Scientific Journal of Informatics
Vol. 10, No. 4, Nov 2023
p-ISSN 2407-7658 http://journal.unnes.ac.id/nju/index.php/sji e-ISSN 2460-0040

Voting Classifier Technique and Count Vectorizer with N-gram to Identify Hate Speech and Abusive Tweets in Indonesian

Riza Arifudin[1*] , Dandi Indra Wijaya[2], Budi Warsito[3], Adi Wibowo[4]
[1,2]Department of Computer Science, Faculty of Mathematics and Natural Sciences,
Universitas Negeri Semarang, Indonesia
[3]Department of Statistics, Faculty of Sciences and Mathematics,
Diponegoro University, Indonesia
[4]Department of Informatics, Faculty of Sciences and Mathematics,
Diponegoro University, Indonesia

Abstract.
Purpose: The objective of this study is to identify hate speech and abusive tweets in Indonesian using a Voting Classifier technique and Count Vectorizer with N-grams. Voting Classifier technique involves combining multiple classifiers like Random Forest and Support Vector Machines to improve classification accuracy.
Methods: This research begins by preprocessing the data. Voting classifier uses Support Vector Machine algorithm and Random Forest algorithm. Support Vector Machine and Random Forest serve as the estimators for the voting classifier. As for feature extraction, N-gram and count vectorizer were employed. The effectiveness of the suggested procedures is the desired outcome.
Result: Combining the Voting Classifier approach with Count Vectorizer feature extraction and using 1 gram of N-grams, or 82.50%, resulted in the best accuracy. From this study, it can be inferred that the approach employed to identify hate speech and abusive tweets is extremely practical.
Novelty: Combining multiple classifiers and using feature extraction techniques like count vectorizer and N-gram with machine learning algorithms can be used for sentiment analysis to differentiate between hate speech and abusive tweets.

Keywords: Voting classifier, Count vectorizer, Support vector machine, Random forest, Sentiment analysis
Received August 2023 / **Revised** November 2023 / **Accepted** November 2023

INTRODUCTION

Social media use is widely accepted in modern life. Around two billion users were active in 2015 [1]. The increasing adoption of the internet and smartphones has given rise to new social media platforms that enable online connections and friendships. The use of social media facilitates communication [2]. Communication is the act of one person expressing sentiments or thoughts to another via the use of appropriate symbols and media. Twitter is a popular social media platform. Tweets containing hate speech are one issue with Twitter [3].

Hate speech is defined as language used to show scorn towards an individual or a group based on characteristics such as race, ethnicity, gender, disability, or sexual orientation [4]. Hate speech inherently aims to have a specific impact, whether it is direct or indirect. The problem of hate speech exists within social media platforms. Consequently, it is important to note that while a statement may possess a strong or offensive tone, it may not necessarily reflect hatred and could instead be regarded as a form of disrespectful communication. These distinctions must be taken into account since they may result in negative bias in daily life. Some researchers argue for using the term "discriminatory speech" instead of "hate speech" to better capture the harm caused by such speech [5]. Efforts are being made to develop AI and deep learning techniques to automatically detect hate speech messages in real-time [4].

Since online information is produced so quickly, hate speech content cannot be manually reduced [6]. Hence, an automated system capable of identifying hate speech tweets is necessary. The technique may be

*Corresponding author.
Email addresses: rizaarifudin@mail.unnes.ac.id (Riza)
DOI: 10.15294/sji.v10i4.46633

used to discern between abusive and hateful communication. Sentiment analysis may be used to create the system.

Social media sentiment analysis is a research area that centers on extracting individuals' opinions, attitudes, and emotions from social networks [7], [8]. Attempting to automatically identify the attitudes included in the text, sentiment analysis is an emerging area of computer science, according to Taboada [9]. Sentiment analysis, also known as opinion mining, is the process of analyzing and categorizing emotions and perceptions expressed in text or reviews [10], [11]. Hussein [12] claims that sentiment analysis is a machine learning application of natural language processing, sometimes known as NLP or just NLP. Large volumes of natural language data may be processed and analyzed by computers using the NLP approach.

Numerous studies have focused on identifying hate speech tweets in Indonesian language. Ibrohim and Budi [13] conducted a research that dealt with the classification of multi-label text to detect abusive speech and identify hate speech on Twitter Indonesia. They utilized machine learning techniques, including support vector machine, naive Bayes, random forest decision tree, and Binary Relevance, to accomplish this task. Prior to training, various feature extraction methods were applied. Among these, the random forest algorithm with the LP data transformation approach demonstrated the highest performance when utilizing the word unigram feature, achieving an accuracy of 76.16%.

In a similar vein, Fauzi and Yuniarti [14] employed a comparable machine learning approach but adopted a distinct feature extraction technique to examine hate speech in the Indonesian language. The Term Frequency - Inverse Document Frequency (TF-IDF) approach was employed in his study. With the use of TF-IDF, we are able to link every word in a document to a numerical representation of how pertinent it is. There are discrepancies between the two studies' feature extraction stages. The feature extraction step has a significant impact on how well the classification system performs. The N-gram approach, which can boost performance and was used in the research by Ibrohim & Budi, was not used in the study by Fauzi & Yuniarti. According to Laoh et.al research's [15], the N-gram approach may improve accuracy in numerous sentiment analysis literatures by an average of 94% when used in dataset feature extraction.

The ensemble approach, namely voting, is also used in research by Fauzi and Yuniarti. The study's findings suggest that by employing the ensemble approach with soft voting on the three leading classifiers - naive Bayes, SVM, and RF - can potentially enhance the classification performance. One ensemble strategy where we may mix many models for improved classification is classifier voting [16]. The voting classifier selects one of many options depending on the projected class that receives the majority of votes [17].

According to the preceding definition, random forests and support vector machines are two learning algorithms that are often employed and excel at sentiment analysis. Random Forest (RF) is a set classifier that generates multiple decision tree algorithms using a subset of randomly selected training samples and variables [18]. A machine learning technique based on statistical learning theory is the support vector machine (SVM) [19]. SVM's basic concept is the use of a hyperplane to divide several classes [20]. The study concentrates on the utilization of ensemble approaches, such as voting, with the random forest and support vector machine, as well as the preprocessing stage involving Count vectorizer and N-grams.

METHODS

In this paper, the classification of hate speech and abusive tweets was performed using a RF method, a SVM, and a combination of the two using the Voting Classifier Method. The feature extraction techniques employed were Count vectorizer and N-gram. Figure 1 illustrates the flowchart for the preprocessing phase.

The preprocessing phase starts with a Twitter dataset obtained from Kaggle. The collected data underwent several preprocessing steps to ensure cleaner and more reliable results. The first step involved case folding, which converted all text to lowercase to eliminate inconsistencies caused by different capitalizations. Next, the punctional remover step removed any punctuation marks from the text. Following that, tokenization was performed to split the text into individual tokens or words. Stopword removal was then conducted to eliminate commonly used words that do not carry significant meaning for sentiment analysis. The next step involved stemming, where words were reduced to their base or root form to reduce variations. Finally, the cleaned dataset from the preprocessing phase was ready for further analysis and classification.

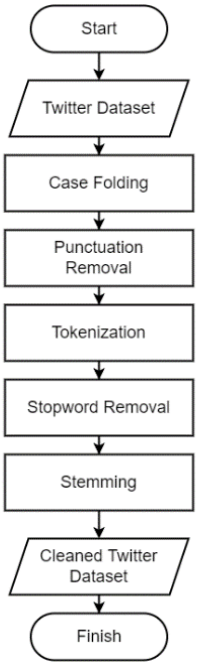

Figure 1. Dataset preprocessing flowchart

The procedure employed in this investigation is illustrated in Figure 2, depicting the flowchart of the steps involved.

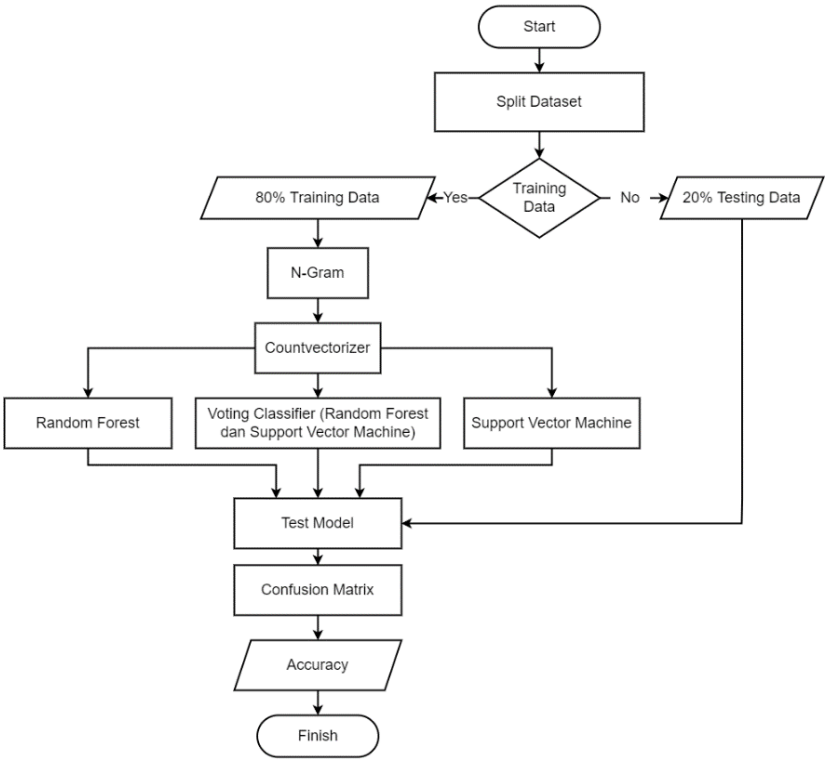

Figure 2. Flowchart of the proposed method

In the initial stage of this study, the dataset is prepared. The dataset employed in this research is derived from Kaggle.com, specifically the Indonesian language Twitter Hate Speech dataset, which contains instances of hate speech and abusive content. Subsequently, the resulting dataset undergoes a preprocessing

stage consisting of five phases: (1) case folding, (2) punctuation removal, (3) tokenization, (4) stopword removal, and (5) stemming. Case folding involves converting all characters to lowercase. The punctuation removal stage eliminates any punctuation marks present in the data. At the tokenization phase, each tweet's data, whether in the form of phrases or paragraphs, is split into individual tokens.Words or tokens that are deemed to be of minor importance are eliminated during the stopword removal step. During the stemming step, affix-containing words are converted to basic words.

After completing the preprocessing stage, the next step involves feature extraction, which entails utilizing count vectorizer and N-grams (where n=1, n=2, and n=3). By employing the splitter approach, the dataset, along with the selected features obtained from the feature extraction phase, is divided into training data and test data. The dataset is split in an 80:20 ratio, with 80% allocated as training data and 20% as test data. Subsequently, the training data is used to train models such as support vector machine, voting classifier, and random forest algorithm.

Data Collection

The dataset used in this study consists of 13,169 tweet data regarding 2019 Indonesian hate speech which has 2 labels, where each label contains the number 0 and number 1. The value 0 is for FALSE and the value 1 is for TRUE according to the label. This dataset is sourced from the Kaggle online community which can be downloaded via the link kaggle.com/ilhamfp31/indonesian-abusive-and-hate-speech-twitter-text.

Support Vector Machine

Support Vector Machines (SVMs) are widely used for classification tasks. Here are the key steps for using SVMs in classification: (1) Understanding the concept: SVMs are binary classifiers that aim to maximize the margin between classes in the feature space [21]. They use the principle of structural risk minimization to fit small data samples [21]. (2) Feature weighting: To improve classification accuracy, a method based on Relief-F feature weighting can be used. This method calculates the weight value of each feature and weights the inner product in the SVM kernel function accordingly [22]. (3) Optimization: Various optimization techniques have been proposed to enhance SVMs. These include model selection, novelty detection, and advancements in kernel variants [23]. These optimizations aim to improve computational efficiency and classification accuracy [24].

Random Forest

To use the random forest algorithm for classification, follow these steps: (1) Train individual decision tree classifiers: Random forest is a method that combines multiple decision tree classifiers. Each decision tree is trained on a subset of the data using the bootstrap method [25]. (2) Aggregate the predictions: Once the decision trees are trained, the predictions from each tree are aggregated to make the final classification decision [26]. (3) Optimize the algorithm: Researchers have proposed various optimization techniques to improve the performance of random forest. These include forest pruning to reduce the number of trees and improve efficiency [27], unequal weight voting to assign weights based on the performance of individual trees [28].

Voting Classifier

The Voting Classifier using Random Forest and SVM involves combining multiple classifiers to make predictions. The steps to implement this approach: (1) Train individual classifiers: Train a Random Forest classifier and an SVM classifier using the training data [29] [30] [31]. (2) Obtain predictions: Use the trained classifiers to obtain predictions for the test data [29] [30] [31]. (3) Combine predictions: Combine the predictions from the Random Forest and SVM classifiers using a voting mechanism. This can be done by taking the majority vote of the predictions [30]. (4) Make final prediction: Assign the class label with the highest number of votes as the final prediction [30].

The Ensemble Voting Classifier facilitates both "hard" and "soft" voting methods in ensemble learning [32]. In "hard voting," the input data is classified by taking the average of predictions made by multiple classifiers. If the classifiers' weights are identical or majority votes are different, they are considered distinct.

If there are three classifiers, namely clf 1, clf 2, and clf 3, and their classification results are [1, 1, 0]. In this case, the voting result would be 1. This serves as an illustration of a hard voting approach with a majority of votes, all carrying equal weight.. An illustration of a hard vote where the majority votes with the same

weight The difference is that the prediction outcomes are averaged to [0, 0, 0, 0, 1, 1, 1, 1, 1, 1] if there are three classifiers (clf 1, clf 2, clf 3), each with a weight of [0.1, 0.3, 0.6] and a prediction result of [0, 0, 1]. So, 1 votes were cast.

Based on the probability of each prediction provided by each classifier, the input data is classified using soft voting. Each classifier's specified weights are applied correctly. For instance, the three binary classifiers clf 1, clf 2, and clf 3 are available. The classifier uses the class [0,1] to predict values for a given record. The outputs of each classifier's probability prediction, for instance, are clf 1 -> [0.3, 0.7], clf 2 -> [0.2, 0.8], and clf 3 -> [0.8, 0.2]. The likelihood of each class having the same weight will be determined using the following formula.

Class 0 = 0.33*0.3 + 0.33*0.2 + 0.33*0.8 = 0.429
Class 1 = 0.33*0.7 + 0.33*0.8 + 0.33*0.2 = 0.561

With the aforementioned outcomes, the class [0, 1] with three classifiers has a voting outcome of 1. After the individual classifier has been trained, no parameters need to be provided for the majority decision [33].

Count vectorizer

The count vectorizer library has established itself as a reliable technique for feature extraction. The result of count vectorization yields a typical sparse matrix in terms of counts [34]. When working with text data, the count vectorizer is an effective tool as it converts each word into a vector. With count vectorizer, every unique word is assigned a column in the resulting matrix, while each text sample from the document is represented by a row in the matrix [35]. The value in each cell simply denotes the frequency of the corresponding word in that particular text sample. Table 1 can include this.

document = [“Jackson went to the market”, “The market is close”]

Table 1. Count vectorizer example

	jackson	went	to	the	market	is	close
document[0]	1	1	1	1	1	0	0
document[1]	0	0	0	1	1	1	1

N-gram

The steps of the N-Gram model are: (1) Building the N-Gram corpus: To construct an N-Gram model, a large pre-processed dataset called N-Grams is needed. This dataset consists of continuous sequences of words of length N [36]. (2) Estimating N-Gram probabilities: Language models are trained by estimating the probabilities of N-Grams based on observed sequences of words in a text corpus [37]. (3) Predicting the next word: Once the N-Gram model is built and probabilities are estimated, it can be used to predict the next word in a given sequence of words [38]. This prediction is based on the probabilities assigned to different N-Grams in the model.

Table 2. Example for N-gram

Texts	Terms
Apapun Profesinya, Masyarakat Indo itu umumnya Dah CERDAS	Sentence/N-gram
"Apapun","Profesinya,","Masyarakat","Indo" "itu", "umumnya", "Dah", "CERDAS"	Unigram
"Apapun Profesinya,", "Profesinya, Masyarakat", "Masyarakat Indo", "Indo itu", "itu umumnya" "umumnya Dah", "Dah CERDAS",	Bigram
"Apapun Profesinya, Masyarakat", "Profesinya, Masyarakat Indo", "Masyarakat Indo itu", "Indo itu umumnya", "itu umumnya Dah", "umumnya Dah CERDAS"	Trigram

Based on the information presented in Table 2, the attributes of n-grams can be categorized as features related to words and features related to letters. Terms like Unigrams, bigrams, trigrams, and so forth are employed to describe n-grams of various lengths, such as n-grams of length 1, which are also referred to as unigrams, n-grams of length 2, commonly known as bigrams, and so forth.

RESULTS AND DISCUSSIONS

In this study, the initial phase involves preprocessing the dataset. The preprocessing steps include case folding, removing punctuation, tokenization, removing stop words, and stemming. One of these preprocessing techniques is applied to generate the tweet.

" Setuju bro... #GANTI Presiden, maka otomatis akan tergantikan : menteri2 yg konyol, kapolri yg dzolim, komisaris2 BUMN yg oportunis, dan seluruh bani cebi dari puncak beringin hingga akar rumputnya. Buktikan."

can be seen in Table 3.

Table 3. Preprocessing method result

Tweet	Method
setuju bro... #ganti presiden, maka otomatis akan tergantikan : menteri2 yg konyol, kapolri yg dzolim, komisaris2 bumn yg oportunis, dan seluruh bani cebi dari puncak beringin hingga akar rumputnya. buktikan.	Case Folding
setuju bro ganti presiden maka otomatis akan tergantikan menteri2 yg konyol kapolri yg dzolim komisaris2 bumn yg oportunis dan seluruh bani cebi dari puncak beringin hingga akar rumputnya buktikan	Punctuation Removal
['setuju', 'bro', 'ganti', 'presiden', 'maka', 'otomatis', 'akan', 'tergantikan', 'menteri2', 'yg', 'konyol', 'kapolri', 'yg', 'dzolim', 'komisaris2', 'bumn', 'yg', 'oportunis', 'dan', 'seluruh', 'bani', 'cebi', 'dari', 'puncak', 'beringin', 'hingga', 'akar', 'rumputnya', 'buktikan']	Tokenization
['setuju', 'bro', 'ganti', 'presiden', 'otomatis', 'tergantikan', 'menteri2', 'konyol', 'kapolri', 'dzolim', 'komisaris2', 'bumn', 'oportunis', 'bani', 'cebi', 'puncak', 'beringin', 'akar', 'rumputnya', 'buktikan']	Stop word Removal
['setuju', 'bro', 'ganti', 'presiden', 'otomatis', 'ganti', 'menteri2', 'konyol', 'kapolri', 'dzolim', 'komisaris2', 'bumn', 'oportunis', 'bani', 'cebi', 'puncak', 'beringin', 'akar', 'rumput', 'bukti']	Stemming

In this research, the Sastrawi technique is utilized for stemming. Sastrawi is a user-friendly Python package designed to streamline the process of reducing inflectional Indonesian words to their simplest form. Following stemming, the labeling process takes place. Each tweet within the dataset is assigned a label indicated by the numbers 0 and 1. The label 0 signifies that the tweet does not belong to the category, whereas the label 1 signifies its inclusion. The tweets with their corresponding labels are presented in Table 4.

Table 4. Labeled tweets

Tweet	Hate Speech	Abusive
Ngomongnya gini ?nyatanya minta 2 periode.,; Gak ngurus copras capres tp ngotot 2 periode.; Cukup 1 periode saja.....; #2019GantiPresiden	1	0
Negara harus tetap sbg wasit yg berdiri ditengah sehingga Keadilan itu tetap tegak. Tdk akan ada keadilan ketika ada negara dlm negara...! #presiden #calonpresiden	0	0
Klo ada yg ngomong Deklarasi #ganti presiden adlh makar, itu bloon alias dungu. Mkx klo g ngerti bernegara mending diam spy g keliat bodox...	1	1
Apapun Profesinya, Masyarakat Indo itu umumnya Dah CERDAS Mereka Sdh Jijik Dg Cara2 KOTOR Petahana yg Menghalalkan Semua Cara (Gaya Komunis)	1	1

To facilitate research, the table mentioned above has been modified to include three categories: neutral tweets are labeled 0, tweets containing harsh sentences are labeled 1, while tweets containing hate speech are labeled 2. In addition, 0 has been assigned to the category for hate speech. Specifically, hate speech

tweets are defined as tweets labeled 1 for hate speech and 0 for abusive speech, or as tweets labeled 1 for hate speech and 1 for abusive speech. The revised labels for the tweets are presented in Table 5.

Table 5. Tweets with updated labels

Preprocessing results	Label
['ngomong', 'gini', 'nyata', 'minta', '2', 'periode', 'gak', 'urus', 'copras', 'capres', 'tp', 'ngotot', '2', 'periode', 'cukup', '1', 'periode', 'saja', '2019gantipresiden']	1
['negara', 'tetap', 'wasit', 'berdiri', 'tengah', 'keadilan', 'tetap', 'tegak', 'tdk', 'keadilan', 'negara', 'negara', 'presiden', 'calonpresiden']	0
['Klo', 'ngomong', 'Deklarasi', '#ganti', 'presiden', 'adlh', 'makar', 'bloon', 'alias', 'dungu', 'Mkx', 'g', 'ngerti', 'beregara', 'mending', 'diam', 'spy', 'g', 'keliat', 'bodox']	2
['apapun', 'profesi', 'masyarakat', 'indo', 'umumnya', 'cerdas', 'mereka', 'sdh', 'jijik', 'dg', 'cara2', 'kotor', 'petahana', 'yg', 'menghalalkan', 'semua', 'cara', 'gaya', 'komunis']	2

To train the models, the dataset is utilized in conjunction with several algorithms including support vector machine, random forest, voting classifier with a RF estimator, SVM with count vectorizer, and an N-gram feature extraction algorithm. The precision of these models is evaluated using the Confusion Matrix, which provides insights into their performance. It is crucial to measure a model's accuracy using the Confusion Matrix. Table 6 displays the findings of the accuracy of the current models.

Table 6. Models accuracy

Feature Extraction	Gram	Classifier	Accuracy(%)
Count Vectorizer	1	Random Forest	81.70
Count Vectorizer	2	Random Forest	80.64
Count Vectorizer	3	Random Forest	79.31
Count Vectorizer	1	Support Vector Machine	80.90
Count Vectorizer	2	Support Vector Machine	80.11
Count Vectorizer	3	Support Vector Machine	78.55
Count Vectorizer	1	Voting Classifier	82.50
Count Vectorizer	2	Voting Classifier	80.75
Count Vectorizer	3	Voting Classifier	78.74

The combination of the soft voting classifier approach with support vector machine and random forest estimators, along with the utilization of count vectorizer and N-gram feature extraction up to 1 gram, achieved the highest accuracy of 82.50%, as depicted in Table 5. The random forest technique obtained the second-highest accuracy of 81.70% with 1 gram of N-gram. In comparison, the support vector machine yielded a lower accuracy of 80.90% with 1 gram of N-gram. Notably, the use of N-grams alongside count vectorizer demonstrated a decrease in accuracy for each method as the gram size increased.

The results of the study indicate that sentiment analysis on the Indonesian hate speech Twitter dataset was effectively conducted by employing count vectorizer and N-gram feature extraction with support vector machine and random forest algorithms. The approach taken in this study is superior to those currently being employed in similar studies. Table 7 displays comparisons with earlier research using the same dataset and technique.

Table 7. Comparing the findings with prior studies

Writer	Method	Accuracy(%)
Ibrohim & Budi [13]	Label power set & Unigram & Random forest	76.16
Proposed Method	Count Vectorizer & Unigram & Voting Classifier with estimator Support Vector Machine and Random Forest	82.50
Proposed Method	count vectorizer & Unigram & Random forest	81.70

Table 7 illustrates the highest accuracy achieved at 82.50% through the utilization of a combination of the soft voting method in the voting classifier, support vector machine estimator, random forest feature extraction with count vectorizer, and N-grams up to 1 gram. In the research conducted by Ibrohim and Budi [13], they solely employed N-grams with different algorithms and achieved a maximum accuracy of 76.16% using the random forest method with 1 gram of N-grams. It is worth noting that the random forest approach with count vectorizer feature extraction yields higher accuracy compared to this study. The findings of this research suggest that employing count vectorizer and N-gram feature extraction positively impacts the accuracy of algorithms, as observed in previous studies. Additionally, the voting classifier algorithm can contribute to higher accuracy.

To enhance the accuracy of techniques utilized in related research, incorporating support vector machine, random forest, and voting classifier algorithms with SVM estimators, along with random forest utilizing count vectorizer and N-gram feature extraction, can be beneficial.

CONCLUSION

To analyze the sentiment of hate speech and abusive speech in the Indonesian Twitter dataset, this study employed various techniques, including the RF algorithm, SVM, Voting Classifier with Count Vectorizer, and N-gram feature extraction. The Voting Classifier technique utilized SVM and RF as estimators. Both "hard" and "soft" voting were performed using the voting classifier. In this study, the author opted for soft voting during the training phase as it considers more information and takes into account the uncertainty of each classifier. The highest accuracy results, at 82.50%, were achieved by employing the voting classifier method with a RF estimator, SVM with count vectorizer, and N-gram feature extraction up to 1 gram. This outcome outperformed the accuracy of 76.16% obtained in a related study that used the random forest and unigram technique.

REFERENCES

[1] D. Kim and S. S. Jang, "The psychological and motivational aspects of restaurant experience sharing behavior on social networking sites," *Serv. Bus.*, vol. 13, no. 1, pp. 25–49, Mar. 2019, doi: 10.1007/s11628-018-0367-8.

[2] T. L. Nikmah, M. Z. Ammar, and Y. R. Allatif, "Comparison of LSTM, SVM, and naive bayes for classifying sexual harassment tweets," *J. Soft Comput. Explor.*, vol. 3, no. 2, pp. 131–137, 2022, doi: 10.52465/joscex.v3i2.85.

[3] N. R. Fatahillah, P. Suryati, and C. Haryawan, "Implementation of Naive Bayes classifier algorithm on social media (Twitter) to the teaching of Indonesian hate speech," in *2017 International Conference on Sustainable Information Engineering and Technology (SIET)*, Nov. 2017, pp. 128–131. doi: 10.1109/SIET.2017.8304122.

[4] I. Karthika, G. Boomika, R. Nisha, M. Shalini, and S. P. Srivarshini, "A Survey on Detecting and Preventing Hateful Comments on Social Media Using Deep Learning," in *Smart Innovation, Systems and Technologies*, Springer Science and Business Media Deutschland GmbH, 2023, pp. 285–298. doi: 10.1007/978-981-19-3575-6_30.

[5] D. Kindermann, "Against 'Hate Speech,'" *J. Appl. Philos.*, Apr. 2023, doi: 10.1111/japp.12648.

[6] S. Mishra, S. Prasad, and S. Mishra, "Exploring Multi-Task Multi-Lingual Learning of Transformer Models for Hate Speech and Offensive Speech Identification in Social Media," *SN Comput. Sci.*, vol. 2, no. 2, pp. 1–19, 2021, doi: 10.1007/s42979-021-00455-5.

[7] Z. Li, Y. Fan, B. Jiang, T. Lei, and W. Liu, "A survey on sentiment analysis and opinion mining for social multimedia," *Multimed. Tools Appl.*, vol. 78, no. 6, pp. 6939–6967, 2019, doi: 10.1007/s11042-018-6445-z.

[8] J. Jumanto, M. A. Muslim, Y. Dasril, and T. Mustaqim, "Accuracy of Malaysia Public Response to Economic Factors During the Covid-19 Pandemic Using Vader and Random Forest," *J. Inf. Syst. Explor. Res.*, vol. 1, no. 1, pp. 49–70, 2022, doi: 10.52465/joiser.v1i1.104.

[9] M. Taboada, "Sentiment Analysis: An Overview from Linguistics," *Annu. Rev. Linguist.*, vol. 2, pp. 325–347, 2016, doi: 10.1146/annurev-linguistics-011415-040518.

[10] G. I. Ahmad and J. Singla, "Machine Learning Techniques for Sentiment Analysis of Indian Languages," *Int. J. Recent Technol. Eng.*, vol. 8, no. 2S11, pp. 3630–3636, Nov. 2019, doi: 10.35940/ijrte.B1456.0982S1119.

[11] W. Budiawan Zulfikar, A. Rialdy Atmadja, and S. F. Pratama, "Sentiment Analysis on Social Media Against Public Policy Using Multinomial Naive Bayes," *Sci. J. Informatics*, vol. 10, no. 1, pp. 25–34, 2023, doi: 10.15294/sji.v10i1.39952.

[12] D. M. E.-D. M. Hussein, "A survey on sentiment analysis challenges," *J. King Saud Univ. - Eng. Sci.*, vol. 30, no. 4, pp. 330–338, Oct. 2018, doi: 10.1016/j.jksues.2016.04.002.
[13] M. O. Ibrohim and I. Budi, "Multi-label Hate Speech and Abusive Language Detection in Indonesian Twitter," in *Proceedings of the Third Workshop on Abusive Language Online*, 2019, pp. 46–57. doi: 10.18653/v1/W19-3506.
[14] M. A. Fauzi and A. Yuniarti, "Ensemble Method for Indonesian Twitter Hate Speech Detection," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 11, no. 1, p. 294, Jul. 2018, doi: 10.11591/ijeecs.v11.i1.pp294-299.
[15] E. Laoh, I. Surjandari, and N. I. Prabaningtyas, "Enhancing Hospitality Sentiment Reviews Analysis Performance using SVM N-Grams Method," in *2019 16th International Conference on Service Systems and Service Management (ICSSSM)*, Jul. 2019, pp. 1–5. doi: 10.1109/ICSSSM.2019.8887662.
[16] U. K. Kumar, M. B. S. Nikhil, and K. Sumangali, "Prediction of breast cancer using voting classifier technique," in *2017 IEEE International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM)*, Aug. 2017, pp. 108–114. doi: 10.1109/ICSTM.2017.8089135.
[17] Y. Zhang, H. Zhang, J. Cai, and B. Yang, "A Weighted Voting Classifier Based on Differential Evolution," *Abstr. Appl. Anal.*, vol. 2014, pp. 1–6, 2014, doi: 10.1155/2014/376950.
[18] M. Belgiu and L. Drăguţ, "Random forest in remote sensing: A review of applications and future directions," *ISPRS J. Photogramm. Remote Sens.*, vol. 114, pp. 24–31, Apr. 2016, doi: 10.1016/j.isprsjprs.2016.01.011.
[19] S. Ding, Z. Zhu, and X. Zhang, "An overview on semi-supervised support vector machine," *Neural Comput. Appl.*, vol. 28, no. 5, pp. 969–978, 2017, doi: 10.1007/s00521-015-2113-7.
[20] A. Tharwat, A. E. Hassanien, and B. E. Elnaghi, "A BA-based algorithm for parameter optimization of Support Vector Machine," *Pattern Recognit. Lett.*, vol. 93, pp. 13–22, Jul. 2017, doi: 10.1016/j.patrec.2016.10.007.
[21] Q. Wang, "Support Vector Machine Algorithm in Machine Learning," in *2022 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, Jun. 2022, pp. 750–756. doi: 10.1109/ICAICA54878.2022.9844516.
[22] J. Huang, J. Zhou, and L. Zheng, "Support Vector Machine Classification Algorithm Based on Relief-F Feature Weighting," in *2020 International Conference on Computer Engineering and Application (ICCEA)*, Mar. 2020, pp. 547–553. doi: 10.1109/ICCEA50009.2020.00121.
[23] U. S. Bist and N. Singh, "Analysis of recent advancements in support vector machine," *Concurr. Comput. Pract. Exp.*, vol. 34, no. 25, Nov. 2022, doi: 10.1002/cpe.7270.
[24] K. Guo, J. Song, Y. Wang, and D. Yan, "An Improved Directed Acyclic Graph SVM," 2023, pp. 6410–6419. doi: 10.1007/978-981-19-6613-2_618.
[25] N. E. I. Karabadji, A. Amara Korba, A. Assi, H. Seridi, S. Aridhi, and W. Dhifli, "Accuracy and diversity-aware multi-objective approach for random forest construction," *Expert Syst. Appl.*, vol. 225, p. 120138, Sep. 2023, doi: 10.1016/j.eswa.2023.120138.
[26] M. Azhari, A. Alaoui, Z. Achraoui, B. Ettaki, and J. Zerouaoui, "Adaptation of the random forest method," in *Proceedings of the 4th International Conference on Smart City Applications*, Oct. 2019, pp. 1–6. doi: 10.1145/3368756.3369004.
[27] Y. Manzali and M. Elfar, "Random Forest Pruning Techniques: A Recent Review," *Oper. Res. Forum*, vol. 4, no. 2, p. 43, May 2023, doi: 10.1007/s43069-023-00223-6.
[28] N. Mohapatra, K. Shreya, and A. Chinmay, "Optimization of the Random Forest Algorithm," in *Lecture Notes on Data Engineering and Communications Technologies*, Springer Science and Business Media Deutschland GmbH, 2020, pp. 201–208. doi: 10.1007/978-981-15-0978-0_19.
[29] R. K. Sevakula and N. K. Verma, "MVPC—A Classifier with Very Low VC Dimension," in *Studies in Computational Intelligence*, Springer Science and Business Media Deutschland GmbH, 2023, pp. 23–39. doi: 10.1007/978-981-19-5073-5_3.
[30] C. Atik, R. A. Kut, R. Yilmaz, and D. Birant, "Support Vector Machine Chains with a Novel Tournament Voting," *Electronics*, vol. 12, no. 11, p. 2485, May 2023, doi: 10.3390/electronics12112485.
[31] E. Alfaro, M. Gámez, and N. García, "Ensemble Classifiers Methods," in *Ensemble Classification Methods with Applicationsin R*, Wiley, 2018, pp. 31–50. doi: 10.1002/9781119421566.ch3.
[32] S. Wu, J. Li, and W. Ding, "A geometric framework for multiclass ensemble classifiers," *Mach. Learn.*, Sep. 2023, doi: 10.1007/s10994-023-06406-w.
[33] L. I. Kuncheva and J. J. Rodríguez, "A weighted voting framework for classifiers ensembles,"

Knowl. Inf. Syst., vol. 38, no. 2, pp. 259–275, 2014, doi: 10.1007/s10115-012-0586-6.
[34] T. Turki and S. S. Roy, "Novel Hate Speech Detection Using Word Cloud Visualization and Ensemble Learning Coupled with Count Vectorizer," *Appl. Sci.*, vol. 12, no. 13, p. 6611, Jun. 2022, doi: 10.3390/app12136611.
[35] S. Kang, L. Kong, B. Luo, C. Zheng, and J. Wu, "Principle research of word vector representation in natural language processing," in *International Conference on Electronic Information Engineering and Computer Science (EIECS 2022)*, Apr. 2023, p. 133. doi: 10.1117/12.2668487.
[36] A. Esmaeilzadeh, J. R. F. Cacho, K. Taghva, M. E. Z. N. Kambar, and M. Hajiali, "Building Wikipedia N-grams with Apache Spark," Springer Science and Business Media Deutschland GmbH, 2022, pp. 672–684. doi: 10.1007/978-3-031-10464-0_45.
[37] N. S. Mamatov, N. A. Niyozmatova, A. N. Samijonov, and B. N. Samijonov, "Construction of Language Models for Uzbek Language," in *2022 International Conference on Information Science and Communications Technologies (ICISCT)*, Sep. 2022, pp. 1–4. doi: 10.1109/ICISCT55600.2022.10146788.
[38] S. Avasthi, R. Chauhan, and D. P. Acharjya, "Processing Large Text Corpus Using N-Gram Language Modeling and Smoothing," Springer Science and Business Media Deutschland GmbH, 2021, pp. 21–32. doi: 10.1007/978-981-15-9689-6_3.