www.dergipark.gov.tr
ISSN:2148-3736

El-Cezerî Journal of Science and Engineering
Vol: 10, No: 3, 2023 (600-611)
DOI: 10.31202/ecjse.1325078

ECJSE

Research Paper

Comparison of SVM and Naïve Bayes Algorithms with InNER Enriched to Predict Hate Speech

Isnen Hadi AL GHOZALI[1,a], Arif PIRMAN[1,b], Indra INDRA[1,c]

[1]Magister Ilmu Komputer, Fakultas Teknologi Informasi, Universitas Budi Luhur, Jakarta, Indonesia
[a]2111601163@student.budiluhur.ac.id

Received: 10.07.2023 **Accepted:** 25.09.2023

Abstract: Hate speech is one of the negative sides of social media abuse. Hate speech can be classified into insults, defamation, unpleasant acts, provoking, inciting, and spreading fake news (hoax). The purpose of this study is to compare the SVM and Naïve Bayes methods with feature extraction in the form of Indonesian NER (InNER) for detecting hate speech. To obtain the best model, this study applies five steps: a) data collection; b) data preprocessing; c) feature engineering; d) model development; and e) evaluating and comparing models. In this study, we have collected 7100 tweets as an initial dataset. After manual annotation, this study produced 1681 tweets: 548 insult tweets, 288 blasphemy tweets, 272 provocative tweets, and 573 neutral tweets. This study use two Python libraries that accommodate NER in Indonesian, namely the NLTK library and the Polyglot library. Based on the results of the evaluation of the proposed model, model 5, which develops the SVM algorithm with the NLTK library, is the best model proposed. This model shows an accuracy score of 92.88% with a precision of 0.93, a recall of 0.93, and an F-1 score of 0.92.

Keywords: SVM, Naïve Bayes, NER, Hate Speech

1. Introduction

In recent times, information and communication technology (ICT) has developed rapidly. The ICT sector has created social media as a force to be reckoned with in everyday life. Social media provides an open space for every connected individual to voice anything, including hate speech. Hate speech is legally defined as words, behavior, writing, or performances that are prohibited because they can trigger acts of violence and prejudice, either on the part of the perpetrator of the statement or the victim of the action. Hate speech is one of the negative sides of social media abuse. Based on the consensus of NLP researchers, hate speech is subjective and demeaning speech against protected characteristics that is expressed indirectly or directly to certain groups in textual form [1]. In the legal context, hate speech is a criminal act regulated in the Criminal Code (KUHP). According to Chief of Police Circular Letter Number SE/06/X/2015, hate speech can be classified into insults, blasphemy, unpleasant acts, provoking, inciting, and spreading fake news (hoax). Natural language processing (NLP) is one of the techniques used to detect hate speech in cyberspace. The NLP technique is a combination of artificial intelligence (AI) and linguistics, which is specified so that computers understand statements or words written in human language [2].

Based on research by the Pew Research Center in 2014, as many as 73% of adult internet users know someone is being harassed in cyberspace, and 40% have directly victims [3]. These results have sparked many studies to define post typologies that contain hate speech on social media, especially Twitter. This research conducted generally related to the development of machine learning for detecting hate speech. At a complex level, researchers can utilize neural networks to detect hate speech. Popular algorithms used are Long Short Term Memory (LSTM) [4] [5] [6] [7], Convolutional

How to cite this article
I. H. Al Ghozali, A. Pirman, Indra, "Comparison of SVM and Naïve Bayes Algorithms with InNER enriched to Predict Hate Speech," *El-Cezerî Journal of Science and Engineering*, Vol: 10, No: 3, 2023, pp. 600-611.
ORCID: [a]0000-0001-8155-6438; [b]0009-0005-5818-3991; [c]0000-0002-8180-4849

Neural Network (CNN) [7], and Gated Recurrent Unit (GRU) [8]. The classification algorithms used include Logistic Regression [9] [10] [11], Support Vector Machine (SVM) [11] [10] [12], Naive Bayes (NB) [11] [12] [13], Random Forest (RF) [9], and Gradient Boosting (GB) [9] [10]. The result of study [14] show that by using an open hate speech dataset, the Random Forest (RF) algorithm has the potential to be used for generic purposes. SVM is one of the supervised algorithms that uses non-linear mapping to transform the initial training data to a higher dimension. Meanwhile, Naïve Bayes uses the basic Bayesian theorem, which performs well when the data dimensions are high. The Bayesian classifier can be relied upon to calculate the most probable output based on the input. Apart from using classification algorithms, there is research using different methods, such as lexicon dictionaries [15], TweetNLP [16], and named entity recognition (NER) [17]. NER is used to solve the problem of extracting and classifying attributes in text, such as the name of a person, organization, or location. In the context of detecting hate speech, NER explores information and classifies the identity of the author, victim, or location that is the focus of hate-triggering events. This is after going through a process of identifying explicit or implicit expressions of hate and violations. Table 1 shows the results of the literature study that we carried out.

The results of the study [6] show that the proposed LSTM model's F-1 score is 0.63. This result is lower than the study [4], which reached 0.84. While study [5] achieved the highest score of 0.97 using the same method base. The use of the CNN method was only able to achieve an F-1 score of 0.72 for studies that used Arabic objects [7].]. For the hate speech detection method using the GRU method, the resulting F-1 score is 0.79 for Indonesian-language objects [8].]. The results of these studies generally show a lower F-1 score than the classification algorithm. Study [10] shows an F-1 score for the SVM method of 0.98 and the Naïve Bayes method of 0.97. The results of the study [11] show results that are close to the acquisition of an SVM score of 0.90 and a Naïve Bayes score of 0.93. Contrasting results were shown by the study [12] , which showed a high F-1 score in the SVM method, amounting to 0.99, but the Naïve Bayes method showed a very low value, amounting to 0.50. The results of this study prompted us to continue studies related to the detection of hate speech using the SVM and Naïve Bayes methods. We will examine the results of this contrasting model evaluation further by using different objects and adding Indonesian NER (InNER) features to develop the model.

The aims of this study is to compare the SVM and Naïve Bayes methods with feature extraction in the form of InNER for detecting hate speech. The use of the SVM and Naïve Bayes methods has been discussed in study [12]. However, this study was not optimal in developing the Naïve Bayes method so as to produce a small F-1 score. If the study [8] used Indonesian language tweet objects, but the feature was word embedding, this study added feature extraction in the form of InNER. InNER was added after preprocessing the data in order to get more in-depth information from a tweet. InNER can help the developed model understand the context of words in a sentence. For example, InNER can help detect the word "dog" in a tweet as having the meaning "pet" or "insult" in Indonesian language.

2. Methodology

The methodology used to detect hate speech for the insulting, blasphemy, and provocative fractions, using machine learning is shown in Figure 1. To obtain the best model, this study applies five steps: a) data collection; b) data preprocessing; c) feature engineering; d) model development; and e) evaluating and comparing models. At the data collection stage, the Twitter Application Programming Interface (API) is needed to obtain tweet data. This study uses the Twitter API v2 with a free license. The preprocessing stage involves humans and the Natural Language Toolkit (NLTK) library. This human role is one of the critical stages in getting the best input when building a model. After preprocessing the data, we add the InNER feature before processing the data into the model. The purpose of InNER is to search for and identify named entity types in text into predefined categories such as location, event, name of person, time, and organization.

Table 1. Systematic Literature Review

Research	Dataset	Method	Evaluation	Result
Pandey et al. [4]	Hate Speech and Offensive Language Dataset from Kaggle	MLP and Bi-LSTM	precision, recall and F-1 score	The MLP and Bi-LSTM models produce an F-1 score of 0.84. This result is higher than that of pure deep learning models.
Roy et al. [5]	Facebook dataset	LSTM	precision, recall and F-1 score	The proposed LSTM models produce an F-1 score of 0.98. This result is higher than that of pure machine learning models.
Abarna et al. [6]	Chelmis and Yao dataset	bi-LSTM, Intention detection model	precision, recall and F-1 score	The proposed Intention detection model produce F-1 score of 0.6327 and training/testing time of 0.14 s. The time for processing is lower due to better memory management by adding fast text to the model.
Faris et al. [7]	3696 Arabic tweet dataset	CNN and LSTM	accuracy, precision, recall and F-1 score	The proposed model produce accuracy of 0.67 and F-1 score of 0.72. The AraVec word embedding approach gets competent and good results with the model.
Patihullah and Winarko [8]	Twitter hate speech in Indonesian	GRU	accuracy	Experiment results reveal that the combination of word2vec and GRU provides the greatest accuracy of 0.93.
Oriola and Kotza [9]	21,350 tweets of South African discourses on Twitter	LogReg, Random Forest, Gradient Boosting, GRU	accuracy, precision, recall and F-1 score	Experiment results reveal that the Gradient Boosting provides the greatest accuracy of 0.881 and F-1 score of 0.63.
Khanday et al. [10]	Twitter dataset	LogReg, SVM, Gradient Boosting	accuracy, precision, recall and F-1 score	Gradient Boosting has the best performance, with 0.99 precision, 0.97 recall, 0.98 F-1 score, and 0.98 accuracy.
Viswapriya et al. [11]	Twitter dataset	LogReg, SVM, Naïve Bayes	accuracy	The findings indicated that Logistic Regression performed better, with an accuracy of 0.96.
Asogwa et al. [12]	Hate Speech Dataset	SVM and Naïve Bayes	precision, recall and F-1 score	Empirical testing of this approach yielded classification accuracy of around 0.99 and 0.50 for SVM and NB, respectively, over the test set.
Ivan et al. [13]	250 tweets hate speech in Indonesian	Naïve Bayes	precision, recall and F-1 score	The greatest accuracy result is 0.98, the highest precision result is 1.0, the highest recall result is 0.96, and the highest f-measure value is 0.98.
Fortuna et al. [14]	Hate speech datasets	Random Forest, SVM, BERT	F-1 score	The Random Forest (RF) algorithm has the potential to be used for generic purposes. SVM was the model that fared the poorest when it came to detecting hate speech.
Wang et al. [15]	11,917 comments to political news	Lexicon dictionary	precision, recall and F-1 score	Lexicon dictionary has the best performance, with 0.55 precision, 0.60 recall, 0.57 F-1 score, and 0.54 accuracy.
Camacho-Collados et al. [16]	-	TweetNLP	F-1 score	The findings indicated that TweetNLP performed better, with a F-1 Score of 0.55.
Englmeier and Mothe [17]	German tweets related to “refugees”	NER	human recognition	Hate speech on social media may be automatically classified using named-entity recognition.

The models developed in this study are SVM and Naïve Bayes. In research [12], the Naïve Bayes method produced an accuracy of 0.50, in contrast to research [13] , which produced an accuracy of 0.98. We identified a research gap in the two studies. As a comparison, the SVM method showed an accuracy above 0.90 in studies [10], [11], and [12]. These results made us interested in using the SVM and Naïve Bayes methods in the model we developed. SVM is used to find the best hyperplane by maximizing the distance between classes. Hyperplane is a function that can be used to separate classes. In SVM, the outer data object closest to the hyperplane is called a support vector. Objects called support vectors are the most difficult to classify because of their position, which almost overlaps with other classes. Given its critical nature, only this support vector is calculated to find the most optimal hyperplane by SVM. Meanwhile, Naïve Bayes is an algorithm based on the Bayes theorem, which is formulated as follows:

$$P(A \mid B) = P(B \mid A)P(A)P(B) \quad (1)$$

with:
P(A | B) : Probability that A occurs with evidence that B has occurred (superior probability)
P(B | A) : The probability that B will occur given the evidence that A has occurred.
P(A) : The probability that A occurs
P(B) : The probability that B occurs

The Naïve Bayes Classifier is a straightforward and very efficient classification technique that facilitates the development of rapid machine learning models capable of making swift predictions. This algorithm assumes that object attributes are independent. The probabilities involved in producing the final estimate are calculated as the sum of the frequencies from the "master" decision table.

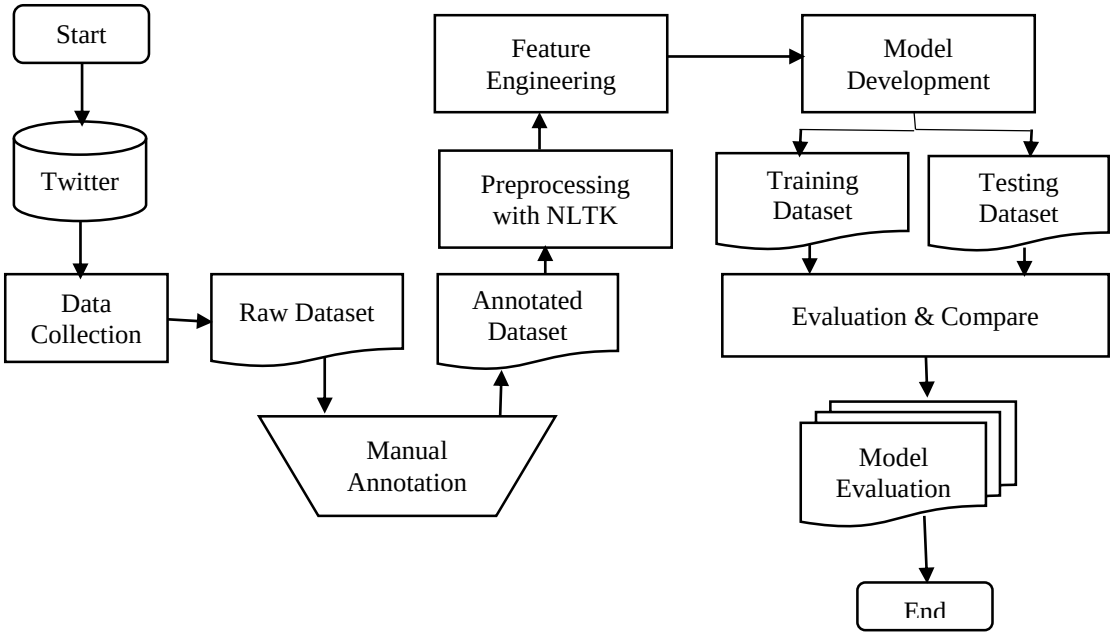


Figure 1. Research Framework

After the model was developed, we evaluated both models. Evaluation is done by using a confusion matrix. The study also measures precision, recall, and F-1 scores to analyze the performance of the developed model in detecting hate speech. The formula used for evaluation is as follows:

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

$$f1\ score = 2 \times \frac{recall \times precision}{recall + precision} \quad (4)$$

$$\text{Accuracy} = \frac{TN + TP}{TN + FN + TP + FN} \quad (5)$$

with:

True Positive (TP): The number of predictions where the classification is done correctly predicts the positive class as positive.

True Negative (TN): The number of predictions where the classification is done correctly predicts the negative class as negative.

False Positive (FP): The number of predictions where the classifier incorrectly predicts a negative class as positive.

False Negative (FN): Number of predictions where the classification incorrectly predicts a positive class as negative.

In summary, the metrics of accuracy, recall, and F1 score have significant importance in the evaluation of classification models. Performance evaluation metrics are valuable tools for assessing the effectiveness of a model, particularly in scenarios where there is imbalance distribution of data. The selection of a measure should be in accordance with the particular objectives and criteria of our model.

3. Results and Discussion

3.1. Data Collection

This study uses a dataset sourced from Twitter. The dataset was obtained using Orange Canvas and Twitter API. The keywords used to obtain the dataset related to insult, namely "*pelecehan* (harassment)", "*seksual* (sexual)", "*penghinaan* (insult)", "Ganjar Pranowo", "Puan Maharani", "*lambang negara* (national symbol)", and "*anjing* (dog)". To obtain datasets related to blasphemy, the keywords used were "*agama* (religion)", "*penghinaan* (blasphemy)", "*Kadrun* (in Indonesian abbreviation; kadal gurun (desert lizard))", "*rasis* (racist)", "*Islamofobia* (Islamophobia)". To obtain datasets related to provocative, the keywords used were "*provokasi* (provocation)", “provocateur (provokator)”, “*demo* (demo)”, “buzzer”, “Anies Baswedan”. Data collection was carried out on April 12–14, 2023. The data collected was 7100 tweets, consisting of 2100 insulting tweets, 2500 blasphemous tweets, and 2500 provocative tweets.

3.2. Preprocessing Data

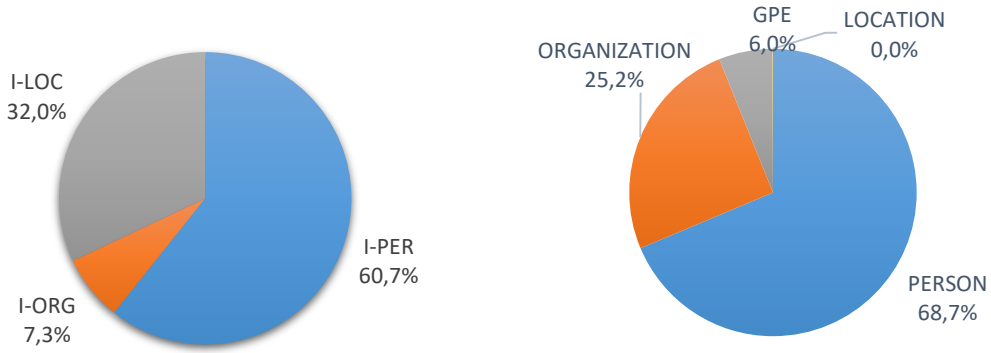
The data preprocessing stage consists of human annotation and the NLTK platform. Human annotation is the process of labeling collected tweets based on the context of the meaning of the words used in the sentence (semantic approach). At this stage, we check each tweet to determine its eligibility, meaning, and labeling. The labeling of humiliation and defamation is guided by the Indonesian Dictionary and Chief of Police Circular Letter Number SE/06/X/2015. For the insult dataset, out of 2100 tweets, there were 1514 tweets that were out of context; 38 tweets were duplicated, so 548 tweets of insults were obtained. As for the blasphemy dataset, out of 2500 tweets, there were 2159 tweets that were out of context, and 53 tweets were duplicated, resulting in 288 tweets of blasphemy. For provocative tweets, out of 2500 sample tweets, there are 36 tweets that are duplicated and 2192 tweets that are out of context, so 272 tweets are labeled as provocative tweets. Tweets that are out of context and duplicates that are not used will be eliminated from the dataset. The dataset is still imbalanced, so we added 573 tweets that were not included in the category of insults, blasphemy, or provocative, labeled as neutral tweets. The addition of this sample is because we used an oversampling approach. We are trying to balance the insult tweet, which is almost twice

the number of blasphemy tweets and provocative tweets. If the proportion of the sample category insults: blasphemy: provocative: neutral close to 30:20:20:30, overfitting on the model can be reduced.

After the human annotation process, the dataset is further processed using the NLTK library in Python. At this stage, case folding, tokenizing, frequency distribution, stop word removal, and normalization are carried out. The tokenizing stage eliminates numbers, URLs, and special characters (punctuation) until sentences are divided into tokens. At the stop word removal stage, the word "wkwk" is added to the source code. The normalization process uses an Indonesian slang word dictionary with the addition of 131 new words based on observations on the Stemmer results. The addition of data to the normalization dictionary is expected to increase the accuracy score of the model to be developed. Study [18] suggests stemming data as one of the bases for normalizing words. After the normalization process, the dataset is divided into two, namely the training and testing datasets, with a ratio of 80:20. This study uses a random state of 7 in dividing training and testing data.

3.3. Feature Engineering

Named entity recognition (NER) refers to lexical and semantic problems [19]. NER is a word or part of text that has personal data. In general, NER performs two important subtasks. First, a word or piece of text must act as a named entity. Second, the introduction of named entity types, for example, the name of the person, the name of the organization, and the name of the place or location. In the Python library, there are two libraries that accommodate NER in Indonesian, namely the NLTK library and the Polyglot library. The entity tags generated with the NLTK library are 2119 entities, consisting of 1455 Person tags, 128 GPE (Location) tags, 535 Organization tags, and 1 Location tag. Meanwhile, using the Polyglot library, 331 entity tags were identified: 201 I-PER (Person), 106 I-LOC (Location), and 24 I-ORG (Organization). Figure 2 shows the results of NER tags on datasets that have passed the data preprocessing stage. Figure 2 (a) shows the distribution of NER tag results using the Polyglot library, there are 60.7% tagging I-PER (Person), 32.0% I-LOC (Location), and 7.3% I-ORG (Organization). Figure 2 (b) shows the distribution of NER tag results using the NLTK library: there are 68.7 Person, 6.0% GPE, 25.2% Organization, and 0.0% Location. These libraries show the largest tagging on the Person entity; the identification results for both are above 60%. The difference is in the Organization and Location entities; the Polyglot library identifies more Location entities than Organization entities. The opposite happens when using the NLTK library.



(a) (b)

Figure 2. NER tag (a) with library Polyglot (b) with library NLTK

3.4. Model Developing

In developing the model, there are several important parameters, namely data preprocessing, InNER, and the algorithm used. Table 2 shows the six models developed based on parameter variations. In model 2 and model 5, the preprocessing stage does not apply case folding. This is due to the use of

the NLTK library, which is case sensitive. The NLTK library has difficulty identifying a named entity if all words are in lower case. For feature engineering parameters, in general, the study developed three schemes, namely the use of the Polyglot library, the NLTK library, and without NER. For the proportion of training and testing data, this study uses a ratio of 80:20 for all models. We chose this proportion because it produces the best outcomes and is supported by research [13]. Preliminary test results on the proportion of 60:40 using the SVM and Naïve Bayes basic models without InNER show an accuracy of 83.95% and 71.32%, respectively. While the preliminary test of the proportion of 70:30 using the basic model of SVM and Naïve Bayes without InNER shows an accuracy of 88.71% and 73.66%. The selection of these proportions is in accordance with the results of the study [13]. The algorithms to be compared in this study are Naïve Bayes and SVM. In general, the dataset to be processed will undergo four important stages: data preprocessing, feature extraction with InNER, split testing training, and algorithm processing. The results of model development will be evaluated using precision, recall, f-1 score, and accuracy values.

Table 2. Model development schematic

Parameter	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
Preprocessing data:						
a. Case folding	Yes	No	Yes	Yes	No	Yes
b. Tokenizing	Yes	Yes	Yes	Yes	Yes	Yes
c. Frequency distribution	Yes	Yes	Yes	Yes	Yes	Yes
d. Stop word removal	Yes	Yes	Yes	Yes	Yes	Yes
e. Normalization	Yes	Yes	Yes	Yes	Yes	Yes
InNER:						
a. Library Polyglot	Yes	No	No	Yes	No	No
b. Library NLTK	No	Yes	No	No	Yes	No
Training Testing split	80:20	80:20	80:20	80:20	80:20	80:20
Algoritms:						
Naïve Bayes	Yes	Yes	Yes	No	No	No
SVM	No	No	No	Yes	Yes	Yes

For the SVM algorithm, the kernel parameters are 'linear', C is 1.0, and gamma is 'scale'. The test data used has been converted into a TF-IDF vector representation. Meanwhile, the Naive Bayes algorithm uses a multinomial model that can process discrete features of text data represented by TF-IDF values. Next, the fit model will train a Naive Bayes model using the given training data and be ready to make predictions on new data.

3.5. Model Evaluating and Comparing

Figure 3 shows the confusion matrix for each developed model. The confusion matrix uses 337 data points of testing data. The proportion of the number of labels for each type of hate speech 116 tweets insult, 40 tweets blasphemy, 40 tweets are provocative, and 141 tweets are neutral. Model 1, model 2 and model 3, which uses the Naïve Bayes algorithm, can predict tweet insults at the same level. These models are better at predicting tweet insults than models using the SVM algorithm. This is different from blasphemy tweets and provocative tweets; models that use the SVM algorithm are better at predicting than models that use the Naïve Bayes algorithm. This also applies to neutral tweet predictions. Model 1 and model 2 mostly make provocative tweet prediction errors, which are

predicted as neutral tweets. Model 3 mispredicts more neutral tweets as insult tweets. Model 4, model 5, and model 6 show the most prediction errors in blasphemy tweets, which are predicted to be neutral tweets. The three models built using the SVM algorithm show good predictive results. Meanwhile, the other three models that use the Naïve Bayes algorithm show quite high prediction errors.

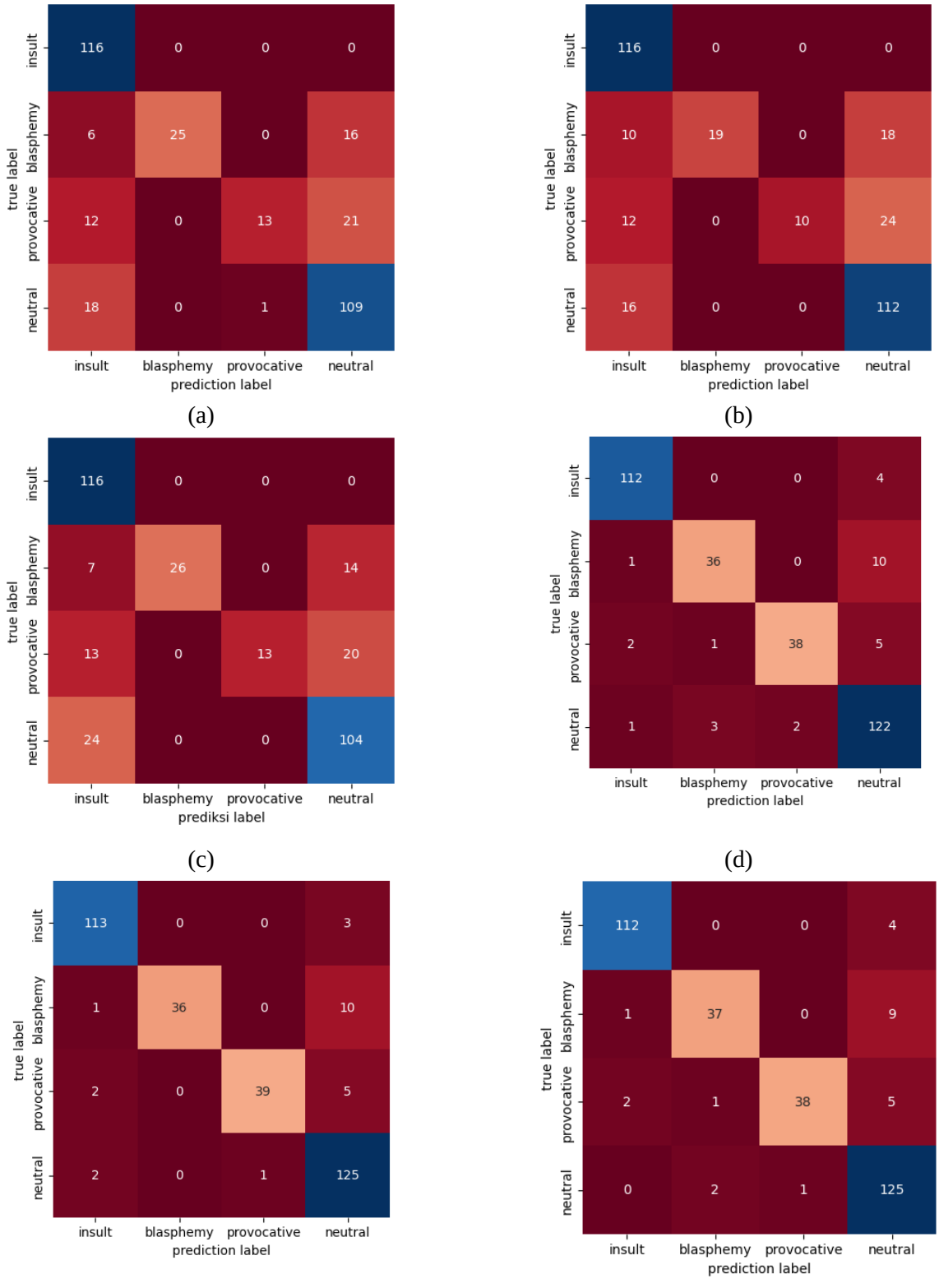


Figure 3. Confusion matrix: (a) Model 1; (b) Model 2; (c) Model 3; (d) Model 4; (e) Model 5; and (f) Model 6.

The results of the evaluation of the developed model are shown in Table 3. The range of precision scores is 0.81–0.93. The highest precision scores are found in model 5 and model 6. Both models use the SVM algorithm; the difference between the two is in the NER feature. Model 5 uses the NLTK library, while model 6 does not use the NER feature. The lowest precision scores are found in models that use the Naïve Bayes algorithm, namely model 1, model 2, and model 3. Recall scores are in the range of 0.76–0.93. The highest recall score of 0.93 is found in model 5 and model 6. The lowest recall score is found in model 2, which uses a combination of the Naïve Bayes algorithm with the InNER feature using the NLTK library. While the F-1 score is in the range of 0.73–0.93. Model 2 shows the lowest F-1. For F-1, the highest score is found in model 5. Model 5 uses the SVM algorithm with the InNER feature using the NLTK library. The evaluation results show various results for the accuracy score, the score range is 76.26–92.88%. The lowest accuracy score is found in model 2 and the highest in model 5. Based on the results of the evaluation parameter measurements, model 2 shows the lowest values for three of the four test parameters. Different things are shown by model 5 which shows the highest value on the four test parameters. This indicates that the InNER feature has a different effect on the Naïve Bayes and SVM algorithms. Model 3 and model 6 which are the basic models of the Naïve Bayes and SVM algorithms, tend to show results that are between the lowest and highest extreme values.

Table 3. Model evaluation

Evaluation parameter	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
Precision	0.81	0.81	0.81	0.92	0.93	0.93
Recall	0.78	0.76	0.77	0.91	0.93	0.93
F-1 score	0.76	0.73	0.75	0.91	0.93	0.92
Accuracy	78.04%	76.26%	76.85%	91.39%	92.88%	92.58%

3.6. Discussion

Research [20] started to introduce a class 4 annotation for hate speech research. In that study, the annotations used were: acceptable, inappropriate, offensive, and violent. The use of four types of annotation is less popular because of its complexity. The issue of complexity in manual annotation causes many studies to prefer binary or ternary annotations. The survey paper conducted by the study [21] only found three articles using this annotation. This study also notes that the comparative model is the most popular topic.

At the data preprocessing stage, there are two models that do not apply the case folding stage, namely model 2 and model 5. This is related to the nature of NLTK library, which is case-sensitive to uppercase and lowercase. The entity tags generated by the NLTK library contain 1541 entities. The results of entity tags that use a large number of NLTK libraries indicate a high error rate in identifying named entities. For example, the word "Because (Because)" is identified as a Person entity, even though this word is a conjunction. By using the Polyglot library, the level of identification errors is lower, although there are still identification errors with named entities, for example, the word "Indonesia" is identified as I-ORG, even though this word is a location. Even though there are still identification errors, the evaluation results show that InNER can improve the accuracy level. The combination of the NLTK library with the SVM algorithm shows the best level of accuracy in this study, as shown by model 5. Meanwhile, the Polyglot library can increase the accuracy rate by up to 1.19% compared to models that do not use NER. These results support the studies [22] and [19]. Although the combination of the NLTK library with the Naïve Bayes algorithm can reduce the accuracy of the model. This also applies to the combination of the Polyglot library with the SVM algorithm. The effect of reducing the accuracy score of the model reaches 1.49%, higher than the combination of the NLTK library and the Naïve Bayes algorithm.

The accuracy score of the model using the Naïve Bayes algorithm in this study is higher than that of study [23], which only achieved 57%, and study [24] , which achieved 74%. However, the highest accuracy score of model 1 using the Naïve Bayes algorithm and the Polyglot library is still lower than study [25] , which reached 83.1%, and study [13] , which reached 98%. The accuracy of the developed model is not as good as in the study [13] because the number of datasets used is 6.7 times greater and there are more hate speech categories. This is inversely proportional to the model developed with the SVM algorithm. The model developed using the SVM algorithm produces the highest accuracy score of 92.88%. This score was higher than study [26] (87.4%), study [9] (89.7%), and study [27] (81.3%). Although this result is still lower than the study [12], which reached 98.9%, When compared to studies using deep learning, the models developed show mixed results. A higher level of accuracy was generated using the BERT algorithm in the study [27], resulting in an accuracy rate of 88.5%, higher than the model developed using the Naïve Bayes algorithm, but lower than the model using SVM. Whereas study [28] with the LSTM algorithm and study [8] with the Gated Recurrent Unit (GRU) algorithm produced a higher accuracy score than all research models, 97.9% and 92.96%, respectively. Based on these results, model 5, which develops the SVM algorithm with the NLTK library, is the best model proposed. This result is empirical evidence that the combination of the SVM algorithm and the NLTK library can be developed as a hate speech detector with high accuracy. Apart from that, the results of this research can be a trigger to develop a combination of the Naïve Bayes algorithm and the Polyglot library in order to close the research gap found in this study.

4. Conclusions

In this study, we have collected 7100 tweets as an initial dataset. After manual annotation, this study produced 1681 tweets: 548 insult tweets, 288 blasphemy tweets, 272 provocative tweets, and 573 neutral tweets. Tweets that have been annotated are followed by data preprocessing, including case folding, tokenizing, frequency distribution, stop word removal, and normalization. At this stage, the dataset is ready to be assigned an entity tag using the Polyglot library or the NLTK library. The dataset that has received the entity tag will be used to develop a model based on the Naïve Bayes or SVM algorithm.

Based on the results of the evaluation of the developed model, model 5, which develops the SVM algorithm with the NLTK library, is the best model proposed. This model shows an accuracy rate of 92.88% with a precision of 0.93, a recall of 0.93, and an F-1 score of 0.92. The addition of the InNER feature to the NLTK library has been proven to increase model accuracy by 0.3 compared to models that do not use InNER. The addition of the InNER feature to the Polyglot library has a negative impact on the accuracy of the model based on the SVM algorithm. Therefore, adding the InNER feature requires considering the right library.

For future studies, model 2 is the model with the most potential for further development. This is because the potential of the Polyglot library to improve the accuracy of models based on the Naïve Bayes algorithm is still large. Future research can expand the scope of hate speech annotations, especially for hoax tweet detectors. The model in this study can be developed to detect hate speech on other social media platforms such as YouTube, Facebook, and Instagram comments. Hate speech detection methods can also be developed with a combination of NER and deep learning, which have not been explored in more depth.

Authors' Contributions

The IHA collects tweet data and performs an initial annual annotation for insult and blasphemy tweets. AF performs manual annotation for provocative and neutral tweets and adds words to the normalization dictionary. IHA and AF conducted a manual annotation peer review. IHA, in collaboration with AF, wrote up the article. IHA developed source code for preprocessing and

creating models. I reviewed the model that had been developed and the initial draft of the article. All authors read and approved the final manuscript.

Competing Interests

The authors declare that they have no competing interests.

References

[1]. J. Govers, P. Feldman, A. Dant, and P. Patros, "Down the Rabbit Hole: Detecting Online Extremism, Radicalisation, and Politicised Hate Speech," *ACM Comput. Surv.*, p. 3583067, Feb. 2023, doi: 10.1145/3583067.

[2]. D. Khurana, A. Koli, K. Khatter, and S. Singh, "Natural language processing: state of the art, current trends and challenges," *Multimed. Tools Appl.*, vol. 82, no. 3, pp. 3713-3744, Jan. 2023, doi: 10.1007/s11042-022-13428-4.

[3]. A. Shvets, P. Fortuna, J. Soler, and L. Wanner, "Targets and Aspects in Social Media Hate Speech," in *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, Online: Association for Computational Linguistics, Aug. 2021, pp. 179-190. doi: 10.18653/v1/2021.woah-1.19.

[4]. S. S. Pandey, I. Chhabra, R. Garg, and S. Sahu, "Hate Speech Detection," *Int. J. Adv. Eng. Manag. IJAEM*, vol. 5, no. 4, pp. 897–903, 2023, doi: 10.35629/5252-0504897903.

[5]. S. S. Roy, A. Roy, P. Samui, M. Gandomi, and A. H. Gandomi, "Hateful Sentiment Detection in Real-Time Tweets: An LSTM-Based Comparative Approach," *IEEE Trans. Comput. Soc. Syst.*, pp. 1-10, 2023, doi: 10.1109/TCSS.2023.3260217.

[6]. S. Abarna, J. I. Sheeba, S. Jayasrilakshmi, and S. P. Devaneyan, "Identification of cyber harassment and intention of target users on social media platforms," *Eng. Appl. Artif. Intell.*, vol. 115, p. 105283, Oct. 2022, doi: 10.1016/j.engappai.2022.105283.

[7]. H. Faris, I. Aljarah, M. Habib, and P. Castillo, "Hate Speech Detection using Word Embedding and Deep Learning in the Arabic Language Context:," in *Proceedings of the 9th International Conference on Pattern Recognition Applications and Methods*, Valletta, Malta: SCITEPRESS - Science and Technology Publications, 2020, pp. 453-460. doi: 10.5220/0008954004530460.

[8]. J. Patihullah and E. Winarko, "Hate Speech Detection for Indonesia Tweets Using Word Embedding And Gated Recurrent Unit," *IJCCS Indones. J. Comput. Cybern. Syst.*, vol. 13, no. 1, p. 43, Jan. 2019, doi: 10.22146/ijccs.40125.

[9]. O. Oriola and E. Kotze, "Evaluating Machine Learning Techniques for Detecting Offensive and Hate Speech in South African Tweets," *IEEE Access*, vol. 8, pp. 21496-21509, 2020, doi: 10.1109/ACCESS.2020.2968173.

[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," *Int. J. Inf. Manag. Data Insights*, vol. 2, no. 2, p. 100120, Nov. 2022, doi: 10.1016/j.jjimei.2022.100120.

[11]. S. E. Viswapriya, A. Gour, and B. G. Chand, "Detecting Hate Speech and Offensive Language on Twitter using Machine Learning," *Int. J. Comput. Sci. Mob. Comput.*, vol. 10, no. 4, pp. 22-27, Apr. 2021, doi: 10.47760/ijcsmc.2021.v10i04.004.

[12]. D. C. Asogwa, C. I. Chukwuneke, C. C. Ngene, and G. N. Anigbogu, "Hate Speech Classification Using SVM and Naive BAYES." Mar. 21, 2022. doi: 10.9790/0050-09012734.

[13]. I. Ivan, Y. A. Sari, and P. P. Adikara, "Klasifikasi Hate Speech Berbahasa IndonesiadiTwitterMenggunakan Naive Bayes dan Seleksi Fitur Information Gain dengan Normalisasi Kata," *J. Pengemb. Teknol. Inf. Dan Ilmu Komput.*, vol. 3, no. 5, pp. 4914-4922, 2019.

[14]. P. Fortuna, J. Soler-Company, and L. Wanner, "How well do hate speech, toxicity, abusive and offensive language classification models generalize across datasets?," *Inf. Process. Manag.*, vol. 58, no. 3, p. 102524, May 2021, doi: 10.1016/j.ipm.2021.102524.

[15]. C.-C. Wang, M.-Y. Day, and C.-L. Wu, "Political Hate Speech Detection and Lexicon Building: A Study in Taiwan," *IEEE Access*, vol. 10, pp. 44337-44346, 2022, doi: 10.1109/ACCESS.2022.3160712.
[16]. J. Camacho-Collados *et al.*, "TweetNLP: Cutting-Edge Natural Language Processing for Social Media." arXiv, Oct. 25, 2022. doi: 10.48550/arXiv.2206.14774.
[17]. K. Englmeier and J. Mothe, "Application-Oriented Approach for DetectingCyberaggression in Social Media".
[18]. R. Rianto, A. B. Mutiara, E. P. Wibowo, and P. I. Santosa, "Improving the accuracy of text classification using stemming method, a case of non-formal Indonesian conversation," *J. Bg Data*, vol. 8, no. 26, pp. 1-26, 2021, doi: https://doi.org/10.1186/s40537-021-00413-1.
[19]. A. A. Gultiaev and J. V. Domashova, "Developing a named entity recognition model for text documents in Russian to detect personal data using machine learning methods," *Procedia Comput. Sci.*, vol. 213, pp. 127-135, 2022, doi: 10.1016/j.procs.2022.11.047.
[20]. B. Evkoski, N. Ljubešić, A. Pelicon, I. Mozetič, and P. Kralj Novak, "Evolution of topics and hate speech in retweet network communities," *Appl. Netw. Sci.*, vol. 6, no. 1, p. 96, Dec. 2021, doi: 10.1007/s41109-021-00439-7.
[21]. Z. Mansur, N. Omar, and S. Tiun, "Twitter Hate Speech Detection: A Systematic Review of Methods, Taxonomy Analysis, Challenges, and Opportunities," *IEEE Access*, vol. 11, pp. 16226-16249, 2023, doi: 10.1109/ACCESS.2023.3239375.
[22]. J. M. Pérez *et al.*, "Assessing the Impact of Contextual Information in Hate Speech Detection," *IEEE Access*, vol. 11, pp. 30575–30590, 2023, doi: 10.1109/ACCESS.2023.3258973.
[23]. A. U. R. Khan, M. Khan, and M. B. Khan, "Naïve Multi-label Classification of YouTube Comments Using Comparative Opinion Mining," *Procedia Comput. Sci.*, vol. 82, pp. 57-64, 2016, doi: 10.1016/j.procs.2016.04.009.
[24]. R. Jain, D. Goel, P. Sahu, A. Kumar, and J. P. Singh, "Profiling Hate Speech Spreaders on Twitter," in *Conference and Labs of the Evaluation Forum*, Bucharest, Romania, Sep. 2021.
[25]. K. K. Kiilu, "Sentiment Classification for Hate Tweet Detection in Kenya on Twitter Data Using Naïve Bayes Algorithm," Jomo Kenyatta University of Agriculture and Technology, Juja, 2020. Accessed: Jun. 03, 2023. [Online]. Available: http://ir.jkuat.ac.ke/bitstream/handle/123456789/5521/Project%20formatted.pdf?sequence=1&isAllowed=y
[26]. 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.
[27]. M. P. Geetha and D. Karthika Renuka, "Improving the performance of aspect based sentiment analysis using fine-tuned Bert Base Uncased model," *Int. J. Intell. Netw.*, vol. 2, pp. 64-69, 2021, doi: 10.1016/j.ijin.2021.06.005.
[28]. L. H. Son, A. Kumar, S. R. Sangwan, A. Arora, A. Nayyar, and M. Abdel-Basset, "Sarcasm Detection Using Soft Attention-Based Bidirectional Long Short-Term Memory Model With Convolution Network," *IEEE Access*, vol. 7, pp. 23319-23328, 2019, doi: 10.1109/ACCESS.2019.2899260.