e-ISSN 2549-7472
Edumatic: Jurnal Pendidikan Informatika
Vol. 9 No. 1, April, 2025, **Hal. 11-20**
DOI: 10.29408/edumatic.v9i1.29087

Comparative Analysis of Naive Bayes and SVM for Improved Emotion Classification on Social Media

Rio Ferdinand Putra Pratama [1] , Warih Maharani [2, *]

[1] Department of informatics, Telkom University, Indonesia
[2] Data Science, Telkom University, Indonesia
[*] Correspondence: wmaharani@telkomuniversity.ac.id

Received: 28 December 2024 | Revised: 2 January 2025 | Accepted: 10 January 2025 | Published: 9 April 2025

Abstract
Emotion classification is important for understanding user interactions on social media, especially in identifying emotions such as happy, angry, sad, and fear. However, Indonesian text processing faces challenges due to language complexity and slang. This research aims to compare Naive Bayes and SVM models, focusing on evaluating the impact of preprocessing, feature extraction, and parameter optimization to improve emotion classification. The dataset was collected from API X using crawling techniques and manually annotated by six annotators. The training process used full and half preprocessing datasets with TF-IDF, BoW, and Word2Vec feature extraction. Naive Bayes and SVM models were evaluated using accuracy, precision, recall, and F1 score. Our results show that full preprocessing improves accuracy, with TF-IDF + BoW achieving 78.01% with SVM and outperforming Naïve Bayes at 75.53%. The results classify emotions into four classes: happy, sad, angry, and fear. This study demonstrates the value of preprocessing and feature selection to deal with slang and complexity in Indonesian texts. These results provide insights for developing optimal emotion classification models and offer applications in sentiment analysis, social media monitoring, and mental health detection.

Keyword: emotion classification; feature extraction; naïve bayes; preprocessing; svm

INTRODUCTION

Social media has become the main platform for sharing activities and expressions in daily life. One of the most popular social media, Twitter, now called X, is often used to discuss current issues. The conversations made by X users when giving a review or opinion have various emotions, such as anger, sadness, fear, or joy (Regita et al., 2024). Identifying emotions on social media is critical to understanding user interactions, improving user experience, shaping public policy and refining business strategies (Sudianto, 2022).

Emotions have an essential role in understanding human expression. In the context of sentiment analysis, the feelings felt at that moment can also be one of the indicators to measure people's response to an issue (Putri et al., 2023). These feelings appear in different conditions, such as happy, fear, anger, and sadness. The conditions experienced affect how people think and act, resulting in certain expressions (Zhang et al., 2024). This shows how important it is to accurately detect and classify such expressions in texts, especially in social media (Fudholi, 2021). Emotions coming from texts are difficult to understand. So, further text processing techniques are needed, namely text mining methods, to extract unstructured words and create machine learning models that can classify emotions more quickly (Kavithasubramani et al., 2024). The selection of the right algorithm will significantly affect the results of sentiment analysis, especially in the context of complex text. besides the algorithm, preprocessing and selection of extraction features also greatly affect the classification results (Nugroho & Cholissodin, 2021).

Preprocessing cleans the raw text by handling informal language, misspellings and removing irrelevant symbols that are often found in posts (Akbar et al., 2022). For example, in the context of Bahasa Indonesia, preprocessing must handle the diversity of languages and informal expressions that often appear in social media data. Feature extraction techniques, such as Bag of Word (BoW), term frequency-inverse document frequency (TF-IDF), and Word2Vec, transform textual data into numerical representations that machine learning algorithms can process. While BoW captures word frequency (Galke & Scherp, 2021), TF-IDF emphasizes word importance in the dataset (Haya et al., 2024), and Word2Vec provides semantic relationships between words (Indira, 2021). These steps significantly impact model performance by enhancing the quality of the input data (Mokari et al., 2023).

Naive Bayes is a popular algorithm due to its simplicity and speed in processing text data. However, this algorithm is less effective when dealing with datasets that have too many features, which can cause its accuracy to decrease. (Putri et al., 2023). Compared to Naive Bayes, SVM often provides better accuracy in processing text data because SVM can analyze patterns well, so SVM is effective in analyzing sentiment. (Ningsih et al., 2024). Although SVM excels in accuracy, Naive Bayes also has the advantage of coping with large datasets with independent features, where Naive Bayes efficiently enables fast processing without requiring large resources (Zhang et al., 2024). Naive Bayes is often used for simple text analysis such as spam email classification, where the features are independent and the complexity is low (Siddique et al., 2021).

In a study conducted by (Sarimole & Kudrat, 2024) who compared the performance of SVM and Naive Bayes. Their research resulted in SVM accuracy of 87.95% and Naive Bayes of 65% on 1081 total data and using TF-IDF extraction features. The results of their study indicate that SVM has superior performance compared to Naive Bayes. Then in research conducted by (Supian et al., 2024), SVM is also superior to 94% compared to Naive Bayes which is only 91% in analyzing sentiment about the National Capital on twitter which has 2,130 total datasets. in their study using TF-IDF for feature extraction. Datasets in their study go through several preprocessing stages, namely cleansing, case folding, tokenizing, stopword removal, stemming, and normalization. Both studies used the TF-IDF feature because of its ability to emphasize words that are relevant in a particular context, reduce the influence of common words, and provide a more informative representation of the text to improve the model's ability to detect emotional patterns.

This research was conducted to overcome the shortcomings of previous research. As in the research conducted by (Supian et al., 2024) where their research only focuses on TF-IDF extraction features so that it is less illustrated if using other extraction features. then in the research conducted by (Sarimole & Kudrat, 2024) which is less than optimal in handling the complexity of social media text, such as slang, hashtags, or emojis, which can affect sentiment classification accuracy. Both studies also did not discuss what parameters and kernels were used. This research aims to explore some important aspects that were not discussed previously, namely the effect of preprocessing on the final result, the performance of various feature extraction methods and the impact of parameters on the performance of classification models. In addition, this research will also compare the performance of Naïve Bayes and SVM algorithms in classifying emotions into four classes using the most optimal parameters. The findings of this research are expected to provide insights for building optimal models in analyzing emotions, and checking mental health through posts.

METHOD

Figure 1 shows that this research uses the Naive Bayes and SVM methods by going through several stages including data crawling, manual labeling, preprocessing, feature extraction using three extraction features namely TF-IDF, BoW and Word2Vec, emotion classification using SVM and Naive Bayes and evaluation as shown in Figure 1. The first stage is data crawling, where data will be collected using the X API. This process is carried out using the username obtained after the user is willing to fill in the username to retrieve tweets through the Microsoft form. Data is retrieved based on certain keywords such as "*seneng*" (Happy), "*Sedih*"(Sad),"*Ngeri*" (Fear), and "*Kesel*" (Angry) to ensure representation of a wide range of emotions. The data collected was 8,978 tweets that were ready to be manually labeled

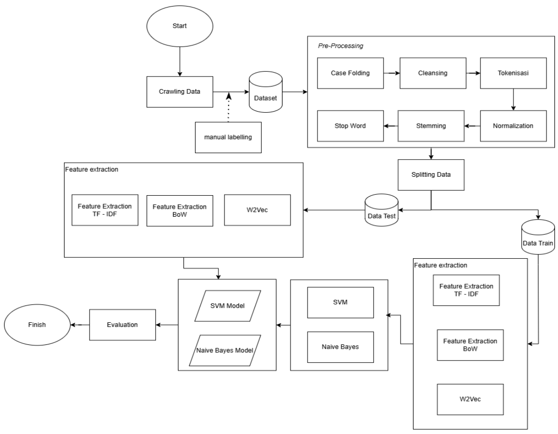

Figure 1. Flowchart

The manual labeling stage is carried out by six annotators who are fluent in Indonesian, categorizing tweets into four emotions: happy, sad, angry, and afraid, using clear criteria such as "*Senangnya hari ini*!" for happy, "*Kesal banget sama pelayanan ini*." for angry. The labels will be decided through voting; if there is a disagreement on a sentence with ambiguous emotions, the sentence will be reviewed. In this process, they pay attention to the words, punctuation, and implicit context.

In the preprocessing stage, the data goes through several processes, namely case folding to convert letters into small letters, cleansing to clean punctuation marks, emoticons, and URLs, tokenization to break sentences into tokens, normalization to convert slang into standards, stemming to return words to their basic form, and stopword removal to remove connecting words such as "yang" and "di" that are less relevant to emotions.

This research will use 3 extraction features namely BoW, TF-IDF, and Word2Vec to compare their performance both individually and combined. Each extraction feature has its advantages such as BoW captures word frequency but lacks in semantic understanding, while TF-IDF highlights important words by reducing noise from common terms. Word2Vec adds semantic relationships by using window=5 and size=100 parameters. Combining these methods improves the feature representation by integrating word frequency, importance, and meaning.

Naive Bayes and SVM were chosen for emotion classification because both models can handle complex text patterns efficiently. In this research, Multinomial Naive Bayes will be used because it is efficient to handle large datasets and its features represent the number of words. SVM will be tested with all its kernels (linear, polynomial, and radial basis function) to find out which kernel has the best performance for emotion classification.

This research will split the data with a ratio of 20:80 as this provides enough training data to build a robust model, while also providing a sufficient proportion of test data to evaluate the model's capabilities. The evaluation is done using accuracy, precision, recall, and F1-score to measure the effectiveness of the model. Analysis of the model results provides insight into the model's performance in classifying emotions.

RESULTS AND DISCUSSION

Result

The data used in this study was obtained from user tweets who were willing to give permission through filling in the username on Microsoft Form, then the data was collected using the crawling method. The total tweets collected were 8978 tweets. After obtaining the tweet dataset, the next step is manual labelling carried out by six annotators who master the Indonesian language. The selection of the four classes is based on their relevance to emotional expressions commonly observed in social media interactions. for the labelling results can be seen in table 1.

Table 1. Labelling result

Label	Full Text
Happy	*@namiawui Sekolah aku ga pake SPP sih fully gratis*
Sad	*Sampe kapan yaa harus terbiassa dengan perpisahan???*
Angry	*Mau marah bgt sama feren woi rambut gue dia potongin tapi ga rataaa*
Fear	*eh serius deh sedih bgt kayanya klo sedih begini di spam foto cantik sedihnya ilang jadi mana pap hari ini? @R_AdelJKT48*

After the labelling process, the label distribution consists of Happy with 2,723 data, Sad with 2,492 data, Angry with 2,001 data, and Fear with 1,762 data. It can be seen in Figure 2 that there is data imbalance where Happy class has the highest number and Fear has the lowest number. Data imbalance can affect the accuracy because the model learns more about classes that have more numbers, the model will not learn too much on the model that has a small number.

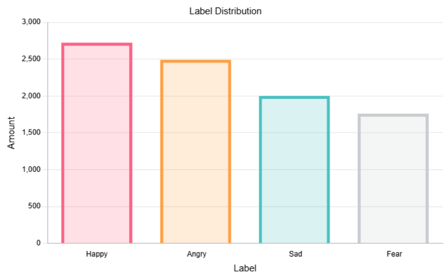

Figure 2. Label distribution

After labelling 8,978 data tweets that already have happy, sad, angry, and fear labels, the tweet data goes through a preprocessing stage, starting with case folding stage, the text will be converted to all lowercase letters, and then, in cleansing, it will clean the text from URLs, mentions, hashtags, and symbols; the tokenizing process breaks the text into individual tokens. Normalization converts slang to formal language. Stemming converts words into their basic form to reduce variations of words with the same root. Finally, stopword removal is carried out using the NLTK library and the Indonesian language stopwords list, ensuring that only relevant keywords are retained, resulting in simpler text that is ready for classification. Table 3 highlights how this comprehensive cleaning process reduced noise and standardized text data, resulting in better feature extraction and improved model performance.

Table 2. Preprocessing

Steps	Result
Full Text	*Serem banget liat berita hari ini!!!, semoga pelakunya cepet ketangkep*
Case Folding	*serem banget liat berita hari ini!!!, semoga pelakunya cepet ketangkep*
Cleansing	*serem banget liat berita hari ini semoga pelakunya cepet ketangkep*
Tokenization	*[serem, banget, liat, berita, hari, ini, semoga, pelakunya, cepet, ketangkep]*
Normalization	*[serem, banget, lihat, berita, hari, ini, semoga, pelaku, cepat, tertangkap]*
Stemming	*[serem, banget, lihat, berita, hari, ini, semoga, pelaku, cepat, tangkap]*
Stopword Removal	*[serem, banget, lihat, berita, semoga, pelaku, cepat, tangkap]*

After the data passed the preprocessing stage as a text data preparation step, the data was divided into 80% for training (7,182 data) and 20% for testing (1,796 data) using the train_test_split function from scikit-learn. The split was done with random_state = 42 to maintain consistency of results. in this testing stage will be divided into two, half preprocessing where the text will only pass the case folding, cleansing, and tokenizing stages and full preprocessing where the text passes all pre-processing processes. In addition, this test uses extraction features, such as TF-IDF, BoW and Word2Vec, which are tested individually or combined using the numpy library, np.hastack. The purpose of this test is to evaluate the effect of preprocessing (half preprocessing and full preprocessing) on the performance of the model in emotion classification, as well as to analyze the performance and suitability of feature extraction methods individually and in combination with SVM and Naive Bayes. Table 3 shows the results of half preprocessing and Table 4 shows the results of full preprocessing.

Table 3. Half preprocessing

Model	Feature Extraction	Precision	Recall	F1-Score	Accuracy
SVM	TF-IDF	0.67	0.67	0.67	0.6715
	BoW	0.66	0.66	0.65	0.6531
	Word2Vec	0.55	0.34	0.26	0.3875
	TF-IDF + BoW	0.68	0.68	0.68	0.6837
	TF-IDF + Word2Vec	0.66	0.57	0.57	0.5952
	BoW + Word2Vec	0.64	0.57	0.58	0.5941
	TF-IDF + BoW + Word2Vec	0.69	0.66	0.66	0.6665
Naïve Bayes	TF-IDF	0.67	0.65	0.66	0.6581
	BoW	0.67	0.65	0.65	0.6570
	Word2Vec	0.26	0.34	0.29	0.3569
	TF-IDF + BoW	0.68	0.66	0.67	0.6637
	TF-IDF + Word2Vec	0.66	0.60	0.61	0.6125
	BoW + Word2Vec	0.66	0.57	0.57	0.5707
	TF-IDF + BoW + Word2Vec	0.68	0.61	0.61	0.6102

The results in Tables 3 and 4 show that the combination of TF-IDF + BoW with full preprocessing overall gives better accuracy. SVM achieved Precision 0.71, Recall 0.70, F1-Score 0.71, and Accuracy 0.7021, while Naïve Bayes achieved Precision 0.67, Recall 0.67, F1-

Score 0.67, and Accuracy 0.6765. Full preprocessing (normalizing slang, stemming, and removing irrelevant words) is proven to make the model more effective in recognizing patterns because the noise in the dataset is reduced and the model becomes more focused on detecting patterns accurately. The combination of TF-IDF + BoW excels because TF-IDF emphasizes weight on important words, while BoW presents word frequency as a simple yet effective representation. Word2Vec is less optimal due to dense embedding and requires large datasets, making it more suitable for neural network-based models. Naïve Bayes struggles with Word2Vec due to its assumption of independent features, and SVM does not get high-value discrete features from Word2Vec. These limitations, plus Word2Vec's lack of context-specificity in short texts such as tweets, lead to a decrease in precision, recall, and F1-score. Next, parameter tests were conducted to show that parameter selection is important.

Table 4. Full preprocessing

Model	Feature Extraction	Precision	Recall	F1-Score	Accuracy
SVM	TF-IDF	0.69	0.69	0.69	0.6893
	BoW	0.71	0.70	0.70	0.6971
	Word2Vec	0.50	0.35	0.27	0.3914
	TF-IDF + BoW	0.71	0.70	0.71	0.7021
	TF-IDF + Word2Vec	0.70	0.60	0.59	0.6186
	BoW + Word2Vec	0.65	0.58	0.58	0.5997
	TF-IDF + BoW + Word2Vec	0.70	0.67	0.68	0.6826
Naïve Bayes	TF-IDF	0.67	0.66	0.66	0.6698
	BoW	0.67	0.66	0.66	0.6665
	Word2Vec	0.29	0.34	0.30	0.3602
	TF-IDF + BoW	0.67	0.67	0.67	0.6765
	TF-IDF + Word2Vec	0.67	0.61	0.62	0.6219
	BoW + Word2Vec	0.67	0.59	0.60	0.5952
	TF-IDF + BoW + Word2Vec	0.69	0.62	0.63	0.6297

Table 5. Parameter test result

Kernel	C	Precision	Recall	F1-Score	Acuracy
Linear	0.1	0.7322	0.7116	0.7188	0.7121
Rbf	1	0.7094	0.7035	0.7059	0.7021
Poly	10	0.6514	0.5234	0.5037	0.5518

To find the best parameter and kernel, all C values (0.01 to 100) and kernels (Linear, RBF and Poly) were tested. Table 5 shows the best results of each kernel along with their C values. The results of SVM parameter testing show that kernel selection and C value greatly affect the performance of the model. The linear kernel with C=0.1 gives the highest accuracy of 71.21% and F1-score of 0.7188, proving its superiority in handling linearly separable data well, while maintaining a balance between precision and recall. C value creates a wider decision margin, allowing the model to be more tolerant to errors in the training data and improving generalization ability. On the other hand, the RBF kernel at C=1 also shows competitive performance, but it is less optimal than the linear kernel for simpler datasets. The polynomial kernel with C=10 has a lower performance 0.5518 accuracy and F1-score 0.5037, as its complexity does not match the data characteristics. This finding underscores the importance of systematic parameter tuning, such as grid search, to determine the appropriate kernel and C value. In future model development, these results can guide the selection of appropriate parameters to improve classification performance.

To ensure data balance, the label distribution was flattened by reducing the number of samples in larger classes to match the smallest class (Fear, with 1,750 samples). This undersampling approach aimed to eliminate biases caused by class dominance, allowing for a fair comparison of SVM and Naïve Bayes models. Figure 3 illustrates the balanced label distribution after this process.

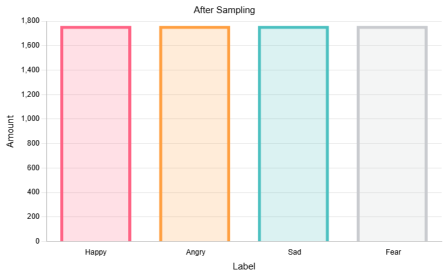

Figure 3. After Sampling Label Distribution

Both models were evaluated using their best-performing parameters and feature extraction methods identified in previous tests. Table 7 presents the results of this comparison. SVM utilized a linear kernel with C=0.1 and TF-IDF + BoW features, while Naïve Bayes also used TF-IDF + BoW. The balanced data further highlights the models' capabilities in handling equally distributed classes.

Table 6. Comparison of Naïve Bayes and SVM result

Metric	SVM TF-IDF + BoW	Naïve Bayes TF-IDF + BoW
Precision	0.80	0.76
Recall	0.780	0.76
F1-score	0.780	0.76
Accuracy	0.780	0.755

After label balancing, the SVM model with TF-IDF + BoW feature extraction outperforms Naïve Bayes, achieving precision 0.80, recall 0.78, F1-score 0.78, and accuracy 0.780, compared to Naïve Bayes which has precision, recall, and F1-score 0.76 and accuracy 0.755. The performance of SVM is superior to Naive Bayes because SVM is able to handle complex patterns through kernel functions and high features provided by TF-IDF and BoW and optimized parameters. Naïve Bayes assumes feature independence, which limits its effectiveness in capturing complex relationships. These results explain the importance of preprocessing, selection of appropriate feature extraction and parameter optimization so as to broaden the insight into what factors influence classification.

Discussion

This research achieves its goal by examining preprocessing, feature extraction, parameter settings, and comparing the performance of Naïve Bayes and SVM in classifying emotions. Full preprocessing can improve the performance of the model in recognizing emotion patterns by reducing bias in the data. Normalization (converting slang into standard language) normalization reduces the complexity of text data thereby making word distribution more uniform. The results of research conducted by Prasetija et al. (2022) revealed that the more slang words that are normalized, the better the accuracy results produced by the model. Stemming can make emotion patterns easier to recognize because the model does not need to deal with variations of the same word. For example, "kemarahan", and "memarahi" can all be associated with negative emotions if they have been stemmed into "*marah*". Research conducted by Paskahningrum et al. (2023) stemming reduces word variation by returning the affixed word to its base form. This makes it easier for the model to learn a word. Stopword removal removes words that have no meaning such as "yang", "di", "dan" so as to reduce words that are considered unimportant in the text. Research conducted by Habberrih & Ali (2024) states that stopwords can improve accuracy by removing words that are considered noise in the model. By going through the stages above, the model will recognize patterns more accurately than text that only goes through half the preprocessing process. The combination of TF-IDF + BoW proves to be effective as it provides a balanced representation of word frequency and importance, allowing SVM to make optimal use of these features for emotion classification. with its ability to handle high-dimensional feature spaces, SVM certainly benefits from this combination of feature extraction, allowing it to recognize complex relationships between features and deliver superior performance. on the other hand, although computationally efficient and suitable for simpler datasets, Naive Bayes has limitations in processing complex patterns due to its assumption of feature independence. Word2Vec is less optimal when combined with SVM and Naive Bayes because these two models are not designed to utilize processing that captures word relationships or context, which is Word2Vec's advantage in representing word meanings based on their patterns in the text. Word2Vec is better suited with models that can understand text patterns such as deep learning models or Neural Network-Based Models.

The results of this study are in line with previous research conducted by Sarimole & Kudrat (2024), showing that SVM gets an accuracy of 87.95% superior to Naive Bayes which only gets 65% accuracy. Both models use TF-IDF extraction features with a dataset of 1081 data. However, their research does not explain whether the completeness of preprocessing affects the performance of the model. The dataset they used, which consisted of user opinions on a particular topic, was also relatively small, with preprocessing limited to standard steps such as tokenization and stopword removal, without handling complexities such as slang or emojis. Their SVM implementation also did not include details about the parameters or kernels used. Similarly, research conducted by Supian et al. (2024) showed the ability of SVM to achieve 94% accuracy which managed to outperform the accuracy of Naive Bayes which only reached 91%. Their study also only relies on TF-IDF extraction features, thus limiting insight into the performance of other extraction features, such as Word2Vec or even a combination of extraction features. Their preprocessing steps were similar but did not evaluate the impact of full preprocessing techniques, such as stemming or normalization to handle language complexity especially in tweets. This research emphasizes the importance of full preprocessing, which normalizes, stems, and removes irrelevant data so as to better improve accuracy. By combining TF-IDF and BoW, this research achieves richer feature representation and higher accuracy compared to single methods. Variations in dataset size, preprocessing approaches, and extraction methods most likely explain the performance differences between these studies. This research can be replicated by using the same approach even if the model is different, with complete preprocessing, extracting features appropriate to the dataset and model and the best parameters of the model.

SVM with a linear kernel and C = 0.1 achieved an accuracy of 0.7121 and F1-score of 0.7188, highlighting the importance of parameter tuning, indicating an optimal balance between complexity and performance. After balancing the label distribution, SVM further improved with an accuracy of 0.7801 and precision of 0.80, outperforming Naïve Bayes with accuracy: 0.755 and precision of 0.76. These results confirm the superiority of SVM in emotion classification, especially on balanced data.

This study confirms that SVM is more suitable than Naïve Bayes for emotion classification tasks, filling the gap in previous research by emphasizing the importance of preprocessing steps, such as normalization, stemming, and stopword removal, as well as the selection of suitable feature extraction techniques such as TF-IDF + BoW, which significantly improve the performance of the model. To address data imbalance, data balancing techniques such as SMOTE or similar can be used. In addition, the findings also underscore how influential parameter tuning can be, in that a small value of C in the linear kernel helps the model provide a more balanced decision by allowing a wider margin around the decision line. This makes the model more tolerant of small errors in the training data, so it is better able to recognize common errors in the data, which makes it better at handling new data. Not only in SVM, finding the best parameters is also important to do in models that have many parameters such as Random Forest, Decision Trees and even in deep learning to make the model can produce its best potential.

This research highlights the crucial significance of preprocessing, feature extraction, and parameter adjustment in enhancing model efficacy for emotion classification. The outcomes are applicable to enhancing models for analyzing customer service sentiment, monitoring social media activities, and evaluating mental health. Through integrating diverse preprocessing methods, feature extraction techniques, and parameter optimization, this study provides a comprehensive framework to advance emotion classification and address prior research constraints.

CONCLUSION

The results of this study show that the application of full preprocessing, a combination of TF-IDF + BoW feature extraction, and optimal parameter selection can improve the performance of emotion classification in Indonesian tweets. The Support Vector Machine (SVM) algorithm achieved the highest accuracy of 78.01%, exceeding the accuracy of Naïve Bayes which only reached 75.53%. This method proved effective in handling the complexity of the Indonesian language on social media. The findings have the potential to be applied to customer sentiment analysis, social media monitoring, and emotion pattern detection to support mental health. Further research is recommended to explore more advanced data balancing techniques and deep learning approaches to capture more complex emotion patterns.

REFERENCES

Akbar, B. M., Akbar, A. T., & Husaini, R. (2022). Analysis of Sentiments and Emotions about Sinovac Vaccine Using Naive Bayes. *Telematika: Jurnal Informatika Dan Teknologi Informasi*, *19*(2), 185–200. https://doi.org/10.31315/telematika.v19i2.7601

Fudholi, D. H. (2021). Klasifikasi Emosi pada Teks dengan Menggunakan Metode Deep Learning. *Syntax Literate; Jurnal Ilmiah Indonesia*, *6*(1), 546-553.

Galke, L., & Scherp, A. (2021). Bag-of-words vs. graph vs. sequence in text classification: Questioning the necessity of text-graphs and the surprising strength of a wide MLP. *ArXiv Preprint ArXiv:2109.03777*.

Habberrih, A., & Abuzaraida, M. A. (2024). Sentiment Analysis of Libyan Dialect Using Machine Learning with Stemming and Stop-words Removal. *International conference on communication engineering and computer science*, 259-264. https://doi.org/10.24086/cocos2024/paper.1171

Indira, R., & Maharani, W. (2021). Personality detection on social media twitter using long short-term memory with word2vec. *IEEE International Conference on Communication, Networks and Satellite (COMNETSAT)*, 64-69. IEEE. https://doi.org/10.1109/COMNETSAT53002.2021.9530820

Jabasheela, L. (2024). Secured Text-Based Emotion Classification Using Machine Learning With NLP. *Educational Administration: Theory and Practice*, *30*(5), 901-910.

Mokari, A., Guo, S., & Bocklitz, T. (2023). Exploring the steps of infrared (IR) spectral analysis: Pre-processing, (classical) data modelling, and deep learning. *Molecules*, *28*(19), 6886. https://doi.org/10.3390/molecules28196886

Ningsih, M. R., Unjung, J., Pertiwi, D. A. A., Prasetiyo, B., & Muslim, M. A. (2024). Optimized support vector machine with particle swarm optimization to improve the accuracy amazon sentiment analysis classification. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, 101-108. https://doi.org/10.22219/kinetik.v9i1.1888

Nugroho, R. A., Cholissodin, I., & Indriati, I. (2021). Implementasi Naïve Bayes Classifier untuk Klasifikasi Emosi Tweet Berbahasa Indonesia pada Spark. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, *5*(1), 301-310.

Paskahningrum, Y. K., Utami, E., & Yaqin, A. (2023). A Systematic Literature Review of Stemming in Non-Formal Indonesian Language. *International Journal of Innovative Science and Research Technology*, *8*(1), 62-69.

Prasetija, Z. R. N. S., Romadhony, A., & Setiawan, E. B. (2022). Analisis Pengaruh Normalisasi Teks pada Klasifikasi Sentimen Ulasan Produk Kecantikan. *eProceedings of Engineering*, *9*(3), 1769.

Regita, E., Luthfiyyah, N., & Marsuki, N. R. (2024). Pengaruh Media Sosial Terhadap Persepsi Diri dan Pembentukan Identitas Remaja di Indonesia. *Jurnal Kajian Dan Penelitian Umum*, *2*(1), 46–52. https://doi.org/10.47861/jkpu-nalanda.v2i1.830

Riani, A. P., Sulistyowati, N., Ridwan, T., & Voutama, A. (2023). Klasifikasi Emosi Publik Terhadap Larangan Penggunaan Obat Sirup Menggunakan Algoritma Naive Bayes. *METHOMIKA: Jurnal Manajemen Informatika & Komputerisasi Akuntansi*, *7*(2), 325-339. https://doi.org/10.46880/jmika.Vol7No2.pp325-339

Sarimole, F. M., & Kudrat, K. (2024). Analisis Sentimen Terhadap Aplikasi Satu Sehat Pada Twitter Menggunakan Algoritma Naive Bayes Dan Support Vector Machine. *Jurnal Sains Dan Teknologi*, *5*(3), 783–790. https://doi.org/10.55338/saintek.v5i3.2702

Siddique, Z. B., Khan, M. A., Din, I. U., Almogren, A., Mohiuddin, I., & Nazir, S. (2021). Machine Learning-Based Detection of Spam Emails. *Scientific Programming*, *2021*(1), 6508784. https://doi.org/10.1155/2021/6508784

Sudianto, S. (2022). Analisis Kinerja Algoritma Machine Learning Untuk Klasifikasi Emosi. *Building of Informatics, Technology and Science (BITS)*, *4*(2), 1027-1034. https://doi.org/10.47065/bits.v4i2.2261

Supian, A., Revaldo, B. T., Marhadi, N., Efrizoni, L., & Rahmaddeni, R. (2024). Perbandingan Kinerja Naïve Bayes dan SVM pada Analisis Sentimen Twitter Ibukota Nusantara. *JURNAL ILMIAH INFORMATIKA*, *12*(01), 15-21. https://doi.org/10.33884/jif.v12i01.8721

Zhang, F., Chen, J., Tang, Q., & Tian, Y. (2024). Evaluation of emotion classification schemes in social media text: an annotation-based approach. *BMC Psychology*, *12*(1), 503. https://doi.org/10.1186/s40359-024-02008-w

Zhang, P., Ma, Z., Ren, Z., Wang, H., Zhang, C., Wan, Q., & Sun, D. (2024). Design of an Automatic Classification System for Educational Reform Documents Based on Naive Bayes Algorithm. *Mathematics*, *12*(8), 1127. https://doi.org/10.3390/math12081127

Zharifa, A. H. A., & Ujianto, E. I. H. (2024). Analisis Sentimen Publik di Twitter Pasca Debat Kelima Pilpres 2024 dengan Naive Bayes. *Edumatic: Jurnal Pendidikan Informatika*, *8*(2), 754-763. https://doi.org/10.29408/edumatic.v8i2.28048