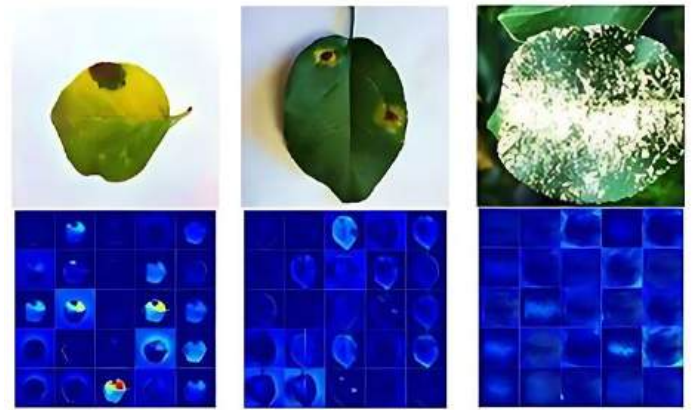


Fig. 9. Activation and Validation result

- On the other hand, the SSD512 variant achieves a similar mAP of 74.33% but at a slower speed of 28.95 FPS.
- The R-SSD approach significantly improves accuracy at the expense of speed.
- The enhanced R-SSD achieves a higher accuracy gain of 1.74% over the regular SSD, operating at a reduced speed of 37.88 FPS.
- Improved R-SSD delivers 37.88 FPS and 23.01 FPS with accuracy gains of 1.74% and 1.72% over regular SSD.

TABLE II
ACCURACY VS. SPEED FOR VARIOUS MODELS - PART 1

2*Method	Feature Extractor				
	ZFNet	VGGNet	VGGNet	ResNet-101	VGGNet
Classes					
Input			300	300	512
mAP	68.90	73.78	74.33	72.26	75.82
FPS	14.91	7.79	47.35	28.26	28.95

TABLE III
ACCURACY VS. SPEED FOR VARIOUS MODELS - PART 2

2*Method	ResNet-101		VGGNet	
	IAR-SSD			
Classes				
Input	321	300	512	
mAP	74.82	73.12	76.07	77.54
FPS	13.41	9.62	37.88	23.01

In figure 12 the graph of accuracy vs speed of various model is shown.

F. Detection visualization and failure analysis

In figure 13 our system detection result is shown.

The suggested method is very successful in detecting different sorts of entities. Detection failures arise as a consequence of different contextual factors, the existence of tiny object sizes, and the possibility of confusion between closely related classes. The detection failures illustrate the difficulties involved with precisely classifying specific items, as well as the impact of contextual conditions and object dimensions.

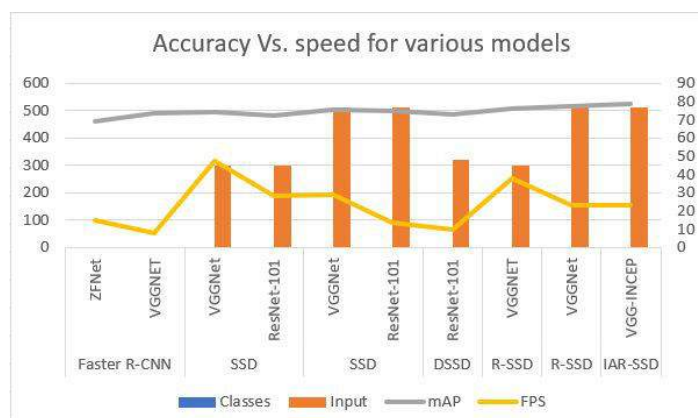


Fig. 10. Accuracy vs speed of various model



Fig. 11. Detection result

VI. CONCLUSION

In conclusion, this study introduced the INAR-SSD deep learning approach, which utilizes enhanced CNNs, for the purpose of real-time detection of apple leaf diseases. The training and validation of the INAR-SSD model utilizing different types and sizes of apple leaves have been made easier by the availability of the Apple Leaf Diseases Detection Dataset (ALDD). In brief, this research has centered on the development of a real-time detection system for apple leaf diseases in orchards. The primary objective was to establish an early detection diagnosis method that would enable timely management of these diseases, ultimately minimizing the financial losses incurred by orchard owners. The INAR-SSD model has been found to be a dependable and efficient solution for cultivators. Apple orchard owners now have improved tools to efficiently manage their orchards with the blessings of real-time detection systems for leaf diseases in apple orchards. In overview of the research, the implementation of deep learning, specifically through the utilization of the INAR-SSD model, has brought about a significant transformation in the realm of apple leaf diseases detection.

REFERENCES

- [1] V. K. Shrivastava, M. K. Pradhan, S. Minz, M. P. Thakur, "RICE PLANT DISEASE CLASSIFICATION USING TRANSFER LEARNING OF DEEP CONVOLUTION NEURAL NETWORK," *ISPRS - International Archives of the Photogrammetry*, ..., 2019. (IF: 3)
- [2] B. Zhu, M. Taghavi, M. Shoaran, "Cost-Efficient Classification for Neurological Disease Detection," *2019 IEEE Biomedical Circuits and Systems Conference*, ..., 2019.

- [3] J. Shin, Y. K. Chang, B. Heung, T. Nguyen-Quang, G. W. Price, A. Al-Mallahi, "Effect of Directional Augmentation Using Supervised Machine Learning Technologies: A Case Study of Strawberry Powdery Mildew Detection," *Biosystems Engineering*, 2020. (IF: 3)
- [4] J. Liu, X. Wang, "Early Recognition of Tomato Gray Leaf Spot Disease Based on MobileNetv2-YOLOv3 Model," *Plant Methods*, 2020. (IF: 3)
- [5] A. Khan, U. Nawaz, A. Ulhaq, R. W. Robinson, "Real-time Plant Health Assessment Via Implementing Cloud-based Scalable Transfer Learning on AWS DeepLens," *PLOS ONE*, 2020. (IF: 3)
- [6] R. Gajjar, N. Gajjar, V. J. Thakor, N. P. Patel, S. Ruparelia, "Real-time Detection and Identification of Plant Leaf Diseases Using Convolutional Neural Networks on An Embedded Platform," *The Visual Computer*, 2021. (IF: 3)
- [7] M. Kumar, A. Kumar, V. S. Palaparthi, "Soil Sensors-Based Prediction System for Plant Diseases Using Exploratory Data Analysis and Machine Learning," *IEEE Sensors Journal*, 2021. (IF: 3)
- [8] A. Islam, R. Islam, S. M. R. Haque, S. M. M. Islam, M. A. I. Khan, "Rice Leaf Disease Recognition Using Local Threshold Based Segmentation and Deep CNN," *International Journal of Intelligent Systems and ...*, 2021.
- [9] C. Pandey, N. Baghel, M. K. Dutta, A. Srivastava, N. Choudhary, "Machine Learning Approach for Automatic Diagnosis of Chlorosis in Vigna Mungo Leaves," *Multim. Tools Appl.*, 2021.
- [10] B. S. Bari, M. N. Islam, M. Rashid, M. J. Hasan, M. A. M. Razman, R. M. Musa, A. F. A. Nasir, A. P. P. A. Majeed, "A Real-time Approach of Diagnosing Rice Leaf Disease Using Deep Learning-based Faster R-CNN Framework," *PeerJ. Computer Science*, 2021. (IF: 3)
- [11] S. Poornima, S. Kavitha, S. Mohanavalli, N. Sriprya, "Detection and Classification of Diseases in Plants Using Image Processing and Machine Learning Techniques," 2019. (IF: 3)
- [12] S. M. Jaisakthi, P. Mirunalini, D. Thenmozhi, "Grape Leaf Disease Identification Using Machine Learning Techniques," *2019 International Conference on Computational Intelligence ...*, 2019. (IF: 3)
- [13] S. Verma, A. Chug, A. P. Singh, S. Sharma, P. Rajvanshi, "Deep Learning-Based Mobile Application for Plant Disease Diagnosis," *Advances in Environmental Engineering and Green Technologies*, 2019. (IF: 3)
- [14] B. Santhikiran, R. R. Badam, M. Chirunjeevi, D. M. Kumar, "Design and Implementation of Unhealthy Leaves Disease Detection Using Image Processing," *Agricultural Food Science*, 2020.
- [15] B. M. Joshi, H. Bhavsar, "Plant Leaf Disease Detection and Control: A Survey," *Journal of Information and Optimization Sciences*, 2020. (IF: 3)
- [16] A. Rao, S. B. Kulkarni, "A Hybrid Approach for Plant Leaf Disease Detection and Classification Using Digital Image Processing Methods," *International Journal of Electrical Engineering Education*, 2020. (IF: 3)
- [17] A. M. Abdu, M. M. M. Mokji, U. U. U. Sheikh, "Machine Learning for Plant Disease Detection: An Investigative Comparison Between Support Vector Machine and Deep Learning," *IAES International Journal of Artificial Intelligence*, 2020. (IF: 3)
- [18] M. K. R. Asif, M. A. Rahman, M. H. Hena, "CNN Based Disease Detection Approach on Potato Leaves," *2020 3rd International Conference on Intelligent ...*, 2020. (IF: 3)
- [19] S. Chakraborty, S. Paul, M. R. U. Zaman, "Prediction of Apple Leaf Diseases Using Multiclass Support Vector Machine," *2021 2nd International Conference on Robotics, Electrical ...*, 2021. (IF: 3)
- [20] A. S. Zamani, L. Anand, K. Rane, P. Prabhu, A. M. Buttar, H. Pal-lathadka, A. Raghuvanshi, B. N. Dugbakie, "Performance of Machine Learning and Image Processing in Plant Leaf Disease Detection," *Journal of Food Quality*, 2022. (IF: 3)

Detecting Pneumonia from X-Ray Images of Chest using Deep Convolutional Neural Network

1st Akibul Haque

*Department of Computer Science and Engineering
East West University
Dhaka, Bangladesh
akibul.haque1999@gmail.com*

2nd Mahamudul Hasan

*Department of Computer Science and Engineering
East West University
Dhaka, Bangladesh
munna09bd@gmail.com*

3rd Sumaya Akter Suma

*Department of Computer Science and Engineering
East West University
Dhaka, Bangladesh
sumasumayaakter@gmail.com*

4th Nishat Vasker

*Department of Computer Science and Engineering
East West University
Dhaka, Bangladesh
nishatvasker@gmail.com*

5th Md. Mahir Ashhab

*Department of Computer Science and Engineering
East West University
Dhaka, Bangladesh
mahir.ashhab@ewubd.edu*

6th Md.Sawkat Ali

*Department of Computer Science and Engineering
East West University
Dhaka, Bangladesh
alim@ewubd.edu*

7th Mohammad Rifat Ahmmad Rashid

*Department of Computer Science and Engineering
East West University
Dhaka, Bangladesh
rifat.abir@gmail.com*

8th M. Saddam Hossain Khan

*Department of Computer Science and Engineering
East West University
Dhaka, Bangladesh
saddam.cse@ewubd.edu*

Abstract—Coughing, chest pain, and exhaustion are symptoms of the common respiratory infection known as pneumonia. It is harmful to young children, the elderly, and anyone with compromised immune systems. The diagnosis process includes physical examination, a review of medical history, imaging testing, and treatment with antibiotics, antiviral drugs, and supportive care. This study proposes using three models—VGG19, DenseNet201, and CheXNet—of convolutional neural networks (CNN) for the identification of pneumonia. The goal is to evaluate the effectiveness of different models and determine which one is most reliable for detecting pneumonia. On a sizable dataset of chest X-ray pictures, the VGG19 and DenseNet201 models were trained and assessed. The best training and tuning accuracy were achieved by our proposed model, which received a score of 98.22%. Numerous patterns and irregularities in chest X-ray images associated with pneumonia were successfully identified using the improved CheXNet model. These results highlight the enormous potential of convolutional neural networks for automating the diagnosis of pneumonia. More research and validation are needed to demonstrate its stability and generalizability across different patient populations and imaging types.

Index Terms—CNN, VGG19, CheXNet, DenseNet201, Pneumonia.

I. INTRODUCTION

Around the world, pneumonia is a serious respiratory infection that affects millions of people [5]. A timely and correct diagnosis is necessary for optimal therapy and care. Chest X-ray scans are often used for this purpose in clinical settings, but radiologists must manually interpret the pictures, which can be laborious and error-prone [13]. Convolutional neural networks (CNN's) have shown potential for use in medical imaging applications as efficient tools for automated picture recognition [7][8]. The study's objectives are to train and assess the VGG19, DenseNet201, and CheXNet models on a dataset of chest X-ray images to detect pneumonia, to modify the CheXNet model in order to enhance performance, to compare the accuracy and effectiveness of the three models, and to look into the effects of various modifications on the CheXNet model's performance [10]. By comparing the performance of the three models, analyzing three cutting-edge CNN models, enhancing the CheXNet model, and evaluating the performance of the three models, the work advances the field of pneumonia identification in chest X-ray pictures [9][11]. An overview of pneumonia detection using CNN's and the identification of knowledge gaps are provided in this

9

study. It explains the research's methodology, experimental set-up, dataset, CNN architectures, and training techniques. Results and analysis contrasting the updated CheXNet model, DenseNet201, and VGG19 are presented. Discussions of the findings and their ramifications are made, as are conclusions drawn from changes made to the CheXNet model [12][14]. A summary of major contributions and recommendations for future directions in automated CNN-based pneumonia detection are provided in the paper's conclusion.

II. LITERATURE REVIEW

A well-known area of research in image processing for a long time is picture classification. This section has reviewed pertinent literature for utilizing CNN to categorize photos. We have examined a lot of articles that belong under the category of photographs to deepen our grasp of CNN. This study created a model to recognize pneumonia in chest X-ray images using the CheXNet model, DenseNet201, and VGG19. Dejun Zhang et al. [2] proposes a multi-layer architecture comprising convolutional, pooling, and fully connected layers for pneumonia diagnosis from chest X-ray images. A sizable dataset of chest X-ray pictures, comprising instances with and without pneumonia, is used to train the model. To boost variety, data augmentation methods and transfer learning are used. Metrics including accuracy, precision, recall, and F1-score are used to evaluate performance. The study seeks to address shortcomings in the literature, including small datasets, lack of comprehensibility, sensitivity to image alterations, and restricted transferability.

Cheng Wang et al. [3] outlines an innovative method for diagnosing pneumonia of the chest that uses graph reasoning. To depict the relationships between various lung regions in an X-ray image, the authors build a graph representation of those regions. The diagnostic model performs better because to this graph reasoning process, which can capture both local and global contextual data.

Alhassan Mabrouk et al. [4] uses a group of deep convolutional neural networks (CNN's) to provide a novel method for successfully diagnosing pneumonia in chest X-ray pictures. To overcome the difficulty of correct diagnosis, the authors suggest a strong ensemble model that incorporates the predictions of various CNN's. When it came to accuracy, sensitivity, specificity, and F1 score, the ensemble model performed remarkably well. The study's findings show how effective the ensemble approach is, how resilient it is to changes in image quality, and how generalizable it is to various datasets. When compared to single-model solutions, the ensemble model shows enhanced accuracy and reliability, highlighting the value of model diversity. The study does, however, have certain drawbacks, including the retrospective dataset, potential bias, and the need for additional validation on bigger and more varied datasets.

Ejaz Khan et al. [6] presents a deep learning-based approach for accurate COVID-19 detection from chest X-ray images. The authors present a detailed methodology using a convolutional neural network (CNN) model and an extensive dataset

of positive and negative X-ray images. The model undergoes comprehensive training, evaluation, and comparison against other state-of-the-art models to assess its performance.

III. SYSTEM ARCHITECTURE AND DESIGN

This section goes into detail on the proposed improved pneumonia detection method's system architecture and design process. Color and form are utilized to separate pneumonia from images. The phases of pneumonia identification approach are described in the sections that follow.

A. Dataset Description

The 5,863 X-ray images and two classifications (pneumonia or normal) were available in the Kaggle dataset. Each image category has its own subfolder within the dataset, which is divided into three folders (train, test, and val) [1]. The dataset was divided into three folders at random, trained, and then evaluated using 70%, 10%, and 20%, respectively. Using a hyper-parameter of 50, 100, 224, or 300, all photos were scaled to a set shape. The validation set has 579 photos divided into normal and pneumonia sub-sets, while the training set has 4037 images with those two sub-sets. The test set had 1170 photos in total, with subsets for normal and pneumonia having 316 and 854 images, respectively. Due to the low exposure dose, chest X-ray images only have a limited amount of contrast and are composed primarily of black, white, and gray. It is challenging for X-rays to get through the heart, lungs, and thick proteins in bones because of their white borders. Fig. 1 shows the input image data for this project.



Fig. 1. Sample inputs of the dataset

B. Dataset Preprocessing

Data preprocessing aims to make the data more suitable for the learning algorithms, improve model performance, reduce computational complexity, and address any issues or challenges present in the original data [15]. It plays a crucial role in ensuring that the input data is representative, informative, and well-prepared for the subsequent stages of model training and evaluation. We prepare the training, validation, and test data by applying various preprocessing steps, such as rescaling pixel values, applying data augmentation techniques (zooming, shearing, and horizontal flipping) to the training data, and resizing the images to a target size of (224, 224) for all the images. This shape is selected so that images can be feed into all the pre-trained models.

C. Proposed Methodology

The methodology uses three deep convolutional neural networks (CNNs): VGG19, DenseNet201, CheXNet, and our proposed model. VGG19 has fully connected layers, while DenseNet201 has dense connections. The proposed model incorporates DenseNet201's dense connectivity and pre-trained weights for improving feature extraction capacity. The pre-trained DenseNet201 weights are initialized, and the target dataset for pneumonia detection is used for training. Data augmentation techniques are employed, and model parameters are initialized. Training data is fed into the network, and parameters are adjusted using backpropagation and gradient descent optimization algorithms. Performance on the validation set is tracked, and the proposed model surpasses other models in terms of precision, recall, and F1-score. Hyperparameter tuning involves determining the best values for various parameters, including learning rate, batch size, optimizer selection, regularization methods, and the number of layers in the fully linked portion of the model. Advanced optimization algorithms, such as Bayesian optimization, grid search, and random search techniques are used for hyperparameter tuning. The final model is trained using these hyperparameters and assessed on the testing set. In our proposed model, we take an image (224x224x3) as input. We used DenseNet201 as the base model and added some extra layers such as GlobalAveragepooling2D, Dense (1024 units), and Dense (2 units) for our proposed model. Fig. 2 shows the workflow of the proposed methodology.

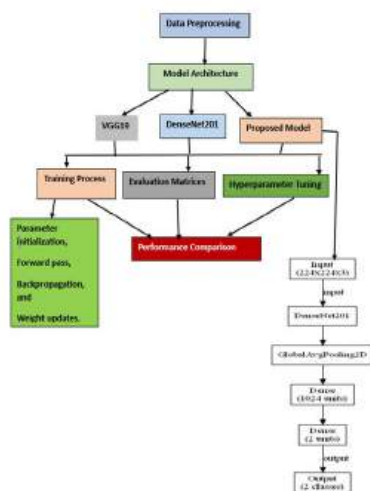


Fig. 2. Workflow of proposed methodology

IV. EXPERIMENTATION AND EXPERIMENTAL RESULT

This part includes an evaluation of the model's performance as well as the implementation process and experimental setup in detail.

A. Experimental Setup

Our hardware configuration consists of a Ryzen 5 2400G CPU and 8GB RAM. For software, we used a robust Python development tool called Google Colab which offers zero-configuration, free access to GPU's, and simple sharing. Colab is a tool supplied by Google for writing Python code with all the libraries.

B. Implementation

The DenseNet201 model is implemented and has been trained on a dataset of chest X-ray images for predicting two classes. The implementation involves importing required libraries, setting the image size, loading the model with pre-trained weights, freezing the layers, creating a custom model architecture, setting up data generators for training, validation, and testing, compiling the model, training the model, evaluating the model on the test set, and calculating additional metrics. The model is trained for 20 epochs because afterward, we observed that there have been no changes. Then the model has been evaluated on the test set, and visualized using plots. The trained model is saved and a confusion matrix is created using the true labels (y_{true}) and predicted labels (y_{pred}). The confusion matrix is visualized using a heatmap. The model's predictions are obtained on the test set using 'predict', and the model's predictions are visualized using a heatmap.

The DenseNet201 model is integrated into the CheXNet architecture by replacing the original convolutional layers of CheXNet with DenseNet201's dense blocks. This integration allows the proposed model to benefit from the dense connectivity and feature extraction capabilities of DenseNet201. The implementation of our proposed model using the DenseNet201 architecture involves importing necessary libraries, defining data paths, setting up data augmentation and generators, creating the base model, building the classification model, compiling the model, training the model, saving the trained model, evaluating the model, plotting the training history, and generating a confusion matrix. The base model, 'DenseNet201', is pre-trained on the ImageNet dataset and has proven effective for image classification tasks. Additional layers are added to perform classification, and global average pooling is applied to reduce spatial dimensions. A fully connected (dense) layer with 1024 units and ReLU activation is added, and a final dense layer with 2 units along with sigmoid activation is added for binary classification. The model is trained using the 'fit' function, with steps_per_epoch set to the length of the training generator. The model is evaluated using the test generator to obtain losses and metrics, and the calculated metrics are printed for analysis.

For high performance in computer vision tasks, the VGG19 architecture is utilized for binary image classification, leveraging pre-trained convolutional neural networks (CNN). To avoid modifications during training, we loaded a pre-trained model and froze all the weights. The final classification model is constructed by adding a fully connected dense layer, which uses the sigmoid activation function to output probabilities for binary classification. Data paths and data generators are set up,

9

and the 'ImageDataGenerator' class is used to augment and pre-process images during training. The model is compiled with appropriate loss, optimizer, and metrics, using the binary cross-entropy loss function and the Adam optimizer for efficiency and effectiveness. The model is trained using the 'fit' method with specified epochs and a validation data generator. We have plotted the accuracy and loss curves based on training and validation history. The model is evaluated on the test set using the 'evaluate' method, providing metrics such as test loss, accuracy, precision, and recall.

C. Experimental Result

Table 1 demonstrates the experimental result analysis of the three models that we implemented for this project. The proposed model using DenseNet201 model performs well and obtains the highest accuracy. In figure 3, we perform the

TABLE I
RESULT ANALYSIS

Model	Accuracy	Val_Accuracy	Loss	Val_Loss
Plain CheXNet	95.71%	93.75%	10.12%	19.13%
DenseNet201	97.26%	87.50%	20.99%	98.17%
VGG19	96.15%	93.75%	10.46%	19.26%
Proposed CheXNet	98.22%	93.75%	4.77%	13.65%

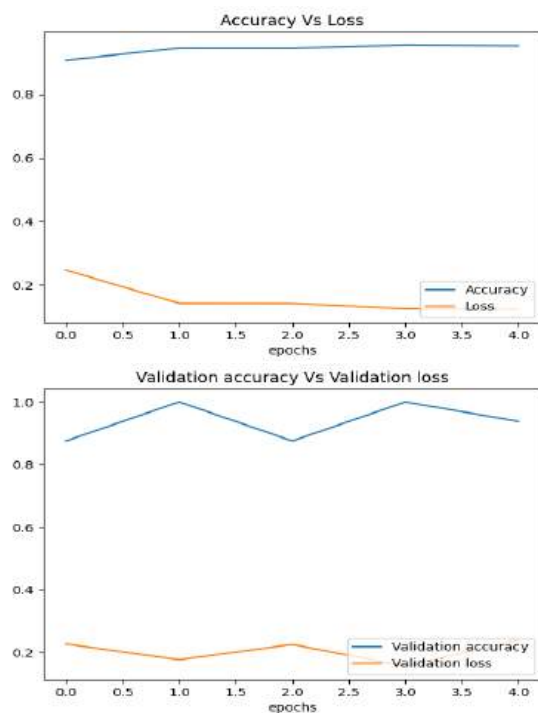


Fig. 3. Plain CheXNet model (epoch vs. accuracy)

CheXNet model with 20 epochs and show the accuracy vs. loss and validation accuracy vs. validation loss plots, where the accuracy is 95.71% and the loss is 10.12%. Also, validation accuracy is 93.75%, and validation loss is 19.13%. In figure 4, we perform our model with 50 epochs and show the accuracy vs. loss and validation accuracy vs. validation loss plots, where

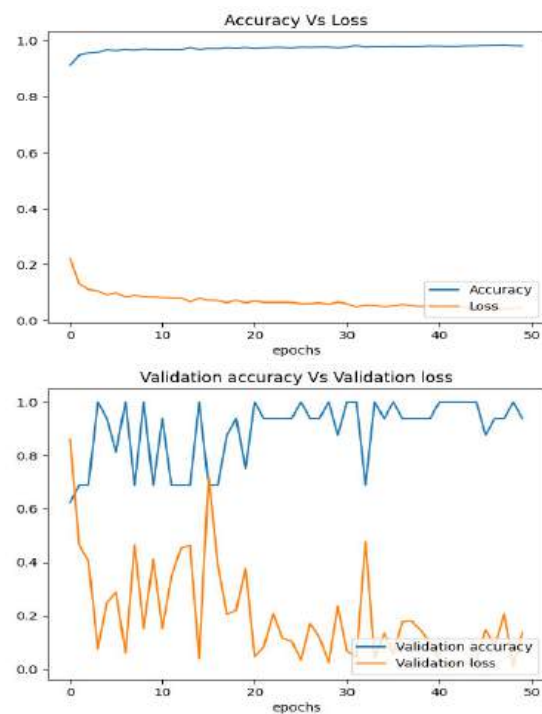


Fig. 4. Our proposed model (epoch vs. accuracy)

the accuracy is 98.22% and the loss is 4.77%. Also, validation accuracy is 93.75%, and validation loss is 13.65%. In figure 5,

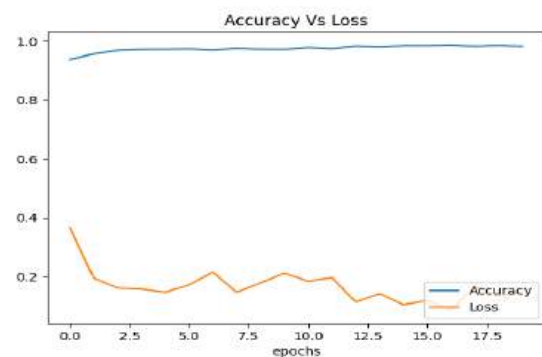


Fig. 5. DenseNet201 model (epoch vs. accuracy)

we perform the DenseNet201 model with 20 epochs and show the accuracy vs. loss and validation accuracy vs. validation loss plots, where the accuracy is 97.26% and the loss is 20.99%. Also, validation accuracy is 87.50%, and validation loss is 98.17%. In figure 6, we perform the VGG19 model with 20 epochs and show the accuracy vs. loss and validation accuracy vs. validation loss plots, where the accuracy is 96.15% and the loss is 10.46%. Also, validation accuracy is 93.75%, and validation loss is 19.26%. With our data, all models work satisfactorily. This allowed us to visualize the data with greater accuracy by excluding the graph from some of the results.

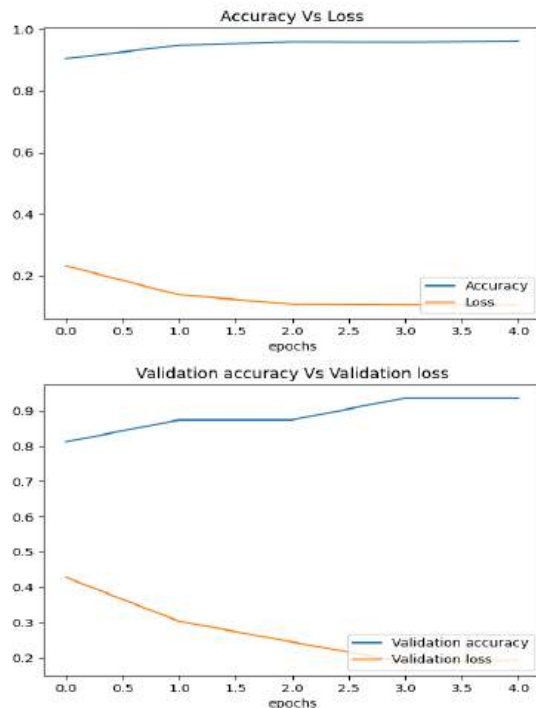


Fig. 6. The VGG19 model (epoch vs. accuracy)

D. Performance Evaluation

Table 2 shows the Summary of several models' performances. We have implemented three models named CheXNet, VGG19, and DenseNet201. Among them, we proposed our model using the DenseNet201 architecture. and it performs well with the highest accuracy among other models.

TABLE II
SUMMARY OF SEVERAL MODELS' PERFORMANCES

Model	Precision	Recall	F1-score
Plain CheXNet	95.70%	95.63%	95.66%
DenseNet201	97.1%	96.95%	97.02%
VGG19	96.16%	96.13%	96.14%
Proposed CheXNet	98.20%	98.22%	98.21%

A performance evaluation was conducted on three models: CheXNet, VGG19, DenseNet 201 and our proposed model using the DenseNet201 architecture has been introduced. Key findings showed that Plain CheXNet achieved a precision of 95.70%, a recall of 95.63%, and an F1-score of 95.66%. With a precision of 97.1%, a recall of 96.95%, and an F1-score of 97.02%, DenseNet201 surpassed Plain CheXNet, with a precision of 96.16%, a recall of 96.13%, and an F1-score of 96.14%, VGG19 likewise did admirably. With a precision of 98.20%, recall of 98.22%, and F1-score of 98.21%, our proposed model showed the highest accuracy among all the other models. When choosing a model, it is essential to consider elements like computational effectiveness, model complexity, and dataset features. Figures 7, 8, 9, and 10 show the confusion matrix for each model.

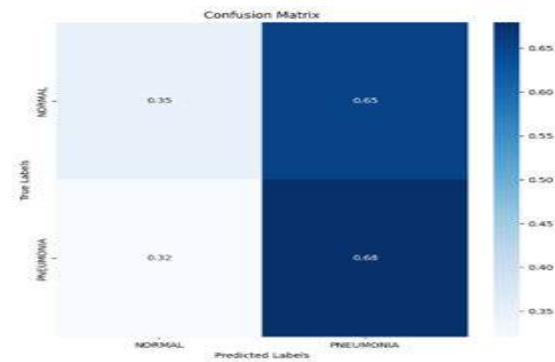


Fig. 7. CheXNet model

There were 35% true positives, 65% false positives, 32% false negatives, and 68% false negatives in the confusion matrix represented in Figure 7. In the confusion matrix shown



Fig. 8. Our proposed model

in Figure 8, there were 33% true positives, 67% false positives, 32% false negatives, and 68% false negatives. Figure 9's

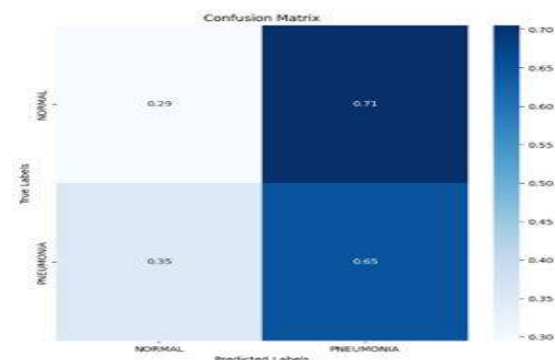


Fig. 9. DenseNet201 model

confusion matrix presented 29% true positives, 71% false positives, 35% false negatives, and 66% false negatives. . There were 33% true positives, 67% false positives, 32% false negatives, and 68% false negatives in the confusion matrix in Figure 10.

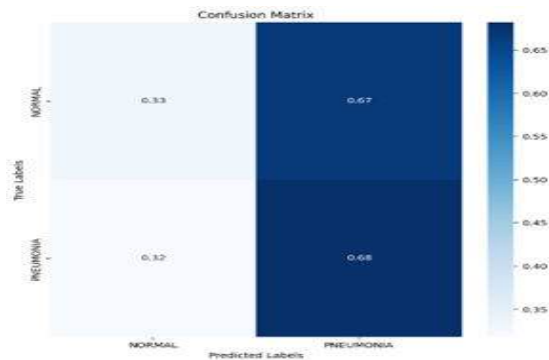


Fig. 10. VGG19 model

CONCLUSION

In our research, chest X-ray images were analyzed to detect pneumonia using convolutional neural networks (CNN's). Out of the three models we used—VGG19, DenseNet201, and CheXNet—our model got the maximum degree of accuracy. Our model outperformed VGG19, DenseNet201, and CheXNet, proving that CNNs are useful tools for spotting pneumonia in chest X-ray pictures. Our model's excellent accuracy demonstrates the potential of deep learning approaches for pneumonia identification and will help radiologists and healthcare professionals diagnose patients correctly. Early diagnosis and treatment of pneumonia can prevent complications and perhaps save lives. The study shows how CNN's can be used in healthcare and adds to the growing body of knowledge in medical imaging. Potential uses include incorporating CNN models into computer-aided diagnostic (CAD) systems, enhancing the effectiveness of medical personnel in complex cases, and helping with the screening and prioritization of individuals who may have pneumonia.

REFERENCES

- 1) Mabrouk, A., Díaz Redondo, R. P., Dahou, A., Elaziz, M. A., & Kayed, M. (2022, June 25). Pneumonia Detection on Chest X-ray Images Using Ensemble of Deep Convolutional Neural Networks. MDPI. <https://doi.org/10.3390/app12136448>
- 2) Wang, C., Xu, C., Zhang, Y., & Lu, P. (2023, June 20). Diagnosis of Chest Pneumonia with X-ray Images Based on Graph Reasoning. MDPI. <https://doi.org/10.3390/diagnostics13122125>
- 3) Mabrouk, A., Díaz Redondo, R. P., Dahou, A., Elaziz, M. A., & Kayed, M. (2022, June 25). Pneumonia Detection on Chest X-ray Images Using Ensemble of Deep Convolutional Neural Networks. MDPI. <https://doi.org/10.3390/app12136448>
- 4) Khan, E., Ur Rehman, M. Z., Ahmed, F., Alfouzan, F. A., Alzahrani, N. M., & Ahmad, J. (2022, February 5). Chest X-ray Classification for the Detection of COVID-19 Using Deep Learning Techniques. MDPI. <https://doi.org/10.3390/s22031211>
- 5) Tiu, E., Talus, E., Patel, P., Langlotz, C. P., Ng, A. Y., & Rajpurkar, P. (2022, September 15). Expert-level detection of pathologies from unannotated chest X-ray images via self-supervised learning - Nature Biomedical Engineering. Nature. <https://doi.org/10.1038/s41551-022-00936-9>
- 6) Dinh, T. L., Lee, S. H., Kwon, S. G., & Kwon, K. R. (2022, May 11). COVID-19 Chest X-ray Classification and Severity Assessment Using Convolutional and Transformer Neural Networks. MDPI. <https://doi.org/10.3390/app12104861>
- 7) H., Ji, D., Zhang, Z., Zhao, Y., & Zhao, Q. (2021, August 25). Research on Classification of COVID-19 Chest X-Ray Image Modal Feature Fusion Based on Deep Learning. Research on Classification of COVID-19 Chest X-Ray Image Modal Feature Fusion Based on Deep Learning. <https://doi.org/10.1155/2021/6799202>
- 8) Sogancioglu, E., Çallı, E., Ginneken, B. V., van Leeuwen, K. G., & Murphy, K. (2021, March 15). Deep Learning for Chest X-ray Analysis: A Survey. arXiv.org. <https://doi.org/10.1016/j.media.2021.102125>
- 9) Deep learning for chest X-ray analysis: A survey. (2021, June 5). Deep Learning for Chest X-ray Analysis: A Survey - ScienceDirect. <https://doi.org/10.1016/j.media.2021.102125>
- 10) Meedeniya, D., Kumarasinghe, H., Kolonne, S., Fernando, C., la Torre Díez, I. D., & Marques, G. (2022, July 18). Chest X-ray analysis empowered with deep learning: A systematic review. PubMed Central (PMC). <https://doi.org/10.1016/j.asoc.2022.109319>
- 11) Tiu, E., Talus, E., Patel, P., Langlotz, C. P., Ng, A. Y., & Rajpurkar, P. (2022, September 15). Expert-level detection of pathologies from unannotated chest X-ray images via self-supervised learning - Nature Biomedical Engineering. Nature. <https://doi.org/10.1038/s41551-022-00936-9>
- 12) COVID-19 Identification from Chest X-Rays. (n.d.). COVID-19 Identification From Chest X-Rays — IEEE Conference Publication — IEEE Xplore. <https://ieeexplore.ieee.org/document/9244509>
- 13) Deep Learning Based Chest X-Ray Image as a Diagnostic Tool for COVID-19. (n.d.). Deep Learning Based Chest X-Ray Image as a Diagnostic Tool for COVID-19 — IEEE Conference Publication — IEEE Xplore. <https://ieeexplore.ieee.org/document/9215257>
- 14) Artificial Intelligence Applied to Chest X-Ray Images for the Automatic Detection of COVID-19. A Thoughtful Evaluation Approach. (n.d.). Artificial Intelligence Applied to Chest X-Ray Images for the Automatic Detection of COVID-19. A Thoughtful Evaluation Approach — IEEE Journals & Magazine — IEEE Xplore. <https://ieeexplore.ieee.org/document/9293268>
- 15) Covid Prediction from X-ray Images. (n.d.). Covid Prediction From X-ray Images — IEEE Conference Publication — IEEE Xplore. <https://ieeexplore.ieee.org/document/9276795>

Using Deep Learning Systems for Imaging Methods for Recognising Brain Tumors

Monjurul Islam Sumon
Computer Science and Engineering
East West University
Dhaka, Bangladesh
123monjurul@gmail.com

Md Tanvir Chowdhury
Computer Science and Engineering
East West University
Dhaka, Bangladesh
mdtanvirchowdhury015@gmail.com

Syed Salman Hayat
Computer Science and Engineering
East West University
Dhaka, Bangladesh
salman.sourov40@gmail.com

Mahir Tagwar
Computer Science and Engineering
East West University
Dhaka, Bangladesh
onontaj1999@gmail.com

Habibur Rahman,
Computer Science and Engineering
East West University
Dhaka, Bangladesh
habibresarchmail@gmail.com

Md. Sabbir Hossain
Computer Science and Engineering
East West University
Dhaka, Bangladesh
sabbirhossain@gmail.com

Mahamudul Hasan
Computer Science and Engineering
East West University
Dhaka, Bangladesh
munna09bd@gmail.com

Taskeed Jabid
Computer Science and Engineering
East West University
Dhaka, Bangladesh
taskeed@ewubd.edu

Maheen Islam
Computer Science and Engineering
East West University
Dhaka, Bangladesh
maheen@ewubd.edu

Abstract—Uncontrolled and abnormal cell proliferation is the cause of a brain tumor, a type of cancer. The healthcare industry has benefited from recent advancements in deep learning by using diagnostic imaging to identify various disorders. Task If you want to learn about images or identify images, CNN is the machine learning algorithm you should use. In a similar vein, we have used the convolutional neural network (CNN) method, Data Augmentation, and Image Processing to determine whether or not images from brain MRI scans show cancer. Using the transfer learning method, we compared the performance of our shredded CNN model to that of the already trained Xception, which ResNet-50, and Inception-v3 models. Our model achieved a perfect score of 100% on a sizable dataset, demonstrating its high accuracy and low complexity, while Xception, ResNet-50, and Inception-V3 only managed to achieve 99.3%, 98.4%, and 99.4% accuracy, respectively. In comparison to other pre-trained models, our model only requires a little amount of processing resources and achieves substantially higher accuracy levels.

Keywords— *CNN Architectures, ResNet50, Inception V3, Deep Learning.*

I. INTRODUCTION

A brain tumor is an accumulation of unruly cells in the brain or central spinal canal. Our skull, which is quite rigid, protects our brain. Any expansion in such a constrained area will result in issues. People have various problems to deal with. Cancerous or non-cancerous brain tumors are also possible. Internal tension Tumors, whether benign or cancerous, enlarge the cranium as they grow. It may result in brain damage, which could be fatal. There are numerous locations where brain tumors can develop, and they can take many different forms. There are two main categories of brain tumors: primary and secondary. Our brain is the site of origin for an initial brain tumor. Primary brain tumors are typically not dangerous. When cancer cells from another organ, such

as the lung or breast, move to our brain, it results in a secondary brain tumor, also referred to as a metastatic brain tumor. A significant amount of human lives will be saved if tumor cells are detected early. The method of detecting a brain tumor and its level is so tough and time-consuming. When signs linked to tumors occur, the patient requests an MRI. If a tumor is suspected after analyzing the brain scans, the patient's brain biopsy is performed.

II. LITERATURE REVIEW

A. Convolutional Neural Network (CNN)

A class of deep learning methods known as convolutional neural networks has gained popularity and influence in a variety of computer vision problems. Convolution layers, pooling layers, and fully connected layers are some of the constituent parts. It is created using a backpropagation method to automatically and adaptively learn spatial hierarchies of features.

B. Related Work

To locate brain tumors in MRI scans, image processing software employs AI along with deep learning to separate and categorize the images. The study and segmentation of MRI scans of the brain have received a lot of attention and work. They have been used alongside MRI in this study [1]. Magnetic resonance imaging (MRI) is a type of scan that uses massive magnets and radio frequencies to produce clear pictures of soft tissue layers deep within the human body. Strong magnets are found in MRI scanners. Total 500 samples data were collected from different hospitals and they have been collected two types of data normal and abnormal cases. Then they apply image pre-processing in their dataset and sharpening images to take place using the Gabor filter. In medical images, it's difficult to perform but image segmentation is important is an important step. By using the OTSU method brain tumor is detected. There is a wide range of signal strengths inside the same tissue due to the volume

effect.[2] Classify multi-grade tumors using data augmentation techniques applied to MRI scans, and then fine-tune the model using a pre-trained VGG-19 CNN. [3] described a method for classifying Pituitary adenoma MRIs analyzed with multinomial logistic regression techniques and k-nearest neighbor techniques. They have an AUC curve of 98.4% accuracy, 83% accuracy in multinomial logistic regression, and 92% accuracy in k-nearest neighbor. [4] presented an innovative method for classifying healthy and sick brain pictures. We propose a unique method for classifying brain medical images into healthy and unhealthy brain images using ConvNet, which includes unhealthy images of brain tumors. It is possible to rate the unhealthy images of brain tumors into low and high categories. By changing the deep learning architecture model of the Alex Krizhevsky network (AlexNet), accuracy of 91.16%, sensitivity of 92.05%, and specificity of 92.25% were attained. The accuracy of the model developed in [5], which used a convolutional neural network as an image processing technique to distinguish between different varieties of brain tumors, was 94.39%, with a standard precision of 93.33%. Using a machine learning technique based on Naive Bayes and k-nearest neighbors, researchers were able to accurately predict tumor grades and nodal activity from CT scans of primary tumor lesions [6]. Deep transfer learning in 613 normal and diseased brain MR images was performed using the ResNet34 model, which is based on a convolutional neural network (CNN). Deep learning approaches were utilized to train the model with 100% accuracy, including data augmentation, the optimal learning rate finding, and fine-tuning. [7] Three convolutional neural network designs (AlexNet, GoogleNet, and VGGNet) were utilised to categorise brain tumours. When applying transfer learning strategies for classification and detection, the VGG16 architecture achieved the highest accuracy (98.69%). [8] Accuracy is compared between pre-trained models and the following networks: GoogleNet, AlexNet, ResNet50, DenseNet201, and InceptionV3. The pre-trained ResNet50 CNN model was modified by removing the last five layers. The designed model has a 97.2% success rate. The new model operates quickly and precisely. [9] 253 brain MRI scans in total were trained using convolutional neural networks. Compared to the previously trained VGG16, ResNet50, and Inception V3 models, see how well the CNN model performs. While the model only needs a little amount of processing power, VGG16 obtained 96% accuracy, ResNet50 achieved 89% accuracy, and InceptionV3 reached 75% accuracy. The maximum accuracy, 96%, was reached by the VGG16 architecture.[10] To apply the suggested tactic, the Densenet201 Pre-Trained Deep Learning Model is adjusted and subsequently trained via a deep transfer of unbalanced data learning. Finally, both EKbHFV and MGA-based features are fused and classified using a multiclass SVM cubic classifier utilising a non-redundant serial-based approach. For the experimental method, two datasets—BRATS2018 and BRATS2019—that have more than 95% correctness are employed without change.[11] In this research, we describe a novel Mask Region-based Convolution neural network (Mask RCNN) for precise brain tumour identification and segmentation, which is trained via transfer learning. Mask RCNN has a densenet-41 backbone architecture. Compared to cutting-edge techniques, our Segmentation accuracy for the proposed model was 96.3 percent, while classification accuracy was 98.34 percent.[12] In this comparative study, four deep learning methods—

AlexNet, VGG16, GoogleNet, and ResNet50—are used to categorize brain tumors. The results show that ResNet50 is the best model with a 95.8% accuracy rate, while AlexNet performs the fastest with a 1.2 second processing time.

Reference	Dataset	Method and Accuracy	Finding	Limitation / Gap
[11]	Total: 253 brain images (2 classes) Nor: 135 images Nor: 118 images	Inception V3: 75% ResNet50: 89% VGG-16: 96%	VGG-16 works better than Inception and ResNet V3	Lack of data model is indicated with more architectures.
[12]	Total: 253 brain images (2 classes) Nor: 135 images Nor: 118 images	Densenet: 93.33% ResNet50: 89%	Densenet performed better	Small dataset and not applying the other deep learning models.
[13]	Total: 253 brain images (2 classes) Nor: 135 images Nor: 118 images	Densenet: 96.3% ResNet50: 95.8%	Densenet achieved better accuracy than ResNet	Not using other deep learning models.
[14]	Total: 3000 brain images (2 classes) Nor: 1500 images Nor: 1500 images	AlexNet: 82.2% VGG-16: 98.6% GoogleNet: 93.8% ResNet50: 95.8%	ResNet performed better than AlexNet, VGG-16 and GoogleNet	Did not get better accuracy.

Fig. 1. A overview of some literature review.

III. METHODOLOGY

The proposed methodologies for identifying brain tumors using brain imaging are described in detail in this section. The section would go over the procedure and formally present the techniques after going over the dataset specifics.

A. Dataset Description

There are numerous freely available datasets of medical images on various platforms. We gathered brain MRI pictures from the Kaggle database. By [13], the dataset was created. After collecting we have balanced both datasets using domain knowledge and modified the datasets. For the convenience of work, here the dataset is divided into Dataset-1 and Dataset-2 and dataset-3. Both datasets have two classes of Brain tumor and Non Brain tumor(normal). Firstly, Dataset-1 contains total 3000 MRI images. Dataset-1 is divided into two classes one is Brain tumor MRI Images and another class is Non Tumor MRI Images(Normal). For Brain tumor 1500 MRI images and another 1500 MRI images for Normal. Secondly, Dataset-2 contains total 2550 MRI images. Dataset-2 is divided into two classes one is Brain tumor MRI Images and another class is Non Tumor MRI Images(Normal). For Brain tumor 1300 MRI images and another 1250 MRI images for Normal. Thirdly, Dataset-3 contains total 2550 MRI images. Dataset-3 is divided into two classes one is Brain tumor MRI Images and another class is Non Tumor MRI Images(Normal). For Brain tumor 1200 MRI images and another 1350 MRI images for Normal.

B. Data Augmentation

Image augmentation is a powerful approach for increasing the size of the training set without having to acquire additional pictures while developing convolutional neural networks. We rotated, cropped, and flipped the photos as needed as part of the data augmentation. We can rearrange the pixels completely by changing the picture horizontally, vertically, and rotating it. You can fit models with image data using the The Image Data Generator function class in the Keras deep learning neural network framework. The training dataset is expanded using picture data augmentation to improve the model's accuracy and ability to generalise.

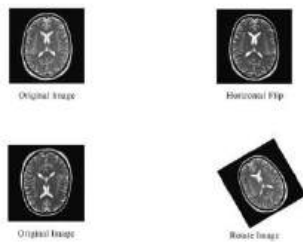


Fig.2. Data Augmentation Process.

C. Pre-Trained models

Academics utilizing medical data have a significant challenge due to a lack of appropriate datasets. In most cases, deep learning models need a large amount of training data. Researchers need to put in a lot of time and effort to designate this data. One major benefit of using transfer learning approach is that it requires fewer datasets and less processing power to train data. We used one from Keras as pre-trained models are often built-in features. Each of them has a value or weight assigned to it. modelled after conditions Train the model ahead of time with a large dataset of photos. It is often used in deep learning, and the pre-trained model transmits to the model being trained the knowledge it has learned from a large dataset. The weights or numbers in pre-trained models are in a range, making it simple to attain the desired fixed value. Using transfer learning, large-scale annotated picture datasets like ImageNet give answers to such a task, requiring the training of tens of millions of parameters of CNN architectures. We utilized the pre-trained ImageNet Inception V3, Resnet, and Xception models, which performed exceptionally well on ImageNet. This convolutional neural network is based on theories about how neurons in the visual brain function. As a result, this network's design differs from that of traditional neural networks. The main difference between a regular neural network and a convulsive neural network is how they operate, and the former is more linked than the latter. The word convolution comes from arithmetic which means a mathematical combination of two functions whose output is the third function. Convolution is to bring two sets from input in one place.

D. Performance Metric

Pre-trained models can take the form of a matrix or a 3D geometry. We must flatten it to 1D as a result. We flatten the last layer's output across the remaining architecture layers. Each subsequent convolutional layer carries the weights that were previously trained from the ImageNet classification challenge. These designs therefore already understand the foundations of image processing. After flattening, the value size substantially increases. If we map anything into the three categories, the model starts to overfit. Many neurons lack the ability to learn anything. Then, we attempt to reduce the size of the neuron to 1024, 512, or 128 utilizing the dense layer. We used dropouts in the algorithms, which required occasionally missing certain neurons, to prevent overfitting. We then build two tightly linked layers with a real activation mechanism and a softmax activation function on top of that. We used Adam Optimizer because it's a well-liked optimizer.

An evaluation of our model was conducted using the [33] multi-class confusion matrix. The adopted performance metrics are:

Number of accurately recognized disease-related X-ray pictures, or true positive (TP).

Number of accurately detected Normal X-ray cases that were True Negative (TN).

False Positive (FP): Cases of normal X-rays that were wrongly identified.

Number of X-ray images of diseases that were wrongly labelled as false negatives (FN).

P, N stands for the positive and negative samples (Yes and No) for each class. One popular classification metric is accuracy. According to (Eq. 1), How successfully the model overall distinguishes between the three groups (Yes and No) is measured by a metric called accuracy. Comparatively, precision (Eq. 2) measures the proportion of true positives to the total number of labels predicted, whereas recall (Eq. 3) measures the proportion of true positives to the total number of labels predicted. There is often no distinction made between recall and sensitivity (sometimes called true positive rate). the F1 rating (Eq. 5) is the cumulative average of Precision and Recall; The efficacy (also known as real negative rate) is the proportion of genuine negatives that are correctly classified as such (Eq. 4).

E. CNN Architectures

In our work, we used a fundamental CNN model that was provided. With RGB color channels and a batch size of 32, we were able to extract enhanced MRI image data with an input size of 224*224. We began by including a single convolutional layer with 16 filters and a 3-3 filter size. 16 Only a few filters are employed to find lines, corners, and edges. To obtain the highest summary of the image, we then added a max-pooling layer with two-filtering, followed by an increase in the number of layers of convolution and screens to 32, 64, and 128 using the same three-stage filter size. As the amount of filters rises, these individual shapes, such as a circular one or a square, are blended together. We piled on top of the convolution layers with max-pooling layers to get the most out of it. We used the Quantitative Biosciences and Engineering, The amount of volume 17, Issue 5, 6203-6216, to create a connected dense layer of 256 neurons. The 6208 soft maximum output layer decides whether an MRI picture has cancer by computing the probability rating for every category and labeling the result as Yes or No. Figure 3 is a diagrammatic representation of the proposed CNN architecture.

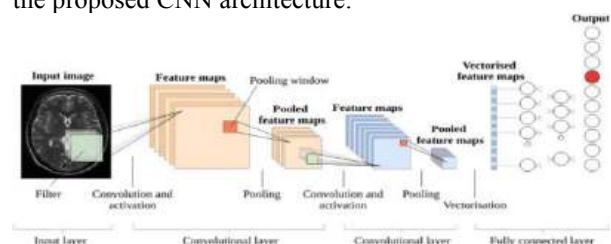


Fig. 3. An overview of the architecture of a CNN

F. ResNet50

A residual network called ResNet50 has 50 layers and 26 million elements. In 2015 [14], Microsoft unveiled the residual network, an advanced convolutional neural network model. In a residual network, we acquire residuals instead of features, which are the result of subtracting the features that were previously learned from the inputs to the layer. ResNet used a skip connection to exchange information across its many levels. ResNet enables the stacking of additional layers and the construction of a deep network by connecting the n th layer input to some $(n+x)$ layer. In our experiment, we improved a ResNet50 model that had already been trained. Figure 4 depicts the architecture of ResNet50.

G. Inception V3.

Launched in 2014 [15], GoogleNet is a pre-trained network model that resembles the Inception network. With a 22-layer network and 5M parameters, Inception V3 used max pooling, 1×1 , 3×3 , and 5×5 filter sizes to extract features at various scales. The requirement to decrease calculation time drives the use of 1×1 filters. InceptionV3 [16], a later version of Google's Inception model that factorizes Convolutional layers to reduce parameters, was released later in 2015. The two 3×3 filters were used in place of the five 5×5 Convolutional filters to reduce computing without compromising network performance. 48 layers make up the InceptionV3 model. In order to reduce overfitting, we used the InceptionV3 model in our experiment and adjusted it in accordance with the target data. Figure 5 shows the architecture of InceptionV3.

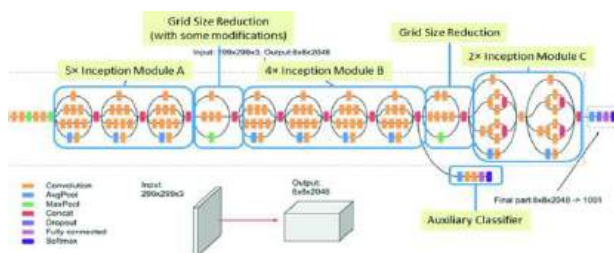


Fig 4. An overview of the architecture of a Inception-V3.

IV. RESULT ANALYSIS

In this part, we examine the suggested model's performance using the approaches and methodologies that have been examined.

A. Classification Accuracy

We know there are total three datasets. We have applied InceptionV3, Resnet50, Xception on those two datasets and attained: Precision, Recall (Sensitivity), Specificity, F1-Score, and Accuracy. Now for complete understanding of result analysis we need to know how these terms can define a model. Accuracy is the most well-known deep learning algorithm for comparison. The percentage of accuracy in a forecast indicates how well a model is performing. By dividing the entire number of occurrences by the combination of the actual favorable and genuine adverse values, we may arrive at this proportion.

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{P} + \text{N}) \quad (1)$$

Now, recall and precision and recall are two metrics that measure the efficacy of a classification or retrieval system. Precision refers to the fraction of useful results among all results obtained. Review is the limited number of recoverable examples within every single major occurrence, which is occasionally referred to as "affectability." Precision P and recall R for the ideal classifier are both equal to 1.

$$\text{P} = (\text{TP}) / (\text{TP} + \text{TN}) \quad (2)$$

$$\text{R} = \text{TP} / \text{P} \quad (3)$$

Specificity measures how many true negatives are caught by the model.

$$\text{Specificity} = (\text{TN}) / (\text{TN} + \text{FP}) \quad (4)$$

The F1-score quantifies a model's performance on a given dataset. It's used to evaluate systems of two-way classification in which cases are either "positive" or "negative." The F1-score is a metric that takes into account both the accuracy and the reliability of the simulation and takes into account its harmonic mean.

$$\text{F1-Score} = 2((\text{P} * \text{R}) / (\text{P} + \text{R})) \quad (5)$$

For improved prediction, the true positive rate, true negative rate, and accuracy all need to be higher. To assess the performance, we calculated each of these three measures over three datasets and four classifiers.

B. Classification accuracy on Dataset_1

Table 1: Classification results from various previously trained CNN models

Model	Label	Sensitivity (recall)	F1-Score	Accuracy
InceptionV3	Tumor	0.99	0.99	.994
	Normal	0.99	0.99	
Xception	Tumor	0.99	0.99	.988
	Normal	0.99	0.99	
Resnet50	Tumor	0.97	0.97	.984
	Normal	0.98	0.97	
Nesnet	Tumor	0.97	0.97	.985
	Normal	0.98	0.97	
DenseNet201	Tumor	0.95	0.96	.960
	Normal	0.96	0.97	
VGG16	Tumor	0.90	0.91	.917

Table 1 shows that when we enlarge our architectures, they all perform at their highest level of accuracy on the test set. According to the data, the InceptionV3 and the Xception had the best categorization accuracy, scoring 99%. Nesnet and Resnet50 both demonstrated 98% accuracy. Densent201 achieved 96% accuracy, VGG16 92% accuracy, and VGG19 91% accuracy for the other models. For the tumour class, the InceptionV3 and Xception models have a sensitivity (recall) of 99%. 99% Sensitivity denotes the classification of the majority of positive data points as positive. When it comes to specificity, Inception V3, Xception, and Resnet50 have been identified as 98%, 99%, and 98%, respectively. This means that 98% of the negative data points have been correctly categorised as negative. Densenet201, Vgg16, and Vgg19, in descending order of moderate sensitivity, all have values of 95%. Inception V3, Resnet50, Xception, and Nesnet will be

among the best performers on Dataset_1. the great worth associated with the tumor subtype's particularity, sensitivity, accuracy, and A score of F1. In comparison to other models, the architectures Inception V3, Resnet50, Xception, and Nesnet performed around averagely on the dataset_1. In the initial step of image processing, we resized each and every training and test image. We received the anticipated results in the rough release. In the end, we performed picture augmentation and obtained exceptional results. After the data processing stage and the augmentation of the data set, the pre-trained models showed high training and testing accuracy, as shown in Table 2. Additionally, it is important to note that the training and validation loss is minimal. This improves the models' performance even further.

Table 2: Comparative Performance of Modern CNN Architectures in Quantitative Terms

Model	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss
Inception V3	0.9938	0.0320	0.9986	0.0174
Xception	0.9826	0.0583	0.9979	0.0121
Resnet50	0.9842	0.0623	0.9971	0.0121
Nesnet	0.9883	0.0683	0.9928	0.0172
DenseNet201	0.9660	0.0852	0.9822	0.0184
VGG16	0.9212	0.0991	0.9652	0.0417
VGG19	0.9081	0.1012	0.9542	0.0421

C. Classification accuracy on Dataset_2

Dataset_2 contains 1350 normal images and 1200 tumor images. Similarly, to evaluate the results for each CNN for Dataset_2 are illustrated in Table 3.

Table 3: Classification results from various previously trained CNN models

Model	Label	Sensitivity (recall)	F1-Score	Accuracy
InceptionV3	Tumor	0.99	0.98	.989
	Normal	0.99	0.98	
Xception	Tumor	0.99	0.98	.993
	Normal	0.98	0.98	
Resnet50	Tumor	0.95	0.95	.892
	Normal	0.96	0.96	
Nesnet	Tumor	0.96	0.94	.942
	Normal	0.95	0.94	
DenseNet201	Tumor	0.91	0.91	.908
	Normal	0.90	0.91	
VGG16	Tumor	0.90	0.91	.907
	Normal	0.91	0.91	
VGG19	Tumor	0.91	0.91	.911
	Normal	0.91	0.91	

Comparative Performance of Modern CNN Architectures in Quantitative Terms The overall accuracy for the three classes, as well as the F1-score for brain tumor-related sensitivity, specificity, and precision, are provided. The table

above demonstrates that all CNN models can be used to detect brain tumors successfully, with a sensitivity of above 98%, and that the specificity of several models, including InceptionV3, Xception, and Nestnet, is moderate at 95%. The sensitivity of VGG19 is 91%, VGG16 is 90%, and Densenet201 is 91%. If we examine the accuracy of InceptionV3, Nesnet, Resnet50, Xception, Densenet, VGG16, and VGG19, it is 98.9%, 98.5%, 90.8%, and 91.1% respectively.

Table 4: Comparative Performance of Modern CNN Architectures in Quantitative Terms.

Model	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss
InceptionV3	0.9828	0.0241	0.9986	0.0174
Xception	0.9949	0.0283	0.9979	0.0121
Resnet50	0.8941	0.0923	0.9971	0.0121
Nesnet	0.9481	0.0614	0.9928	0.0172
DenseNet201	0.9124	0.0871	0.9822	0.0184
VGG16	0.9071	0.0841	0.9652	0.0417
VGG19	0.9121	0.0912	0.9542	0.0421

D. Classification accuracy on Dataset_3

Dataset_3 contains 1200 normal images and 1350 tumor images. Similarly, to evaluate the results for each CNN for Dataset_3 are illustrated in Table 5.

Table 5: Comparing the results of several pre-trained Convolutional Neural Network models for classification

Model	Label	Sensitivity (recall)	F1-Score	Accuracy
InceptionV3	Tumor	0.99	0.99	.989
	Normal	0.98	0.99	
Xception	Tumor	0.98	0.98	.985
	Normal	0.98	0.99	
Resnet50	Tumor	0.97	0.97	.966
	Normal	0.96	0.96	
Nesnet	Tumor	0.97	0.97	.974
	Normal	0.98	0.97	
DenseNet201	Tumor	0.90	0.90	.900
	Normal	0.90	0.89	
VGG16	Tumor	0.89	0.89	.893
	Normal	0.88	0.89	
VGG19	Tumor	0.93	0.93	.934
	Normal	0.93	0.93	

Statistical Comparison of High-Performance CNN Architectures Brain tumor sensitivity, specificity, F1-score, and total classification accuracy are all shown. It can be seen from the table above that all of the CNN models can be deployed successfully to detect Brain tumor with a sensitivity of above 99% and specificity of above 98% in some of the models InceptionV3, Xception. Nestnet shows specificity of 97%. Densenet201 show 91%, VGG16 show

90%, VGG19 show 91% sensitivity. If we look at the accuracy of InceptionV3, Xception are show 98.9% and 99.3% accuracy. Vgg16, DenseNet201 95% of both, and the other two models InceptionResNetV2, Xception, show 94% accuracy.

Table 6: Comparative Performance of Modern CNN Architectures in Quantitative Terms:

Model	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss
InceptionV3	0.9791	0.0219	0.9986	0.0174
Xception	0.9831	0.0583	0.9979	0.0121
Resnet50	0.9642	0.0622	0.9971	0.0121
Nesnet	0.9883	0.0683	0.9928	0.0172
DenseNet201	0.9002	0.0412	0.9822	0.0184
VGG16	0.9031	0.0512	0.9652	0.0417
VGG19	0.9343	0.0612	0.9542	0.0421

Although Inception V3 and Xception has performed very well in all respects, Nesnet, Resnet50 will also be ahead. One of the new and prevalent algorithms we used which is Xception. It goes without saying that very little work has been done on it before, or it has not been used for image classification. It is much smaller and faster. Depth wise scaling is most popular amongst all the conventional CNN architectures. But compound scaling has been done in Xception. Using this compound scaling method; it achieved brilliant improvement for InceptionV3 model architecture. According to Table 7, Inception, Xception, Resnet50, Densenet201, VGG16, and VGG19 all take longer than 213 milliseconds to complete. We have run only 22 epoch and batch size 15 for turning of each model. Testing time will be greatly reduced and we can emphasize that it's possible to get the results up to 99% accurately in a swift and short time

Table 7: Performance and Latency of Various Modern CNN Architectures Accuracy Graph:

Architecture	Latency Per Step
InceptionV3	213 ms
Xception	272 sec
Resnet50	375 ms
Densenet201	570 ms
VGG16	652 ms
VGG19	575 ms



Fig. 5. Inception V3 Model

REFERENCES

- [1] P. Saini, and M. Singh, "Brain Tumour Detection in Medical Imaging Using Matlab", 2015
- [2] M. Sajjad, S. Khan, M. Khan, W. Wu, A. Ullah, S. W. Baik, Multi-grade brain tumor classification using deep CNN with extensive data augmentation, J. Comput. Sci., 30 (2019), 174–182
- [3] R. Carlo, C. Renato, C. Giuseppe, U. Lorenzo, I. Giovanni, S. Domenico, Distinguishing Functional from Non-functional Pituitary Macroadenomas with a Machine Learning Analysis, Mediterranean Conference on Medical and Biological Engineering and Computing, Springer, 2019, 1822–1829.
- [4] S. Khawaldeh, U. Pervaiz, A. Rafiq, R. Alkhawaldeh, Noninvasive grading of glioma tumor using magnetic resonance imaging with convolutional neural networks, J. Appl. Sci., 8 (2018), 27.
- [5] S. Das, R. Aranya, N. Labiba, Brain tumor classification using convolutional neural network, 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), 2019.s
- [6] V. Romeo, R. Cuocolo, C. Ricciardi, L. Ugga, S. Cocozza, F. Verde, et al., Prediction of tumor grade and nodal status in oropharyngeal and oral cavity squamous-cell carcinoma using a radiomic approach, Anticancer Res., 40 (2020), 271–280.
- [7] A. Rehman, S. Naz, M. I. Razzak, F. Akram, M. Imran, A Deep Learning-Based Framework for Automatic Brain Tumors Classification Using Transfer Learning, Circuits Syst. Signal Process., 39 (2020), 757–775.
- [8] A. Cinar, M. Yldrm, Detection of tumors on brain MRI images using the hybrid convolutional neural network architecture, Med. Hypotheses, 139 (2020), 109684.
- [9] J. F. Canny, Canny Edge Detection, Open Source Computer Vision, OpenCV. Available from: https://docs.opencv.org/trunk/da/d22/tutorial_py_canny.html.
- [10] Khan, Hassan Ali, et al. "Brain tumor classification in MRI image using convolutional neural network." Mathematical Biosciences and Engineering 17.5 (2020): 6203-6216.
- [11] Sharif, Muhammad Imran, et al. "A decision support system for multimodal brain tumor classification using deep learning." Complex & Intelligent Systems (2021): 1-14.
- [12] Masood, Momina, et al. "A Novel Deep Learning Method for Recognition and Classification of Brain Tumors from MRI Images." Diagnostics 11.5 (2021): 744.
- [13] FAbbood, Alaa Ahmed, Qahtan Makki Shallal, and Mohammed A. Fadhel. "Automated brain tumor classification using various deep learning models: a comparative study." Indonesian Journal of Electrical Engineering and Computer Science 22.1 (2021): 252-259.
- [14] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, Proceedings of the IEEE conference on computer vision and pattern recognition, 2016
- [15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, et al., Going deeper with convolutions, Proceedings of the IEEE conference on computer vision and pattern recognition, 2015.
- [16] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, Proceedings of the IEEE conference on computer vision and pattern recognition, 2016.

Enhancing IoT Forensics through Deep Learning: Investigating Cyber-Attacks and Analyzing Big Data for Improved Security Measures

1st Karam Muhammed Mahdi Salih
Department of Computer Network
Ninevah University
Mosul, Iraq
Karam.mahdi@uoninevah.edu.iq

2nd Najla Badi Ibrahim
Department of Computer Science
University of Mosul
Mosul, Iraq
Najlabadie@uomosul.edu.iq

Abstract— This paper discovers IoT (Internet of Things) forensics and how the deep learning is improving the efficiency of digital investigations. With the exponential growing of IoT, effective security measures and protocols are obligatory to protect from cyber risks and threats. However, IoT devices are remain vulnerable to attacks, So, this led us to data breaches, loss of privacy, and other harmful consequences. IoT forensics is investigates and analyzes digital evidences related to IoT devices and then identify the source of cyber-attack. This paper has been discussed the fundamentals' of IoT forensics also the important role it plays in the realm of cybersecurity. Furthermore, this paper explores the different kinds of IoT datasets and how we can automate the analysis of big data by using deep learning. Also, it helps in identify potential sources of evidence, and construct predictive models to prevent future attacks. The paper also shows experiments of two deep learning models, LSTM and RNN, on a binary, 6 class, and 15 class classification. Different evaluation metrics have been used like: precision, recall, F1-score, and ROC which allow investigators to objectively evaluate the forensic model's effectiveness. The Edge-IIoTset dataset developers who used deep neural networks (DNNs) were compared to the research findings, and it was discovered that the RNN model with the given architecture behaved the best on the dataset.

Keywords—IoT Forensics, Digital Evidence, Deep Learning

I. INTRODUCTION

The Internet of Things (IoT) is a rapidly developing area of technology that includes connecting various objects together across networks and the internet. Effective security measures must be implemented to safeguard against cyber risks due to the IoT's exponential growth pace. In spite of the best security measures IoT devices still vulnerable to threats and attacks which can cause data breaches, loss of privacy and other harmful consequences. IoT forensics is a field of digital forensics which is focuses on investigating and analyzing the digital evidences related to IoT devices to identify the source of a security breach or cyber attacks [1]. The mission of IoT forensics is to provide critical information for legal and regulatory requirements, also to help organizations to better understanding the security risks which is related

with IoT devices. In this article, the basics of IoT forensics has been explored as well as the important role of it in cybersecurity world [2]. In IoT Forensics conventional learning can be operative in certain cases but deep learning offers a more powerful and adaptable approaches. Deep learning is a powerful tool that can facilitate this process by automating the analysis of big volumes of data. Deep learning algorithms can be trained to identify the patterns of data, such as anomalies in network traffic or changes in device activities, that may allocate malicious activity. By using deep learning models, investigators can identify potential sources of evidence in more quick and accurate manner, such as compromised devices or networks, and analyze big data to identify relevant information. Additionally, deep learning models can be used to create predictive models that can help investigators anticipate potential security attacks and prevent future threats [3]. Altogether, the combination of IoT forensic and deep learning can considerably enhance the efficiency and effectiveness of digital investigations.

In Section 2 we conduct a literature review on IoT forensics, specifically focus on the use of Artificial intelligence (AI) techniques for example machine learning (ML) and deep learning. Section 3 presents an overview of the general stages of IoT forensics, Section 4 offer different types of IoT forensic tools. In Section 5, a comprehensive analysis of various IoT datasets have been done. After selecting a dataset and completing preprocessing steps, we elaborate our methodology in Section 6. Section 7 show the results of our experiments, and Section 8 concludes the study with a discussion of future research opportunities.

II. RELATED WORK

The implementation of deep learning in IoT forensics has been the topic of several studies over the past few years. Deep learning models have been created by researchers that can look at network data and device activity patterns to find malware. Algorithms based on deep learning may be trained to recognize the type of device and its particular behaviors by assessing device activity, which can be helpful in recognizing possible security issues. Researchers

have also investigated the application of deep learning for modeling predictions in IoT security. Deep learning algorithms may find trends and foresee potential security problems by evaluating massive amounts of data.

A special Convolutional Neural Network (CNN) called Optimal-Deep-Learning-based CNN (ODL-CNN) is suggested in [4] for forensic suspect search (FSS) which is employing IoT technology. An extensive quantitative and qualitative analysis was done to deaminate its efficacy. In the study [5], Autopsy, an open-source program, is used to do the initial forensic analysis of a PinePhone running Ubuntu-Touch OS. The goal of this study is to establish a framework for inspecting and evaluating devices that run the Ubuntu-Touch OS. This research gives digital forensic investigators a step-by-step guide for conducting their professional investigations utilizing PinePhone and/or Ubuntu-Touch OS as the evidence's cause. In order to (1) explain their architectures to investigators, (2) identify pertinent forensic artifacts, and (3) determine the Android and iOS operating systems (3) identify privacy and security issues in general applications found on both OSs, and (4) authenticate the forensic examination on the chosen tools for duplicability and authentication purposes, this research [6] thoroughly analyses both platforms.

IoT forensics are thoroughly investigated in this article [7]. It offers an overview of IoT forensics and contrasts it with traditional digital forensics (DF). A blockchain-based architecture is also recommended for IoT forensics. For an integrated IoT-Cloud computing environment, [8] suggested a conceptual event recording model and design architecture to facilitate digital forensics in IoT. [9] A comprehensive framework for digital forensic readiness (DFR) is offered, and its usefulness is qualitatively assessed. A novel methodology has been presented in order to address the problems in IoT forensics. In [10], this model contains various requirements such as security, integrity, availability, confidentiality, authentication, and non-repudiation. In [11], a new network forensic basis called the Particle-Deep-Framework is proposed for IoT networks. This framework employs Particle-Swarm-Optimization to optimize the hyperparameters of MLP model and enhance its performance. The proposed framework in [12] defines the digital investigation stages for recognizing and tracing attacks activities in IoT networks, and the results show that it has a high performance for determining and tracing cyberattack actions compared to other techniques. In this article [13], the feature dimensionality of large scale IoT network traffic data is reduced using the encoding stage of long-short-term-memory autoencoder (LAE). Additionally, a new dataset called Bot-IoT is presented in [14], which consists of genuine and virtual IoT network traffic, along with numerous types of attacks. This dataset provides a baseline for botnet identification across IoT-specific networks.

In [15], the performance of several popular Machine Learning (ML) methods and a Deep Learning model is evaluated for intrusion detection purposes using the proposed Telemetry dataset in both binary and multi-class

classification problems. Using cutting-edge deep learning techniques like Artificial Neural Networks (ANN), Gated Recurrent Units (GRU), and Long- or Short-Term Memory (LSTM) models, the proposed method in [16] aims to distinguish between malicious Botnet traffic and legitimate traffic. This represents a significant advancement over earlier works. At the end, the use of deep learning in IoT forensics has produced a very encouraging results and holds the promise of considerably enhancing the effectiveness and efficiency of digital investigations.

III. IOT FORENSICS

IoT forensics, sometimes are referred to as Internet of Things forensics, it is the process of investigating and analyzing the digital systems and things that are a part of the IoT to detect the evidence of criminal conduct. Identifying, preserving, and analyzing digital evidence from IoT devices including wearables, industrial machinery and smart home appliances may be necessary. Finding and analyzing digital evidence to uncover any potential criminal conduct involving Internet of Things (IoT) networks or devices is the main goal of IoT forensics. This can help law enforcement agencies and other stakeholders to identify and prosecute individuals or groups involved in illegal IoT behavior[4].

IoT forensics often requires a number of steps. Identification and preservation of prospective digital evidence constitute the initial stage. This could entail grabbing tangible objects or taking forensic photos of digital storage devices. Analyzing the digital evidence is the second step. To do this, data must be extracted from the saved devices or photos and analyzed to find pertinent details about the illicit conduct. To recover erased data or find concealed information, this may need the employment of sophisticated forensic instruments or methods. The evaluation of the data is the third stage. Reporting along with presenting the evidence is the last step. This entails summarizing the investigation's results in a manner that non-technical stakeholders will readily comprehend. To communicate the investigation's findings, this could involve producing reports or visualization.

IV. IOT FORENSICS TOOLS

IoT forensics uses a variety of tools and methods to examine digital evidence from IoT systems and devices. Here are a few instances [18] from Table (I):

TABLE I. IOT FORENSICS TOOLS

IoT Forensics Tool	Description
F-Response	A tool that allows investigators to access and analyze remote systems and storage devices.
Autopsy	A forensic investigation tool that allows investigators to examine digital media in a forensically sound manner.
Volatility	A memory analysis tool that can be used to analyze data from volatile memory in IoT devices.

Wireshark	A network protocol analyzer that can be used to capture and analyze network traffic from IoT devices.
Axiom	A digital forensics tool that can be used to analyze data from mobile devices, including IoT devices.
EnCase	A digital forensics tool that can be used to investigate IoT devices and systems by extracting and analyzing data from storage devices.
OSForensics	A digital forensics tool that can be used to search for and analyze digital evidence on IoT devices.

These tools are designed to support forensically sound identification and analysis of digital evidence from IoT devices and systems by the investigators. The usages of this technologies for IoT forensics may also require some particular training and knowledge.

V. IIOT DATASETS

Industrial applications of machine learning and artificial intelligence are relying more and more on edge IIoT (Industrial Internet of Things) datasets. These databases are generated by sensors and other IoT devices that are set up at the outer boundaries of industrial networks, where data is continuously gathered and analyzed. Edge IIoT datasets are frequently huge and complex, demanding the use of specialist tools and methodologies for effective analysis. They can, however, offer useful insights into manufacturing processes, allowing businesses to organize their procedures and decrease downtime. As the number of IoT devices deployed in industrial settings continues to grow, the importance of edge IIoT datasets for machine learning generally and deep learning especially is only expected to increase. There are several publicly available datasets related to IoT and IIoT cybersecurity as shown in Table (II). IIoT connectivity, security, privacy, machine learning, identification, and reliability are just a few of the many IoT-related challenges that these technologies, platforms, and methods aim to address. Therefore, without a specific scenario or application in mind, a direct comparison between them might not be special.

TABLE II. COMPREHENSIVE ANALYSIS OF IIOT DATASETS

IIoT Dataset	Source & year	Description	Main drawback & Techniques
IoT-23 [5]	UNSW and UC Berkeley 2018	Dataset of network traffic from 23 IoT devices in a controlled environment for intrusion detection	Small, ML
CIC-IDS2018 [6]	CIC at UNB 2018	Dataset of network traffic from IT and IIoT networks in a controlled environment for intrusion detection	simulated dataset, ML, DL
UNSW-	UNSW	Dataset of network traffic from	controlled ,

NB15[5]	2015	various networks, including IoT, in a realistic environment for intrusion detection	ML
Stratosphere IoT[6]	University of Lorraine. 2018	Dataset of network traffic from IoT devices in a realistic environment for intrusion detection	Small, ML
N-Balot[7]	3GPP 2016	Narrowband IoT standard for low-power wide-area networks in IoT applications	Low data rates, ML, DL
Bot-IoT[8]	UCSB 2018	Botnet detection framework for IoT devices using machine learning	Customization, ML, DL
MQTTset[9]	University of Padua 2019	Extensions to the MQTT protocol for enhanced security and reliability in IoT applications	New protocol, ML, DL
Federated IOT[5]	University of Glasgow 2020	Federated learning framework for collaborative machine learning on IoT devices with privacy and security	Large number of devices, Federated learning
X-IIoTID[9]	NIST 2020	Standard for identifying and tracking IoT devices in industrial settings for device heterogeneity and interoperability	New standard, ML, DL
WUSTL-IIOT[10]	Washington University 2017	Research project for developing secure and reliable IoT systems for critical infrastructure	Expertise and resources, ML, DL

IIoT/IIoT cybersecurity tools need realistic data to evaluate efficacy & test AI algorithms. IIoT datasets increase as more devices are deployed & collected, with focus on specific apps (e.g. smart homes, healthcare). Some datasets are publicly accessible, others proprietary. They contain sensor/device data, metadata & vary in size. Choosing the right dataset is important for performance & security, and analysis must be secure/privacy-preserving to protect sensitive information like digital evidences in IIoT forensics.

VI. DATA PREPROCESSING

In this study, a new IIoT dataset has been chosen. Edge-IIoTset is a cybersecurity dataset for IIoT and IIoT applications designed for centralized and federated learning. It includes realistic data from sensors, controllers, and actuators, focusing on the industrial domain, and features several protocols and communication technologies. The dataset also contains various attack scenarios to enable the development and evaluation of cybersecurity solutions. Edge-IIoTset is suitable for intrusion detection, digital forensics, and predictive maintenance in IIoT systems, and it's a valuable resource for researchers and practitioners. The dataset is also suitable for machine learning, deep learning, and federated learning applications and can be used for IIoT forensics to develop and improve IIoT forensic techniques and tools.

Edge-IIoTset is publicly available and contains sensor readings collected from an industrial IoT environment in multiple CSV files. To preprocess the EdgeIoT dataset, follow these steps as shown in figure (1):

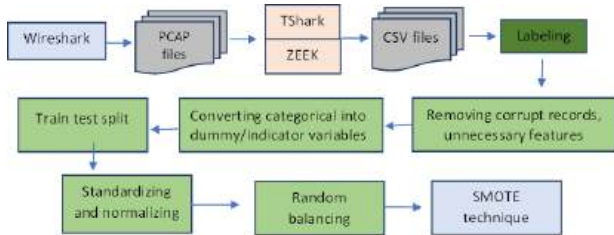


Figure 1. Preprocessing of EdgeIoT dataset

- Load CSV files into a data analysis tool like Python pandas or R.
- Merge CSV files if necessary to combine data from multiple sensors.
- Handle missing values by removing rows with missing values or using mean or regression imputation.
- Normalize or standardize the data depending on the machine learning algorithms.
- Engineer new features like statistical or time-based features from existing sensor readings.
- Split preprocessed data into training and testing sets for machine learning model development.

Overall, Preprocessing steps for the EdgeIoT dataset depend on analysis goals and ML algorithms. Careful consideration is key to ensure data suitability and accurate, reliable ML models.

VII. OVERVIEW OF PROPOSED IOT FORENSICS FRAMEWORK

We can divide our model into four components as shown in figure (2):

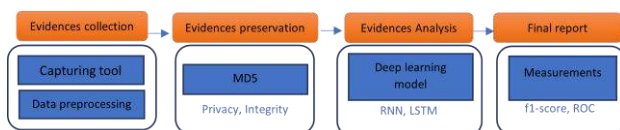


Figure 2. IoT forensics framework

A. Evidences Collection:

IoT devices in the network under investigation are configured in promiscuous mode to capture all network traffic using tools such as Wireshark to collect PCAP files. The PCAP files are then converted to CSV files and preprocessed.

B. Evidences Preservation:

MD5 hashing algorithm ensures the authenticity and integrity of digital evidence by generating a unique digital fingerprint of the original data. Using MD5 to preserve evidence in CSV files helps to ensure the privacy and integrity of the data. Record the MD5 hash value in a secure location.

C. Evidences Analysis:

Deep learning models, such as Long Short-Term Memory (LSTM), and Recurrent Neural Networks (RNN), are increasingly used to analyze digital evidence, such as social media activity and system logs, to identify patterns of malicious activity. If we want to choose any algorithm we should depend on specific analysis goals that we want and also it depend on the nature of the data. The output data from the second stage are used for deep learning algorithms training and testing. The best outputs were produced with 2 hidden layers in both LSTM and RNN with 91, 30, 60, 15 neurons. The hyperparameters used for train and validate the MLP algorithms include epochs of 100, a learning rate of 0.0001, and a batch size of 800. The Adam optimizer was also used with a softmax activation function. Adam optimizer is a popular optimization algorithm that is widely used in deep learning. It is a stochastic gradient descent optimization algorithm that is designed to converge quickly and efficiently. Softmax activation function is used in the output layer of the MLP algorithm. It is a popular activation function used for multi-class classification problems. The softmax function outputs a probability distribution over the classes, ensuring that the sum of the probabilities is equal to one. Overall, these hyperparameters and the use of Adam optimizer with softmax activation function suggest a well-optimized MLP algorithm for the given problem.

D. Final Report:

A deep MLP system is run on data validation to obtain performance measurement. The final report is crucial for IoT forensic models, providing a summary of investigation, methodology, findings, and recommendations. IoT forensic investigations entail analyzing voluminous data from various sources, necessitating appropriate metrics like precision, F1 score, recall, and ROC curve. Precision and recall measure its capability to categorize positive and negative results. F1 score is an overall performance metric combining precision and recall, while the ROC curve shows the model's ability to distinguish between classes. These metrics enable investigators to objectively evaluate the forensic model's effectiveness in the final report, informing stakeholders' decisions. A well-written final report can ensure the investigation's integrity and offer useful insights for future IoT forensic investigations.

VIII. EXPERIMENT RESULTS

LSTM and RNN have been tested on EdgeIoT data with varying numbers of classes (2, 6, and 15). These algorithms are commonly used for time-series analysis to detect patterns over time, which can be useful in the investigations of security threats in IoT.

As shown in table (III), The experiment results show the performance of two different models, LSTM and RNN, on a binary classification task where the classes are "Normal" and "Abnormal". The estimation metrics used are precision, recall, and F1-score, along with the support values. From the results, we can see that both models achieved perfect precision, recall, and F1-score for both

classes, indicating that they were able to correctly classify all instances in the dataset. The macro average and weighted average scores are also perfect, which indicates that the models are well-balanced and perform equally well on both classes. These results suggest that the models are highly accurate and have achieved a very high level of performance on the given task.

TABLE III. 2- CLASS LSTM AND RNN

		Precision	recall	f1-score	support
LSTM	Normal	1.00	1.00	1.00	4816
	Abnormal	1.00	1.00	1.00	25620
	macro avg	1.00	1.00	1.00	30436
	weighted avg	1.00	1.00	1.00	30436
RNN	Normal	1.00	1.00	1.00	4816
	Abnormal	1.00	1.00	1.00	25620
	macro avg	1.00	1.00	1.00	30436
	weighted avg	1.00	1.00	1.00	30436

Table (IV) display the Experiment results on 6-class multiclass classification: both models achieved high scores for some classes, lower for others (e.g., Malware, Scanning). Normal and MITM had perfect scores. Figure (3) display the confusion matrix of RNN. ROC AUC scores indicate good performance in distinguishing between classes, with RNN (0.920) slightly outperforming LSTM (0.905). Overall, models achieved decent performance, with RNN slightly better than LSTM. figure (4) show a graphical representation of ROC, Model accuracy and model loss.

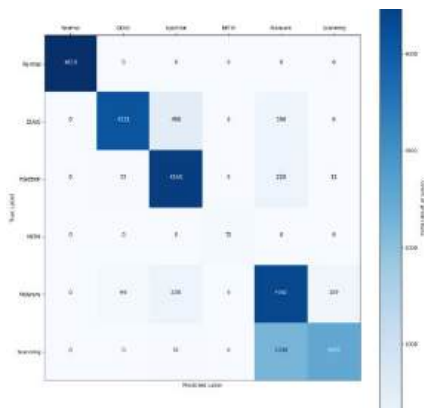


Figure 3. 6-class RNN confusion matrix

TABLE IV. 6- CLASS LSTM AND RNN

		precision	recall	f1-score	support
LSTM	Normal	1.00	1.00	1.00	4816
	DDoS	0.90	0.83	0.87	4816
	Injection	0.78	0.95	0.85	4816
	MITM	1.00	1.00	1.00	72
	Malware	0.77	0.36	0.49	4816
	Scanning	0.63	0.88	0.74	4816
	macro avg	0.85	0.84	0.82	24156
	weighted avg	0.82	0.81	0.79	24156
ROC AUC score= 0.900430303					
RNN	Normal	1.00	1.00	1.00	4816
	DDoS	0.98	0.86	0.91	4816
	Injection	0.86	0.94	0.90	4816
	MITM	1.00	1.00	1.00	72
	Malware	0.62	0.91	0.74	4816
	Scanning	0.94	0.53	0.67	4816
	macro avg	0.90	0.87	0.87	24156
	weighted avg	0.88	0.85	0.84	24156
ROC AUC score: 0.9253719989					

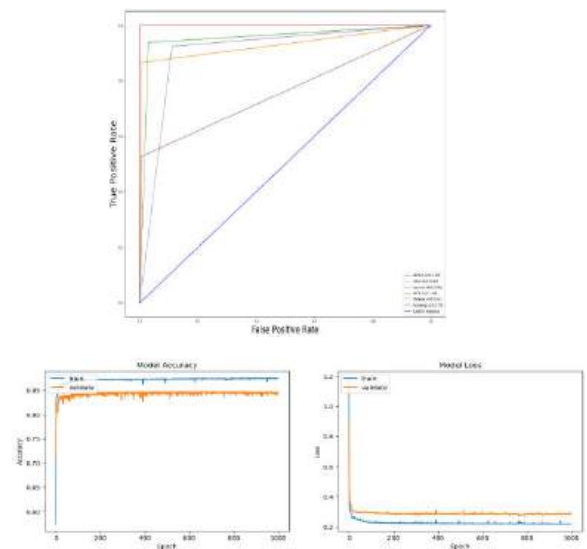


Figure 4. class RNN ROC, Model Accuracy and Loss

In 15 class classification as shown in table (V) looking at the results, we can see that the performance of the two models varies across different classes. For some classes like DDoS_ICMP and DDoS_UDP, both models achieve perfect precision, recall, and f1-scores. For other classes like Fingerprinting and XSS, the performance is much lower, with f1-scores below 0.4 for both models. In general, the RNN model looks to accomplish better than the LSTM model. For example, in the macro and weighted average metrics, the RNN model achieves higher precision, recall, and f1-scores. Additionally, the RNN model achieves a higher ROC AUC score of 0.902 compared to the LSTM model's score of 0.855. Figure (5), (6), (7) show the model accuracy and model loss, confusion matrix, and ROC graphical representation of RNN 15-class model respectively.

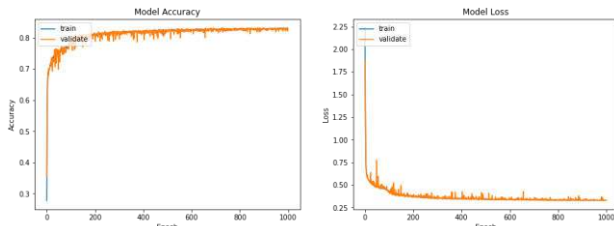


Figure 5. 15-class RNN Model Accuracy and Loss

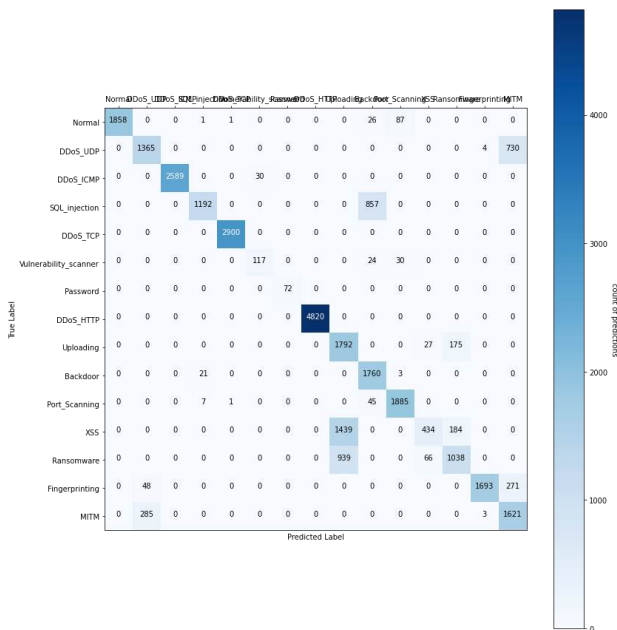


Figure 6. 15-class RNN confusion matrix

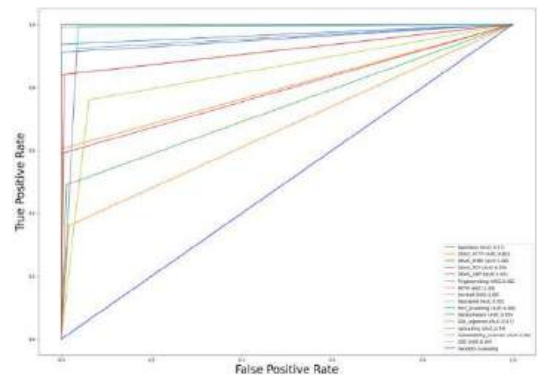


Figure 7. 15-class RNN ROC

Finally, we compare our results with those of the researchers who created the new dataset (Edge-IIoTset). We are focusing on comparing the results of our research with those of the researchers who created the Edge-IIoTset dataset [23] using deep neural networks (DNNs), as DNNs are a core component of our research field. The details are presented in the TABLE V. In result, the RNN models outperform the DNN model consistently across the majority of classes. The RNN model has an average precision and recall value and the greatest ROC AUC score. On the other hand, depending on the class, the DNN model performs very differently. For some classes, such as DDoS_ICMP, DDoS_UDP, and Vuln_scanner, it functions in a manner comparable to the LSTM and RNN models. For classes like Password, Fingerprinting, and XSS, it has much worse recall, precision, and F1 scores. The ROC AUC score shows that when comparing the overall performance of the classifier, the RNN models outperform the LSTM models. Overall, across most classes, the RNN model has the highest recall and precision values and average precision. In conclusion, across most classes, The RNN model has an average precision and recall value and the greatest ROC AUC score. So, the results of our RNN model with the stated architecture are the best for this dataset, according to the TABLE VI.

TABLE V . OUR RNN MODEL ARCHITECTURE

Layer (type)	Output Shape	Param #
simple_rnn (SimpleRNN)	(None, 91)	8463
dense (Dense)	(None, 20)	1840
dense_1 (Dense)	(None, 60)	1260
dense_2 (Dense)	(None, 80)	4880
dense_3 (Dense)	(None, 90)	7290
dense_4 (Dense)	(None, 15)	1365
Total	-	25,098
Trainable	-	25,098
Non-trainable	-	0

TABLE VI. 15- CLASS LSTM AND RNN WITH RESULTS COMPARISON

	LSTM (our model)			RNN (our model)			DNN [23]		
	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score
Normal	1.00	1.00	1.00	1.00	0.94	0.97	1.00	1.00	1.00
Backdoor	0.62	0.86	0.72	0.65	0.99	0.78	0.95	0.86	0.90
DDoS_HTTP	0.58	0.94	0.71	1.00	1.00	1.00	0.76	0.92	0.83
DDoS_ICMP	1.00	0.95	0.97	1.00	0.99	0.99	1.00	0.99	1.00
DDoS_TCP	0.82	0.65	0.72	1.00	1.00	1.00	0.82	1.00	0.90
DDoS_UDP	1.00	1.00	1.00	0.80	0.65	1.00	1.00	1.00	1.00
Fingerprinting	0.25	0.75	0.38	1.00	0.84	0.91	0.59	0.64	0.61
MITM	0.91	1.00	0.95	0.62	0.85	0.72	1.00	1.00	1.00
Password	0.42	0.56	0.48	1.00	1.00	1.00	0.55	0.38	0.45
Port_Scanning	0.64	0.85	0.73	0.94	0.97	0.96	1.00	0.50	0.66
Ransomware	0.82	0.35	0.49	0.74	0.51	0.60	0.73	0.85	0.79
SQL_injection	0.48	0.34	0.40	0.98	0.98	0.58	0.47	0.71	0.57
Uploading	0.50	0.50	0.50	0.43	0.90	0.58	0.67	0.48	0.56
Vuln_scanner	0.85	0.79	0.82	0.80	0.68	0.74	0.96	0.85	0.90
XSS	0.61	0.24	0.35	0.82	0.21	0.34	0.53	0.37	0.43
macro avg	0.70	0.72	0.68	0.85	0.80	0.78	Unknown	Unknown	Unknown
weighted avg	0.76	0.74	0.73	0.87	0.81	0.81	Unknown	Unknown	Unknown
ROC AUC score	0.85502807			0.90229589			Unknown		
Avg Precision	70%			85.2%			80.2%		
Avg recall	71.8%			83.4%			77%		
Avg f1-score	68%			81.1%			77.3%		

IX. CONCLUSION AND FUTURE WORK

Deep learning has potential in IoT forensics by enabling efficient analysis of complex datasets. The use of deep learning models in IoT forensics with edgeIoT dataset has the potential to enhance the accuracy and efficiency of forensics analysis, improve the security and consistency of IoT systems, and facilitate the development of new applications and services. In our case, RNNs may be a more efficient and effective choice. LSTMs are more complex than RNNs, which can make them more difficult to train effectively. The additional complexity of the LSTM may lead to overfitting, resulting in poor generalization performance.

Future work could aim to create more efficient deep learning models that are suitable for training and deployment on resource-constrained devices. Approaches

such as federated learning and edge computing hold promise in addressing these challenges. Furthermore, the integration of additional deep learning techniques, such as CNNs and autoencoders, could be explored for edgeIoT forensics tasks. Federated learning and edge computing could also be utilized to improve privacy and security and mitigate challenges related to limited resources and data availability. Additionally, new techniques can be added to the Evidences preservation stage of our model, instead of relying solely on the MD5 technique, such as blockchain.

REFERENCES

- [1] S. Mrdovic, "IoT Forensics," 2021, pp. 215–229. doi: 10.1007/978-3-030-10591-4_13.
- [2] T. Janarthanan, M. Bagheri, and S. Zargari, "IoT Forensics: An Overview of the Current Issues and Challenges," in Advanced Sciences and Technologies for Security Applications, 2021, pp. 223–254. doi: 10.1007/978-3-030-60425-7_10.

- [3] M. S. Mazhar et al., "Forensic Analysis on Internet of Things (IoT) Device Using Machine-to-Machine (M2M) Framework," *Electronics (Switzerland)*, vol. 11, no. 7, Apr. 2022, doi: 10.3390/electronics11071126.
- [4] M. Elhoseny, M. M. Selim, and K. Shankar, "Optimal Deep Learning based Convolution Neural Network for digital forensics Face Sketch Synthesis in internet of things (IoT)," *International Journal of Machine Learning and Cybernetics*, vol. 12, pp. 3249–3260, 2020.
- [5] Y. Keim, Y. H. Yoon, and U. Karabiyik, "Digital Forensics Analysis of Ubuntu Touch on PinePhone," *Electronics (Basel)*, vol. 10, no. 3, p. 343, 2021.
- [6] F. E. Salamh, M. M. Mirza, S. Hutchinson, Y. H. Yoon, and U. Karabiyik, "What's on the horizon? an in-depth forensic analysis of android and iOS applications," *IEEE Access*, vol. 9, pp. 99421–99454, 2021.
- [7] G. Liang, J. Xin, Q. Wang, X. Ni, and X. Guo, "Research on IoT Forensics System Based on Blockchain Technology," *Security and Communication Networks*, vol. 2022, 2022.
- [8] A. Pichan, M. Lazarescu, and S. T. Soh, "A logging model for enabling digital forensics in iot, in an inter-connected iot, cloud eco-systems," in *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*, 2020, pp. 478–483.
- [9] V. R. Kebande, P. P. Mudau, R. A. Ikuesan, H. S. Venter, and K.-K. R. Choo, "Holistic digital forensic readiness framework for IoT-enabled organizations," *Forensic Science International: Reports*, vol. 2, p. 100117, 2020.
- [10] A. Hambouz, Y. Shaheen, and M. Ababneh, "An Internet of Things (IoT) forensics model using third-party logs-vault," in *International Conference on Data Science, E-learning and Information Systems 2021*, 2021, pp. 143–146.
- [11] N. Koroniotis and N. Moustafa, "Enhancing network forensics with particle swarm and deep learning: The particle deep framework," *arXiv preprint arXiv:2005.00722*, 2020.
- [12] N. Koroniotis, N. Moustafa, and E. Sitnikova, "A new network forensic framework based on deep learning for Internet of Things networks: A particle deep framework," *Future Generation Computer Systems*, vol. 110, pp. 91–106, 2020.
- [13] S. I. Popoola, B. Adebisi, M. Hammoudeh, G. Gui, and H. Gacanin, "Hybrid deep learning for botnet attack detection in the internet-of-things networks," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4944–4956, 2020.
- [14] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset," *Future Generation Computer Systems*, vol. 100, pp. 779–796, 2019.
- [15] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, "TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems," *Ieee Access*, vol. 8, pp. 165130–165150, 2020.
- [16] R. Biswas and S. Roy, "Botnet traffic identification using neural networks," *Multimed Tools Appl*, vol. 80, pp. 24147–24171, 2021.
- [17] L. Sadineni, E. S. Pilli, and R. B. Battula, "ProvNet-IoT: Provenance based network layer forensics in Internet of Things," *Forensic Science International: Digital Investigation*, vol. 43, p. 301441, Sep. 2022, doi: 10.1016/j.fsidi.2022.301441.
- [18] K. Salih and N. Dabagh, "Digital Forensic Tools: A Literature Review," *Journal of Education and Science*, vol. 32, no. 1, pp. 109–124, Mar. 2023, doi: 10.33899/edusj.2023.137420.1304.
- [19] I. Tareq, B. M. Elbagoury, S. El-Regaily, and E. S. M. El-Horbaty, "Analysis of ToN-IoT, UNW-NB15, and Edge-IIoT Datasets Using DL in Cybersecurity for IoT," *Applied Sciences (Switzerland)*, vol. 12, no. 19, Oct. 2022, doi: 10.3390/app12199572.
- [20] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and Adna N Anwar, "TON-IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems," *IEEE Access*, vol. 8, pp. 165130–165150, 2020, doi: 10.1109/ACCESS.2020.3022862.
- [21] S. I. Popoola, "Federated Deep Learning for Botnet Attack Detection in IoT Networks."
- [22] I. Vaccari, G. Chiola, M. Aiello, M. Mongelli, and E. Cambiaso, "Mqttset, a new dataset for machine learning techniques on mqtt," *Sensors (Switzerland)*, vol. 20, no. 22, pp. 1–17, Nov. 2020, doi: 10.3390/s20226578.
- [23] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke, "Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications for Centralized and Federated Learning," *IEEE Access*, vol. 10, pp. 40281–40306, 2022, doi: 10.1109/ACCESS.2022.3165809.



Karam M. Salih Assistant lecturer in Computer Network Dept., Collage of Information technology, Ninevah university, Phd student in University of Mosul.



Najla Badi Ibrahim Assistant prof. in Department of Computer Science, Computer Science and Mathematics College, University of Mosul, Mosul, Iraq

IBDAP 2023, August 25-27, Bangkok, Thailand, 2023

IBDAP 2023

การประชุมวิชาการระดับนานาชาติ
ด้านการวิเคราะห์ข้อมูลขนาดใหญ่และการประยุกต์ใช้ ครั้งที่ 4
The 4th International Conference on
Big Data Analytics and Practices (IBDAP 2023)

Organized by

Big Data Institute (Public Organization)
Digital Economy Promotion Agency
Ministry of Digital Economy and Society

Website IBDAP: <https://www.ibdap.org/>

IBDAP²⁰²³

The 4th International Conference
on Big Data Analytics and Practices

Organizing Committee
Big Data Institute (Public Organization)