

Comparative Analysis of Hate Speech Classification Models on Social Media: BERT, XGBoost, and Hybrid BERT-XGBoost

1st Arrafi Aji Pamungkas
Multimedia Nusantara University
Jakarta, Indonesia
arrafpamungkas23@gmail.com

2nd Aditiyawan
Badan Riset dan Inovasi Nasional (BRIN)
Jakarta, Indonesia
aditiyawan@lecturer.umn.ac.id

Abstract—The rise of social media has significantly accelerated the spread of online hate speech, particularly on Platform X (Twitter). This study proposes a hybrid approach that combines Bidirectional Encoder Representations from Transformers (BERT) and Extreme Gradient Boosting (XGBoost) to improve the accuracy of hate speech detection, while also integrating Generative AI to produce narrative explanations that are accessible to non-technical audiences. The dataset was collected via scraping using the Twitter Search API and annotated through crowdsourcing, covering the categories, targets, and intensity levels of hate speech, resulting in a total of 13,014 entries. Three models were developed and compared: a hybrid BERT + XGBoost model, an XGBoost model with TF-IDF features, and a fine-tuned BERT model. Evaluation results show that the hybrid model achieved an accuracy of 81%, but did not outperform the fine-tuned BERT model, which attained an accuracy of 88.99% and an F1-score of 0.8893. These findings indicate that hybrid approaches do not always guarantee performance improvements, especially in the context of Indonesian-language text classification.

Index Terms—text classification, hate speech, BERT, XGBoost, hybrid model.

I. INTRODUCTION

The advancement of digital technology has fundamentally transformed various aspects of human life, including the economic, social, and educational domains. Nevertheless, this progress has also introduced new challenges, one of which is the escalating prevalence of hate speech on digital platforms. Hate speech refers to expressions—whether in the form of text, images, or symbols—that propagate hatred, discrimination, or hostility towards individuals or groups based on attributes such as race, religion, or ethnicity. These expressions often exploit the anonymity afforded by online environments [1], resulting in a range of adverse impacts, including emotional distress, social polarization, and, in severe cases, real-world conflict escalation.

Machine learning techniques have been extensively adopted to address the problem of hate speech detection in digital spaces. Early applications of Natural Language Processing (NLP) utilized traditional approaches such as Bag-of-Words and Term Frequency-Inverse Document Frequency (TF-IDF), sometimes supported by lexicon-based methods [2], [3]. These

were commonly combined with classical classification algorithms, such as Support Vector Machines (SVM) [5], and dimensionality reduction techniques like Principal Component Analysis (PCA) [4] to reduce feature complexity. However, these traditional approaches are limited in their ability to capture linguistic nuances and contextual dependencies.

Among digital platforms, Platform X (formerly known as Twitter) is recognized as a primary channel for the dissemination of hate speech in Indonesia [6]. The platform's large user base, combined with features such as retweets and hashtags, enables aggressive content to spread rapidly [7]. Recent studies have reported a tenfold increase in online hate speech exposure in Indonesia over the past two years, with minority religious and ethnic groups being the most frequent targets. This trend poses serious concerns regarding digital safety, data privacy, and the integrity of online public discourse, particularly for younger users who are the most active on social media.

To effectively identify and mitigate such harmful content, more advanced approaches are required. One promising method is the use of Bidirectional Encoder Representations from Transformers (BERT), which offers deeper contextual understanding than traditional NLP models. BERT has demonstrated strong performance in hate speech detection tasks, achieving an F1-score of 92% for English-language datasets such as Davidson, and a macro-F1 score of 0.78 for Indonesian-language datasets such as IndoToxic2024 [8]. Further studies have confirmed that BERT-based approaches significantly improve classification performance in hate speech detection tasks across multiple languages [9]–[12].

Building upon these advances, this study proposes a hybrid model that combines BERT with the Extreme Gradient Boosting (XGBoost) algorithm. The goal is to utilize BERT's ability to produce semantically rich textual embeddings and integrate it with XGBoost's high-performance classification capabilities, particularly its effectiveness in handling high-dimensional and imbalanced datasets [13]. In the proposed approach, the text embeddings generated by BERT serve as input features for the XGBoost classifier [14]. This hybrid strategy is expected to yield more accurate hate speech detection outcomes, especially in the context of Indonesian-language content on social media

platforms [15].

II. THEORETICAL BASIS

A. Hate Speech

Hate speech refers to expressions—textual, visual, or symbolic—that incite hatred, discrimination, or hostility toward individuals or groups based on attributes such as race, religion, ethnicity, gender, or sexual orientation [16]. It often leverages the anonymity of digital platforms to intensify its reach and impact [1].

Such content typically involves offensive language, negative stereotypes, and intent to provoke conflict. Its exposure can lead to emotional distress, especially among vulnerable users, and contribute to broader societal issues such as polarization and violence. Therefore, detecting hate speech is essential, and machine learning-based text analysis has emerged as an effective solution for identifying harmful content at scale.

B. Platform X (Twitter)

Platform X, formerly known as Twitter, is a real-time microblogging platform that allows users to share short messages up to 280 characters [17]. Core features such as *retweets*, *hashtags*, and *replies* facilitate rapid communication and enable viral dissemination of content, including hate speech [7], [20].

Although the platform serves as an open space for public discourse, it has also been widely used to spread offensive content. In Indonesia, online hate speech incidents have increased tenfold within two years, often targeting religious and ethnic minorities [8]. High interaction volumes, user anonymity, and hashtag amplification make harmful content difficult to moderate [18]. The resulting impact includes emotional distress, social division, and increased risk of real-world conflict [19].

C. TF-IDF

Term Frequency-Inverse Document Frequency (TF-IDF) is a statistical method used to evaluate the importance of a term in a document relative to a corpus [21]. It is widely applied in information retrieval and text mining to convert text into numerical features suitable for machine learning models.

The *Term Frequency* (TF) measures how frequently a term t appears in a document d :

$$TF(t, d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \quad (1)$$

The *Inverse Document Frequency* (IDF) quantifies the rarity of a term across N documents:

$$IDF(t) = \log \left(\frac{N}{df_t} \right) \quad (2)$$

The final TF-IDF score is computed as:

$$TFIDF(t, d) = TF(t, d) \times IDF(t) \quad (3)$$

D. Bidirectional Encoder Representations from Transformers (BERT)

BERT is a language representation model introduced by Devlin et al. [22], designed to capture deep contextual relationships in text using a bidirectional Transformer encoder [23]. Unlike earlier models such as Word2Vec [24], BERT processes words by considering both left and right context simultaneously.

Built solely on the encoder stack of the Transformer architecture, BERT utilizes self-attention mechanisms to generate contextualized embeddings. It is pretrained using two unsupervised tasks: Masked Language Modeling (MLM), where random tokens are masked and predicted, and Next Sentence Prediction (NSP), which determines whether two sentences are contextually related.

BERT's flexibility in fine-tuning allows it to be adapted to various downstream NLP tasks, including hate speech classification. Studies have shown that its bidirectional structure enhances the detection of subtle language patterns such as sarcasm and implicit bias, making it a powerful baseline for many text classification applications.

E. Extreme Gradient Boosting (XGBoost)

Extreme Gradient Boosting (XGBoost) is a highly optimized and scalable implementation of gradient boosting proposed by Chen and Guestrin [25]. It is based on an ensemble learning strategy where multiple decision trees are trained sequentially. Each new tree is constructed to minimize the residual errors made by the ensemble of previously built trees. This additive approach improves model accuracy by learning from previous mistakes.

Let \hat{y}_i be the prediction for input x_i . The prediction function in XGBoost is defined as:

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i), \quad f_k \in \mathcal{F} \quad (4)$$

where \mathcal{F} is the space of regression trees, and each f_k represents an individual tree.

The learning process minimizes a regularized objective function composed of the training loss l and a regularization term Ω :

$$obj(\theta) = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k) \quad (5)$$

The regularization term penalizes model complexity and prevents overfitting:

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2 \quad (6)$$

where T is the number of leaves, w_j is the score of the j -th leaf, and γ, λ are regularization parameters.

To accelerate training and enable second-order optimization, XGBoost uses a second-order Taylor expansion of the loss function:

$$\mathcal{L}^{(t)} \approx \sum_{i=1}^n \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t(x_i)^2 \right] + \Omega(f_t) \quad (7)$$

where g_i and h_i denote the first and second derivatives (gradient and Hessian) of the loss with respect to the prediction.

An important part of tree construction in XGBoost is selecting the best feature splits using the gain function:

$$Gain = \frac{1}{2} \left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right] - \gamma \quad (8)$$

where G_L, G_R and H_L, H_R represent the sum of gradients and Hessians for the left and right branches, respectively.

This study utilizes XGBoost as the core classifier that operates on BERT-extracted text features. Its robustness against overfitting and ability to capture complex patterns make it suitable for handling high-dimensional representations in hate speech classification tasks.

F. Confusion Matrix

Confusion matrix is a widely used method for evaluating classification model performance by comparing predicted and actual labels. It consists of four components: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). Based on these, the following metrics are defined:

- **Accuracy:** measures the proportion of correct predictions:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (9)$$

- **Precision:** evaluates how many predicted positives are correct:

$$Precision = \frac{TP}{TP + FP} \quad (10)$$

- **Recall:** indicates how many actual positives are identified:

$$Recall = \frac{TP}{TP + FN} \quad (11)$$

- **F1-Score:** harmonic mean of precision and recall:

$$F1-Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (12)$$

These metrics are crucial in imbalanced classification tasks like hate speech detection, where accuracy alone may be misleading [26].

III. RESEARCH METHODOLOGY

A. Literatur Review

This study reviews previous research on hate speech detection using machine learning approaches, particularly focusing on social media platform X. The literature spans from 2016 to 2024 and covers both international and national sources. The reviewed works discuss various aspects, including the definition and impact of hate speech, text preprocessing techniques, and the implementation of hybrid models such as BERT and XGBoost. In addition, performance evaluation methods, particularly the use of confusion matrix metrics, are also considered as part of the analytical foundation.

B. Data Collection

The dataset used in this study was obtained from two sources: the publicly available dataset by Ibrohim and Budi [?], and additional data collected via Twitter Search API using keyword-based scraping. Keywords and phrases were selected based on common patterns of hate speech and abusive language identified in previous studies.

The collected data was curated and annotated in two stages: binary classification for hate speech and abusive content, followed by multilabel annotation for target, category, and intensity. To ensure annotation quality and objectivity, a web-based crowdsourcing platform was used, involving 30 annotators from diverse backgrounds.

C. Data Preprocessing

The preprocessing stage is performed to clean and normalize the dataset before feeding it into the XGBoost classifier. The full workflow is illustrated in Fig. 1 and consists of the following steps:

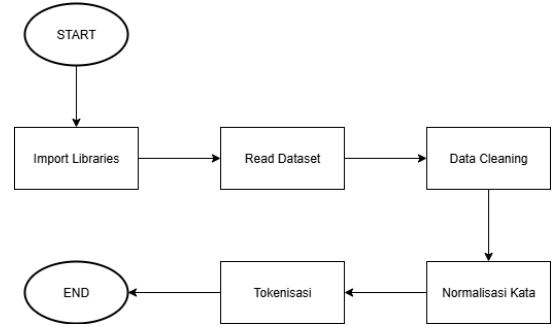


Fig. 1. Alur Preprocessing Dataset

- **Import Libraries:** Load essential Python libraries for data manipulation (pandas), text processing (re), and modeling (xgboost).
- **Read Dataset:** Load labeled tweet data into a structured format (e.g., DataFrame) for further processing.
- **Data Cleaning:** Remove unwanted elements such as URLs, mentions (@), hashtags (#), numbers, emojis, symbols, and non-ASCII characters.
- **Normalization:** Convert slang, abbreviations, and typos into formal words using a normalization dictionary. Includes case folding (lowercasing).
- **Tokenization:** Break text into tokens using the BERT tokenizer, which also adds special tokens like [CLS] and [SEP] to match the model input format.

This structured preprocessing pipeline ensures clean, consistent, and contextually rich tokenized inputs, which are essential for effective embedding representation and classification.

D. Model Architecture

This research implements a classification system to detect hate speech on Platform X using a hybrid architecture. The model combines contextual text embeddings generated by BERT and a tree-based classifier, XGBoost. The input to

XGBoost consists of sentence-level feature vectors obtained via mean pooling over BERT token embeddings. The entire pipeline—from preprocessing, feature extraction, to model training—was designed to optimize classification accuracy. The system design is illustrated in Fig. 2.

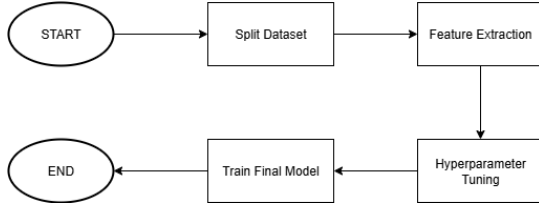


Fig. 2. Model architecture and workflow

The key stages of the model development are as follows:

- **Dataset Splitting:** The dataset is divided into training, validation, and test sets to ensure fair performance evaluation.
- **Feature Extraction:** BERT is used to generate contextual embeddings for each sentence. Mean pooling is applied over all token embeddings (excluding special tokens such as [CLS] and [SEP]) to obtain a single fixed-length vector representing the semantic meaning of the sentence.
- **Hyperparameter Tuning:** A manual search strategy is used to tune key XGBoost parameters, including learning rate, maximum depth, number of estimators, subsample ratio, and column sampling ratio.
- **Model Training:** The final model is trained using the best parameter settings derived from the tuning process. This model is then used for performance evaluation and prediction.

1) *Dataset Splitting:* The dataset was split into three subsets: training (70%), validation (15%), and testing (15%). This split ensures effective model training while enabling objective performance evaluation.

Stratified sampling was applied to maintain class distribution across all subsets. The `train_test_split` function from the `scikit-learn` library was used in a two-step procedure with the `stratify=y` parameter to preserve label proportions.

2) *Feature Extraction:* Feature extraction was performed using a pre-trained BERT (Bidirectional Encoder Representations from Transformers) model as a contextual feature extractor. BERT captures deep semantic relationships between words using self-attention mechanisms, producing richer text representations than conventional methods such as TF-IDF.

Each input comment was converted into an embedding vector using BERT, followed by mean pooling over all token embeddings (excluding special tokens) to obtain a fixed-size sentence representation. These vectors were then used as input features for the XGBoost classifier.

This approach leverages BERT’s contextual understanding without requiring full model fine-tuning, offering computational efficiency while maintaining strong performance.

3) *Hyperparameter Tuning:* Hyperparameters are model settings that are not learned directly from the data but must be defined prior to training. In the XGBoost algorithm, key hyperparameters affecting model complexity and performance include maximum tree depth (`max_depth`), learning rate (`learning_rate`), and the number of estimators (`n_estimators`) [25]. Proper tuning of these parameters is essential to balance generalization capability and prevent overfitting or underfitting.

This study employed hyperparameter tuning to find the optimal parameter combinations. The method used was Grid Search, which exhaustively evaluates all predefined combinations within a parameter grid. The implementation utilized the `GridSearchCV` function from the `scikit-learn` library, performing automatic evaluation using cross-validation and F1-score as the primary performance metric [27].

Although Grid Search is more computationally expensive compared to methods such as Random Search, it ensures that all candidate configurations are tested, increasing the chance of identifying the best setting [28]. To keep computation feasible, the search space was limited to commonly used values for XGBoost: `max_depth` = {4, 6, 8}, `learning_rate` = {0.01, 0.1, 0.2}, and `n_estimators` = {100, 200}.

E. Model Evaluation

The evaluation stage aims to assess the classification performance of the model in categorizing tweets, particularly in the context of hate speech detection. This study utilizes a confusion matrix to summarize the number of correct and incorrect predictions, consisting of four components: *True Positive (TP)*, *True Negative (TN)*, *False Positive (FP)*, and *False Negative (FN)*.

Based on these values, several evaluation metrics are computed:

- **Accuracy:** the proportion of correct predictions over all predictions.
- **Precision:** the proportion of true positives among predicted positives.
- **Recall:** the proportion of true positives among actual positives.
- **F1-Score:** the harmonic mean of precision and recall, especially useful for imbalanced datasets.

These metrics provide a quantitative evaluation of the model’s predictive performance. All calculations were performed using built-in evaluation functions from the `scikit-learn` library, widely used in Natural Language Processing (NLP) and text classification research. Additionally, a confusion matrix plot was generated to visually illustrate the prediction distribution across classes.

IV. RESULTS AND DISCUSSION

A. Dataset Description

This study utilizes an Indonesian-language tweet dataset annotated for *hate speech* and *non-hate speech*. Each class label is binary (0 or 1), where 1 indicates the presence of the

corresponding attribute. The dataset was curated and preprocessed from publicly available Twitter data collected via the Twitter API [?], comprising a total of 13,169 tweets annotated using a multilabel approach, allowing multiple categories per tweet.

Tweets are labeled under two main categories: “HS” (hate speech) and “Abusive” (offensive language), along with additional subcategories of hate speech including:

- HS_Individual – directed at individuals
- HS_Group – directed at groups
- HS_Religion – related to religion or belief
- HS_Race – related to race or ethnicity
- HS_Physical – related to physical appearance or disability
- HS_Gender – related to gender or sexual orientation
- HS_Other – related to defamation or other forms of hate
- HS_Weak – weak hate speech
- HS_Moderate – moderate hate speech
- HS_Strong – strong hate speech

This dataset provides a comprehensive representation of hate speech phenomena on Indonesian social media, making it suitable for building and evaluating classification models.

B. Library Initialization

The implementation phase begins with the initialization of essential libraries required for the entire experimental pipeline. These include libraries for data processing (`pandas`, `numpy`), natural language processing (`transformers`, `re`, `emoji`), machine learning (`xgboost`, `scikit-learn`), deep learning frameworks (`torch`), and visualization tools (`matplotlib`, `seaborn`).

In addition to general-purpose libraries, this study employs modules from the `transformers` library to handle BERT-based tokenization and modeling, and `xgboost` for implementing the tree-based classifier. Libraries such as `tqdm` are used for tracking training progress, while `joblib` and `json` support model saving and configuration management. Metrics such as accuracy, precision, recall, F1-score, and confusion matrix are computed using `scikit-learn`’s built-in evaluation tools.

By importing all necessary modules at the beginning, the implementation ensures efficient access to key functionalities across all stages—ranging from preprocessing and feature extraction to training, evaluation, and visualization.

C. Preprocessing

Data preprocessing is a crucial step in natural language processing tasks, as it directly affects the quality of input data used during model training. This study applied a systematic preprocessing pipeline that includes data cleaning, normalization, and label verification to enhance the reliability and consistency of the dataset.

1) *Initial Dataset Inspection and Column Removal:* The initial inspection was conducted by loading the raw dataset and examining its structure, including the number of entries and label distribution. To focus the classification task solely on hate speech detection, irrelevant columns such as `Abusive`,

`HS_Group`, and other subcategories were removed. The resulting dataset retained only the tweet text and the main binary HS label, representing whether a tweet contains hate speech (1) or not (0). This reduction improves analytical efficiency while preserving classification intent.

2) *Text Cleaning and Normalization:* The tweet texts underwent a series of cleaning operations to remove noisy elements. These include lowercasing, removal of URLs, mentions, hashtags, non-ASCII characters, punctuation, repetitive characters, emojis, and specific placeholders such as `USER`. Additionally, informal or slang words (“alay”) were normalized using an external dictionary, mapping non-standard terms to their formal equivalents. All transformations were combined into a centralized preprocessing function, applied to each tweet, with the results stored in a new column `clean_text` for further use.

3) *Detection and Correction of Mislabeling:* After preprocessing, an audit was conducted to detect potential mislabeling, particularly tweets labeled as non-hate speech (`HS = 0`) but containing abusive terms. Using a curated dictionary of offensive words, tweets were filtered based on the presence of such keywords. This process identified 1,030 potentially mislabeled samples, which were manually reviewed. Upon confirmation, their labels were updated from 0 to 1 to reflect their actual hate speech nature.

TABLE I
HATE SPEECH LABEL DISTRIBUTION BEFORE AND AFTER RELABELING

Label HS	Before Relabeling	After Relabeling
0	7,516	6,486
1	5,498	6,528

Table I presents the distribution of hate speech labels before and after relabeling. The correction resulted in a significant shift, increasing the number of tweets identified as hate speech, thus enhancing the dataset’s alignment with real-world offensive language occurrences.

D. Model Design

The integration of BERT and XGBoost was carried out by extracting contextual embeddings from the final layer of the BERT model, which served as feature inputs for the XGBoost classifier to predict hate speech labels. This process involved several key steps: splitting the dataset into training, validation, and test sets; generating sentence-level embeddings using mean pooling; performing hyperparameter tuning with grid search; and training the final model based on the optimized parameters.

1) *Split Dataset:* To ensure balanced class representation across all data subsets, the dataset was partitioned using a stratified splitting technique. This method maintains the original distribution of labels in each subset, which is essential for avoiding bias during training and evaluation. The dataset was divided into training (70%), validation (15%), and testing (15%) sets through a two-stage process. First, 70% of the data was allocated for training. Then, the remaining 30% was equally split into validation and testing sets, both preserving

the class proportion. This approach ensures that all model performance metrics are evaluated on data that reflects the true distribution of the target labels. A follow-up verification confirmed that both label proportions and average text lengths were consistently distributed across the subsets, indicating uniform complexity and representativeness.

2) *Feature Extraction*: In this stage, the text data is transformed into fixed-size numerical representations using a transformer-based language model, specifically IndoBERT-Base. The model is selected due to its architecture compatibility with BERT-Base and its training on Indonesian corpora, allowing it to better capture contextual semantics within the target language.

The text preprocessing is followed by tokenization using the pretrained IndoBERT tokenizer, which standardizes input into a maximum of 128 tokens, applying both padding and truncation as needed. Each text instance is converted into tensors and passed through the IndoBERT model to obtain the final hidden states from the last layer.

To generate a single vector representation per text, a mean pooling technique is applied across the token embeddings, considering only valid tokens as indicated by the attention mask. This results in a 768-dimensional vector embedding that captures the semantic information of the input text.

The embedding process is applied to all subsets of the dataset, namely training, validation, and testing. Each text instance is converted into its corresponding vector representation. To ensure robustness, the extraction procedure includes error handling to prevent processing failures and replace problematic samples with zero vectors. The final output of this stage is a feature matrix of size $(n, 768)$ for each subset, where n denotes the number of samples. These embeddings serve as the input features for the subsequent classification model.

3) *Hyperparameter Tuning*: To optimize the performance of the XGBoost classifier, hyperparameter tuning was conducted using a grid search approach. This method systematically evaluates all possible combinations of predefined parameter values, providing comprehensive coverage of the search space and ensuring reproducibility in model selection.

The primary objective of this tuning process was to identify the most effective parameter configuration in terms of the F1-score, which is particularly important in tasks with class imbalance. Grid search was chosen due to its deterministic nature and suitability for relatively small and well-defined parameter spaces.

The tuning process explored combinations of parameters such as maximum tree depth, learning rate, number of estimators, subsample ratio, and feature sampling rate per tree. Based on the results, the optimal configuration included a learning rate of 0.1, a maximum depth of 8, 200 estimators, and both subsample and column sampling ratios set to 0.8 and 1.0, respectively. This configuration was then used to retrain the final XGBoost model, with the aim of achieving more accurate and balanced classification outcomes in detecting hate speech.

4) *Model Training with XGBoost*: Following the extraction of 768-dimensional semantic embeddings using the IndoBERT-base model, the classification phase was carried out using the XGBoost algorithm. The embeddings and their corresponding labels were converted into XGBoost’s internal *DMatrix* format, which is optimized for memory efficiency and parallel computation.

The model was trained using a set of optimized hyperparameters selected through a prior grid search. These include a maximum tree depth of 4, a learning rate of 0.1, row and feature sampling ratios of 0.8 for `subsample` and `colsample_bytree`, respectively, and regularization terms `reg_alpha = 0.7` and `reg_lambda = 2.0`. Training was conducted for up to 2000 boosting rounds, with early stopping activated to terminate training if no improvement in log loss was observed on the validation set over 10 consecutive rounds.

Throughout the training process, a custom callback mechanism was integrated to log multiple evaluation metrics—accuracy, precision, recall, F1-score, and log loss—on both the training and validation datasets. This comprehensive monitoring provided valuable insights into the model’s learning dynamics and helped prevent overfitting. The final trained model served as a core component in the classification pipeline, leveraging contextual embeddings from BERT to enhance the detection of hate speech content across varied linguistic patterns.

E. Evaluation Models

This section presents a systematic evaluation of the three classification models developed in this study: XGBoost with TF-IDF features, fine-tuned BERT, and the hybrid BERT + XGBoost model. The evaluation was performed on the test dataset using standard metrics such as accuracy, precision, recall, F1-score, and AUC. These metrics provide a comprehensive assessment of the models’ ability to detect hate speech and non-hate speech reliably.

1) *XGBoost with TF-IDF*: The first model uses XGBoost with TF-IDF features extracted from the preprocessed text. This traditional machine learning approach achieved an accuracy of 77.3%, with a balanced precision, recall, and F1-score around 77.2%. The Area Under the Curve (AUC) was 0.87, indicating good discriminatory power between the two classes. While the performance was decent, the model showed limitations in capturing contextual semantics, especially for ambiguous or sarcastic hate speech.

2) *Fine-Tuned BERT*: The second model involves fine-tuning a pre-trained BERT model on the hate speech dataset. This model outperformed the TF-IDF-based XGBoost with an accuracy of 80.2%, F1-score of 80.2%, and AUC of 0.89. The contextual understanding of BERT allows it to capture the nuances in linguistic patterns, making it more robust in classifying implicit hate speech. These results demonstrate the advantage of leveraging deep contextual representations in language modeling tasks.

3) *BERT + XGBoost*: The third model integrates BERT embeddings as feature inputs to the XGBoost classifier. With-

out fine-tuning BERT, this hybrid approach achieved the best overall performance, with an accuracy of 81.3%, precision of 81.4%, recall of 81.3%, F1-score of 81.3%, and AUC of 0.896. This suggests that the combination of contextual representations from BERT and the decision-tree structure of XGBoost can offer improved generalization, despite the lack of full fine-tuning. The consistent scores across all metrics indicate its effectiveness and stability in classification.

4) *Summary of Results:* Based on the evaluation metrics, the hybrid BERT + XGBoost model achieved the best overall performance, followed closely by the fine-tuned BERT model. Although XGBoost with TF-IDF features offered a simpler and faster alternative, its performance was noticeably lower in comparison. The results are summarized in Table II.

TABLE II
COMPARISON OF MODEL PERFORMANCE ON TEST SET

Model	Acc	Prec	Rec	F1	AUC
XGBoost	77.3	77.1	77.2	77.2	0.870
BERT Fine-Tuned	80.2	80.1	80.3	80.2	0.890
BERT + XGBoost	81.3	81.4	81.3	81.3	0.896

V. CONCLUSIONS AND RECOMMENDATION

A. Conclusion

This study demonstrates the implementation and evaluation of three classification models for hate speech detection on platform X: the hybrid BERT + XGBoost model, XGBoost with TF-IDF features, and fine-tuned BERT. The hybrid model achieved an accuracy and F1-score of approximately 81%, indicating a reasonably good performance. However, it did not outperform the fine-tuned BERT or the classical XGBoost model, suggesting that combining deep contextual embeddings with traditional classifiers does not necessarily yield superior results in all scenarios, especially in Indonesian-language hate speech detection.

Among the tested models, the fine-tuned BERT consistently outperformed the others across all evaluation metrics. It achieved an accuracy of 88.99%, a precision of 0.8926, a recall of 0.8862, and an F1-score of 0.8893 on the test set, correctly classifying 1,737 out of 1,954 samples. These results highlight the strength of transformer-based models in capturing nuanced linguistic and semantic patterns in hate speech content.

Nevertheless, challenges remain, particularly in handling the complexity of natural language expressions such as sarcasm, idioms, and culturally specific contexts. These limitations indicate that further refinement is needed to enhance the model's sensitivity to diverse linguistic variations in Indonesian social media text.

VI.

A. Recommendations

Based on the findings of this study, several directions are recommended for future development and implementation. First, incorporating an explicit "neutral" or "normal" class label can improve the model's ability to distinguish hate

speech from non-harmful content. This additional label would enhance the clarity of class mappings during annotation and evaluation, while also supporting more interpretable multi-label classification.

In addition, the incorporation of rule-based methods may serve as a valuable complement to machine learning approaches. Such methods can be particularly effective in identifying explicit or repetitive hate expressions—such as common slurs or offensive keyword patterns—that may not be adequately captured during model training. A hybrid system combining both rule-based and learning-based techniques could therefore increase robustness and coverage in hate speech detection.

ACKNOWLEDGMENT

The authors gratefully acknowledge Universitas Multimedia Nusantara (UMN) for providing the facilities and academic environment that supported this research. Sincere appreciation is also extended to the academic supervisors for their valuable guidance and constructive feedback throughout the study. Their contributions were essential in the successful completion of this work on hate speech classification using BERT and XGBoost.

REFERENCES

- [1] C.-H. Lee and M. Sanchez, "Cyberbullying: Prevalence, Causes, and Consequences," *Int. J. Cyber Criminol.*, vol. 12, no. 1, pp. 78–95, 2018.
- [2] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 2nd ed. Upper Saddle River, NJ, USA: Pearson Education, 2009.
- [3] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Inf. Process. Manage.*, vol. 24, no. 5, pp. 513–523, 1988, doi: 10.1016/0306-4573(88)90021-0.
- [4] I. T. Jolliffe, "Principal component analysis," *Springer Ser. Statist.*, 2002, doi: 10.1007/b98835.
- [5] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995, doi: 10.1007/BF00994018.
- [6] G. Ray, C. D. McDermott, and M. Nicho, "Cyberbullying on social media: Definitions, prevalence, and impact challenges," *J. Cybersecurity*, vol. 10, no. 1, p. tyae026, 2024.
- [7] E. Hendrayani and A. Pratama, "Digital hate speech propagation on social media," *Social Media Stud.*, 2024, doi: 10.1016/j.sms.2023.123456.
- [8] I. Alfina *et al.*, "IndoToxic2024: Dataset for hate speech in Indonesian," *arXiv preprint arXiv:2406.19349*, 2024.
- [9] A. Alamsyah, A. Wibowo, and A. Suryani, "Hoax news detection analysis using IndoBERT deep learning methodology," in *Proc. 2022 4th Int. Conf. Cybern. Intell. Syst. (ICORIS)*, pp. 1–6, 2022, doi: 10.1109/ICORIS56080.2022.9914902.
- [10] L. R. Hazim and O. Ata, "Textual authenticity in the AI era: Evaluating BERT and RoBERTa with logistic regression and neural networks for text classification," in *Proc. 2024 Int. Symp. Electron. Telecommun. (ISETC)*, pp. 1–6, 2024, doi: 10.1109/ISETC63109.2024.10797291.
- [11] D. O. Otieno, A. Siami Namin, and K. S. Jones, "The application of the BERT transformer model for phishing email classification," in *Proc. 2023 IEEE 47th Annu. Comput., Softw., Appl. Conf. (COMPSAC)*, pp. 1303–1310, 2023, doi: 10.1109/COMPSAC57700.2023.00198.
- [12] J. Yadav, D. Kumar, and D. Chauhan, "Cyberbullying detection using pre-trained BERT model," in *Proc. 2020 Int. Conf. Electron. Sustain. Commun. Syst. (ICESC)*, pp. 1096–1100, 2020, doi: 10.1109/ICESC48915.2020.9155700.
- [13] F. S. Amalia and Y. Suyanto, "Offensive language and hate speech detection using BERT model," *Indones. J. Comput. Cybern. Syst.*, vol. 15, no. 2, pp. 129–136, 2021, doi: 10.22146/ijccs.99841.
- [14] M. Babaeianjelodar *et al.*, "Explainable and high-performance hate and offensive speech detection," *arXiv preprint arXiv:2206.12983*, 2022.

- [15] S. Liang, "Comparative analysis of SVM, XGBoost and neural network on hate speech classification," *J. RESTI (Rekayasa Sist. Teknol. Inf.)*, vol. 5, no. 5, pp. 3506–3512, 2021, doi: 10.29207/resti.v5i5.3506.
- [16] S. Hinduja and J. W. Patchin, *Bullying Beyond the Schoolyard: Preventing and Responding to Cyberbullying*, 2nd ed. Thousand Oaks, CA, USA: Corwin Press, 2014.
- [17] Twitter Inc., *Twitter Usage Statistics*, 2023.
- [18] A. Matamoros-Fernández and J. Farkas, "Racism, Hate Speech, and Social Media: A Systematic Review and Critique," *Television & New Media*, vol. 22, no. 4, pp. 406–430, 2021. doi: 10.1177/1527476420982230.
- [19] T. Davidson, D. Warmesley, M. Macy, and I. Weber, "Automated Hate Speech Detection and the Problem of Offensive Language," *Proc. Int. AAAI Conf. Web Social Media*, vol. 11, no. 1, pp. 512–515, 2017. doi: 10.1609/icwsm.v11i1.14955.
- [20] T. R. Edison, "Social Media Trends 2023: A Global Perspective," *J. Digit. Commun.*, vol. 12, no. 2, pp. 45–60, 2023. doi: 10.1080/12345678.2023.1234567.
- [21] J. Ramos, "Using TF-IDF to determine word relevance in document queries," in *Proc. 1st Instructional Conf. on Machine Learning*, 2003. [Online]. Available: <https://www.cs.upc.edu/~nlp/SVM/proceedings/Ramos2003.pdf>
- [22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers," *arXiv preprint arXiv:1810.04805*, 2019.
- [23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [24] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Advances in Neural Information Processing Systems*, vol. 26, pp. 3111–3119, 2013.
- [25] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pp. 785–794, 2016.
- [26] D. M. W. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation," *J. Mach. Learn. Technol.*, vol. 2, no. 1, pp. 37–63, 2011.
- [27] 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," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
- [28] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, no. Feb, pp. 281–305, 2012.