

# Hate Speech Detection on Platform X Using Word2Vec and Convolutional Neural Network (CNN)

Adhy Ardana Setyawan<sup>1</sup>, Aditiyawan<sup>1,2</sup>

<sup>1</sup>Universitas Multimedia Nusantara

<sup>2</sup>Badan Riset dan Inovasi Nasional (BRIN)

\*Email aditiyawan@lecturer.umn.ac.id

**Abstract**—Hate speech is a form of communication that demeans or attacks individuals or groups based on certain identities such as race, religion, or gender. This phenomenon is increasingly widespread on social media, especially on platform X (Twitter), which allows for the rapid and massive spread of negative content. This condition raises the need for an automatic detection system to identify hate speech on a large scale. This study aims to design and evaluate a Convolutional Neural Network (CNN)-based classification model to detect hate speech in Indonesian. The word representation process is carried out using the Word2Vec word embedding method, which is able to capture the semantic context of the text. The dataset used is the result of a combination of public data and scraping results, then processed through cleaning, normalization, and stemming stages. The evaluation results show that the developed CNN model is able to achieve an accuracy of 88.44%, with a precision value of 89%, a recall of 91%, and an F1-score of 90% in the class of texts containing hate speech. Based on the confusion matrix, the model successfully classified 664 hate speech data correctly, with 57 cases of misprediction. Meanwhile, for non-hate speech texts, 493 data points were correctly classified, while 83 were misclassified. These findings indicate that the model has the potential to be applied in automated and efficient text-based content moderation systems.

**Index Terms**—Convolutional Neural Network, Hate Speech, Machine Learning, Social Media, Word2Vec

## I. INTRODUCTION

Hate speech refers to expressions of hatred toward individuals or groups based on attributes such as race, religion, gender, or sexuality, aiming to demean, intimidate, or incite hatred [1]. It may appear not only in words but also in symbols, images, or harmful behaviors [2], and varies in impact depending on protected characteristics [3]. Legal responses differ by country, from incitement to violence to broader anti-discrimination laws [4].

Regulating hate speech remains imbalanced with freedom of expression, particularly online [5]. Platforms like X (Twitter) allow real-time sharing, yet their structure—through features like retweets and hashtags—enables the rapid spread of harmful content [5]. X's enforcement policies face criticism for failing to prevent abuse and clearly define platform-user responsibility [6].

In Indonesia, digital hate speech has surged tenfold in the past two years, often targeting religious and ethnic minorities [7]. This raises concerns about online safety, especially for youth, and threatens open, democratic discourse [8]. A relevant technical solution to the rise of hate speech content—given the limits of manual moderation—is the development of automated detection systems [9]. Machine learning offers an effective and scalable approach to identifying such content [9].

Machine learning has become essential in recognizing hate speech on social media [10]. To better capture complex language patterns, deep neural networks and transfer learning have shown superior performance over traditional models [11]. Deep learning models like Convolutional Neural Networks (CNNs) are particularly effective in recognizing nuanced language structures [12]. CNNs, originally designed for grid-like data such as images, can automatically extract key features from raw input, making them useful for tasks like image classification, object detection, and even speech recognition [12].

This study aims to design a CNN-based model for detecting Indonesian-language hate speech. CNN is chosen for its ability to effectively identify hate patterns by extracting relevant local features from text [13]. It can be trained from scratch with relatively small datasets, provided the data distribution is representative. Combining CNN with context-based embeddings like Word2Vec is expected to enhance overall model performance. This approach seeks to develop a classification system that is not only accurate but also linguistically aligned with hate speech characteristics in Indonesian.

## II. THEORETICAL BASIS

### A. Hate Speech

Hate speech is a complex social phenomenon where communication is used to demean, mock, or harm individuals or groups based on identity factors such as religion, ethnicity, or gender [14]. It appears both online and offline in various forms—sarcasm, slurs, misinformation, mockery, and unconstructive criticism—often targeting political or religious identities, especially during times of political conflict [2]. Historically, hate speech has manifested in religious contexts through slander, gossip, and jealousy, affecting both individuals and communities [15].

Hate speech systematically undermines marginalized groups identified by gender, sexual orientation, or religious belief, threatening social cohesion and public trust, particularly in ethnically diverse societies [16]. In politics, it is frequently weaponized to mobilize supporters and attack opponents, often exploiting religious and ethnic identities [15]. This intensifies identity politics and endangers democratic values.

### B. Deep Learning

Deep learning is a subset of artificial intelligence and machine learning that utilizes artificial neural networks with multiple interconnected layers to automatically process and extract patterns from large-scale data [17]. It is highly effective in handling complex, high-dimensional data and has led to major advances in fields such as computer vision, speech recognition, and natural language processing.

Deep learning scales well with increasing data volume and complexity, making it well-suited for big data applications across domains like healthcare, security, industry, and government. Among its core architectures, Convolutional Neural Networks (CNNs) are widely used for tasks such as image recognition, classification, and detection [13].

### C. Convolutional Neural Network

Convolutional Neural Networks (CNNs) are a class of deep learning models designed to process data with spatial structures, such as word sequences or images [13]. A typical CNN consists of multiple layers: convolutional, activation, pooling, and fully connected layers. The process begins with a convolution operation, where a filter (or kernel) slides over the input, performing element-wise multiplication followed by summation to produce an output value in the feature map [18].

$$S(i, j) = (I * K)(i, j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I(i+m, j+n) \cdot K(m, n) \quad (1)$$

Here  $S(i, j)$  denotes the convolution result at position  $(i, j)$ , while  $M$  and  $N$  represent the kernel's vertical and horizontal dimensions.

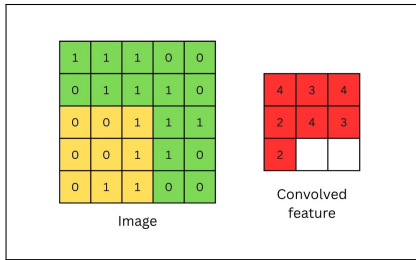


Fig. 1. Convolutional operation  
Source: [18]

Figure 1 illustrates a 2D convolution over a 5×5 input matrix using a 3×3 kernel. At each position, the filter multiplies and sums input values to generate a single output value on the feature map. After convolution, outputs are typically

passed through a non-linear activation function such as ReLU (Rectified Linear Unit), which removes negative values and accelerates network training. The ReLU activation function is designed to eliminate negative values and accelerate training. It is defined as:

$$f(x) = \max(0, x) \quad (2)$$

After convolution, pooling—typically max pooling—is applied to reduce dimensionality and highlight dominant features. Max pooling selects the highest value within a defined window (e.g., 2×2), helping to summarize critical information. Figure 2 visualizes this operation.

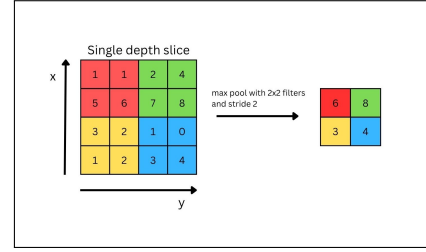


Fig. 2. Max pooling operation  
Source: [18]

As shown in Figure 2, the output is a smaller (2×2) matrix capturing the most significant values from each 2×2 block of the input. After several convolution and pooling layers, CNNs proceed to a fully connected layer, which combines all extracted features into a final vector for classification.

In binary classification tasks, the sigmoid activation function is commonly used in the output layer:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (3)$$

This function converts prediction scores into probabilities between 0 and 1, indicating class membership. The output size of a convolutional layer (without padding) is determined by:

$$\text{Output Size} = \left( \frac{n - f}{s} + 1 \right) \quad (4)$$

During training, CNNs optimize weights by minimizing a loss function. For binary classification, the most widely used is binary cross-entropy, defined as:

$$\mathcal{L} = -[y \cdot \log(\hat{y}) + (1 - y) \cdot \log(1 - \hat{y})] \quad (5)$$

### D. Word2Vec

Word2Vec is a neural network-based model that transforms words into numerical vectors, capturing their meaning and relationships in a way that enables computers to process and understand language for various NLP tasks [19]. Its primary function is to represent similar words with similar vectors, allowing semantic similarity and relationships such as synonyms or analogies to be modeled effectively [20].

Word2Vec uses distributed representation, where a word's meaning is inferred from its surrounding context. It consists of two main architectures: Continuous Bag of Words (CBOW), which predicts a target word from surrounding context words, and Skip-Gram, which predicts context words from a given target word. Both use a shallow neural network where learned weights become the word embeddings [19]. Training efficiency is improved through techniques like negative sampling and hierarchical softmax.

Compared to earlier text representation methods like Bag of Words (BoW) or TF-IDF, Word2Vec offers several advantages:

- It captures both semantic and syntactic meaning, considering word order and context.
- It produces dense and low-dimensional vectors (e.g., 100–300 dimensions) that are computationally efficient.
- It reduces sparsity, unlike BoW or TF-IDF, making it more stable for downstream learning tasks.

#### E. Evaluation Metrics

Evaluasi terhadap performa model klasifikasi merupakan aspek fundamental dalam sistem pembelajaran mesin [?]. Tujuan dari evaluasi ini adalah untuk mengukur seberapa baik model dapat melakukan prediksi terhadap data uji yang tidak pernah digunakan dalam proses pelatihan. Dalam praktiknya, metrik evaluasi digunakan sebagai alat ukur terhadap kinerja model, baik dari segi akurasi keseluruhan maupun kemampuan dalam mengenali kelas tertentu secara spesifik [?]. Empat metrik evaluasi yang paling umum digunakan dalam klasifikasi adalah *accuracy*, *precision*, *recall*, dan *F1-score*. Keempat metrik tersebut memberikan sudut pandang yang berbeda dan saling melengkapi untuk menilai kualitas hasil prediksi.

1) *Accuracy*: Accuracy is the most basic evaluation metric used to measure the proportion of correct predictions over the total number of predictions made [?]. It provides a general overview of how often the model predicts the correct label. The formula for accuracy is as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

Where TP (True Positive) and TN (True Negative) represent correct predictions for positive and negative classes, while FP (False Positive) and FN (False Negative) indicate misclassifications. For example, with 40 TP, 50 TN, 10 FP, and 5 FN, the accuracy is:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{90}{105} \approx 0.857 \text{ (85.7\%)}$$

2) *Precision*: Precision measures how many of the predicted positive instances are actually true positives. This metric is especially important in situations where false positives must be minimized [?]. The formula is given as:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (7)$$

Using the same example data, where there are 40 true positives and 10 false positives, the calculation becomes:

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{40}{40 + 10} = \frac{40}{50} \approx 0.8 \text{ (80\%)}$$

This result indicates that 80% of all positive predictions made by the model are correct, reflecting a good level of trust in positive classification results.

This result indicates that 85.7% of the model's predictions were correct. However, accuracy can be misleading in imbalanced datasets—where one class significantly outweighs another—since high accuracy may still mask poor performance on the minority class.

3) *Recall*: Recall, also known as sensitivity, measures the model's ability to identify all actual positive instances. This metric is particularly important when missing positive cases (false negatives) is considered critical [?]. The formula is:

$$\text{Recall} = \frac{TP}{TP + FN} \quad (8)$$

Given 40 true positives and 5 false negatives, the calculation is:

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{40}{40 + 5} = \frac{40}{45} \approx 0.889 \text{ (88.9\%)}$$

This means the model successfully identified nearly 89% of all actual positive instances. A high recall value indicates strong detection performance, though it may sometimes come at the cost of lower precision.

4) *F1-Score*: F1-score is the harmonic mean of precision and recall. It is particularly useful when dealing with imbalanced datasets, offering a trade-off between minimizing false positives and false negatives [?]. The formula is:

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (9)$$

Using the previously calculated values:

$$\text{F1-score} \approx 2 \cdot \frac{0.8 \cdot 0.889}{0.8 + 0.889} \approx 0.841$$

An F1-score of 84.1% indicates that the model achieves a good balance between identifying relevant positive instances and avoiding false alarms. This metric is widely used for model comparison in both research and real-world applications due to its robustness in evaluating classification performance.

### III. RESEARCH METHODOLOGY

#### A. Literature Review

In this stage, a literature review was conducted to gather insights from previous studies related to hate speech and machine learning algorithms used for classification. The review included scientific journals and articles published between 2018 and 2024, from both national and international sources. This process served as a foundational step to strengthen the conceptual understanding necessary for the study's development.

Based on the findings, Convolutional Neural Network (CNN) was selected as the primary classification model. For feature representation, the Word2Vec embedding technique was used to capture contextual semantic relationships between words. Hyperparameter tuning was performed using Optuna,

an efficient optimization framework based on Bayesian methods. Model performance was evaluated using metrics such as accuracy, precision, recall, F1-score, and the confusion matrix, providing a comprehensive view of the classification effectiveness.

### B. Data Collection and Analysis

The dataset used in this study was sourced from GitHub and developed by Muhammad Okky Ibrohim and Indra Budi in response to the widespread use of hate speech and offensive language on Indonesian-language Twitter [21]. Hate speech poses serious risks of inciting social conflict, discrimination, and even physical violence. The dataset is multi-labeled, containing annotations related to both hate speech and abusive language.

The annotation process was conducted in two stages. The first stage classified tweets as hate speech, abusive language, or neither. The second stage identified the target, category, and level of hate for tweets labeled as hate speech. Annotation involved 30 native Indonesian-speaking Twitter users, selected based on demographic and linguistic criteria to ensure objectivity and high data quality. Annotators represented diverse backgrounds in age, education, ethnicity, religion, and profession, aiming to minimize cultural and social bias.

Tweets included in the final dataset met agreement thresholds: full agreement was required in the first stage, and majority voting was applied in the second stage. The dataset was constructed by combining results from previous studies and seven months of Twitter data crawling, using keywords developed through linguistic studies and consultation with sociolinguistic experts. The classification approach adopted multi-label text classification, employing algorithms such as Naive Bayes, Support Vector Machine, and Random Forest Decision Tree, paired with data transformation techniques like Binary Relevance, Label Power-set, and Classifier Chains.

Features used included word and character n-grams, orthographic features, sentiment lexicons, and profanity lists. Experimental results showed that the combination of word unigrams, Random Forest, and Label Power-set yielded the highest performance 77.36% accuracy for detecting hate speech and abusive language. However, performance dropped to 66.12% when detecting additional dimensions such as target, category, and hate level, due to label complexity and data imbalance, leading to dominant false negatives.

To address these issues, the authors recommended hierarchical multi-label classification and the integration of semantic features like word embeddings to improve contextual understanding. The dataset, along with detailed annotation guidelines, has been openly published to support further research on Indonesian-language hate speech and abusive language detection.

### C. Data Processing

In this stage, data preprocessing was carried out to prepare the dataset for model training. The process included removing duplicates, handling missing values, and cleaning the text by

eliminating URLs, hashtags, Unicode characters, punctuation, repeated words or characters, excessive spaces, and emojis. Additional steps included slang word normalization and stemming, ensuring the data aligns with model requirements.

### D. Model Design

After preprocessing, the next step involved designing a Convolutional Neural Network (CNN) model aimed at effectively detecting hate speech on the X platform. The model development phase included several key steps: importing libraries, tokenization, data splitting, embedding, model training, hyperparameter tuning, and evaluation.

Hyperparameter tuning is a crucial process in machine learning, as it significantly influences model performance [22]. Hyperparameters—such as learning rate, number of neurons, number of layers, and batch size—are predefined before training and not learned from the data. Selecting optimal values ensures the model generalizes well to unseen data.

In this study, Optuna was employed as an efficient and flexible framework for hyperparameter optimization. Unlike conventional methods like random search, Optuna leverages Bayesian optimization through the Tree-structured Parzen Estimator (TPE) algorithm, which improves efficiency by modeling the distribution of promising hyperparameter values based on prior trials.

Optuna has demonstrated superior tuning efficiency and the ability to discover optimal configurations across diverse machine learning tasks. According to Akiba et al. (2019), in "Optuna: A Next-generation Hyperparameter Optimization Framework", the use of Optuna leads to better model performance in a shorter time compared to other methods [23]. Hence, Optuna contributes significantly to the effectiveness and efficiency of predictive model development.

### E. Evaluation

In the evaluation phase, the trained model was tested using a validation dataset to assess its performance in detecting hate speech on the X platform. The evaluation employed common performance metrics, including accuracy, precision, recall, and F1-score. Additionally, a confusion matrix was used to analyze misclassifications and understand how the model distributed its predictions across the actual classes.

## IV. RESULTS AND DISCUSSION

### A. Dataset Description

The dataset consists of 13,169 Indonesian-language tweets that were labeled using a multi-label annotation scheme. The categories include hate speech (HS) and abusive content, with sub-labels for more specific attributes such as religion, race, gender, and others.

The dataset was collected by combining pre-existing data with tweets scraped using Twitter Search API. It includes both hate and non-hate speech samples and was reviewed to ensure consistency and validity before training. Table I provides sample rows from the annotated dataset.

Label 1 indicates hate speech or abusive content, while 0 represents non-hate speech.

TABLE I  
EXAMPLE OF RAW DATASET ENTRIES

Tweet	Label
"Disaat semua cowok berusaha melacak perhatian gue. loe lantas remehkan perhatian yg gue kasih khusus ke elo. basic elo cowok bego !"	1
"RT USER: USER siapa yang telat ngasih tau elu? edan sarap gue bergaul dengan cigax jifla calis sama siapa noh licew juga"	1
"Kadang aku berfikir, kenapa aku tetap percaya pada Tuhan padahal aku selalu jatuh berkali-kali..."	0
"USER USER AKU ITU AKU KU TAU MATAMU SIPIT TAPI DILIAT DARI MANA ITU AKU"	1
"USER USER Kaum cebong kapir udah keliatan dongoknya dari awal tambah dongok lagi hahahah"	1

### B. Preprocessing

1) *Data Labeling*: All individual hate speech and abusive indicators were merged into a single binary label named *label*, where '1' represents negative content (hate or abusive speech) and '0' denotes neutral or non-hate content. This simplification facilitates binary classification and reduces label complexity.

2) *Text Cleaning*: Text cleaning included lowercasing, removal of URLs, hashtags, user mentions, non-alphanumeric characters, and excessive whitespace. Furthermore, slang words were normalized using an external slang dictionary to standardize informal expressions commonly found in social media text.

3) *Stemming*: Stemming was conducted using the *Sastrawi* library to convert inflected words into their root forms. The result was stored in a new column called *stemmed*.

### C. Model Design

The model was built using a Convolutional Neural Network (CNN) with an initial embedding layer trained using Word2Vec vectors. The training pipeline included:

- Tokenization and sequence padding
- Data splitting into training, validation, and testing subsets
- Word2Vec training on preprocessed corpus
- CNN architecture with Conv1D, Dropout, Dense layers
- Use of early stopping to prevent overfitting

1) *Hyperparameter Optimization*: Optuna was employed to optimize hyperparameters such as the number of filters, kernel size, dropout rate, learning rate, and batch size. The best configuration achieved through this process was:

- Filters: 160
- Kernel Size: 7
- Dropout Rate: 0.2013
- Learning Rate: 0.00056
- Batch Size: 32

### D. Model Testing and Evaluation

1) *Testing Results*: Testing was conducted using the unseen test data. The model predicted binary outcomes based on sigmoid activation outputs. The training history is illustrated in Fig. 3, showing trends in accuracy and loss.

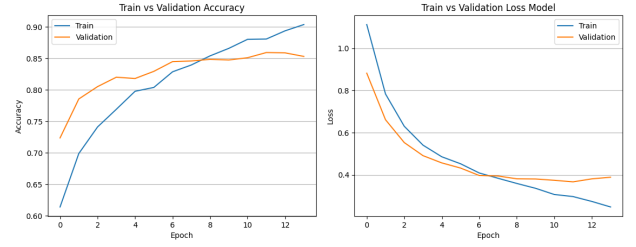


Fig. 3. Training and Validation Accuracy and Loss

TABLE II  
EVALUATION METRICS ACROSS SPLIT SCENARIOS

Split	Class	Precision	Recall	F1-score	Accuracy
80:20	Non-HS	0.88	0.86	0.87	0.88
	HS	0.89	0.91	0.90	
70:30	Non-HS	0.90	0.80	0.85	0.87
	HS	0.85	0.92	0.89	
60:40	Non-HS	0.88	0.78	0.83	0.85
	HS	0.84	0.92	0.88	

2) *Evaluation Metrics*: Table II presents the model performance across three data split scenarios.

The model achieves its highest accuracy (0.88) on the 80:20 split, indicating that a larger training portion improves generalization.

3) *Impact of Word2Vec*: The impact of using Word2Vec embeddings is shown in Table III. It clearly enhances all performance metrics.

TABLE III  
COMPARISON: WITH VS. WITHOUT WORD2VEC

Method	Class	Precision	Recall	F1-score	Accuracy
Without	Non-HS	0.86	0.68	0.76	0.81
	HS	0.78	0.91	0.84	
With	Non-HS	0.88	0.86	0.87	0.88
	HS	0.89	0.91	0.90	

4) *Confusion Matrix*: The confusion matrix in Fig. 4 further confirms the model's strong capability to distinguish between hate and non-hate speech, with low false positives and false negatives.

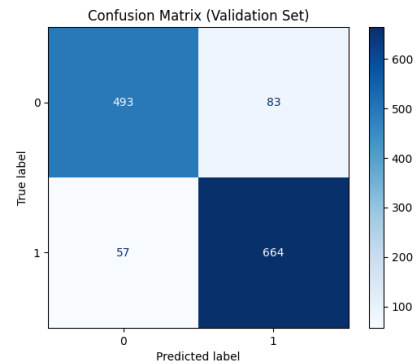


Fig. 4. Confusion Matrix on Validation Set

The CNN model integrated with Word2Vec embeddings performs effectively in detecting Indonesian hate speech, with robust metrics across evaluation scenarios. This method demonstrates a practical approach for automated moderation in social platforms.

## V. CONCLUSIONS AND RECOMMENDATIONS

### A. Conclusions

This research successfully developed a model to detect hate speech in the Indonesian language using a Convolutional Neural Network (CNN) algorithm combined with Word2Vec embedding. The model training process employed the best combination of hyperparameters obtained through tuning, including 160 filters, a kernel size of 7, a dropout rate of 0.2013, a learning rate of 0.00056, and a batch size of 32. Model performance was evaluated using a dataset split scenario with an 80:20 ratio between training and testing data.

The results show that the proposed model achieved an overall accuracy of 88.44%, with balanced classification performance across both text classes. For the non-hate speech class (positive label), the model attained a precision of 88%, a recall of 86%, and an F1-score of 87%. On the other hand, for the hate speech class (negative label), the model demonstrated even higher performance, with a precision of 89%, a recall of 91%, and an F1-score of 90%.

These evaluation results indicate that the model exhibits stable and consistent classification performance, despite a slight imbalance in the number of samples per class. Therefore, this model can be effectively utilized as part of an automated classification system in text-based content moderation applications.

### B. Recommendations

- 1) **Increasing the Dataset Size:** Expanding the dataset with more diverse and extensive data is necessary to improve the model's accuracy in detecting hate speech. A larger and more representative dataset will enhance the model's generalization ability, allowing it to identify various hate speech variations, including emerging types due to social and cultural changes.
- 2) **Experimenting with More Complex CNN Architectures:** This study employed a simple Convolutional Neural Network (CNN) architecture. Exploring more complex CNN designs is recommended to potentially achieve better performance.
- 3) **Exploring Alternative Algorithms:** Although this research used CNN, future work should consider other methods such as transformers or hybrid models combining CNN with other algorithms for further improvement.

## ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to Universitas Multimedia Nusantara (UMN) for providing the support and resources that made this research possible. Special appreciation is extended to the experts and annotators who contributed their knowledge and effort in validating the hate

speech dataset and labeling process, which are fundamental to the development of this detection model. Their expertise and dedication significantly enhanced the accuracy and reliability of this research on hate speech detection using Word2Vec and Convolutional Neural Network.

## REFERENCES

- [1] S. Vilar-Lluch, "Understanding and appraising 'hate speech'," *Journal of Language Aggression and Conflict*, vol. 11, pp. 279–306, Sep. 2023.
- [2] H. Watanabe, M. Bouazizi, and T. Ohtsuki, "Hate speech on Twitter: A pragmatic approach to collect hateful and offensive expressions and perform hate speech detection," *IEEE Access*, vol. 6, pp. 13825–13835, 2018.
- [3] K. Gelber, "Differentiating hate speech: A systemic discrimination approach," *Critical Review of International Social and Political Philosophy*, vol. 24, pp. 393–414, Jun. 2021.
- [4] E. Fino, "Defining hate speech," *Journal of International Criminal Justice*, vol. 18, pp. 31–57, Mar. 2020.
- [5] Z. Liu, "Online hate speech on Twitter from the perspective of pragmatics," *International Journal of Social Sciences and Public Administration*, vol. 4, pp. 322–326, Aug. 2024.
- [6] N. Paradis, M. A. Knoll, C. Shah, C. Lambert, G. Delouya, H. Bahig, and D. Taussky, "Twitter," *American Journal of Clinical Oncology*, vol. 43, pp. 442–445, Jun. 2020.
- [7] S. Riyadi, A. D. Andriyani, A. M. Masyhur, C. Damarjati, and M. I. Solihin, "Detection of Indonesian hate speech on Twitter using hybrid CNN-RNN," in *Proc. 2023 Int. Conf. on Information Technology and Computing (ICITCOM)*, IEEE, Dec. 2023, pp. 352–356.
- [8] N. M. Andini, Y. Findawati, I. R. I. Astutik, and A. Eviyanti, "Implementasi convolutional neural network (CNN) untuk mendeteksi ujaran kebencian dan emosi di Twitter," *SMATIKA JURNAL*, vol. 14, pp. 314–325, Dec. 2024. [Online]. Available: <https://jurnal.stiki.ac.id/SMATIKA/article/view/1346>
- [9] N. S. Mullah and W. M. N. W. Zainon, "Advances in machine learning algorithms for hate speech detection in social media: A review," pp. 88364–88376, 2021.
- [10] P. Kagne, "Political hate speech detection using machine learning," *International Journal of Scientific Research in Engineering and Management*, vol. 07, pp. 1–11, Oct. 2023.
- [11] Y. Zhou, Y. Yang, H. Liu, X. Liu, and N. Savage, "Deep learning based fusion approach for hate speech detection," *IEEE Access*, vol. 8, pp. 128923–128929, 2020.
- [12] C. D. Putra and H.-C. Wang, "Advanced BERT-CNN for hate speech detection," *Procedia Computer Science*, vol. 234, pp. 239–246, 2024.
- [13] J. Fan, C. Ma, and Y. Zhong, "A selective overview of deep learning," *Statistical Science*, vol. 36, May 2021.
- [14] A. Tontodimamma, E. Nissi, A. Sarra, and L. Fontanella, "Thirty years of research into hate speech: Topics of interest and their evolution," *Scientometrics*, vol. 126, pp. 157–179, Jan. 2021.
- [15] W. W. Utami and D. Darmaiza, "Hate speech, agama, dan kontestasi politik di Indonesia," *Indonesian Journal of Religion and Society*, vol. 2, pp. 113–128, Dec. 2020.
- [16] D. C. C. H. Olajide Muili Folaranmi, "The role of adult and non-formal education in the eradication of hate speech as a catalyst for national disintegration in Nigeria," *Journal of Education and Practice*, Mar. 2019.
- [17] A. Shrestha and A. Mahmood, "Review of deep learning algorithms and architectures," *IEEE Access*, vol. 7, pp. 53040–53065, 2019.
- [18] R. Refianti, A. Benny, and R. Poetri, "Classification of melanoma skin cancer using convolutional neural network," *International Journal of Advanced Computer Science and Applications*, vol. 10, 2019.
- [19] S. Sivakumar, L. S. Videla, T. R. Kumar, J. Nagaraj, S. Itnal, and D. Hariitha, "Review on Word2Vec word embedding neural net," pp. 282–290, Sep. 2020.
- [20] D. Jatnika, M. A. Bijaksana, and A. A. Suryani, "Word2Vec model analysis for semantic similarities in English words," *Procedia Computer Science*, vol. 157, pp. 160–167, 2019.
- [21] M. O. Ibrohim and I. Budi, "Multi-label hate speech and abusive language detection in Indonesian Twitter," in *Proc. of the Third Workshop on Abusive Language Online*, Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 46–57. [Online]. Available: <https://www.aclweb.org/anthology/W19-3506>

- [22] S. Shekhar, A. Bansode, and A. Salim, “A comparative study of hyper-parameter optimization tools,” Jan. 2022. [Online]. Available: <http://arxiv.org/abs/2201.06433>
- [23] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, “Optuna,” in *Proc. of the 25th ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining*, ACM, Jul. 2019, pp. 2623–2631.