khazanah informatika

Jurnal Ilmu Komputer dan Informatika

Identifying Hate Speech in Tweets with Sentiment Analysis on Indonesian Twitter Utilizing Support Vector Machine Algorithm

Imam Riadi[1*], Abdul Fadlil[2], Murni[3]
[1]Department of Information System
Universitas Ahmad Dahlan
Yogyakarta, Indonesia
[2]Department of Electrical Engineering
Universitas Ahmad Dahlan
Yogyakarta, Indonesia
[3]Master Program of Informatics
Universitas Ahmad Dahlan
Yogyakarta, Indonesia
[*]imam.riadi@is.uad.ac.id

Abstract-Twitter had 24 million users in Indonesia at the beginning of 2023. Despite having fewer users than other platforms, its fast and instant nature makes Twitter a significant source of information dissemination. Tweets shared on Twitter offer various advantages. However, it also has negative consequences, including the dissemination of fake news, instances of cyberbullying, and the expression of hate speech. Specifically, hate speech employs offensive language to discriminate against an individual or group based on race, ethnicity, nationality, religion, gender, sexual orientation, or other personal attributes, leading to discord. Such behavior comes under the jurisdiction of various legal statutes, including the Constitution, the Criminal Code, and the ITE Law. The primary objective of this research is to categorize tweets shared on Twitter into hate speech and non-hate speech sentiments, utilizing a Support Vector Machine (SVM) algorithm based on a dataset of 5,000 tweets. This research involved data preprocessing, labeling, feature extraction using TF-IDF, model training (80%), and testing (20%). The final stage includes enhancing SVM parameters through GridSearch and cross-validation methods (GridSearchCV), followed by analysis using a Confusion Matrix with the Matplotlib Library. Radial Basis Function (RBF) kernels, defined by parameters C=10 and gamma=0.1, exhibited the highest performance among SVM models, boasting an 84% accuracy. The RBF kernel also attained 85% precision, 97% recall, and a 91% F1-score for hate speech identification. In conclusion, the evaluation of SVM kernel performance highlights the superiority of RBF kernels in achieving the highest accuracy, complemented by nuanced insights into hate speech precision, recall, and F1-score values across various kernel types.

Keywords: Sentiment Analysis, Hate Speech, Indonesian Twitter, Support Vector Machine Algorithm, Machine Learning

Article info: *submitted May, 13, 2023, revised August, 3, 2023, accepted August, 28, 2023*

1. Introduction

Social media is crucial in the digital age because it has become a primary means of communication and interaction with others. It enables users to connect with friends, family, and people from different countries. Moreover, social media provides various benefits, including the latest information, business promotion, entertainment, and education. However, social media also has negative impacts, such as the spread of fake news, cyberbullying, and hate speech.

The number of active social media users in Indonesia is increasing rapidly. As of early 2023, there are 212.9 million active users, representing an increase of 5.2% (10 million) from 2022. The average time spent online daily by Indonesian internet users is 3 hours and 28 minutes. Social Media Facebook boasts the highest number of users among social media platforms in Indonesia, with 119.9 million, while Twitter, known as X, has 24 million users [1]. Despite having fewer users than other social media platforms, Twitter is known for its rapid dissemination of information through short and instantaneous tweets. [2] making it ideal for sharing news and opinions. However, this rapid spread of tweets or posts has also resulted in a surge of hate speech, including sensitive issues such as ethnicity, religion, race, and inter-group (SARA) [3], which has been the cause of depression and suicide [4].

Indonesia regulates hate speech through various laws and regulations, including the Constitution, Criminal Code, and ITE Law. While every Indonesian citizen has the right to express opinions and disseminate information freely, this right is subject to certain restrictions, such as the prohibition of speech that goes against people's moral values, promotes violence, or damages the dignity of others. The Criminal Code contains provisions that criminalize hate speech and incitement to violence, such as Article 156a, which prohibits the propagation of information that may incite hatred or hostility against certain groups based on SARA. The ITE Law of 2008 also regulates hate speech and other content on the Internet [5]. Specifically, Article 28(2) of the ITE Law prohibits the propagation of information that incites hatred or hostility against individuals or groups based on SARA. Violators may face imprisonment and fines. The government and other agencies have issued regulations and guidelines to combat hate speech and promote responsible social media use in addition to these three laws.

Sentiment analysis is required to identify hate speech on Twitter. Sentiment analysis involves evaluating the content of words or sentences in a text to determine whether it contains hate speech or non-hate speech. Machine learning algorithms provide a way to detect hate speech efficiently on social media platforms, reducing the need for extensive human effort and time-consuming data processing. Commonly used algorithms for sentiment analysis include Naïve Bayes [6], Support Vector Machine (SVM), and Decision Tree.

Numerous researches, such as [7]–[11], have extensively discussed hate speech on Twitter. Hate speech and offensive language have also been the focus of some of this research [12]–[15] and employed the SVM model for hate speech detection on social media, particularly Twitter. Based on the findings of these researches, it is evident that hate speech is prevalent on Twitter. Therefore, conducting regular sentiment analysis is crucial to monitor trends and determine whether the incidence of hate speech is decreasing or increasing.

This research utilized the SVM algorithm due to its previous success in achieving accuracy rates of 82% [16]commonly called PP 25 Tapera 2020, is one of the government's efforts to ensure that Indonesian people can afford houses. Tapera is a deposit of workers for house financing, which is refundable after the term expires. Immediately after enaction, there were many public responses regarding the ordinance. We investigate public sentiments commenting on the regulation and use Support Vector Machine (SVM and 70% [17]. The SVM classification process for hate speech detection involves several steps, such as preprocessing, labeling, feature extraction (TF-IDF), training and testing, GridSearchCV, and evaluation using a confusion matrix processed using Python programming language by utilizing Python-provided libraries and modules such as NumPy, Pandas, Scikit-learn, and Matplotlib.

This research utilizes the Scikit-learn library for analyzing SVM algorithms. Various SVM algorithms, including linear kernel, radial basis function (RBF), polynomial, and sigmoid, are examined to determine the most appropriate. The performance of the model is analyzed using a confusion matrix. Results include accuracy, precision, recall, and F1-score. In addition, the Matplotlib library helps to visualize the results of the SVM algorithm.

This research aimed to analyze 5,000 tweets on Twitter using the SVM algorithm and Python tools to classify them as either containing hate speech or not containing hate speech. By evaluating different SVM kernels, the research identified the most efficient models to detect hate speech in social media, especially Twitter.

2. Methods

The research employed a methodological approach to analyze hate speech and non-hate speech sentiments toward Twitter service users. The method involved collecting tweet data related to hate speech from Twitter users, followed by preprocessing, labeling, and term weighting using the TF-IDF method. The data was then split for analysis using the SVM algorithm, GridSearchCV, and evaluated using a confusion matrix. Figure 1 depicts the details of the research steps.

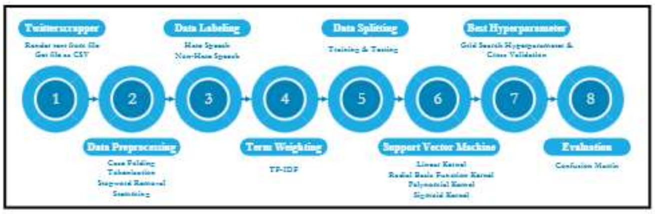
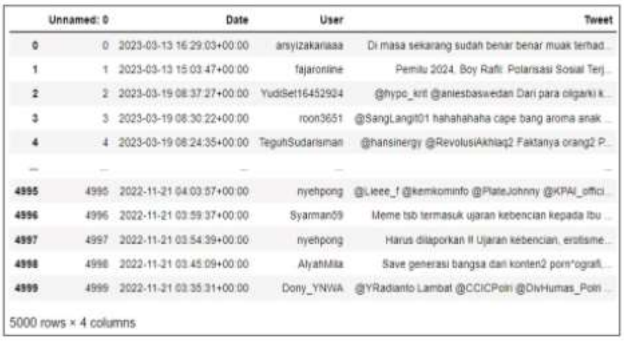


Figure 1. Research Steps

1. Sentiment Analysis

   Sentiment analysis is the technique of Natural Language Processing (NLP) and Machine Learning techniques to extract subjective information from textual data. This process is crucial for numerous applications, including social media monitoring, market research, and customer feedback. It enables the analysis of people's emotions and attitudes toward a specific topic, brand, product, or person by generating a label or score that indicates whether the sentiment conveyed in the text is positive, negative, or neutral [18]. The accuracy of sentiment analysis is affected by the quality and quantity of training data, the complexity of the language used, and the specific nuances of the industry or domain under examination.

2. Hate Speech
   Hate speech includes any form of expression or communication that demeans or poses a threat to an individual or group based on race, ethnicity, nationality, religion, gender, sexual orientation, or other personal attributes, including verbal abuse, offensive gestures, written messages, images, and actions that promote or incite violence against a particular group [19]. The detrimental impact of hate speech is concerning because it creates a hostile environment and can lead to discrimination, harassment, and acts of violence. It's important to distinguish between hate speech and legitimate criticism or expressions of disapproval.
3. Data Collection
   The research used 5,000 datasets collected from Twitter. The data was collected using the Twitter API and the Twitterscraper tool in Python with keywords related to hate speech [20]. It's vital to use the Twitterscraper responsibly and ethically, adhering to Twitter's guidelines and applicable laws. Figure 2 shows the dataset successfully collected using the Twitterscraper tool during the data collection process.



	Unnamed: 0	Date	User	Tweet
0	0	2023-03-13 16:29:03+00:00	arsyizakariaaa	Di masa sekarang sudah benar benar muak terhad...
1	1	2023-03-13 15:03:47+00:00	fajaronline	Pemilu 2024, Boy Rafli: Polarisasi Sosial Terj...
2	2	2023-03-19 08:37:27+00:00	YudiSet16452924	@hypo_krit @aniesbaswedan Dari para oligarki k...
3	3	2023-03-19 08:30:22+00:00	roon3651	@SangLangit01 hahahahaha cape bang aroma anak ...
4	4	2023-03-19 08:24:35+00:00	TeguhSudarisman	@hansinergy @RevolusiAkhlaq2 Faktanya orang2 P...
...	...	...	...	...
4995	4995	2022-11-21 04:03:57+00:00	nyehpong	@Lieee_f @kemkominfo @PlateJohnny @KPAI_offici...
4996	4996	2022-11-21 03:59:37+00:00	Syarman09	Meme tsb termasuk ujaran kebencian kepada ibu ...
4997	4997	2022-11-21 03:54:39+00:00	nyehpong	Harus dilaporkan !! Ujaran kebencian, erotisme...
4998	4998	2022-11-21 03:45:09+00:00	AlyahMila	Save generasi bangsa dari konten2 porn*ografi...
4999	4999	2022-11-21 03:35:31+00:00	Dony_YNWA	@YRadianto Lambat @CCICPolri @DivHumas_Polri ...

5000 rows × 4 columns

Figure 2. Data Collection

4. Data Preprocessing
   The collected Twitter datasets undergo preprocessing techniques that involve cleaning, transforming, and preparing the text data for sentiment analysis. This step is critical because text data often contains noise, irrelevant information, and formatting that requires removal before performing analysis.

   The main objectives of preprocessing in sentiment analysis are:

   1. Case folding: This step converts all letters in text to lowercase to reduce the complexity of text data and make it more consistent for further analysis.
   2. Tokenization: Text is parsed into words or individual tokens by separating text at space characters or using more sophisticated techniques such as regular expressions.
   3. Stopword removal: This step removes common words like 'the,' 'and,' and 'a' that lack significant meaning and do not contribute to sentiment analysis.
   4. Stemming: This step simplifies tokens to their base form by stripping away affixes, allowing words with similar meanings to be categorized more effectively.

   The preprocessing steps are presented in Figure 3.

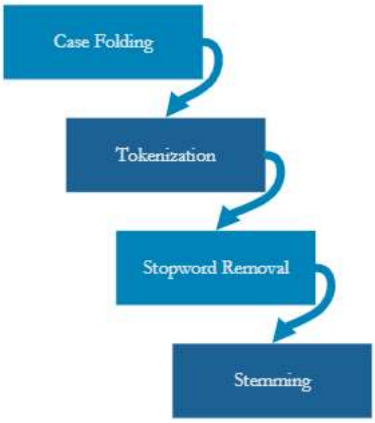


Figure 3. Data Preprocessing Steps

   The preprocessing steps in sentiment analysis can transform raw text data into a format that is easier to analyze using machine learning and natural language processing algorithms, thus improving the accuracy and reliability of sentiment analysis results.
5. Data Labeling
   In labeling data for sentiment analysis research, researchers often use a pre-existing dataset or create their dataset by manually Labeling a set of text documents or sentences. Machine learning algorithms can undergo training on the labeled dataset to perform the automatic sentiment classification of new, unlabeled text data.
   However, in this research, the labeling used was hate speech (HS) and non-hate speech (NHS) [21]. Sentiment analysis often involves assigning numerical values to text to quantify the sentiment expressed. These numerical values typically fall within a predefined range, often from -1 to 1. In this context, HS sentiment scores are 1, while NHS sentiment scores are 0.
   The formula for evaluating sentiment analysis labels is shown in equation (1). This approach allows the sentiment of the text to be quantified and serves as the basis for distinguishing between hate speech and non-hate speech in the dataset.

$$if \begin{cases} Hate\ Speech\ = 1 \\ Non - Hate\ Speech\ = 0 \end{cases} \tag{1}$$

   The process of data labeling is presented in Figure 4.

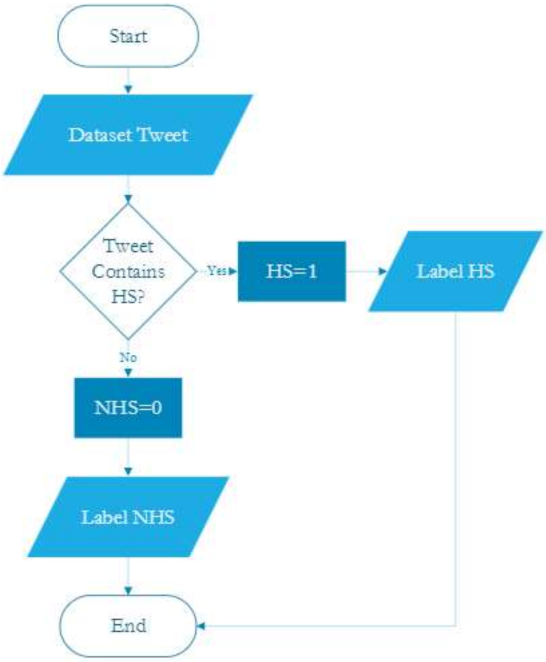


Figure 4. Data Labeling Steps

6. Term Weighting

Natural language processing (NLP) refers to a word or phrase that appears in a document or text corpus. The term can be a single word, such as 'cat' or 'house,' or a combination of words that form a phrase. Weighting and identifying these terms are crucial in text analysis, as they enable a more accurate representation of the content and meaning of the document.

The term frequency-inverse document frequency (TF-IDF) method is applied to achieve term weighting. This technique uses the term's frequency within a specific document or across the entire corpus to calculate weights that precisely indicate the term's significance within that particular document [22]. TF-IDF weights are determined by considering stemmed words.

By utilizing TF-IDF, the representation of text features becomes more informative and accurate. This approach emphasizes significant terms while mitigating the influence of frequent or irrelevant terms, thereby improving the performance of classification tasks and deriving more meaningful insights from text data.

The TF-IDF formula is presented in Equations (2), (3), and (4).

a) Term Frequency (TF)

$$TF_{(t,d)} = \frac{t}{D} \tag{2}$$

The variable t is the number of occurrences of term t in document d, and D is the total number of terms in document d.

b) Inverse Document Frequency (IDF)

$$IDF_{(t)} = \log \frac{N}{df_{(t)}} \tag{3}$$

N represents the total number of documents in the corpus, while $df_{(t)}$ represents the number of documents in which the term t appears.

c) TF-IDF

$$TF - IDF = TF_{(t,d)} * IDF_{(t)} \tag{4}$$

The calculation process of TF-IDF is presented in Figure 5.

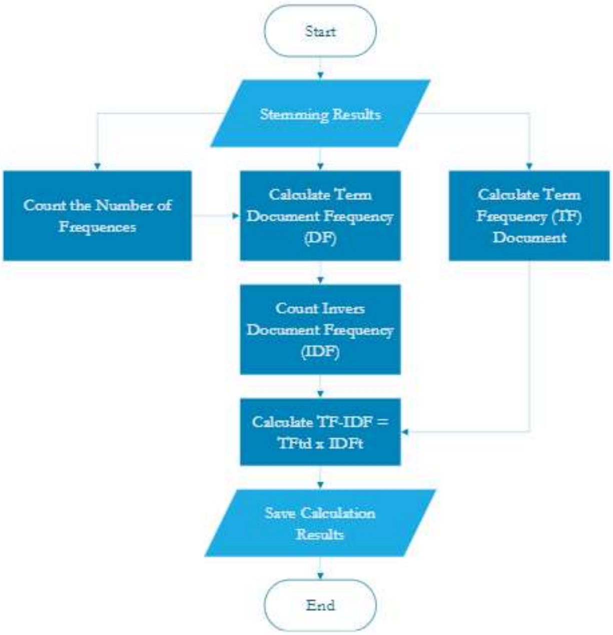

Start
Stemming Results
Count the Number of Frequences
Calculate Term Document Frequency (DF)
Calculate Term Frequency (TF) Document
Count Invers Document Frequency (IDF)
Calculate TF-IDF = TFtd x IDFt
Save Calculation Results
End

Figure 5. TF-IDF Steps

7. Data Splitting

Data splitting is a critical step in machine learning that involves partitioning a dataset into two distinct subsets: the training set and the test set. The training set is used for training machine learning models to identify patterns in the data and learn how to classify sentiment based on those patterns. The test set evaluates the performance of the trained model. By assessing the model with new and unseen data, data splitting provides a more accurate estimate of overall performance and helps prevent overfitting to training data.

The size of the test set depends on factors such as the dataset's size and the number of categories to be classified. Training and test sets must be similar regarding sentiment label distribution, text length, and other relevant features to ensure accurate model evaluation. This practice helps guarantee that the machine learning model is trained and evaluated on a representative sample of data, ultimately leading to more precise and reliable overall performance scores.

8. Support Vector Machine

Support Vector Machine (SVM) is a machine learning algorithm commonly applied in sentiment analysis. SVM is employed to create predictive models using supervised learning techniques and classify text documents, such as product reviews or social media messages, as positive, negative, or neutral [23]. SVM works by finding an optimal decision boundary that separates positive and negative data in the feature space. An illustration of SVM is presented in Figure 6.

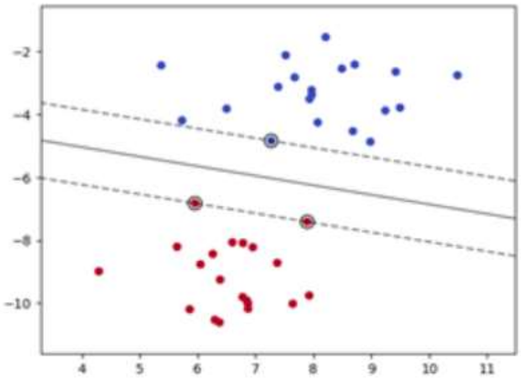


Figure 6. Support Vector Machine

Equation (5) is the SVM hyperplane equation:

$$w^{T}x + b = 0 \quad (5)$$

Where w is the weight vector, x is the feature vector, and b is the refractive constant. To find the best hyperplane, SVM minimizes objective functions called loss functions. Equation (6) shows this function.

$$w^{T}x + b = 0 \quad (6)$$

Where is the Euclidean norm of the weight vector, C is the penalty parameter, yi is the class label (1 for Hate Speech and 0 for Non-Hate Speech), xi is the feature vector, and the function Σ(max(0, 1 - y_i(w^Tx + b))) represents the objective function.SVM can easily find the optimal hyperplane separating the two classes when the data can be separated linearly [24]. However, when the data is not linear, the SVM uses an approach that is called kernel tricking.

The kernel trick in SVM is a mapping technique for data into higher-dimensional feature spaces that allow SVM to handle non-linear data. Some commonly used kernel functions for non-linear SVM are radial basis function (RBF), polynomial, and sigmoid kernels. Table 1 shows the formulas used for each SVM kernel.

Table 1. SVM Kernel Mapping Function

	Kernel	Mapping Function
1	Linear	K(x, y) =
2	RBF	K(x, y) =
3	Polynomial	K(x, y) =
4	Sigmoid	K(x, y) =

Where x and y represent the two input vectors that are to be mapped into the higher feature space, SVM kernel model performance is affected by hyperparameters.

The hyperparameters have to be determined before the training of the model and cannot be learned from the dataset but have to be defined by the researcher. The choice of hyperparameters can significantly affect the performance of the model. The hyperparameters used in the SVM kernel are:

a) C is a hyperparameter used for regularization that controls the trade-off between achieving low training error and testing error, also known as balancing overfitting and underfitting. A higher C value results in narrower margins and fewer misclassifications, while a lower C value results in wider margins and more classification errors. This parameter is common to all types of SVM kernels.

b) Gamma is a hyperparameter in SVM that controls the width of the kernel. A higher gamma value results in a more complex decision boundary, which can lead to overfitting if set too high. Conversely, a smaller gamma value leads to a smoother decision boundary, potentially resulting in underfitting if set too low. This hyperparameter applies only to RBF and Sigmoid kernels.

c) Degree is a hyperparameter exclusively used for polynomial kernels, controlling the degree of the polynomial function.

d) Coef0 is a hyperparameter used for Polynomial and Sigmoid kernels only to control the scaling factor of the kernel functions.

One common approach to achieving good SVM kernel model performance is to choose the correct hyperparameters. Grid search, a popular method, involves determining the optimal hyperparameter values from a range based on their performance on the validation set.

9. GridSearchCV

GridSearchCV is a technique used to find the optimal hyperparameters for a machine learning model. It involves exhaustively searching through a pre-defined range of hyperparameter values to find the combination that gives the best performance [25]. Finding the optimal hyperparameters for a machine learning model using GridSearchCV involves a systematic process. It starts with defining a grid of hyperparameter values to explore, including parameters such as learning rates and regularization strengths. Next, select a machine learning algorithm and create a GridSearchCV object that includes the model, hyperparameter grid, cross-validation settings, and the performance metric to be optimized. Fit this object to the training data, and GridSearchCV will exhaustively search through the hyperparameter combinations and identify the set that maximizes the chosen metric. Once the search is complete, the optimal hyperparameters appear along with the corresponding model. Finally, this model is evaluated on the test data to assess its generalization performance. This meticulous process effectively fine-tunes the model, improving its predictive accuracy and reliability.

10. Evaluation

The confusion matrix is a tabular representation used to assess the performance of a classification algorithm model [26]. It provides a means to compare the predicted results of the algorithm model with the actual values from the test set data. The confusion matrix consists of four primary labels: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). TP indicates the number of tweets accurately identified as containing hate speech, while TN represents the number of tweets correctly identified as non-hate speech. Conversely, FP refers to the number of tweets misclassified as including hate speech when the tweets are not. Finally, FN is the number of tweets misclassified as non-hate speech when the tweets contain hate speech. Table 2 shows the confusion matrix for predicting HS (Hate Speech) and NHS (Non-Hate Speech).

Table 3. Confusion Matrix

Actual	**Predicted**	
	HS	**NHS**
HS	TP	FN
NHS	FP	TN

The results of evaluating the SVM algorithm using the confusion matrix include accuracy, precision, recall or the true positive rate, and F1-score [27]–[29].

Accuracy measures the ratio of correctly classified tweets (TP+TN) to the total number of tweets, as represented by Equation (7).

$$Accuracy \;=\; \frac{TP+TN}{TP+TN+FP+FN} \qquad (7)$$

Precision quantifies the ratio of correctly classified tweets with hate speech (TP) to the total number of tweets predicted to have hate speech (TP+FP), expressed in Equation (8).

$$Precision \;=\; \frac{TP}{TP+FP} \qquad (8)$$

Recall assesses the ratio of correctly classified hate speech tweets (TP) to the total number of actual hate speech tweets, as outlined in Equation (9).

$$Recall \;=\; \frac{TP}{TP+FN} \qquad (9)$$

The F1-score, or the F-measure, quantifies the balance between precision and recall. Equation (10) outlines the formula used to calculate the F1-score.

$$F1-score \;=\; 2*\frac{precision*recall}{precision+recall} \qquad (10)$$

3. Result and Discussion

The report includes a detailed description of each step taken throughout the research.

a. Data Preprocessing

The Twitter dataset used in this research was unstructured and noisy due to its raw nature and the fact that it contained a significant number of characters. Preprocessing is a crucial step to convert the raw data into a format suitable for further processing. The preprocessing stage includes cleaning the data by removing unnecessary characters and punctuation. Table 3 presents case folding steps whose function is to convert all uppercase and lowercase letters to lowercase letters for ease of text processing and to handle unique cases where some characters may not have lowercase equivalents or may be represented differently by some characters.

Table 3. Case Folding Step Results

Before Case Folding	After Case Folding
Mendukung PDIP itu sama saja mendukung pengkhianat bangsa. PDIP menampung anak2 PKI Partai Komunis Indonesia. Slogan merakyat, kebijakan2 memiskinkan rakyat. https://t.co/aQnzfurgym https://t.co/oQx00XiYyu #NegaraKorupRepIndonesia #NegaraKorupRepIndonesia	mendukung pdip itu sama saja mendukung pengkhianat bangsa pdip menampung anak pki partai komunis indonesia slogan merakyat kebijakan memiskinkan rakyat

Before Case Folding	After Case Folding
In english Supporting PDIP is tantamount to supporting traitors to the nation. PDIP accommodated PKI children of the Indonesian Communist Party. Popular slogans, policies2 impoverish the people. https://t.co/aQnzfurgym https://t.co/oQx00XiYyu #CorruptStateRepIndonesia #CorruptStateRepIndonesia	In english supporting pdip is tantamount to supporting traitors to the nation, pdip accommodates pki children indonesian communist party popular slogan impoverishing the people policy

Table 4. Tokenization Step Results

Before Tokenization	After Tokenization
mendukung pdip itu sama saja mendukung pengkhianat bangsa pdip menampung anak pki partai komunis indonesia slogan merakyat kebijakan memiskinkan rakyat	['mendukung', 'pdip', 'itu', 'sama', 'saja', 'mendukung', 'pengkhianat', 'bangsa', 'pdip', 'menampung', 'anak', 'pki', 'partai', 'komunis', 'indonesia', 'slogan', 'merakyat', 'kebijakan', 'memiskinkan', 'rakyat']
In english supporting pdip is tantamount to supporting traitors to the nation, pdip accommodates pki children indonesian communist party popular slogan impoverishing the people policy	In english ['support', 'pdip', 'it', 'same', 'only', 'support', 'traitor', 'nation', 'pdip', 'accommodating', 'child', 'pki', 'party', 'communist', 'indonesia', 'slogan', 'populist', 'policy', 'impoverish', 'people']

Table 5. Stopword Removal Step Results

Before Stopword Removal	After Stopword Removal
['mendukung', 'pdip', 'itu', 'sama', 'saja', 'mendukung', 'pengkhianat', 'bangsa', 'pdip', 'menampung', 'anak', 'pki', 'partai', 'komunis', 'indonesia', 'slogan', 'merakyat', 'kebijakan', 'memiskinkan', 'rakyat']	['mendukung', 'pdip', 'mendukung', 'pengkhianat', 'bangsa', 'pdip', 'menampung', 'anak', 'pki', 'partai', 'komunis', 'indonesia', 'slogan', 'merakyat', 'kebijakan', 'memiskinkan', 'rakyat']
In english ['support', 'pdip', 'it', 'same', 'only', 'support', 'traitor', 'nation', 'pdip', 'accommodating', 'child', 'pki', 'party', 'communist', 'indonesia', 'slogan', 'populist', 'policy', 'impoverish', 'people']	In english ['support', 'pdip', 'support', 'traitor', 'nation', 'pdip', 'accommodating', 'child', 'pki', 'party', 'communist', 'indonesia', 'slogan', 'populist', 'policy', 'impoverish', 'people']

Table 4 presents tokenization steps whose function is to break down text into individual words or phrases for analysis.

Table 5 shows the stopword removal step, which removes common words having little meaning in a given language.

Table 6 shows the stemming step, which reduces words to their base form by removing prefixes, suffixes, and other inflectional endings. It is used in natural language processing to improve efficiency and accuracy by reducing the dimensionality of text. However, stemming may cause loss of information and errors due to the potential creation of non-existent or differing-context stems. Sastrawi is a popular stemming algorithm used for Indonesian language text processing.

Table 6. Stemming Step Results

Before Stemming	After Stemming
['mendukung', 'pdip', 'mendukung', 'pengkhianat', 'bangsa', 'pdip', 'menampung', 'anak', 'pki', 'partai', 'komunis', 'indonesia', 'slogan', 'merakyat', 'kebijakan', 'memiskinkan', 'rakyat']	['dukung', 'pdip', 'dukung', 'khianat', 'bangsa', 'pdip', 'tampung', 'anak', 'pki', 'partai', 'komunis', 'indonesia', 'slogan', 'rakyat', 'bijak', 'miskin', 'rakyat']
In english ['support', 'pdip', 'support', 'traitor', 'nation', 'pdip', 'accommodating', 'child', 'pki', 'party', 'communist', 'indonesia', 'slogan', 'populist', 'policy', 'impoverish', 'people']	In english ['support', 'pdip', 'support', 'betrayal', 'nation', 'pdip', 'tampung', 'children', 'pki', 'party', 'communist', 'indonesia', 'slogan', 'people', 'wise', 'poor', 'people']

The dataset is now in a format suitable for labeling after several preprocessing steps have been applied, such as case folding, tokenization, stopword removal, and stemming.

b. Data Labeling

Tweets are labeled or classified using two variables, HS and NHS. Based on the GitHub repository of Okkyibrohim, which can be found at https://github.com/okkyibrohim/id-multi-label-hate-speech-and-abusive-language-detection [30], the labeling in this research was done based on words containing hate speech. The repository contains text data containing hate speech and abusive language.

Table 7 categorizes commonly used hate speech words by their respective categories, including politics, race, religion, and gender. Hate speech and abusive language can be harmful and offensive, so it is necessary to recognize and address these forms of speech to create a safe and inclusive environment.

Table 7. Hate Speech Keywords

No	Politics	Race	Religion	Gender
1.	Antek	China	Hindu	Homo
2.	Asing	Komunis	Katolik	Sange
3.	Cebong	Sitip	Kristen	Transgender
4.	Komunis		Yahudi	
5.	Pendatang		Budha	
6.	PKI			
7.	Presiden			
8.	Rezim			

The labeling results of HS and NHS are presented in Table 8.

Table 8. Tweets Labeling Results

Tweets	Label
mendukung pdip itu sama saja mendukung pengkhianat bangsa pdip menampung anak pki partai komunis indonesia slogan merakyat kebijakan memiskinkan rakyat In english supporting pdip is tantamount to supporting traitors to the nation, pdip accommodates pki children indonesian communist party popular slogan impoverishing the people policy	HS
di masa sekarang sudah benar benar muak terhadap orang yang menyebar ujaran kebencian dan kelemahan didepan orang banyak dan dipermalukan di banyak pikiran orang percaya sii hukum karma pasti ada buat kedepan harus hati hati In english in the present time, it is really disgusted with people who spread hate speech and weakness in front of many people and are humiliated in many minds of believers sii the law of karma must exist for the future must be careful	NHS

Table 9 shows the final results of the labeling process on the Twitter dataset. The label conversion is using equation (1).

Table 9. Results of labelling conversion

Dataset	Label Description	Conversion	Total
Tweets	HS	1	3,988
	NHS	0	1,012
Total Dataset			5,000

c. Term Weighting

The TF-IDF method is applied to the data set to assign weights to each term. The researchers used two sample tweets from the dataset to demonstrate the manual TF-IDF calculations.

- Tweet 1: mendukung pdip itu sama saja mendukung pengkhianat bangsa pdip menampung anak pki partai komunis indonesia slogan merakyat kebijakan memiskinkan rakyat.
- Tweet 2: pdip kemarin ngomongnya partai ideologi islam bukan ideologi ideologi itu komunis.

TF is defined using equation (2), IDF in equation (3), and TF-IDF is determined using equation (4). Table 10 shows the results of TF-IDF calculations based on two example tweets.

Table 10. TF-IDF Calculation Results

TF-IDF Calculation								
Words	Count		TF		DF	IDF	TF * IDF	
	T1	T2	T1	T2			T1	T2
mendukung	2/20	0	0.100	0	1	0.301	0.030	0
pdip	2/20	1/11	0.100	0.091	2	0	0	0
itu	1/20	1/11	0.050	0.091	2	0	0	0
sama	1/20	0	0.050	0	1	0.301	0.015	0
saja	1/20	0	0.050	0	1	0.301	0.015	0
pengkhianat	1/20	0	0.050	0	1	0.301	0.015	0
bangsa	1/20	0	0.050	0	1	0.301	0.015	0
menampung	1/20	0	0.050	0	1	0.301	0.015	0
anak	1/20	0	0.050	0	1	0.301	0.015	0
pki	1/20	0	0.050	0	1	0.301	0.015	0
partai	1/20	1/11	0.050	0.091	2	0	0	0
komunis	1/20	1/11	0.050	0.091	2	0	0	0
indonesia	1/20	0	0.050	0	1	0.301	0.015	0
slogan	1/20	0	0.050	0	1	0.301	0.015	0
merakyat	1/20	0	0.050	0	1	0.301	0.015	0
kebijakan	1/20	0	0.050	0	1	0.301	0.015	0
memiskinkan	1/20	0	0.050	0	1	0.301	0.015	0
rakyat	1/20	0	0.050	0	1	0.301	0.015	0
kemarin	0	1/11	0	0.091	1	0.301	0	0
ngomongnya	0	1/11	0	0.091	1	0.301	0	0
ideologi	0	3/11	0	0.273	1	0.301	0	0
islam	0	1/11	0	0.091	1	0.301	0	0
bukan	0	1/11	0	0.091	1	0.301	0	0

d. Data Splitting

Sharing the training and testing datasets in this research is an 80/20 data split, where 80% of the data is used to train the model, and the remaining 20% is used to evaluate its performance. Therefore, in this research, SVM models were trained and tested using 80/20 data separation to provide reliable and accurate results.

e. Support Vector Machine and GridSearchCV

SVM is used to build a model that can separate data into two classes, HS and NHS, on the text that already has sentiment, and then this model is used to predict sentiment on new text data. The formula used is Equation (6). To improve model performance, researchers use linear kernels, RBF, polynomial, and sigmoid, then perform hyperparameter optimization with the Grid Search method to choose the best parameters that can produce the best model performance. The grid search technique is a powerful tool for finding the best combination of SVM hyperparameters by combining cross-validation with kernel functions. Through cross-validation, it's possible to estimate the performance of the model for different hyperparameter combinations. This research used cross-validation with a value of 10 (cv=10). Table 11 shows the combinations of the grid search for SVM kernel hyperparameters.

Table 11. Grid Search Hyperparameters

Kernel	Hyperparameters	Values
Linear	C	[0.1, 1, 10, 100, 1000]
RBF	C	[0.1, 1, 10, 100, 1000]
	Gamma	[0.1, 0.01, 0.001, 0.0001]
Polynomial	C	[0.1, 1, 10, 100, 1000]
	Gamma	[0.1, 0.01, 0.001, 0.0001]
	Degree	[2, 3, 4, 5]
	Coef	[0.0, 0.1, 0.5, 1.0]
Sigmoid	C	[0.1, 1, 10, 100, 1000]
	Gamma	[0.1, 0.01, 0.001, 0.0001]
	Coef	[0.0, 0.1, 0.5, 1.0]

Based on the combination of grid search hyperparameters and cross-validation, the best hyperparameters and the best model accuracy are obtained for each kernel. Here are the results of the optimal hyperparameters and accuracy for each SVM kernel model:

- Linear Kernel
  Best hyperparameters: {C: 1.9952623149688797}
  Accuracy of the best model: 0.78
- RBF Kernel
  Best hyperparameters: {C: 10, gamma: 0.1}
  Accuracy of the best model: 0.84
- Polynomial Kernel
  Best hyperpameters: {C: 10, coef0: 0.5, degree: 2, gamma: 0.01}
  Accuracy of the best model: 0.83
- Sigmoid Kernel
  Best hyperpameters: {C: 10, coef0: 0.0, gamma: 0.01}
  Accuracy of the best model: 0.83

The optimal combination of hyperparameters was determined for each kernel, and the test results showed that the RBF kernel achieved the highest accuracy rate of 0.84, making it the best performing model. Thus, using RBF kernels in SVM can significantly improve the accuracy of sentiment prediction on text data. Table 12 presents the best hyperparameters of each SVM model.

Table 12. Grid Search Hyperparameters Results

Kernel	Hyperparameters			
	C	gamma	degree	coef0
Linear	1.99	-	-	-
RBF	10	0.1	-	-
Polynomial	10	0.01	2	0.5
Sigmoid	10	0.01	-	0.0

f. Evaluation

The evaluation of SVM kernel models using a confusion matrix provides valuable insight into the ability of the model to correctly classify TP, TN, FP, and FN instances. By analyzing performance results using a confusion matrix, researchers gain insight into the accuracy, precision, recall, and F1-score, which provide a comprehensive assessment of the model's effectiveness and highlight potential areas for improvement. These metrics are determined using equations (7), (8), (9), and (10). Here are the Confusion Matrix analysis results for the linear kernel model: The model achieved 78% accuracy, 86% precision, 87% recall, and an 86% F1-score for HS detection. Table 13 summarizes those findings.

Table 13. Linear Kernel Test Results

	Precision	Recall	F1-score	Testing Data
1	0.86	0.87	0.86	799
0	0.45	0.44	0.44	201
Accuracy			0.78	1000

The confusion matrix analysis of the RBF kernel model shows an accuracy of 84%, a precision of 85%, a recall of 97%, and an F1-score of 91% for HS detection. Table 14 shows these results.

Table 14. RBF Kernel Test Results

	Precision	Recall	F1-score	Testing Data
1	0.85	0.97	0.91	799
0	0.73	0.31	0.43	201
Accuracy			0.84	1000

The confusion matrix analysis results for the polynomial kernel model show 83% accuracy, 85% precision, 96% recall, and a 90% F1-score for HS detection. These results are shown in Table 15.

Table 15. Polynomial Kernel Test Results

	Precision	Recall	F1-score	Testing Data
1	0.85	0.96	0.90	799
0	0.68	0.30	0.42	201
Accuracy			0.83	1000

The confusion matrix analysis results for the sigmoid kernel model show that the model produces an accuracy rate of 83%, a precision rate of 84%, a recall rate of 96%, and an F1-score of 90% for HS detection. These results are shown in Table 16.

Table 16. Sigmoid Kernel Test Results

	Precision	Recall	F1-score	Testing Data
1	0.84	0.96	0.90	799
0	0.66	0.28	0.39	201
Accuracy			0.83	1000

Plot the confusion matrix is a great way to visualize the performance of the SVM kernel model more clearly. The confusion matrix plot for the Linear kernel is presented in Figure 7.

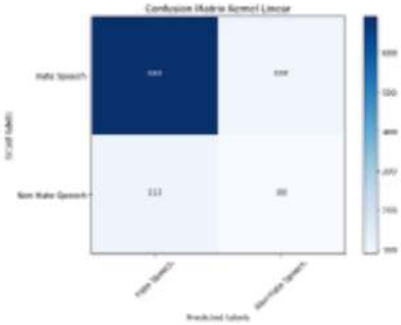


Figure 7. Confusion Matrix for The Linear Kernel

Table 17 shows the TP, TN, FP, and FN linear kernel primary label results.

Table 17. Linear Kernel Primary Label Results

Labeling	TP	TN	FP	FN
HS	692	88	113	107
NHS	88	692	107	113

Figure 8 showcases the confusion matrix plot for the RBF kernel model.

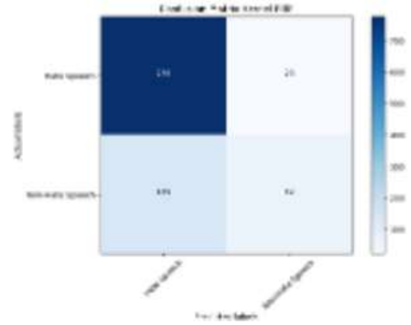


Figure 8. Confusion Matrix for The RBF Kernel

Table 18 shows the TP, TN, FP, and FN RBF kernel primary label results.

Table 18. RBF Kernel Primary Label Results

Labeling	TP	TN	FP	FN
HS	776	62	139	23
NHS	62	776	23	139

Figure 9 displays the confusion matrix plot for the polynomial kernel model.

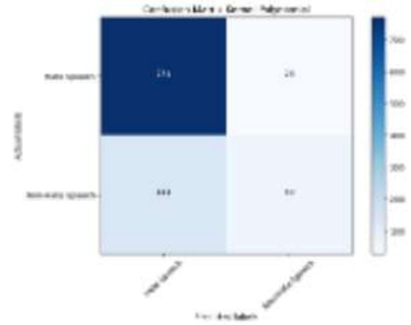


Figure 9. Confusion Matrix for The Polynomial Kernel

Table 19 shows the TP, TN, FP, and FN polynomial kernel primary label results.

Table 19. Polynomial Kernel Primary Label Results

Labeling	TP	TN	FP	FN
HS	771	60	141	28
NHS	60	771	28	141

Figure 10 showcases the confusion matrix plot for the sigmoid kernel model.

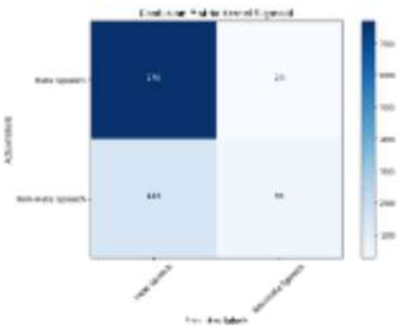


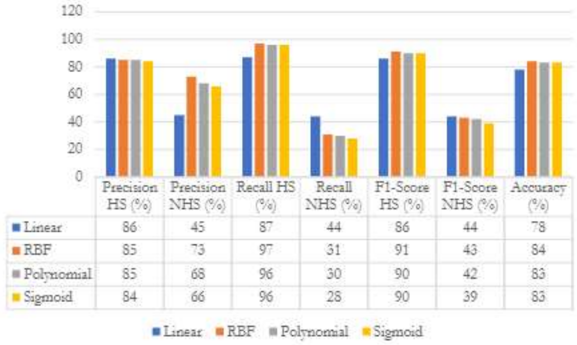
Figure 10. Confusion Matrix for The Sigmoid Kernel

Table 20 shows the TP, TN, FP, and FN sigmoid kernel primary label results.

Table 20. Sigmoid Kernel Primary Label Results

Labeling	TP	TN	FP	FN
HS	770	56	145	29
NHS	56	770	29	145

These findings highlight the model's capability to effectively classify tweets, as indicated by the successful identification of TP, TN, FP, and FN. Figure 11 displays the confusion matrix results presented in a diagram format to facilitate the comparison of performance differences for each kernel SVM testing.



	Precision HS (%)	Precision NHS (%)	Recall HS (%)	Recall NHS (%)	F1-Score HS (%)	F1-Score NHS (%)	Accuracy (%)
■ Linear	86	45	87	44	86	44	78
■ RBF	85	73	97	31	91	43	84
■ Polynomial	85	68	96	30	90	42	83
■ Sigmoid	84	66	96	28	90	39	83

Figure 11. Support Vector Machine Kernel Results Diagram

Evaluation of SVM kernel performance, as shown in Figure 11, shows that RBF kernels outperform other kernels with the highest accuracy value of 84%. In comparison, polynomial and sigmoid kernels achieve 83% accuracy, while linear kernels achieve 78% accuracy. Furthermore, when considering HS precision values, linear kernels stand out with the highest value of 86%, surpassing RBF and polynomial kernels with HS precision values of 85% and sigmoid kernels with HS precision values of 84%. In terms of HS recall values, RBF kernels show superior performance with a value of 97%, compared to polynomial and sigmoid kernels with a value of 96%, while linear kernels show a lower value of 87%. Regarding the HS F1-score, RBF kernels achieve the highest score of 91%, which is 1% higher than polynomial and sigmoid kernels with a HS F1-score of 90%, and the linear kernel with a HS F1-score of 86%.

Comparing these findings to specific related researches that explore hate speech identifying using SVM algorithms reveals meaningful insights. A research conducted in 2023 [7] achieved the highest accuracy of 89% by combining linear SVM with Count Vectors and TF-IDF, supported by precision, recall, and F1-score values of 88%, 89%, and 88%, respectively, employing a dataset of 25,502 instances. Likewise, another research from 2022 [10], based on a dataset of 30,000, demonstrated a remarkable accuracy of 97.71%, complemented by a precision of 98%, recall of 96%, and an F1-score of 97%. Furthermore, a 2022 research [17] achieved an accuracy of 70.03%, precision of 89.74%, recall of 45.24%, and F1-score of 60.15% with a dataset of 1,111 instances. These outcomes highlight the potential for improved performance through advanced methodologies and larger datasets. Additionally, an exploration of the support vector machine algorithm in a research from 2020 [3] resulted in an accuracy of 71.14%, precision of 70.56%, recall of 100%, and F1-score of 82.74% using a dataset of 201 instances. Moreover, another research conducted in 2022 [16]commonly called PP 25 Tapera 2020, is one of the government's efforts to ensure that Indonesian people can afford houses. Tapera is a deposit of workers for house financing, which is refundable after the term expires. Immediately after enaction, there were many public responses regarding the ordinance. We investigate public sentiments commenting on the regulation and use Support Vector Machine (SVM, utilizing a dataset of 519 instances, achieved an accuracy of 81.73%, precision of 78.27%, recall of 81.73%, and F1-score of 79.6%.

In conclusion, this research's findings reveal competitive results compared to related research. They offer promising directions for enhancement by leveraging advanced methodologies and extensive datasets, as highlighted by recent investigations in this domain. Furthermore, the algorithm's performance is notably impacted by the chosen SVM kernel, its associated parameters, and the unique dataset attributes. When the dataset lacks intricate patterns or complexity, the advantages of using a larger dataset might not stand out. It's crucial to stress that machine learning outcomes can exhibit substantial diversity due to specific problem details, dataset characteristics, algorithm intricacies, and experimental setups. The absence of a pronounced distinction between extensive and limited datasets in this research can provide a valuable viewpoint, showcasing the SVM algorithm's resilience and effectiveness in addressing the assigned task.

4. Conclusion and Future Work

Sentiment analysis of hate speech was successfully performed on a dataset of 5,000 Twitter data using the

SVM algorithm. This process involved several steps, including data preprocessing, labeling, feature extraction using the TF-IDF method, data splitting for training (80%) and testing (20%), GridSearchCV which combines cross-validation and SVM model parameters, as well as evaluation using a confusion matrix. The SVM model with the best performance was obtained using the RBF kernel with optimal parameters C=10 and gamma=0.1, resulting in a remarkable accuracy of 84%. In terms of accuracy comparison across different kernels, the RBF kernel outperforms polynomial and sigmoid kernels with respective accuracies of 83%, while the linear kernel achieves 78%. Additionally, for HS precision values, the linear kernel stands out with the highest score of 86%, surpassing the RBF and polynomial kernels with HS precision values of 85% and the sigmoid kernel with a HS precision value of 84%. Concerning HS recall values, the RBF kernel showcases superior performance at 97%, compared to polynomial and sigmoid kernels with values of 96%, while the linear kernel displays a lower value of 87%. In terms of HS F1-score, the RBF kernel attains the highest score of 91%, which is 1% higher than the polynomial and sigmoid kernels achieving HS F1-scores of 90%, and the linear kernel with a HS F1-score of 86%. The results of this research offer a significant point of reference for legal frameworks like the Constitution, the Criminal Code, and the ITE Law. These references contribute to the effective identification of Hate Speech. The knowledge gained can be employed to swiftly recognize and manage cases of hate speech, curbing its rapid dissemination and possible adverse outcomes.

In conclusion, the evaluation of SVM kernel performance highlights the superiority of RBF kernels in achieving the highest accuracy, complemented by nuanced insights into Hate Speech precision, recall, and F1-score values across various kernel types. Comparative analysis with related researches on hate speech identification using SVM algorithms underscores the potential for performance improvement through advanced methodologies and larger datasets, as evidenced by notable outcomes achieved in recent inquiries. Future work could focus on refining SVM parameter tuning, exploring ensemble methods, and exploring alternative approaches to enhance hate speech identification accuracy and robustness.

Reference

[1] S. Kemp, "Digital 2023 : Indonesia," Datareportal, 2023. https://datareportal.com/reports/digital-2023-indonesia?rq=digital 2023 indonesia (accessed May 06, 2023).

[2] F. E. Ayo, O. Folorunso, F. T. Ibharalu, and I. A. Osinuga, "Machine learning techniques for hate speech classification of twitter data: State-of-The-Art, future challenges and research directions," Computer Science Review, vol. 38, p. 100311, 2020, doi: 10.1016/j.cosrev.2020.100311.

[3] W. M. Baihaqi, M. Pinilih, and M. Rohmah, "Kombinasi K-Means Dan Support Vector Machine (Svm) Untuk K-Means and Support Vector Machine (Svm) Combination To Predict Sara Elements on Tweet," Jurnal Teknologi Informasi dan Ilmu Kompututer, vol. 7, no. 3, pp. 501–510, 2020, doi: 10.25126/jtiik.2020732126.

[4] N. Badri, F. Kboubi, and A. H. Chaibi, "Combining FastText and Glove Word Embedding for Offensive and Hate speech Text Detection," Procedia Computer Science, vol. 207, no. Kes, pp. 769–778, 2022, doi: 10.1016/j.procs.2022.09.132.

[5] A. F. dkk Hidayatullah, "Identifikasi konten kasar pada tweet bahasa Indonesia," Jurnal Linguistik Komputasional, vol. 2, no. 1, pp. 1–5, 2019, [Online]. Available: http://inacl.id/journal/index.php/jlk/article/view/15.

[6] Murni, I. Riadi, and A. Fadlil, "Analisis Sentimen HateSpeech pada Pengguna Layanan Twitter dengan Metode Naïve Bayes Classifier (NBC)," JURIKOM (Jurnal Riset Komputer), vol. 10, no. 2, pp. 0–9, 2023, doi: 10.30865/jurikom.v10i2.5984.

[7] L. Tabassum, A. Karim, L. T. Ava, A. Karim, and A. Charles, "Intelligent Identification of Hate Speeches to address the increased rate of Individual Mental Degeneration," Procedia Computer Science, vol. 219, pp. 1527–1537, 2023, doi: 10.1016/j.procs.2023.01.444.

[8] D. Mody, Y. D. Huang, and T. E. Alves de Oliveira, "A curated dataset for hate speech detection on social media text," Data in Brief, vol. 46, p. 108832, 2023, doi: 10.1016/j.dib.2022.108832.

[9] F. E. Ayo, O. Folorunso, F. T. Ibharalu, I. A. Osinuga, and A. Abayomi-Alli, "A probabilistic clustering model for hate speech classification in twitter," Expert Systems with Application, vol. 173, no. February, p. 114762, 2021, doi: 10.1016/j.eswa.2021.114762.

[10] A. M. U. D. Khanday, S. T. Rabani, Q. R. Khan, and S. H. Malik, "Detecting twitter hate speech in COVID-19 era using machine learning and ensemble learning techniques," International Journal of Information Management Data Insights, vol. 2, no. 2, p. 100120, 2022, doi: 10.1016/j.jjimei.2022.100120.

[11] M. A. Fauzi and A. Yuniarti, "Ensemble method for indonesian twitter hate speech detection," Indonesian Journal of Electrical Engineering and Computer Science, vol. 11, no. 1, pp. 294–299, 2018, doi: 10.11591/ijeecs.v11.i1.pp294-299.

[12] G. A. Marchellim and Y. Ruldeviyani, "Sentiment analysis of hate speech as an information tool to prevent riots and environmental damage," IOP Conference Series: Earth and Environmental Science, vol. 700, no. 1, 2021, doi: 10.1088/1755-1315/700/1/012024.

[13] M. Hayaty, S. Adi, and A. D. Hartanto, "Lexicon-Based Indonesian Local Language Abusive Words Dictionary to Detect Hate Speech in Social Media," Journal of Information Systems Engineering and Business Intelligence, vol. 6, no. 1, p. 9, 2020, doi: 10.20473/jisebi.6.1.9-17.

[14] M. Okky Ibrohim, E. Sazany, and I. Budi, "Identify abusive and offensive language in indonesian twitter using deep learning approach," Journal of Physics: Conference Series, vol. 1196, no. 1, 2019, doi: 10.1088/1742-6596/1196/1/012041.

[15] H. Watanabe, M. Bouazizi, and T. Ohtsuki, "Hate Speech on Twitter: A Pragmatic Approach to Collect Hateful and Offensive Expressions and Perform Hate Speech Detection," IEEE Access, vol. 6, pp. 13825–13835, 2018, doi: 10.1109/ACCESS.2018.2806394.

[16] R. H. Muhammadi, T. G. Laksana, and A. B. Arifa, "Combination of Support Vector Machine and Lexicon-Based Algorithm in Twitter Sentiment Analysis," Khazanah Informatika: Jurnal Ilmu Komputer dan Informatika, vol. 8, no. 1, pp. 59–71, 2022, doi: 10.23917/khif.v8i1.15213.

[17] A. Wikandiputra, Afiahayati, and V. M. Sutanto, "Identifying Hate Speech in Bahasa Indonesia With Lexicon-Based Features and Synonym-Based Query Expansion," ICIC Express Letters, vol. 16, no. 8, pp. 811–818, 2022, doi: 10.24507/icicel.16.08.811.

[18] A. N. Muhammad, S. Bukhori and P. Pandunata, "Sentiment Analysis of Positive and Negative of YouTube Comments Using Naïve Bayes – Support Vector Machine (NBSVM) Classifier," 2019 International Conference on Computer Science, Information Technology, and Electrical Engineering (ICOMITEE), Jember, Indonesia, 2019, pp. 199-205, doi: 10.1109/ICOMITEE.2019.8920923.

[19] G. del Valle-Cano, L. Quijano-Sánchez, F. Liberatore, and J. Gómez, "SocialHaterBERT: A dichotomous approach for automatically detecting hate speech on Twitter through textual analysis and user profiles," Expert Systems with Applications, vol. 216, no. June 2022, p. 119446, 2023, doi: 10.1016/j.eswa.2022.119446.

[20] K. Rakshitha, R. H M, M. Pavithra, A. H D, and M. Hegde, "Sentimental analysis of Indian regional languages on social media," Global Transitions Proceedings, vol. 2, no. 2, pp. 414–420, 2021, doi: 10.1016/j.gltp.2021.08.039.

[21] R. Rini, E. Utami and A. D. Hartanto, "Systematic Literature Review Of Hate Speech Detection With Text Mining," 2020 2nd International Conference on Cybernetics and Intelligent System (ICORIS), Manado, Indonesia, 2020, pp. 1-6, doi: 10.1109/ICORIS50180.2020.9320755.

[22] N. Hafidz and D. Yanti Liliana, "Klasifikasi Sentimen pada Twitter Terhadap WHO Terkait Covid-19 Menggunakan SVM, N-Gram, PSO," J. RESTI (Rekayasa Sistem dan Teknologi Informasi), vol. 5, no. 2, pp. 213–219, 2021, doi: 10.29207/resti.v5i2.2960.

[23] M. Rahardi, A. Aminuddin, F. F. Abdulloh, and R. A. Nugroho, "Sentiment Analysis of Covid-19 Vaccination using Support Vector Machine in Indonesia," International Journal of Advanced Computer Science and Applications (IJACSA), vol. 13, no. 6, pp. 534–539, 2022, doi: 10.14569/IJACSA.2022.0130665.

[24] A. A. Firdaus, A. Yudhana, and I. Riadi, "Analisis Sentimen pada Proyeksi Pemilihan Presiden 2024 menggunakan Metode Support Vector Machine," Decode Jurnal Pendidikan Teknologi Informasi, vol. 3, no. 2, pp. 236–245, 2023, [Online]. Available: http://journal.umkendari.ac.id/index.php/decode.

[25] M. Kolev, "XGB-COF: A machine learning software in Python for predicting the friction coefficient of porous Al-based composites with Extreme Gradient Boosting[Formula presented]," Software Impacts, vol. 17, no. June, p. 100531, 2023, doi: 10.1016/j.simpa.2023.100531.

[26] A. M. Pravina, I. Cholissodin, and P. P. Adikara, "Analisis Sentimen Tentang Opini Maskapai Penerbangan pada Dokumen Twitter Menggunakan Algoritme Support Vector Machine (SVM)," Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, vol. 3, no. 3, pp. 2789–2797, 2019, [Online]. Available: http://j-ptiik.ub.ac.id.

[27] I. Riadi, A. Fadlil, I. Julda, and D. E. P. Putra, "Batik Pattern Classification using Naïve Bayes Method Based on Texture Feature Extraction," Khazanah Informatika: Jurnal Ilmu Komputer dan Informatika, vol. 9, no. 1, 2023.

[28] A. Yudhana, I. Riadi, and M. R. Djou, "Determining eligible villages for mobile services using k- NN algorithm," Ilkom Jurnal Ilmiah, vol. 15, no. 1, pp. 11–20, 2023.

[29] H. Herman, I. Riadi, and Y. Kurniawan, "Vulnerability Detection With K-Nearest Neighbor and Naïve Bayes Method Using Machine Learning," International Journal of Artificial Intellegence Research, vol. 7, no. 1, 2023. doi: 10.29099/ijair.v7i1.795

[30] K. M. Hana, Adiwijaya, S. A. Faraby and A. Bramantoro, "Multi-label Classification of Indonesian Hate Speech on Twitter Using Support Vector Machines," 2020 International Conference on Data Science and Its Applications (ICoDSA), Bandung, Indonesia, 2020, pp. 1-7, doi: 10.1109/ICoDSA50139.2020.9212992.