Paradigma, Vol. 27, No. 2, September 2025, P-ISSN 1410-5063, E-ISSN: 2579-3500, Page. 92-98
Published: LPPM Universitas Bina Sarana Informatika

PARADIGMA

website: http://jurnal.bsi.ac.id/index.php/paradigma/

Support Vector Machine with FastText Word Embedding for Hate Speech Aspect Categorization

Aida Millati Mardiana[1], Imam Fahrur Rozi[2], Rudy Arianto[3]

[1,2,3]Information Technology, State Polytechnic of Malang, Malang, Indonesia

ARTICLE INFORMATION

Artikel History:
Received: April 15, 2025
Revised: June 11, 2025
Accepted: August 26, 2025
Available Online: Sept. 30, 2025

Keyword:

Hate Speech
Aspect Categorization
Twitter
Support Vector Machine
Word Embedding FastText

ABSTRACT

Freedom of expression on Twitter often leads to issues such as hate speech, which may include provocation, incitement, or insults based on race, religion, gender, and other aspects. To address this issue, machine learning techniques can be applied to automatically classify hate speech. Therefore, this study aims to implement a machine learning–based approach for automatic hate speech aspect classification and to evaluate the accuracy of the obtained results. Support Vector Machine is used as the classifier method, with FastText as the word embedding method in the categorization process of hate speech aspects. The categorized aspects include abusive, individual, group, religion, race, physical, gender and other. The dataset used in this research is a collection of Indonesian tweets from Kaggle, which have been classified into each aspect. This study also tested combinations of preprocessing methods, namely filtering with stemming and the FastText pre-trained model. From the test results of the application of the Support Vector Machine method with FastText word embedding, with parameters C value = 1.0, gamma = 1.0 and RBF kernel and the ratio between training data and testing data is 90:10, the best results were obtained accuracy 98%, precision 98%, recall 98% and F1-Score 97% on Physical and Gender aspects. In addition, this study also tested if it did not use fasttext word embedding and the accuracy results showed 84%, precision 74%, recall 86% and F1 Score 79% in the abbusive aspect.

Corresponding Author:

Imam Fahrur Rozi,
Information Technology,
State Polytechnic of Malang,
Jl. Soekarno Hatta No.9, Jatimulyo, Kec. Lowokwaru, Kota Malang, Jawa Timur 65141,
Email: imam.rozi@polinema.ac.id

INTRODUCTION

With the advancement of communication technology, social media has increasingly become an important medium for human interaction (Iskandar & Nataliani, 2021). Social media serves as a source of information and as a channel for exchanging messages online (Dewi & Setiawan, 2022)One widely used social media platform is Twitter, recently renamed as X, which allows users to share content with others (Tineges et al., 2020). Twitter is a popular social media platform that gives users the ability to send text messages with a maximum length of 280 characters (Fadhil & S, 2022).

On the Twitter platform (now X), users are free to upload or comment on other people's posts (Sumantri & Marwoto, 2024), While this encourages freedom of expression, it also opens the possibility for problems such as hate speech (Mukti et al., 2023). Hate speech may occur when individuals or groups interact, which can involve provocation, incitement, or insults based on race, gender, ethnicity, religion, nationality, sexual orientation, or other characteristics (Ridwan & Muzakir, 2022). According to the Criminal Investigation Unit of the Indonesian National Police (Bareskrim Polri) in 2022, at least 33 cases of hate speech were recorded, with the North Sumatra Regional Police handling the highest number of cases (8 in total) (Pusiknas Bareskrim Polri, 2021). Hate speech and insults on social media may result in serious

DOI: https://doi.org/10.31294/p.v27i2.5127

consequences such as discrimination, social conflict, and even genocide (Ridwan & Muzakir, 2022).

Given the serious impact of hate speech, it is important to categorize it by specific aspects to help prevent individuals, especially children, from adopting or reproducing inappropriate language on social media (Ridwan & Muzakir, 2022). One approach that can be used is Machine Learning with the Support Vector Machine Method as previously done by Jessica Widyadhana Iskandar and Yessica Nataliani regarding Comparison of Naïve Bayes, SVM, and k-NN for Aspect-Based Gadget Sentiment Analysis carried out categorization based on aspects on the sentiment of a gadget product. In this study, sentiment is categorized based on aspects of design, price, specifications and brand image. In this study, it was proven that the SVM classification model showed the best results (Iskandar & Nataliani, 2021). Similar research was also conducted by Hansen Gunawan Sulistio and Andreas Handojo regarding Aspect-Based Sentiment Analysis on E-Commerce Reviews with the Support Vector Machine Method to Obtain Sentiment Information from Several Aspects ad (Ardiani et al., 2020). In the study, sentiment was categorized based on general aspects, accuracy, quality, service, delivery, packaging, and price. The results of the study showed that the quality aspect was the most frequently discussed (Sulistio, & Handojo, n.d.). Other research was also conducted by Mustasaruddin, Elvia Budianita, M Fikry and Febi Yanto on Sentiment Classification of MyPertamina Application Review Using Word Embedding FastText and Support Vector Machine (SVM). This research aims to assess the MyPertamina application with sentiment grouping. The results of this study show a good SVM model with the ratio between training data and testing data is 90:10 (Mustasaruddin et al., 2023). Classifying hate speech in tweets on Twitter is an important step to filter out negative messages. It aims to understand the type of hate speech that occurs, by considering certain aspects. Acts of hate speech that occur through social media can be considered a violation of the law that can be subject to Article 27 paragraph 3 of the ITE Law related to defamation through electronic systems (Dwi Arjanto, 2023). Based on previous research related to the use of the Support Vector Machine (SVM) method for aspect categorization, which was only applied to a relatively small amount of data and only into a few classes, as well as data obtained from app store applications, this research will be conducted to classify tweets containing elements of hate speech based on various aspect categories such as violence (abusive), individual (individual), group / group (group), religion (religion), race (race), physical (physical), gender (gender) and others (Other) from social media Twitter which has now changed its name to X and whether the application of the Support Vector Machine method with FastText word embedding is accurate for aspect categorization.

RESEARCH METHOD

This research is divided into several stages, namely data collection obtained from Kaggle used by Muhammad Okky Ibrohim and Indra Budi in 2019 (Ibrohim & Budi, 2019) (Ramli & Sibaroni, n.d.) . The data used in this study is multi-label hate speech and abusive language detection on Indonesian Twitter from Kaggle. The data obtained amounted to 13,170 tweets. The data consists of 12 aspects or categories. However, this study only used 8 aspects, namely Abusive, HS_Individual, HS_Group, HS_Religion, HS_Race, HS_Physical, HS_Gender and HS_Other. The data was used by Muhammad Okky Ibrohim and Indra Budi in 2019 (Ibrohim & Budi, 2019). text preprocessing, vectorization, modeling, model evaluation and testing. The research stages can be seen in Figure 1.

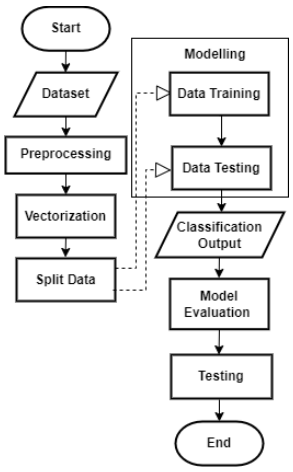

Figure 1 Research Method

(1) The data used in this study are multi-label hate speech and abusive language detection on Indonesian Twitter from Kaggle. The data obtained amounted to 13,170 tweets. The data consists of 12 aspects or categories. However, this research only uses 8 aspects, namely Abbusive, HS_Individual, HS_Group, HS_Religion, HS_Race, HS_Physical, HS_Gender and HS_Other. The data was used by Muhammad Okky Ibrohim and Indra Budi in 2019 (Ibrohim & Budi, 2019). (2) At the preprocessing stage there are several more stages in it, namely Data Cleaning which means simplifying the text by removing irrelevant punctuation, links and special characters (Agustian & Nazir, 2024). Tokenizing wich mean breaking text into smaller units such as words or tokens for further analysis (Agustian & Nazir, 2024). Filtering wich mean remove common words that are less informative in certain contexts, such as "dan", "di", or "untuk" (Agustian & Nazir, 2024) and Stemming wich mean changing words to their basic form by removing special suffixes or prefixes (Agustian & Nazir, 2024). Data needs to be preprocessed so that later the calculation becomes optimal because the data used is clean (Mustasaruddin et al., 2023). In this study, 4 stages of text preprocessing will be carried out as shown in Figure 2.

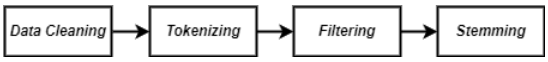

Figure 2 Preprocessing Text

In this study, it is necessary to do Data Cleaning because this stage serves to remove characters in text that have no meaning such as hastags (#), punctuation marks (") etc. and there is a case folding stage to change letters in capitalized data to lowercase letters (Mustasaruddin et al., 2023). Tokenizing aims to get the features of each document after being given a weighted value (Oryza Habibie Rahman et al., 2021). In the Filtering process, words that often appear but have no meaning will be removed to maximize when modeling (Mustasaruddin et al., 2023). The Stemming stage needs to be done to reduce the number of different indices in a document and also to group other words that have the same basic word form, but different forms due to different affixes (Oryza Habibie Rahman et al., 2021). (3) Data that has been processed in the preprocessing process will produce clean data so that vectorization can be done by converting words in alphanumeric format into vector format. In this research, the vectorization process is carried out using FastText word embedding by converting text into a continuous vector that can later be used in any language-related task. In FastText, words are learned by paying attention to a small part of the word, represented as a series of n-gram characters. This approach allows FastText to understand the meaning of short words as well as capturing the pattern of prefixes and suffixes of words (Girsang, 2020). By considering word parts, FastText can understand words that rarely appear in documents and overcome the problem of words that are not recognized (Out of Vocabulary). (Ramadhy & Sibaroni, 2022). (4) The Support Vector Machine (SVM) method works by finding the optimal hyperlane by maximizing the distance between classes. Figure 3 shows that Support Vector Machine (SVM) constructs an optimal hyperplane by maximizing the margin between two classes. This hyperplane is defined by support vectors, which are data points closest to the boundary. Through this approach, SVM achieves reliable classification and strong generalization to previously unseen data.

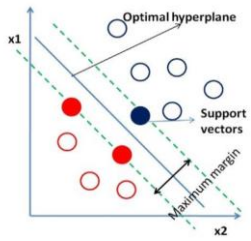

Figure 3 Hyperplane SVM Method

The class assumption can be perfectly separated by the hyperplane defined in equation (1).

$$x_i w + b = 0 \quad (1)$$

Patterns x_i that belong to either class 1 or 0 can be formulated as patterns that satisfy the inequality, as in equations (2) and (3).

$$x_i w + b \geq +1, y_1 = +1 \quad (2)$$

$$x_i w + b \leq +1, y_1 = -1 \quad (3)$$

w is the normal of the plane and b is the position of the alternative plane with respect to the coordinate center. So, the margin value can be found by maximizing the distance value between the hyperplane and its closest point with equation (4).

$$\text{nilai margin} = \frac{1}{||w||} \quad (4)$$

Support Vector Machine has several training processes, but this research uses Sequential Training because the process is simple and does not require a long time. Here is the Sequential Training algorithm (Nugroho et al., 2003). The initialization process begins by setting the value αi = 0. The matrix calculation is performed using equation (5).

$$D_{ij} = y_i y_j (K(x_i, x_j) + \lambda^2) \quad (5)$$

With i, j = 1, ..., ..., n, where:
xi = i-th data
xj = j-th data
yi = i-th data class
yj = j-th data class
n = number of data
K(*xi, xj*) = kernel function used

From the i-th data to the j-th data, it can be calculated using equation (6).

$$E_i = \sum_{j=1}^{n} \alpha_j D_{ij} \quad (6)$$

Where:
αj = j-th alpha
Dij = Hessian Matrix
Eij = Error Rate

The correction of the *αi* value is done using equation (7).

$$\delta ai = \min(\max[\gamma(1 - E_1), -\alpha_1], C - \alpha_1]) \quad (7)$$

Where:
αi = i-th alpha
γ = gamma constant (parameter to control the speed of the learning process)
Eij = Error Rate
C = constant C

The value of *αi* is then updated with equation (8).

$$a_i = a_i + \delta a_i \quad (8)$$

Where:
αi = i-th alpha
δa_i = i-th alpha delta

In the testing process, the SVM method starts by calculating the value of f(x) using equation (9).

$$f(x) = \sum_{i=1}^{m} \alpha_i y_j K(xi, x) + b \quad (9)$$

Where:
αi = i-th alpha
yi = i-th data class
m = the amount of data that is SV
K(xi, x) = kernel function
b = bias value

Bias value can be found using equation (10).

$$b = -\frac{1}{2}[\sum_{i=1}^{m} \alpha_i y_j K(xi, x^+) + \sum_{i=1}^{m} \alpha_i y_j K(xi, x^-)] \quad (10)$$

The value of K(xi, xtestj) is done using equation (11).

$$\sum_{i=1}^{m} y_i y_j K(xi, x) \quad (11)$$

Where:
K(xi, x) = the kernel function to be used

(5) The K-Fold Cross Validation method is used to separate test data and training data with the aim of minimizing randomness when sharing data, such as when using the split data method. The K-Fold Cross Validation method divides the number of records according to the specified "k" value. The "k" value used is determined by the researcher to ensure the accuracy value of each test. After each class calculates its precision, all the precision is summed up and divided by the specified number of "k" (Khatib Sulaiman et al., n.d.).

RESULTS AND DISCUSSION

1) Data Collection

The dataset used in this study is a multi-label hate speech dataset from Kaggle, containing 13,170 Indonesian Twitter tweets classified into eight aspects. The data consists of 12 aspects or categories. However, this study only uses 8 aspects, namely Abbusive, HS_Individual, HS_Group, HS_Religion, HS_Race, HS_Physical, HS_Gender and HS_Other. The data was used by Muhammad Okky Ibrohim and Indra Budi in 2019 (Ibrohim & Budi, 2019). Testing was performed with different training-to-testing ratios: 90:10, 80:20, and 70:30. Results indicated that the 90:10 split yielded the highest accuracy at 82%. This ratio was chosen as it allows more data to train the model, thereby improving its ability to capture patterns and optimize accuracy and generalizability. This approach aligns with the study's objective to maximize model performance through effective data composition. results of split data testing are shown in Table 1.

Tabel 1 Test Results for the Effect of Split Data

Split Data	Results			
	Accurcay	Precision	Recall	F1 Score
90/10	0.82	0.82	0.82	0.82
80/20	0.81	0.81	0.81	0.81
70/30	0.80	0.80	0.80	0.80

2) Preprocessing Data

The preprocessing stage included various data cleaning steps, such as tokenization, filtering, and stemming. Testing different combinations of preprocessing methods revealed that filtering combined with stemming produced the highest accuracy and F1-score (82%) compared to using stemming alone. The filtering process played a key role in removing high-frequency but low-information words, thereby reducing noise in the data and strengthening the signals learned by the model. This is consistent with the methodological aim of optimizing data relevance for model input. the combination of preprocessing is tested which consists of filtering, namely words that often appear will be removed from the meaning and stemming. The test results are shown in Table 2.

Tabel 2 Test Results of the Effect of Preprocessing Combinations

Preprocessing Combinations	Results			
	Accuracy	Precision	Recall	F1 Score
Filtering + Stemming	0.82	0.82	0.82	0.82
Filtering	0.82	0.82	0.82	0.82
Stemming	0.71	0.71	0.71	0.68

Preprocessing tests using filtering and stemming as well as filtering alone get better results than just using stemming because filtering can perform the stopword removal process. In addition, it can reduce noise in the data by removing words that have no important meaning to the model. This can improve the quality of the input data and strengthen the signal learned by the model.

3) Vectorization with FastText

Once preprocessed, the data was vectorized using FastText word embedding to create a continuous vector representation of the tweets. Testing with a pre-trained FastText model yielded higher accuracy compared to a model trained from the dataset itself. This improved performance underscores the benefits of using a larger and more diverse dataset for the FastText pre-trained model, which enhances generalization, especially for out-of-vocabulary words. This process, as outlined in the methodology, aims to leverage robust word representations to improve model accuracy in handling complex language patterns. The test results are shown in Table 3.

Tabel 3 Test Results of the Effect of the Word Embedding Model

Word Embedding Model	Results			
	Accuracy	Precision	Recall	F1 Score
Pre trained model FastText Model	0.82	0.82	0.82	0.82
FastText yang di training dari dataset peneliti	0.78	0.78	0.78	0.78

Testing with the word embedding model provided by FastText (PreTrained Model) produces high scores compared to the FastText model trained from the researcher's dataset because the quality and size of the datasets used are different. FastText's Pre-Trained Model is usually trained on very large datasets, these datasets have more diverse vocabularies and examples than the researcher's dataset.

4) SVM Parameter Optimization with GridSearchCV

SVM parameters, specifically C, gamma, and kernel, were optimized using GridSearchCV. The testing indicated that C=1.0, gamma=1.0, and the RBF kernel provided the highest accuracy, precision, recall, and F1-score. This configuration strikes a balance between margin maximization and classification error,

while gamma extends the influence of data points for broader pattern capture. The RBF kernel, known for handling non-linear data effectively, proved to be a suitable choice for this dataset. These results reflect the methodology's emphasis on selecting optimal parameters to enhance classification performance.

In hyperparameter testing, parameter tunning is carried out using GridSearch to get the best model of each parameter in the Support Vector Machine Method. Tunning SVM parameters using the GridSearchCV method to find the maximum kernel results, C and gamma values. In this study, the kernel used is {RBF, Polynomial} along with the value of C = {0.1, 1.0, 0.001, 0.01} and the value of gamma = {10, 1, 0.1, 0.01}. The results of testing the C value are shown in table 4.

Tabel 4 C-Value Testing Results

C-Value	Accuracy	Precision	Recall	F1 Score
0.1	0.72	0.80	0.72	0.60
1.0	0.79	0.78	0.79	0.77
0.001	0.72	0.51	0.72	0.60
0.01	0.72	0.51	0.72	0.60

The value of C = 1.0 gives high accuracy, precision, recall and F1 Score results because the C parameter in the SVM Method is a regulation parameter that controls the trade-off between maximizing margins and minimizing classification errors. Therefore, the value of C = 1.0 allows SVM to find a good balance between wide margins and keeping the number of misclassifications low.

The gamma value tests in Table 2 were conducted using a value of C = 1.0 and the kernel used was RBF and conducted on the HS_Other aspect. Different gamma values (10, 1.0, 0.1, 0.01) were chosen to explore how far a data point can influence the model decision. The results of testing the gamma values are shown in Table 5.

Tabel 5 Gamma Value Testing Results

Gamma Value	Accuracy	Precision	Recall	F1 Score
10	0.72	0.67	0.72	0.61
1.0	0.79	0.78	0.79	0.77
0.1	0.74	0.79	0.74	0.65
0.01	0.72	0.52	0.72	0.60

The value of gamma = 1.0 gives high accuracy, precision, recall and F1 Score results because the gamma parameter in the SVM Method controls how far the influence of the training example extends. With gamma = 1.0, the influence of one training example will extend widely enough to capture more global patterns in the data.

The kernel tests in Table 3 were conducted using C value = 0.1 and gamma value = 1.0. Different kernels (RBF, Ploynomial) were selected to explore the most suitable kernel used in this study. The results of the kernel testing are shown in Table 6.

Tabel 6 Kernel Testing Results

Kernel	Accuracy	Precision	Recall	F1 Score
RBF	0.79	0.78	0.79	0.77
Polynomial	0.78	0.77	0.78	0.77

The RBF (Radial Basis Function) kernel has the ability to handle non-linearity, this makes the RBF kernel flexible and able to capture complex relationships. Polynomial kernels can model polynomial relationships between features, which can be useful if the data has non-linear relationships and is represented with polynomials.

5) Model Validation with k-Fold Cross Validation

During the validation stage, k-Fold Cross Validation was applied with k values of 3, 5, 7, and 9. Optimal results were observed at k=7 and k=9, where the model achieved a good balance between bias and variance, supporting improved generalization. This process aligns with the study's methodological objective of minimizing overfitting through multiple folds, thus enhancing the model's robustness when encountering new data. The test results are shown in Table 7.

Tabel 7 Results of k-Fold Cross Validation Testing

K-Fold (K)	Results			
	Accuracy	Precision	Recall	F1 Score
3	0.78	0.76	0.74	0.74
5	0.78	0.76	0.74	0.74
7	0.79	0.77	0.74	0.75
9	0.79	0.77	0.75	0.75

At k=7 and 9 the model can achieve a good balance between bias and variance, resulting in optimal accuracy. In addition, the larger the k value, the more folds are used for model training. This can reduce the tendency of the model to overfitting to one or a few folds. Reduction of overfitting can improve the generalization ability of the model to new data.

6) Model Performance Evaluation Across Aspect Categories

Final testing across different hate speech aspects revealed varying levels of accuracy. For example, the HS_Physical and HS_Gender categories demonstrated the highest accuracy at 98%, likely due to clearer and more distinct feature patterns in these categories. Conversely, categories such as HS_Individual and HS_Group yielded lower accuracy, potentially due to greater linguistic variation or a limited number of examples. These findings align with the study's goal to evaluate the effectiveness of the SVM and FastText combination in categorizing hate speech aspects on social media. The test results are shown in Table 8.

Tabel 8 Categorization Aspect Testing Results

Category	Results			
	Accuracy	**Precision**	**Recall**	**F1 Score**
Abbusive	0.82	0.82	0.82	0.82
HS_Individual	0.78	0.77	0.78	0.74

Category	Results			
	Accuracy	**Precision**	**Recall**	**F1 Score**
HS_Group	0.85	0.86	0.85	0.80
HS_Religion	0.95	0.94	0.95	0.93
HS_Race	0.95	0.94	0.95	0.94
HS_Physical	0.98	0.98	0.98	0.97
HS_Gender	0.98	0.96	0.98	0.97
Other	0.79	0.78	0.79	0.76

HS_Physical and HS_Gender aspects because the data characteristics in these aspects have more examples or clear indicators for HS_Physical and HS_Gender categories compared to other categories. This can facilitate the model to learn better patterns in the HS_Physical and HS_Gender categories.

CONCLUSION

Based on the results of the tests, it can be concluded that the Support Vector Machine (SVM) algorithm combined with FastText word embedding achieves high accuracy for aspect-based hate speech detection on Twitter. The best performance was obtained in the HS_Physical and HS_Gender aspects with 98% accuracy from k-Fold testing, demonstrating that this method is highly effective in categories with clear linguistic patterns.

The classification process involved several stages, including preprocessing (data cleaning, tokenizing, filtering, and stemming), feature vectorization with FastText, training, and classification. These steps operated effectively, as reflected in the accuracy results across categories: Abusive (82%), HS_Individual (78%), HS_Group (85%), HS_Religion (95%), HS_Race (95%), HS_Physical (98%), HS_Gender (98%), and Other (79%). This indicates that the proposed approach works optimally, particularly in aspects with well-defined linguistic markers.

Future research is recommended to improve several aspects. First, addressing the imbalance between categories using techniques such as oversampling or SMOTE may improve the performance of minority classes. Second, comparative experiments with deep learning models such as CNN, LSTM, or transformer-based models (e.g., BERT) are needed to benchmark performance. Third, expanding the dataset with more recent and diverse social media content would enhance generalizability. Finally, integrating this model into a practical application, such as a real-time hate speech monitoring system, could provide significant societal benefits.

REFERENCES

Agustian, S., & Nazir, A. (2024). Klasifikasi Sentimen Terhadap Pengangkatan Kaesang Sebagai Ketua Umum Partai PSI Menggunakan Metode Support Vector Machine. *Technology and Science (BITS)*, *6*(1). https://doi.org/10.47065/bits.v6i1.5340

Ardiani, L., Sujaini, H., & Tursina, T. (2020). Implementasi Sentiment Analysis Tanggapan Masyarakat Terhadap Pembangunan di Kota Pontianak. *Jurnal Sistem Dan Teknologi Informasi (Justin)*, *8*(2), 183. https://doi.org/10.26418/justin.v8i2.36776

Dewi, M. P. K., & Setiawan, E. B. (2022). Feature Expansion Using Word2vec for Hate Speech Detection on Indonesian Twitter with Classification Using SVM and Random Forest. *Jurnal Media Informatika Budidarma*, *6*(2), 979. https://doi.org/10.30865/mib.v6i2.3855

Dwi Arjanto. (2023). Begini Bunyi Pasal Ujaran Kebencian di UU ITE yang Digunakan Dalam Pelaporan Rocky Gerung. In Dwi Arjanto (Ed.), *tempo.co*.

Fadhil, I. M., & S, Y. S. (2022). Klasifikasi Topik Pada Tweet Berbahasa Indonesia menggunakan Ekspansi fitur Fast-Text dengan Metode Support Vector Machine (SVM). 9(3).

Girsang, A. (2020, November 17). *Word Embedding dengan Word2vec*. BINUS Education.

Ibrohim, M., & Budi, I. (2019). *Indonesian Abusive and Hate Speech Twitter Text*. Kaggle.

Iskandar, J. W., & Nataliani, Y. (2021). Perbandingan Naïve Bayes, SVM, dan k-NN untuk Analisis Sentimen Gadget Berbasis Aspek. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *5*(6), 1120–1126. https://doi.org/10.29207/resti.v5i6.3588

Khatib Sulaiman, J., Pradema Sanjaya, U., Pribadi, T., Wisma Dwi Prastya, I., Nahdlatul Ulama Sunan Giri, U., Kunci Klasifikasi, K., Bayes, N., & Hibah, D. (n.d.). Klasifkasi Dana Hibah Usaha Mikro Kecil Dan Menengah dengan Metode Naïve Bayes. *Indonesian Journal of Computer Science*.

Mukti, A., Hadiyanti, A. D., Nurlaela, A., & Panjaitan, J. (2023). *Sistem Analisa Sentiment Bakal Calon Presiden 2024 Menggunakan Metode NLP Berbasis Web the Sentiment Analysis System For the 2024 Presidential Candidates Uses Web-Based NLP Method* 6(1), 128-140. https://doi.org/10.32531/jsoscied.v6i1.621.

Mustasaruddin, M., Budianita, E., Fikry, M., & Yanto, F. (2023). Klasifikasi Sentiment Review Aplikasi MyPertamina Menggunakan Word Embedding FastText dan SVM (Support Vector Machine). *Jurnal Sistem Komputer Dan Informatika (JSON)*, *4*(3), 526. https://doi.org/10.30865/json.v4i3.5695

Nugroho, A. S., Witarto, A. B., & Handoko, D. (2003). *Support Vector Machine-Teori dan Aplikasinya dalam Bioinformatika 1*. http://asnugroho.net

Oryza Habibie Rahman, Gunawan Abdillah, & Agus Komarudin. (2021). Klasifikasi Ujaran Kebencian pada Media Sosial Twitter Menggunakan Support Vector Machine. *Jurnal*

RESTI (Rekayasa Sistem Dan Teknologi Informasi), *5*(1), 17–23. https://doi.org/10.29207/resti.v5i1.2700

Pusiknas Bareskrim Polri. (2021). *Berani Unggah Ujaran Kebencian, Siap-siap Dihukum 6 Tahun Penjara*. Pusiknas Bareskrim Polri.

Ramadhy, I. F., & Sibaroni, Y. (2022). Analisis Trending Topik Twitter dengan Fitur Ekspansi FastText Menggunakan Metode Logistic Regression. *JURIKOM (Jurnal Riset Komputer)*, *9*(1), 1. https://doi.org/10.30865/jurikom.v9i1.3791

Ramli, R. G., & Sibaroni, Y. (n.d.). *Klasifikasi Topik Twitter menggunakan Metode Random Forest dan Fitur Ekspansi Word2Vec*.

Ridwan, M., & Muzakir, A. (2022). Model Klasifikasi Ujaran Kebencian pada Data Twitter dengan Menggunakan CNN-LSTM Hate Speech Classification Model on Twitter Data Using CNN-LSTM. *Teknomatika*, *12*(02), 1–5.

Sulistio, H. G., & Handojo, A. (2022). Aspect-Based Sentiment Analysis pada Ulasan ECommerce dengan Metode Support Vector Machine untuk Mendapatkan Informasi Sentimen dari Beberapa Aspek. Jurnal Infra, 10(2), 450-454.

Sumantri, G., & Marwoto, B. S. H. (2024). Analisis Sentimen di Twitter Terkait Tim Nasional Sepak Bola Indonesia Menggunakan Metode Support Vector Machine. Jurnal Kajian Dan Terapan Matematika, 10(2), 96–104. https://doi.org/10.21831/jktm.v10i2.19561

Tineges, R., Triayudi, A., & Sholihati, I. D. (2020). Analisis Sentimen Terhadap Layanan Indihome Berdasarkan Twitter Dengan Metode Klasifikasi Support Vector Machine (SVM). *Jurnal Media Informatika Budidarma*, *4*(3), 650. https://doi.org/10.30865/mib.v4i3.2181