Institut Riset dan Publikasi Indonesia (IRPI)
MALCOM: Indonesian Journal of Machine Learning and Computer Science
Journal Homepage: https://journal.irpi.or.id/index.php/malcom
Vol. 6 Iss. 1 January 2026, pp: 414-422
ISSN(P): 2797-2313 | ISSN(E): 2775-8575

Comparative Analysis of Naive Bayes and Support Vector Machine for Hate Speech Classification

Rolanda Difandana[1*], Ian Imaduddin[2], Indra[3]

[1,2,3]Information Technology, Universitas Budi Luhur, Indonesia

E-Mail: [1]2411600659@student.budiluhur.ac.id,
[2]2411600667@student.budiluhur.ac.id, [3]indra@budiluhur.ac.id

Received Dec 11th 2025; Revised Dec 26th 2025; Accepted Jan 24th 2026; Available Online Jan 31th 2026
Corresponding Author: Rolanda Difandana

Abstract

This study addresses the increasing need for automated hate speech detection in Indonesia due to the rapid growth of social media and the rise of abusive online content. It compares the performance of Naive Bayes (NB) and Support Vector Machine (SVM) algorithms in classifying Indonesian-language tweets into three categories: hate speech (27.52%), abusive language (34.25%), and neutral content (38.23%). The dataset consists of 13,169 manually annotated tweets collected from Twitter (now X), with moderate class imbalance handled using stratified sampling. Text preprocessing included tokenization, case folding, stopword removal, and stemming using the Nazief–Adriani algorithm, followed by TF-IDF feature extraction with a unigram configuration (min_df=3, max_df=0.95). Both algorithms were evaluated using 10-fold stratified cross-validation with accuracy, precision, recall, and F1-score as performance metrics. Experimental results show that SVM with a linear kernel outperformed NB, achieving an accuracy of 93.28%, precision of 92.45%, and F1-score of 92.89%, compared to NB's accuracy of 84.71%, precision of 83.56%, and F1-score of 84.12%. Although effective, this study is limited to classical machine learning approaches with TF-IDF features and does not incorporate deep learning or contextual embeddings, while still providing practical guidance for algorithm selection in Indonesian hate speech detection systems.

Keywords: Abusive Language, Hate Speech Detection, Naive Bayes, Support Vector Machine, Text Classification

1. INTRODUCTION

The rapid growth of social media platforms in Indonesia has fundamentally transformed the way people communicate and express their opinions. As of 2025, Indonesia ranks among the top five countries globally in terms of social media users, with over 191 million active users across various platforms including Twitter (now rebranded as X), Facebook, Instagram, and TikTok [1]. While social media provides unprecedented opportunities for public discourse and information sharing, it has simultaneously become a breeding ground for hate speech and abusive language [2]. The Indonesian Ministry of Communication and Information Technology reported a 45% increase in reported cases of online hate speech between 2022 and 2025, underscoring the urgency of developing automated detection mechanisms [3].

Hate speech, as defined by the Indonesian Electronic Information and Transactions Law (UU ITE) No. 19 of 2016, encompasses any expression intended to incite hatred or hostility against individuals or groups based on ethnicity, religion, race, and intergroup relations (SARA) [4]. Abusive language, while not necessarily targeting specific demographic groups, includes profanity, insults, and degrading expressions that violate community standards [5]. The manual identification and moderation of such content are impractical given the sheer volume of daily social media posts, necessitating the deployment of machine-learning-based automated classification systems [6]. Text classification using machine learning has emerged as a prominent approach to addressing this challenge. Among the various algorithms available, Naive Bayes (NB) and Support Vector Machine (SVM) have been extensively studied for natural language processing (NLP) tasks [7]. Naive Bayes, grounded in Bayes' theorem and assuming feature independence, offers computational efficiency and simplicity, making it particularly suitable for large-scale text classification [8]. Support Vector Machine, on the other hand, constructs optimal hyperplanes in high-dimensional feature spaces to maximize classification margins, demonstrating robust performance in text categorization tasks [9].

Previous studies have investigated hate speech detection in various languages, including English [10], German [11], and Arabic [12]. In the Indonesian context, several notable studies have been conducted. Ibrohim

and Budi [13] developed a multi-label hate speech detection system for Indonesian Twitter data using various machine learning approaches including SVM, Naive Bayes, and Random Forest, achieving accuracies ranging from 77.36% to 93.45%. Alfina et al. [14] explored hate speech detection in Indonesian social media using lexicon-based approaches combined with machine learning, reporting an accuracy of 82.15%. More recently, deep learning and transformer-based approaches have shown promising results for Indonesian hate speech detection. Kusuma and Chowanda [36] combined IndoBERTweet with BiLSTM for Indonesian hate speech detection on Twitter, demonstrating the potential of pretrained language models. Yulfa et al. [37] applied fine-tuned IndoBERT for hate speech identification during the 2024 Indonesian presidential election, achieving improved detection performance. Susanto et al. [38] introduced the IndoToxic2024 dataset comprising 43,692 entries and achieved a macro-F1 score of 0.78 using IndoBERTweet, highlighting that transformer-based models can capture contextual nuances more effectively. Rokhim et al. [39] proposed an IndoBERT-based ensemble learning approach combining Bi-LSTM and Bi-GRU, achieving 86% accuracy for multi-level multi-label hate speech detection in Indonesian social media. Darmawan et al. [40] implemented IndoBERT for multi-label hate speech detection with data resampling through synonym replacement, achieving 88.23% accuracy.

Despite these advances, a comprehensive comparative analysis between classical machine learning algorithms, specifically NB and SVM, for three-class Indonesian hate speech classification with rigorous evaluation methodology remains insufficiently explored. While deep learning approaches such as IndoBERT have shown strong performance, they require substantial computational resources and large training datasets that may not always be available in practical deployment scenarios [40]. Classical algorithms offer advantages in terms of interpretability, training efficiency, and lower computational requirements, making them relevant for resource-constrained content moderation systems [35]. The novelty of this research lies in three aspects: (1) the systematic three-class classification (hate speech, abusive, neutral) rather than binary classification, (2) the rigorous 10-fold stratified cross-validation with detailed per-class metric comparison between NB and SVM, and (3) the comprehensive error analysis including confusion matrix visualization and identification of specific misclassification patterns in Indonesian hate speech text.

This study aims to address the identified gap by conducting a systematic comparative analysis of Naive Bayes and Support Vector Machine algorithms in classifying hate speech and abusive text in Indonesian language. The research questions guiding this study are: (1) How do NB and SVM perform in classifying Indonesian hate speech and abusive text? (2) Which algorithm demonstrates superior classification performance based on accuracy, precision, recall, and F1-score? (3) What factors contribute to the performance differences between the two algorithms? The findings of this study are expected to contribute to the development of more effective content moderation tools for Indonesian digital platforms.

2. MATERIALS AND METHOD

2.1. Literature Review

The selection of preprocessing techniques and feature extraction methods in this study is grounded in established literature on text classification. Stemming using the Nazief-Adriani algorithm was chosen over alternatives such as the Porter stemmer or the Enhanced Confix Stripping (ECS) algorithm because it was specifically designed for Indonesian morphology and has demonstrated superior performance in reducing Indonesian inflected words to their root forms [21]. Tala [20] showed that the Nazief-Adriani algorithm achieves higher stemming accuracy for Indonesian text compared to language-agnostic approaches.

For feature extraction, TF-IDF was selected over alternative representations such as Word2Vec, FastText, and contextual embeddings (e.g., IndoBERT). While Word2Vec and FastText can capture semantic relationships between words through dense vector representations, TF-IDF remains a strong baseline for classical machine learning classifiers due to its simplicity, interpretability, and effectiveness in high-dimensional sparse feature spaces [22]. Previous comparative studies have demonstrated that TF-IDF combined with SVM achieves competitive performance with word embedding approaches for text classification tasks, particularly when the vocabulary is domain-specific [26]. Furthermore, transformer-based contextual embeddings such as IndoBERT, while achieving state-of-the-art results, require significantly more computational resources for training and inference, making TF-IDF a more practical choice for the classical ML comparison scope of this study [36]. The choice of TF-IDF also enables direct comparison with prior hate speech studies that used similar feature representations [13][14].

Recent studies comparing classical ML and deep learning approaches for hate speech detection provide important context for this research. A comprehensive comparative study by Al-Makhadmeh and Tolba [41] demonstrated that while deep learning models generally outperform classical methods, SVM with TF-IDF features remains competitive for smaller to medium-sized datasets. Kusuma and Chowanda [36] showed that IndoBERTweet combined with BiLSTM outperformed classical approaches, but also noted the trade-off between accuracy and computational cost. These findings justify the continued relevance of comparing classical algorithms as a baseline and practical alternative for Indonesian hate speech detection systems.

2.2. Dataset Collection

The dataset used in this study was obtained from the Indonesian hate speech and abusive language detection dataset originally compiled by Ibrohim and Budi [13], supplemented with additional data collected through the Twitter API using Python's Tweepy library. The combined dataset consists of 13,169 tweets in Indonesian language, collected between January 2023 and December 2025. The data collection criteria included: (1) tweets must be written primarily in Indonesian language, (2) tweets must contain at least five words to ensure sufficient textual content for classification, (3) retweets and duplicate content were excluded to prevent data redundancy, and (4) tweets were collected from trending topics, political discussions, and social issues that commonly generate hateful and abusive content in the Indonesian context [15].

The tweets were manually annotated by three independent annotators with expertise in Indonesian linguistics and social media discourse. The annotation followed a predefined guideline based on the Indonesian UU ITE regulations and the hate speech taxonomy proposed by Davidson et al. [16]. The annotation guidelines specified that: (a) hate speech refers to expressions targeting individuals or groups based on SARA attributes with the intent to incite hatred, (b) abusive language includes profanity, insults, and degrading expressions not necessarily targeting demographic groups, and (c) neutral content refers to tweets not containing hateful or abusive elements. Each tweet was independently classified by all three annotators, and the final label was determined by majority voting. Inter-annotator agreement was measured using Fleiss' Kappa coefficient, yielding a value of 0.78, indicating substantial agreement [17]. Table 1 presents the distribution of the annotated dataset, which shows a moderate class imbalance with neutral content comprising the largest proportion (38.23%).

Table 1. Distribution of Annotated Dataset

Category	Number of Tweets	Percentage (%)
Hate Speech (HS)	3,624	27.52
Abusive (AB)	4,511	34.25
Neutral (N)	5,034	38.23
Total	13,169	100.00

2.3. Text Preprocessing

Text preprocessing is a critical step in natural language processing that transforms raw text data into a structured format suitable for machine learning algorithms [18]. The preprocessing pipeline implemented in this study consists of five sequential stages: case folding, tokenization, stopword removal, stemming, and feature extraction. Each stage was implemented using Python 3.10 with the Natural Language Toolkit (NLTK) and Sastrawi libraries [19].

Case folding converts all characters to lowercase to ensure uniformity. Tokenization segments the text into individual word tokens while removing punctuation, URLs, mentions, hashtags, and non-alphanumeric characters. Stopword removal eliminates common Indonesian words that do not carry significant meaning for classification, using a customized stopword list comprising 758 Indonesian stopwords [20]. Stemming reduces words to their root forms using the Sastrawi stemmer, which implements the Nazief-Adriani algorithm specifically designed for Indonesian morphology [21]. The Nazief-Adriani algorithm was selected over alternative Indonesian stemmers because it handles Indonesian affixation rules more comprehensively, including prefixes (me-, ber-, di-, ke-, se-, pe-), suffixes (-kan, -an, -i), and confixes [21].

Term Frequency-Inverse Document Frequency (TF-IDF) was employed for feature extraction, which quantifies the importance of each term relative to the document and the entire corpus [22]. The TF-IDF vectorizer was configured with the following parameters: ngram_range=(1,1) using unigrams only, max_features=None (no limit on vocabulary size before filtering), min_df=3 (terms appearing in fewer than 3 documents were excluded), max_df=0.95 (terms appearing in more than 95% of documents were excluded), and sublinear_tf=True (applying logarithmic term frequency scaling). These parameters were selected based on preliminary experiments and recommendations from prior hate speech classification studies [13][22]. The TF-IDF weight is calculated using the following equations 1 and 2.

$$TF\text{-}IDF(t,d) = TF(t,d) \times IDF(t) \quad (1)$$

$$IDF(t) = log(N / df(t)) \quad (2)$$

where TF(t,d) represents the frequency of term t in document d, N is the total number of documents in the corpus, and df(t) is the number of documents containing term t [22]. After filtering, the TF-IDF vectorization produced a feature matrix with 15,832 unique terms across the entire corpus.

2.4. Naive Bayes Classifier

Naive Bayes is a probabilistic classifier based on Bayes' theorem that assumes conditional independence among features given the class label [8]. Despite this simplifying assumption, NB has demonstrated competitive performance in text classification tasks. The Multinomial Naive Bayes variant was employed in this study, as it is specifically designed for discrete feature counts such as word frequencies in text data [23]. The implementation utilized scikit-learn's MultinomialNB class with the following hyperparameter configuration: alpha (Laplace smoothing parameter) was optimized through grid search over the values {0.01, 0.1, 0.5, 1.0, 2.0}, and the optimal value of alpha=1.0 was selected based on cross-validation performance. The fit_prior parameter was set to True, allowing the classifier to learn class prior probabilities from the training data [24].

The classification is performed by computing the posterior probability of each class given the observed features and selecting the class with the highest probability. The posterior probability is calculated as follows, as equation 3.

$$P(c|d) = P(c) \times \prod P(wi|c) / P(d) \quad (3)$$

where P(c|d) is the posterior probability of class c given document d, P(c) is the prior probability of class c, P(w_i|c) is the likelihood of word w_i given class c, and P(d) is the evidence. Laplace smoothing with alpha=1 was applied to handle zero-probability issues for unseen words [24].

2.5. Support Vector Machine

Support Vector Machine is a supervised learning algorithm that constructs optimal separating hyperplanes in high-dimensional feature spaces to maximize the margin between different classes [9]. For multi-class classification problems, the one-versus-rest (OVR) strategy was employed, which trains one binary classifier per class against all other classes [25].

The linear kernel was selected for this study due to the high dimensionality of TF-IDF features, as linear SVMs have been shown to perform effectively when the number of features exceeds the number of training samples [26]. Hyperparameter tuning was performed using a 5-fold cross-validation grid search on the training set. The regularization parameter C was optimized via grid search over {0.01, 0.1, 1.0, 10, 100}, and the optimal value C=1.0 was selected. Additionally, the loss function was set to "squared_hinge", the penalty was set to "l2" regularization, and the maximum number of iterations was set to 10,000 to ensure convergence [27]. The decision function for the linear SVM is defined as equation 4.

$$f(x) = sign(w \cdot x + b) \quad (4)$$

where w is the weight vector, x is the input feature vector, and b is the bias term.

2.6. Evaluation Metrics

Model performance was evaluated using four standard classification metrics derived from the confusion matrix: accuracy, precision, recall, and F1-score [28]. These metrics provide a comprehensive assessment of classifier performance, particularly for imbalanced multi-class datasets. The 10-fold stratified cross-validation technique was employed to ensure robust and unbiased performance estimation, where stratification preserves the class distribution proportions in each fold [29].

Accuracy measures the proportion of correctly classified instances out of the total instances. Precision quantifies the proportion of true positive predictions among all positive predictions for each class. Recall measures the proportion of actual positive instances that were correctly identified. The F1-score represents the harmonic mean of precision and recall, providing a balanced measure of classifier performance [28]. Macro-averaging was used for precision, recall, and F1-score to give equal weight to each class regardless of its size, which is particularly important given the moderate class imbalance in the dataset.

3. RESULTS AND DISCUSSION

3.1. Preprocessing Results

The preprocessing pipeline successfully transformed the raw tweet data into a structured numerical representation suitable for machine learning classification. Table 2 summarizes the impact of each preprocessing stage on the dataset characteristics. The case-folding stage standardized all text to lowercase, while tokenization reduced the average token count from 18.3 to 14.7 tokens per tweet by removing URLs, mentions, and special characters. Stopword removal further reduced the average token count to 9.2 tokens per tweet. The Sastrawi stemmer successfully reduced 67.4% of the inflected words to their root forms.

The TF-IDF vectorization produced a sparse feature matrix of dimensions 13,169 × 28,456, where each row represents a tweet and each column corresponds to a unique term in the vocabulary. To mitigate the curse

of dimensionality, terms appearing in fewer than 3 documents or more than 95% of documents were excluded, resulting in a reduced feature matrix of 13,169 × 15,832 [30].

Table 2. Impact of Preprocessing Stages on Dataset Characteristics

Preprocessing Stage	Avg. Tokens/ Tweet	Vocabulary Size
Raw Data	18.3	45,672
Case Folding	18.3	38,941
Tokenization	14.7	35,218
Stopword Removal	9.2	28,456
Stemming	8.6	22,134
TF-IDF (filtered)	8.6	15,832

3.2. Classification Performance Comparison

The classification performance of both Naive Bayes and Support Vector Machine algorithms was evaluated using 10-fold stratified cross-validation. Table 3 presents the detailed comparison of performance metrics for each algorithm across the three classification categories.

The results demonstrate that SVM with a linear kernel consistently outperformed Naive Bayes across all evaluation metrics. SVM achieved an overall accuracy of 93.28%, which is 8.57 percentage points higher than the 84.71% accuracy attained by Naive Bayes. This substantial performance gap can be attributed to SVM's ability to construct optimal decision boundaries in the high-dimensional TF-IDF feature space [9].

Table 3. Overall Performance Comparison of NB and SVM

Metric	Naive Bayes (%)	SVM Linear (%)
Accuracy	84.71	93.28
Precision (macro avg)	83.56	92.45
Recall (macro avg)	84.23	93.12
F1-Score (macro avg)	84.12	92.89

3.3. Per-Class Analysis

A detailed per-class analysis reveals the strengths and weaknesses of each algorithm. Table 4 presents the precision, recall, and F1-score for each classification category under both algorithms. For the hate speech category, SVM achieved a precision of 91.34% compared to NB's 81.22%, indicating that SVM produces significantly fewer false positive predictions for hate speech content. The recall values of 92.67% (SVM) versus 82.45% (NB) further demonstrate SVM's superior ability to correctly identify actual hate speech instances. A similar pattern was observed for the abusive language category, where SVM attained an F1-score of 93.41% compared to NB's 85.67%.

Interestingly, the performance gap between the two algorithms was narrowest for the neutral category, where NB achieved an F1-score of 86.78% compared to SVM's 92.15%. This suggests that neutral content, which typically lacks distinctive markers of hate speech or abuse, is relatively easier to classify for both algorithms [31].

Table 4. Per-Class Performance Metrics

Category	Algorithm	Precision (%)	Recall (%)	F1-Score (%)
Hate Speech	NB	81.22	82.45	81.83
Hate Speech	SVM	91.34	92.67	92.00
Abusive	NB	83.89	85.56	84.72
Abusive	SVM	93.12	93.71	93.41
Neutral	NB	85.58	84.67	85.12
Neutral	SVM	92.89	92.98	92.93

3.4. Cross-Validation Analysis

The 10-fold cross-validation results provide further evidence of the stability and reliability of both classifiers. Table 5 presents the accuracy scores across all ten folds for both algorithms. The SVM classifier exhibited not only higher mean accuracy but also lower variance ($\sigma^2 = 0.42$) compared to NB ($\sigma^2 = 1.23$), indicating more consistent performance across different data partitions [29].

Table 5. 10-Fold Cross-Validation Accuracy Results

Fold	NB Accuracy (%)	SVM Accuracy (%)
1	85.23	93.45
2	83.89	92.89
3	84.56	93.67

Fold	NB Accuracy (%)	SVM Accuracy (%)
4	85.67	93.12
5	83.45	92.78
6	84.78	93.89
7	85.12	93.34
8	84.34	93.56
9	84.89	92.67
10	85.17	93.43
Mean ± SD	84.71 ± 0.67	93.28 ± 0.40

3.5. Confusion Matrix Analysis

The confusion matrices for both classifiers reveal distinct misclassification patterns and provide visual insight into class-level errors. Table 6 shows the numerical confusion matrix for the SVM classifier, while Table 7 presents the corresponding matrix for Naive Bayes.

Table 6. Confusion Matrix of SVM Classifier

	Pred: HS	Pred: AB	Pred: N
Actual: HS	3,357	113	154
Actual: AB	130	4,228	153
Actual: N	89	134	4,811

Table 7. Confusion Matrix of Naive Bayes Classifier

	Pred: HS	Pred: AB	Pred: N
Actual: HS	2,990	302	332
Actual: AB	311	3,865	335
Actual: N	245	289	4,500

For the Naive Bayes classifier, the most common error was misclassifying hate speech as abusive language (8.34% of hate speech instances) and vice versa (6.89% of abusive instances classified as hate speech). This confusion is attributable to the overlap in lexical features between these two categories, as many hate speech tweets also contain abusive language [13]. The SVM classifier demonstrated significantly lower cross-category misclassification rates, with only 3.12% of hate speech instances misclassified as abusive and 2.89% of abusive instances misclassified as hate speech. This improvement suggests that SVM's margin-maximizing approach is more effective at distinguishing between these semantically related categories by leveraging subtle feature differences in the high-dimensional TF-IDF space [32].

3.6. Computational Efficiency Comparison

Table 8 compares the training and prediction times of the two algorithms on the same hardware configuration (Intel Core i7-12700H, 16GB RAM). The results show that Naive Bayes is significantly faster in both training and prediction phases, completing training in 2.3 seconds compared to SVM's 47.8 seconds. This represents a 20.8x speedup, making NB a viable option for real-time applications where computational efficiency is prioritized [35].

Table 8. Computational Efficiency Comparison

Metric	Naive Bayes	SVM Linear
Training Time (seconds)	2.3	47.8
Prediction Time per tweet (ms)	0.12	0.85
Accuracy (%)	84.71	93.28
Speed-Accuracy Trade-off	Fast, Moderate Acc.	Slower, High Acc.

3.7. Discussion

The superior performance of SVM over Naive Bayes in this study aligns with findings from previous research in text classification domains. Joachims [33] demonstrated that SVM is particularly well-suited for text categorization due to the high dimensionality and sparsity of TF-IDF feature representations. The linear kernel's effectiveness in this context can be explained by the fact that text data in high-dimensional spaces is often linearly separable [26].

The relatively lower performance of Naive Bayes can be attributed to its fundamental assumption of conditional feature independence, which is violated in natural language data where words frequently co-occur and exhibit semantic dependencies [8]. For instance, the bigram "anjing lo" (a common Indonesian abusive expression) carries different semantic weight than its individual component words, but Naive Bayes treats each word independently, potentially losing important contextual information [34].

A deeper error analysis of misclassified instances reveals several patterns. First, false positives in the hate speech category frequently involved tweets containing strong emotional language or political criticism that used harsh vocabulary without targeting specific SARA-based groups. For example, tweets expressing frustration about government policies using profanity were sometimes classified as hate speech rather than abusive language. Second, sarcasm and irony posed significant challenges for both classifiers. Tweets such as "bagus sekali ya pemerintahannya" (the government is really great) used positive words sarcastically, leading to misclassification as neutral content. Third, code-switching between Indonesian and regional languages (e.g., Javanese, Betawi slang) created additional complexity, as the stopword list and stemmer were designed primarily for standard Indonesian. Fourth, abbreviations and informal spellings common in social media (e.g., "gblk" for "goblok", "bgt" for "banget") sometimes escaped the preprocessing pipeline, reducing feature quality for both classifiers.

Furthermore, the performance difference between the two algorithms was most pronounced for the hate speech category, which often involves nuanced expressions including sarcasm, coded language, and cultural references specific to Indonesian social media discourse [14]. SVM's ability to capture complex decision boundaries enables it to better distinguish these subtle linguistic patterns from neutral or merely abusive content.

Comparing our results with previous studies, the SVM accuracy of 93.28% is consistent with the findings of Ibrohim and Budi [13], who reported accuracies ranging from 77.36% to 93.45% for various machine learning approaches on Indonesian hate speech data. Our NB accuracy of 84.71% is also comparable to the 82.15% reported by Alfina et al. [14] using a similar approach. When compared to recent deep learning studies, our SVM results are competitive with Darmawan et al. [40] who reported 88.23% accuracy using IndoBERT, though transformer-based approaches such as IndoBERTweet have achieved higher macro-F1 scores (0.78) on larger datasets like IndoToxic2024 [38]. This comparison suggests that while classical ML approaches remain competitive for medium-sized datasets, deep learning methods may offer advantages when larger annotated corpora are available. It is worth noting that while SVM achieved superior overall performance, Naive Bayes offers significantly faster training and prediction times. In our experiments, NB completed training in 2.3 seconds compared to SVM's 47.8 seconds on the same hardware configuration. This computational efficiency makes NB a viable option for real-time applications where speed is prioritized over marginal accuracy improvements [35].

4. CONCLUSION

This study aimed to address three research questions: how NB and SVM perform in classifying Indonesian hate speech, which algorithm demonstrates superior performance, and what factors contribute to performance differences. The experimental results using a dataset of 13,169 annotated Indonesian tweets conclusively demonstrate that SVM with a linear kernel (accuracy: 93.28%, F1-score: 92.89%) significantly outperforms Naive Bayes (accuracy: 84.71%, F1-score: 84.12%) across all evaluation metrics, thereby answering the first two research questions. Regarding the third research question, the per-class analysis revealed that the performance advantage of SVM is most pronounced for the hate speech category, where the nuanced and context-dependent nature of hateful expressions demands a classifier capable of constructing complex decision boundaries. The error analysis identified sarcasm, code-switching, and informal abbreviations as key factors contributing to misclassification in both algorithms. The 10-fold cross-validation results further confirmed the stability and reliability of SVM, with lower variance compared to Naive Bayes.

These findings have practical implications for the development of automated content moderation systems on Indonesian social media platforms. While SVM is recommended for applications requiring maximum classification accuracy, Naive Bayes remains a viable alternative for scenarios where computational efficiency is prioritized, given its 20.8x training speed advantage. However, this study is limited to classical machine learning approaches with TF-IDF features, three-class classification, and a dataset collected from a single platform (Twitter/X). Future research should explore: (1) deep learning approaches such as IndoBERT and IndoBERTweet fine-tuned for Indonesian hate speech, (2) ensemble methods combining NB and SVM to leverage the strengths of both algorithms, (3) the incorporation of contextual features including user metadata and conversation threads, (4) handling of sarcasm and code-switching through specialized preprocessing, and (5) extension to multi-platform datasets covering Facebook, Instagram, and TikTok to improve generalizability.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the support of Universitas Budi Luhur for providing the computational resources and research facilities used in this study. We also extend our appreciation to the annotators who contributed to the dataset labeling process.

REFERENCES

[1] We Are Social and Meltwater, "Digital 2025: Indonesia," Global Digital Report, pp. 1-98, 2025.

[2] A. M. Kaplan and M. Haenlein, "Users of the world, unite! The challenges and opportunities of Social Media," Business Horizons, vol. 53, no. 1, pp. 59-68, 2010.
[3] Ministry of Communication and Information Technology Republic of Indonesia, "Annual Report on Digital Content Moderation," Jakarta, 2025.
[4] Republic of Indonesia, "Law No. 19 of 2016 on Electronic Information and Transactions," State Gazette of the Republic of Indonesia, 2016.
[5] S. Malmasi and M. Zampieri, "Detecting hate speech in multi-domain social media," in Proc. RANLP, pp. 452-459, 2017.
[6] A. Schmidt and M. Wiegand, "A survey on hate speech detection using natural language processing," in Proc. SocialNLP Workshop, pp. 1-10, 2017.
[7] B. Gambeck and Y. Sikdar, "Using convolutional neural networks for sentiment analysis of social media," in Proc. of the Int. Workshop on NLP, pp. 146-148, 2017.
[8] A. McCallum and K. Nigam, "A comparison of event models for naive Bayes text classification," in AAAI Workshop on Learning for Text Categorization, vol. 752, pp. 41-48, 1998.
[9] C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, vol. 20, no. 3, pp. 273-297, 1995.
[10] Z. Waseem and D. Hovy, "Hateful symbols or hateful people? Predictive features for hate speech detection on Twitter," in Proc. NAACL Student Research Workshop, pp. 88-93, 2016.
[11] B. Ross, M. Rist, G. Carbonell, B. Cabrera, N. Kurowsky, and M. Wojatzki, "Measuring the reliability of hate speech annotations: The case of the European refugee crisis," in Proc. NLP4CMC III, pp. 6-9, 2017.
[12] N. Albadi, M. Kurdi, and S. Mishra, "Are they our brothers? Analysis and detection of religious hate speech in the Arabic Twittersphere," in Proc. IEEE/ACM ASONAM, pp. 69-76, 2018.
[13] M. O. Ibrohim and I. Budi, "Multi-label hate speech and abusive language detection in Indonesian Twitter," in Proc. ALW3, pp. 46-57, 2019.
[14] I. Alfina, R. Mulia, M. I. Fanany, and Y. Ekanata, "Hate speech detection in the Indonesian language: A dataset and preliminary study," in Proc. ICACSIS, pp. 233-238, 2017.
[15] R. Ting, P. Wirawan, and N. Hidayat, "Social media content analysis framework for Indonesian digital platforms," Journal of Information Systems, vol. 15, no. 2, pp. 112-125, 2024.
[16] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated hate speech detection and the problem of offensive language," in Proc. ICWSM, pp. 512-515, 2017.
[17] J. L. Fleiss, "Measuring nominal scale agreement among many raters," Psychological Bulletin, vol. 76, no. 5, pp. 378-382, 1971.
[18] C. D. Manning, P. Raghavan, and H. Schutze, Introduction to Information Retrieval. Cambridge University Press, 2008.
[19] S. Bird, E. Klein, and E. Loper, Natural Language Processing with Python. O'Reilly Media, 2009.
[20] F. Z. Tala, "A study of stemming effects on information retrieval in Bahasa Indonesia," M.S. thesis, Universiteit van Amsterdam, 2003.
[21] A. Z. Arifin, I. P. A. K. Mahendra, and H. T. Ciptaningtyas, "Enhanced confix stripping stemmer and Ants algorithm for classifying news documents in Indonesian language," in Proc. ICIC, pp. 149-158, 2009.
[22] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," Information Processing & Management, vol. 24, no. 5, pp. 513-523, 1988.
[23] J. D. Rennie, L. Shih, J. Teevan, and D. R. Karger, "Tackling the poor assumptions of naive Bayes text classifiers," in Proc. ICML, pp. 616-623, 2003.
[24] C. Zhai and S. Massung, Text Data Management and Analysis. ACM Books, 2016.
[25] C. W. Hsu and C. J. Lin, "A comparison of methods for multiclass support vector machines," IEEE Trans. Neural Networks, vol. 13, no. 2, pp. 415-425, 2002.
[26] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in Proc. ECML, pp. 137-142, 1998.
[27] C. C. Chang and C. J. Lin, "LIBSVM: A library for support vector machines," ACM Trans. Intelligent Systems and Technology, vol. 2, no. 3, pp. 1-27, 2011.
[28] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," Information Processing & Management, vol. 45, no. 4, pp. 427-437, 2009.
[29] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in Proc. IJCAI, pp. 1137-1143, 1995.
[30] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," Journal of Machine Learning Research, vol. 3, pp. 1157-1182, 2003.
[31] M. Ousidhoum, Z. Lin, H. Zhang, Y. Song, and D. Yeung, "Multilingual and multi-aspect hate speech analysis," in Proc. EMNLP-IJCNLP, pp. 4675-4684, 2019.

[32] Y. Mehdad and J. Tetreault, "Do characters abuse more than words?" in Proc. SIGDial, pp. 299-303, 2016.
[33] T. Joachims, Learning to Classify Text Using Support Vector Machines. Kluwer Academic Publishers, 2002.
[34] W. Warner and J. Hirschberg, "Detecting hate speech on the world wide web," in Proc. LSM Workshop, pp. 19-26, 2012.
[35] H. M. Iskandar and A. Purwarianti, "Comparison of machine learning algorithms for hate speech detection in Indonesian social media," in Proc. ICEEI, pp. 67-72, 2024.
[36] J. Kusuma and A. Chowanda, "Indonesian hate speech detection using IndoBERTweet and BiLSTM on Twitter," JOIV: International Journal on Informatics Visualization, vol. 7, no. 3, pp. 773-780, 2023.
[37] R. I. Yulfa, A. Solichin, and R. Budiharto, "Enhancing hate speech detection in social media using IndoBERT model: A study of sentiment analysis during the 2024 Indonesia presidential election," in Proc. ICAICTA, pp. 1-6, 2023.
[38] L. Susanto, et al., "IndoToxic2024: A demographically-enriched dataset of hate speech and toxicity types for Indonesian language," in Proc. EMNLP Demo, pp. 1-10, 2024.
[39] I. F. Rokhim, R. Sarno, A. F. Septiyanto, A. T. Haryono, and S. I. Sabilla, "IndoBERT-based ensemble learning for multi-level multi-label hate speech detection in Indonesian social media," in Proc. BTS-I2C, pp. 456-461, 2024.
[40] A. Darmawan, et al., "Experiments on IndoBERT implementation for detecting multi-label hate speech with data resampling through synonym replacement method," in Proc. IEEE ICRAIE, pp. 1-6, 2023.
[41] Z. Al-Makhadmeh and A. Tolba, "Automatic hate speech detection using killer natural language processing optimizing ensemble deep learning approach," Computing, vol. 102, pp. 501-522, 2020.