HATE SPEECH DETECTION ON SOCIAL MEDIA CONTENT IN JAVANESE LANGUAGE WITH NAÏVE BAYES ALGORITHM

Juniar Darma Yati; Endang Wahyu Pamungkas
Program Studi Teknik Informatika, Fakultas Ilmu Komunikasi dan Informatika,
Universitas Muhammadiyah Surakarta

Abstrak

Media sosial memungkinkan pengguna untuk menjangkau dan memfasilitasi percakapan positif dan konstruktif antar pengguna di seluruh dunia. Twitter menjadi salah satu media sosial yang memfasilitasi penggunanya untuk berkomunikasi dengan menulis dan mempublikasikan opini secara bebas. Opini tersebut dapat berisi ucapan selamat, bahagia, pujian, dan kebencian yang biasanya ditulis dengan bahasa yang umum digunakan dan sangat beragam, salah satunya adalah bahasa Jawa. Penulis melakukan penelitian terhadap sistem yang didesain untuk menguji kinerja masing-masing model algoritma Naive Bayes yaitu Gaussian Naïve Bayes, Multinomial Naïve Bayes, dan Bernoulli Naïve Bayes dalam mendeteksi dan mengklasifikasikan ujaran kebencian berbahasa Jawa pada Twitter dengan menggunakan bahasa pemrograman Python. Penelitian menggunakan data yang berasal dari dataset Twitter berjumlah 3477 tweet berbahasa Jawa. Data dibagi menjadi dua bagian dengan perbandingan 80%-20%, dengan hasil 2781 data latih dan 696 data uji. Klasifikasi dan evaluasi menghasilkan akurasi sebesar 98%, presisi sebesar 100%, recall sebesar 54%, dan F1-score sebesar 70% dengan menggunakan model Multinomial Naïve Bayes dan melalui tahapan preprocessing.

Kata Kunci: ujaran kebencian, bahasa jawa, naïve bayes, twitter.

Abstract

Social media allows users to reach out and facilitate positive and constructive conversations between users around the world. Twitter is one of the social media that enables its users to communicate by writing and publishing opinions freely. These opinions can contain congratulations, happiness, praise, and hatred, usually written in commonly used and very diverse languages, including Javanese. The authors researched a system designed to test the performance of each Naïve Bayes algorithm model, namely Gaussian Naïve Bayes, Multinomial Naïve Bayes, and Bernoulli Naïve Bayes in detecting and classifying Javanese hate speech on Twitter using the Python programming language. The study used data from a Twitter dataset of 3477 Javanese tweets. The data will split into two parts with a ratio of 80%-20%, with the results of 2781 training data and 696 test data. Classification and evaluation resulted in 98% accuracy, 100% precision, 54% recall, and 70% F1-score using the Naïve Bayes Multinomial model and through preprocessing.

Keywords: hate speech, javanese, naïve bayes, twitter.

1. INTRODUCTION

Social media enables users to reach out and facilitate positive and constructive conversations between users around the world (Chiril et al., 2022). Data Reportal[1] recorded the development of social media use in January 2022, showing that social media users in Indonesia reached 191.4 million users, and

[1] https://datareportal.com/reports/digital-2022-indonesia

where the value increased to around 21 million users (+12.6%) between 2021 and 2022. Data shows Twitter users recorded since early 2022, reaching 18.45 million users.

Twitter is a social media that facilitates users to write and publish opinions freely (Legianto, 2019). In recent years, Twitter has been used to spread hateful messages by denigrating an individual or persons based on their membership in a group, usually determined by race, ethnicity, sexual orientation, gender identity, religion, political affiliation, or views (Kohatsu et al., 2019). Hate speech on social media is a lively issue recently, along with the use of social media by the community as the main medium for communication (Pamungkas et al., 2023). Hate speech generally targets members of minority groups and can incite violence against their real lives (Sap et al., 2020). Hate speech usually uses commonly used language, one of which is Javanese, which expresses using animal names, but not all sentences containing animal names contain hate speech (Ihsan et al., 2021). Written hate speech will influence public opinion in the form of showing prohibited behavior, attitudes, actions, views, and responses, which aim to incite, spread and promote forms of hatred that can lead to acts of violence and prejudice on the part of perpetrators, victims and readers of hate speech (Sri, 2018).

The previous author has conducted similar studies regarding detecting hate speech on social media. Research conducted by (Mutanga et al., 2020) compared the classifier method for detecting hate speech and obtained an accuracy rate of 89%, 75% precision, 75% recall, and F1 score was 75% using the transfer method and a dataset ratio of 80:20. A similar study was conducted by (Feng et al., 2020) to compare classifiers types in detecting hate speech on 4,002 tweet data and yielded 71.2% accuracy, 93.2% recall using the Multinomial Naïve Bayes method.

Research using the Naïve Bayes classifier method conducted by (Legianto, 2019) aims to determine the performance of the Naïve Bayes Classifier algorithm in carrying out the classification process and the results obtained by testing 33% of the training data taken randomly, using 5-fold cross-validation and producing accuracy rate of 71.0%. Similar research that was successfully conducted by (Asogwa et al., 2022), aims to create a system that focuses on the development of machine learning models in the classification of hate speech using the Support Vector Machine and Naïve Bayes and produces an accuracy of around 99% and 55% for Support Vector Machine and Naïve Bayes respectively during the testing process.

Research conducted by (Sutarsih et al., 2022) regarding the Javanese language is a hate speech language often used on social media. The results obtained are in the form of a common vocabulary that contains denotations that turn into rough connotations using Javanese is show to express anger, annoyance, hatred, regret, feelings of shame, disappointment, astonishment, surprise, pride, humiliation, intimacy, joy, sadness, pain, and praise. Another study related to detecting Javanese hates speech on social media by comparing several classification methods (Putri et al., 2021). The research only focuses on detecting and grouping hate speech words into a vocabulary, as well as comparing the level of accuracy based on several algorithm methods used and produces an achievement of 87.5% of F1-score using the Random Forest Decision Tree (RFDT) method.

The purpose of this research is to create and focus on creating a system that can detect hate speech in Javanese on social media Twitter content and determine the classification performance results of each model of the Naïve Bayes algorithm in carrying out the level of accuracy, precision, recall, and F1-Score using or without the preprocessing process using the Python programming language.

This research consists of several parts. The introduction section explains the introduction, background, and research objectives. The related work section explains research related to this research that has been done before and can be used as reference material by the author in conducting research. The method section describes the system and the author's workings in conducting research. The results and discussion section describe the results and explanations of the test metrics obtained using the methods in the previous part. The conclusion section explains the conclusions from the description in the results and discussion sections. The last section contains references used as reference material by the author in conducting research and preparing publication manuscripts.

2. METHOD

The research method includes a workflow from the data collection, preprocessing, data transformation, classification, and evaluation processes Figure 1.

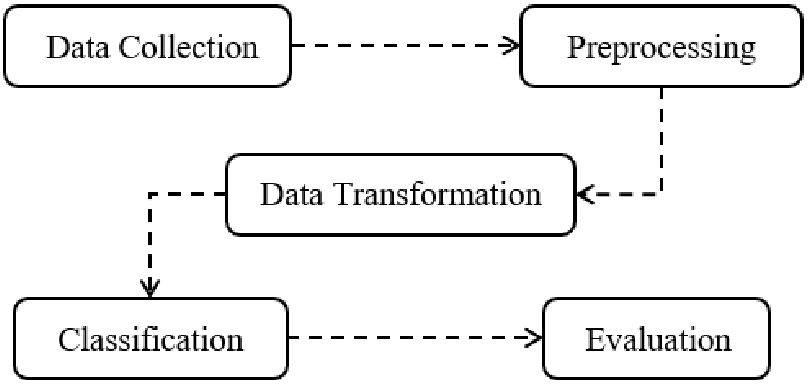

Figure 1. Method flow

2.1 Data Collection

The data used to conduct this research comes from research datasets conducted by (Putri et al., 2021). The researchers previously collected a dataset from social media Twitter which contained comments in Central Javanese with a total of 3477 tweets. Data was collected by crawling data using Twitter API2 and Tweepy Library3. Then the data is filtered and labeled to get tweets that contain hate speech and not hate speech. The attributes in the dataset consist of tweet content, hate speech, violent speech, individual targets, and ridicule. The attribute used is the body text attribute of the tweet. Then manual labeling is done according to the tweet body text for labels 1 (positive) and 0 (negative). The results of manual labeling are 173 sentences labeled 1 and 3304 sentences labeled 0. The labeling results show that the positive and negative class data were not balanced. Table 1 shows some examples of labeled tweets.

Table 1. Illustration of tweet content

Tweet	**Label**
@USER Lu aja anjing! Gausah bawa2 anak kecil lu! Tai bangke **Translation**: @USER Do it yourself, your dumbass! Don’t you dare drag a kid! Fucking ass	1
@USER @USER Mampus lo goblog cepet mati ya tolol **Translation**: @USER @USER You're dead, idiot! Just die you stupid!	1
@USER Tuhkan, emang parte ini biang keroknya, dasar bajingan, goblog kalau masih mau pilih parte ini **Translation**: @USER Damn it, the political parties were the culprit in the beginning! You are freaking stupid bitch if you still choose them	1
@USER Yaelah cuma halu aja disangka penganut bim pekok **Translation**: @USER It just some hallucinations and you assume as bim, dumbass!	0
@USER Asu lupa attendance lagi **Translation**: @USER Shit! forgot to do the attendance again	0

2.2 Preprocessing

The labeled data will then go through the preprocessing stage to convert and process unstructured data into structured data according to data mining needs. At this step, the separation starts from paragraphs which are broken down into sentences and then broken down again into words to remove numbers, symbols, and other characters that are not needed to increase the efficiency of words in the document.

2.2.1 Punctuation-Digit Elimination

Punctuation and digit elimination aims to eliminate unnecessary punctuation marks and digits to improve efficiency in the training and classification process.

2.2.2 Case Folding

Case Folding aims to change uppercase words into lowercase letters so that there is no misinterpretation by the computer because two words have the same meaning but are considered different because of the difference in uppercase and lowercase letters.

2.2.3 Tokenizing

Tokenizing separates existing sentences in the document into words that make up the sentence.

2.2.4 Stopword Removal

Stopword removal is processing text that has been cut into words to eliminate unnecessary words or affixes and makes every important word a basic word that can represent the document's content. The stop-word process is carried out based on the Sastrawi Library. Table 2 shows some examples of the results of the preprocessing process.

Table 2. Illustration of preprocessing process

Tweet	**Preprocessing**
hmm kamu pekok sekali:)	‘hmm’ ‘pekok’
Guru Goblog Ngapain foto jongkok	‘guru’ ‘goblog’ ‘foto’ ‘jongkok’
Muka mu harus seperti celeng dulu a	‘muka’ ‘celeng’ ‘a’

2.3 Data Transformation

Data transformation converts a token with a type string into a numeric vector. The data transformation process aims to enable the classification algorithm to process data because the classification algorithm cannot process the original dataset. After all, it is not of integer or vector numeric type (Hamzah, 2021).

The method used in this process uses the Term Frequency – Inverse Document Frequency (TF – IDF) algorithm adopted from . TF - IDF can be calculated by the formula TF = number of selected word frequencies /number of selected words and values. In contrast, IDF is calculating by the formula IDF = log (number of documents/numbers of selected word frequencies). To calculate TF by calculating the frequency of each word appearing in the document, IDF shows the scarcity of weights. Equation (1) shows the IDF calculation formula, then TF-IDF calculations are carried out in Equation (2) to get the results.

$$idf_j = log\left(\frac{D}{df_j}\right) \quad (1)$$

$$w_{ij} = tf_{ij} \times idf_j \quad (2)$$

Description:

tfij = Number of term occurrences in the doc

wij = Weight of term in a document

D = Number of all documents

idfj = Distribution of words in a document

dfj = Number of docs containing the term

2.4 Classification

Naive Bayes classifier provides a simple approach with precise semantics for the probabilistic representation of Bayes theorem and classifier as a form of Bayesian network called Naive because it relies on two crucial simplifying assumptions (Ismail et al., 2020). Figure 2 shows a graphical representation of Naive Bayes.

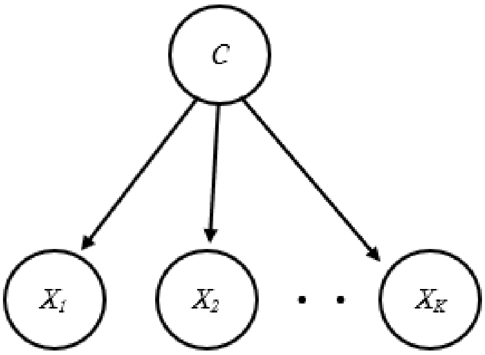

Figure 2. Bayesian Theorem

Data performance testing is processed and measured by comparing training and test data using the Naïve Bayes classifier algorithm. Equation (3), adopted from (Baqi et al., 2023), shows how to calculate class independence based on Bayesian theory.

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)} \tag{3}$$

Description:

c = Class

d = Document

P = Probability

Scientists then develop Bayesian models according to the formulas of each Naive Bayes model, which will be used and compared to find the best model. These models are Gaussian Naïve Bayes, Multinomial Naïve Bayes, and Bernoulli Naïve Bayes.

2.4.1 Gaussian Naïve Bayes

Gaussian Naïve Bayes has a characteristic that is a feature or predictor that matches and takes continuous value. Gaussian implements normal distribution and supports continuous data for classification. The Gaussian model can adjust probabilities across datasets to obtain discrete conditional probability distributions across all intensity classes for each input Ground Motion Parameter value (Cataldi et al., 2021). The goal is to get an alternative way of expressing estimates using ordinal instrumental intensity values with known associated probabilities.

$$P(x_i) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{(x_i - \mu)^2}{2\sigma^2}\right) \tag{4}$$

Description:

xi = Value attribute(xi)

σ = Standard deviation of attribute(xi)

μ = Mean of attribute(xi)

Using Equation (4), we can calculate the probability of classification data in the normal distribution of the Gaussian algorithm.

2.4.2 Multinomial Naïve Bayes

Multinomial is one of the models in the Naïve Bayes algorithm that fits discrete data and fits in classifying text or documents. This model considers the frequency of each word that appears in a particular document (Ashari et al., 2020). This algorithm model helps classify and categorize documents based on specific themes, such as sports, health, education, lifestyle, or socio-politics. The feature such a classifier uses is the frequency of words present in the document. For example, if a document continuously displays the words “gendeng”, “cocot”, “pekok”, then it can be included in the category of hate speech. Equation (5) shows the formula for the Multinomial model, and Equation (6) shows the formula for calculating the parameter P(fk|c).

$$P(c|d) = P(c)\Pi\_(1 \leq k \leq nd)P(fk|c) \tag{5}$$

$$P(fk|c) = \frac{Tct + 1}{\sum_{t\prime \in VTct\prime} + B'} \tag{6}$$

Description:

P(c|d) = Class c likelihood in document d

P(c) = c class probability value

P(fk|c) = Likelihood of the term fk in class c

Tct = Term t’s appearance in a class c doc

B = The number of word variants that still present in train data

$t' \in V\ Tct'$ = The number of terms contained in all documents in all class

2.4.3 Bernoulli Naïve Bayes

Bernoulli is one of the classification algorithm models developed from the Naïve Bayes Classifier algorithm that is suitable for considering the amount of data containing the word term, not the frequency of occurrence of the word (Ashari et al., 2020). Bernoulli Naive Bayes is similar to Multinomial Naive Bayes. Instead of using word frequency, Bernoulli's classifier algorithm uses boolean variables. Parameters used to predict class variables only require a yes or no value. For example, to determine whether a document belongs to the category of hate speech, one can identify whether the words "asu", "picek" appear in the document. If the word appears, the system automatically classifies the document as a document about hate speech. Equation (7) shows Bernoulli's formula.

$$P(c) = P(c)\prod_{i=1}^{N}P(c) \times \prod_{i+1}^{M}(1 - P(fk'|c)) \tag{7}$$

Description:

P(c) = Probability of word in class c

M = Total of words

1-P(fk’|c) = Probability of words that are not in class c

Each model has a different classification work system. The Gaussian model focuses on continuity values and supports continuous prediction for classifying the sample of data under test. The Multinomial model focuses on discrete values to the group and categorizes words that appear on a document into specific classes. Bernoulli's model focuses on counting the data containing the word term and a Boolean variable that can describe only two class values: yes or no. Each model has a different working system.

2.5 Evaluation

Performance evaluation usually aims to compare training and test data using a confusion of performance evaluation metrics. Evaluation of the performance of hate speech detection models typically uses the metrics precision, recall, and F1 score. Performance evaluation is mainly used due to the unbalanced nature of the dataset, while for a balanced dataset, accuracy metrics are the best choice (Mullah & Zainon, 2021).

Accuracy is the degree of closeness between training data and test data. Accuracy is the ratio of predictions to find the amount of data that is classified correctly (positive and negative) with testing all data into the model to get the target value (Anggoro & Kurnia, 2020). Equation (8) shows the formula for calculating accuracy.

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{8}$$

Precision is the level of correctness of the data the user requests with the data generated by system testing. Precision has a ratio of predictions in answering questions correctly compared to the overall positive predicted results. Equation (9) shows the formula for calculating precision.

$$Precision = \frac{TP}{(TP + FP)} \tag{9}$$

The recall is the success rate of the system in classifying. Recall has a correct prediction ratio compared to all correct (positive) data. Equation (10) shows the formula for calculating recall.

$$Recall = \frac{TP}{(TP + FN)} \tag{10}$$

F1-Score is an evaluation with a weighted average value of precision and recall. If the F1-Score scores well, the classification model has good precision and recall values. Equation (11) shows the formula for calculating F1-Score.

$$F1 - Score = \frac{precision\ x\ recall}{precision\ +\ recall} \tag{11}$$

Description:

TP = True Positive

TN = True Negative

FP = False Positive

FN = False Negative

3. RESULT AND DISCUSSION

The data used is 3477 from the Twitter dataset, which contains hate speech in Javanese. Furthermore, researchers process the data obtained using Python and the method used. The data goes through a preprocessing process. In this process, the string type data is clean and deletes punctuation and digit marks, changes words from uppercase to lowercase, divides words, and deletes words that do not represent and represent the contents of a sentence or document. This process aims to improve efficiency during the classification process.

Clean data will go through a data transformation process. At this stage, the data containing string-type documents are converted into integer-type documents containing numeric vectors because documents with string type cannot be processed and classified at the metric level.

Classification and evaluation process stage. Classification and evaluation using three Naïve Bayes classification models. Each model test uses preprocessing and without preprocessing. The highest comparison is obtained by testing the training data and testing data using the metric confusion parameter. Table 3 displays the test results from classification and evaluation.

Table 3. Result of classification and evaluation process

Model	Precision	Recall	F1-Score	Accuracy
Gaussian	0.45	1.00	0.62	0.94
Gaussian with preprocessing	0.44	1.00	0.62	0.94

Multinomial	0.90	0.54	0.68	0.97
Multinomial with preprocessing	1.00	0.54	0.70	0.98
Bernoulli	1.00	0.09	0.16	0.95
Bernoulli with preprocessing	1.00	0.09	0.16	0.95

The data presented in Table 3 shows the results of the accuracy of the tests carried out by each model. The Gaussian model achieves the same accuracy level of 94% when not using a preprocessing process or when using a preprocessing process. The Multinomial model obtains an accuracy rate of 97% without preprocessing and 98% with preprocessing. Meanwhile, the Bernoulli model achieves the same level of accuracy, namely 95%, when not using a preprocessing or preprocessing process. The highest accuracy level obtained was 98%, and also the results of the confusion metrics obtained were 100% precision, 54% recall, and 70% F1-Score when tested using a Multinomial model and a preprocessing process. The results of classification and evaluation tests using confusion metrics show low precision, recall, and F1-Score levels due to the influence of unequal data between hate speech and non-hate speech data.

The multinomial model can produce the highest accuracy because this model focuses on text classification, where the process of labeling each data before the process of classification and evaluation. Labeled data will go through a cleaning stage known as preprocessing.

The preprocessing stage processes data that contains documents, such as cleaning the documents from punctuation or digits, changing words in documents containing uppercase letters into lowercase letters so as not to lead to a different understanding of the same word, and also reducing or deleting words that are not important so that the classification process focuses on the essential words that represent the contents of the document.

Therefore, data that uses the Multinomial model and goes through the preprocessing stage will affect the method's performance in the classification. It obtains higher accuracy than data that does not use the Multinomial model and goes through the preprocessing stage.

4. CLOSSING

The development of the internet has brought several phenomena to social life. Hate speech is a phenomenon that is often found on social media. Detection and classification are done on hate speech by using the three Naïve Bayes models carried out by the author using the Python programming language, it concludes that the Multinomial model using the preprocessing process and the TF-IDF feature to identify hate speech on Twitter obtains satisfactory accuracy performance. The most optimal test performance results obtained are 98%.

Testing the preprocessing process in this study gave positive results, with a tendency for the accuracy value to increase. Because the cleaner and more apparent the document's content, the higher the accuracy obtained.

However, in the Gaussian and Bernoulli models, the preprocessing process does not increase the accuracy value. So, it is not certain that each use of the preprocessing process results in different improvements in accuracy. Unbalanced data also affects the results of the confusion metrics test.

However, in this study, the dataset was data containing Javanese content and tweet comments from previous studies. The results showed that in the case of hate speech detection, using the Multinomial model and applying the preprocessing process resulted in 98% accuracy, 100% precision, 54% recall, and 70% F1-Score.

In future work, it is hoped that research on hate speech in Javanese and other languages can retrieve and use balanced data, so that the results of the confusion metrics are high. In this study, the data used was limited and the unbalanced data factors affected the results of the confusion metrics, which was very low.

REFERENCES

Anggoro, D. A., & Kurnia, N. D. (2020). *Comparison of Accuracy Level of Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) Algorithms in Predicting Heart Disease*. *8*(5).

Ashari, H., Arifianto, D., Azizah, H., & Faruq, A. (2020). Perbandingan Kinerja Algoritma Multinominal Naive Bayes (MNB, Multivariate Bernoulli dan Rocchio Algortihm Dalam Klasifikasi Konten Berita Hoax Berbahasa Indonesia Pada Media Sosial. *Http://Repository.Unmuhjember.Ac.Id*, 1–12.

Asogwa, D., Chukwuneke, C., … C. N. preprint arXiv, & 2022, U. (2022). Hate Speech Classification Using SVM and Naive BAYES. *Arxiv.Org*, *9*(1), 27–34. https://doi.org/10.9790/0050-09012734

Baqi, M. H., Sibaroni, Y., & Prasetiyowati, S. S. (2023). *Comparative Analysis of Naive Bayes Model Performance in Hate Speech Detection in Media Social Twitter*. *10*(1), 1–10. https://doi.org/10.30865/jurikom.v10i1.5493

Cataldi, L., Tiberi, L., & Costa, G. (2021). Estimation of MCS intensity for Italy from high quality accelerometric data, using GMICEs and Gaussian Naïve Bayes Classifiers. *Bulletin of Earthquake Engineering*, *19*(6), 2325–2342. https://doi.org/10.1007/s10518-021-01064-6

Chiril, P., Pamungkas, E. W., Benamara, F., Moriceau, V., & Patti, V. (2022). Emotionally Informed Hate Speech Detection: A Multi-target Perspective. *Cognitive Computation*, *14*(1), 322–352. https://doi.org/10.1007/s12559-021-09862-5

Feng, Y., Li, J., Putri, T. T. A., Sriadhi, S., Sari, R. D., Rahmadani, R., & Hutahaean, H. D. (2020). A comparison of classification algorithms for hate speech detection. *Iopscience.Iop.Org*. https://doi.org/10.1088/1757-899X/830/3/032006

Hamzah, M. (2021). Classification of Movie Review Sentiment Analysis Using Chi-Square and Multinomial Naïve Bayes with Adaptive Boosting. *Journal.Unnes.Ac.Id*, *3*(1). https://journal.unnes.ac.id/sju/index.php/jaist/article/view/49098

Ihsan, F., Iskandar, I., Harahap, N. S., & Agustian, S. (2021). Decision tree algorithm for multi-label hate speech and abusive language detection in Indonesian Twitter. *Jurnal Teknologi Dan Sistem Komputer*, *9*(4), 199–204. https://doi.org/10.14710/jtsiskom.2021.13907

Ismail, M., Hassan, N., & Saleh Bafjaish, S. (2020). Comparative analysis of Naive Bayesian techniques in health-related for classification task. *Penerbit.Uthm.Edu.My*, *1*(2), 1–10. https://doi.org/10.30880/jscdm.2020.01.02.001

Kohatsu, J. C. P., Sánchez, L. Q., Liberatore, F., & Collados, M. C. (2019). Detecting and monitoring hate speech in twitter. *Sensors (Switzerland)*, *19*(21). https://doi.org/10.3390/s19214654

Legianto, S. (2019). *Implementasi Text Mining Untuk Mendeteksi Hate Speech Pada Twitter*. 60.

Mullah, N., & Zainon, W. (2021). Advances in machine learning algorithms for hate speech detection in social media: a review. *Ieeexplore.Ieee.Org*. https://ieeexplore.ieee.org/abstract/document/9455353/

Mutanga, R., … N. N.-… C. S. and, & 2020, undefined. (2020). Hate speech detection in twitter using transformer methods. *Pdfs.Semanticscholar.Org*, *11*(9). https://pdfs.semanticscholar.org/3099/62456a319a36884d98708fa5371139594aaf.pdf

Pamungkas, E. W., Fatmawati, A., Nugroho, Y. S., Gunawan, D., & Sudarmilah, E. (2023). *Hate Speech Detection in Code-Mixed Indonesian Social Media: Exploiting Multilingual Languages Resources*. 1–5. https://doi.org/10.1109/icic56845.2022.10006940

Putri, S. D. A., Ibrohim, M. O., & Budi, I. (2021). Abusive Language and Hate Speech Detection for Indonesian-Local Language in Social Media Text. *Lecture Notes in Networks and Systems*, *251*, 88–98. https://doi.org/10.1007/978-3-030-79757-7_9

Sap, M., Card, D., Gabriel, S., Choi, Y., & Smith, N. A. (2020). The risk of racial bias in hate speech detection. *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, 1668–1678. https://doi.org/10.18653/v1/p19-1163

Sri, M. (2018). Fenomena Hate Speech Dampak Ujaran Kebencian. *Ejournal.Uin-Suska.Ac.Id*, *10*(1). http://ejournal.uin-suska.ac.id/index.php/toleransi/article/view/5722

Sutarsih, S., Ismoyoputro, R., Sudarmanto, B., Susilastri, D., & Subyantoro, S. (2022). *The Using of Javanese Language as a Hate Speech in Sosial Media*. https://doi.org/10.4108/eai.15-9-2021.2315614