Evaluation of Support Vector Machine, Naive Bayes, Decision Tree, and Gradient Boosting Algorithms for Sentiment Analysis on ChatGPT Twitter Dataset

[1]Salsabila Rabbani, [2]Dea Safitri, [3]Farida Try Puspa Siregar, [4]Rahmaddeni, [5]Lusiana Efrizoni
[1,2,3,4,5]Departement of Informatics Engineering, STMIK Amik Riau, Jalan Purwodadi, Pekanbaru, Riau
Email: [1]201003182104@sar.ac.id, [2]2010031802001@sar.ac.id,
[3]2010031802042@sar.ac.id, [4]rahmaddeni@sar.ac.id, [5]lusiana@stmik-amik-riau.co.id

Article Info
Article history:
Received Jul 18th, 2023
Revised Oct 20th, 2023
Accepted Nov 12th, 2023

Keyword:
Decision Tree
Gradient Boosting
Naive Bayes
Sentiment Analysis
Support Vector Machine

ABSTRACT
ChatGPT is a language model used to generate text and interact with users in a conversational format. The model is designed to provide relevant and useful responses based on the context of the ongoing conversation. However, as more and more people use ChatGPT, there is a difficulty in determining whether the responses given by users are positive, negative, or neutral towards the use of ChatGPT. Therefore, sentiment classification of ChatGPT taken from Twitter social media is carried out to determine user responses to the language model. The dataset used is sourced from the kaggle website with a total of 20,000 data. In this study, a comparison of classification algorithms including Support Vector Machine (SVM), Naïve Bayes, Decision Tree, and Gradient Boosting will be conducted. Through the results of the study, it was found that Twitter user responses were negative towards ChatGPT and the Support Vector Machine algorithm with a 90:10 data division had the highest accuracy value with 80% achievement compared to other algorithms. This research is expected to help developers and service providers to improve the performance of ChatGPT and understand user responses better.

Corresponding Author:
Rahmaddeni,
Department of Informatics Engineering,
STMIK Amik Riau,
Jalan Purwodadi, Pekanbaru, Riau.
Email: rahmaddeni@sar.ac.id

DOI: http://dx.doi.org/10.24014/ijaidm.v7i1.24662

1. INTRODUCTION

ChatGPT is an OpenAI-developed language model that utilizes the GPT (Generative Pre-trained Transformer) framework to generate natural and interactive text. The model has undergone thorough training using large volumes of data to generate coherent and contextually suitable replies [2]. ChatGPT possesses remarkable capabilities to advance academia and librarianship in novel ways, which elicit concerns and pique interest [7]. ChatGPT has become a trend in various fields today, especially in the field of technology. This is because ChatGPT can give accurate and thorough responses to a wide range of inquiries from humans, including the ability to rectify incorrect queries [8].

The use of online communication has become an ever-present thing in daily activities. The existence of information technology in the form of platforms that are developing very rapidly allows users to find information quickly and practically. One of the most famous social media platforms is Twitter, where users can share their thoughts, opinions, and experiences with the world. However, with the high volume of tweets, we need to be able to analyze and understand user sentiment towards a particular topic or entity.

To determine whether sentiment text contains positive, negative, or neutral sentiment, sentiment classification is required. Sentiment classification is a methodology employed to recognize and categorize sentiments or opinions expressed in text [1]. To understand how Twitter users respond to ChatGPT, sentiment analysis can provide valuable insights into understanding users' views and getting feedback on their experiences with ChatGPT [3].

In this research, classification will be performed using 4 different algorithm models, namely Support Vector Machine (SVM), Naïve Bayes, Decision Tree, and Gradient Boosting. These four algorithms will be tested using data mining and Natural Language Processing (NLP) processing techniques. Furthermore, they will be combined with Feature Selection, namely K-Fold and Adaboost to help the model increase the accuracy rate. The text of tweets containing sentiment towards ChatGPT will be analyzed and classified into positive, neutral, and negative categories.

Naive Bayes is an efficient algorithm in classification. This can be seen from research [22] proving that the Naive Bayes algorithm has 3% better performance when compared to Decision Tree and K-NN. In another study [23] training in the dataset showed that the Naive Bayes algorithm produced the best results of 93%. It can be said that the Naive Bayes algorithm is one of the best algorithms in classifiers.

Feature Selection Adaboost works by combining several weak learners into one stronger model. Research [9] proves that Feature Selection Adaboost can increase the accuracy of the Naïve Bayes algorithm in every splitting data tested and the SVM algorithm outperforms with an accuracy rate of 82% with 90:10 data splitting.

In the context of feature selection, k-fold cross-validation can help in selecting the most informative or relevant subset of features to improve model performance. Research [4] comparing the performance of Naïve Bayes, Support Vector Machine (SVM), Random Forest, Neural Network, Decision Tree, and K-Nearest Neighbor (K-NN) algorithms using feature extraction selection between TF-IDF, Word2Vec, and Word Embedding (one hot encoding) shows the results of evaluating the performance of the TF-IDF feature extraction model with the SVM method is the highest by utilizing K-Fold cross-validation, an accuracy of 92.6% was achieved for datasets without labels, while labeled datasets obtained 80.3% accuracy.

The Support Vector Machine (SVM) model constructs a hyperplane that possesses the largest possible gap between two categories, to maximize the separation between positive and negative classes. In [5] of the 3 datasets studied, the SVM algorithm outperformed the highest accuracy of 91%-93%. In [24], the highest accuracy of this research is the linearSVC model as a very frequently used algorithm in machine learning which has a performance of 86%.

A decision tree is a decision-making method that organizes each option into a branched form. Another study that discussed the hate speech of twitter users by [6], the Decicion Tree algorithm outperformed other machine learning with an accuracy of 97.96%, followed by the SVM algorithm with 97.71%. In another study [25] on covid-19 vaccine sentiment, it was found that the decision tree classification vectorized using TF-IDF resulted in a performance of 95.77%. While Gradient Boosting is an ensemble algorithm of regression models or classification tree models that combine simple parameter functions. Research [26] shows Gradient Boosting has the best performance in classification with an accuracy value of 90.65%.

Based on the results of several previous studies, research that focuses on evaluating algorithms for ChatGPT sentiment analysis on Twitter datasets has great potential to provide valuable insights in practical applications and the development of better sentiment analysis models. This kind of research can test and evaluate various sentiment analysis algorithms that use the ChatGPT model. By conducting comparisons between various algorithms, this research can help identify the most effective approaches in addressing the challenges of sentiment analysis on Twitter text. This research is expected to explore Twitter users' knowledge of ChatGPT. The results of the analysis can help developers and service providers to improve the performance of ChatGPT and understand users' responses better. This research can present an analysis of the extent to which the algorithm is able to recognize and classify diverse sentiments, including neutral, positive, and negative sentiments.

2. RESEARCH METHODS

The sentiment analysis research about ChatGPT involves a well-planned and systematic approach to address the problem effectively. The dataset, sourced from public opinions on Twitter, needs to be carefully selected to ensure its relevance to the problem at hand, as well as its adequacy in training and evaluating the model. In Figure 1, there is a sequence of steps: collecting data, then pre-processing the data, then splitting the data, by dividing the training data and test data. In the training data, the model classification and feature selection stages are carried out, after which the model evaluation stage is carried out, and ends with a conclusion from the comparison results of the four methods.

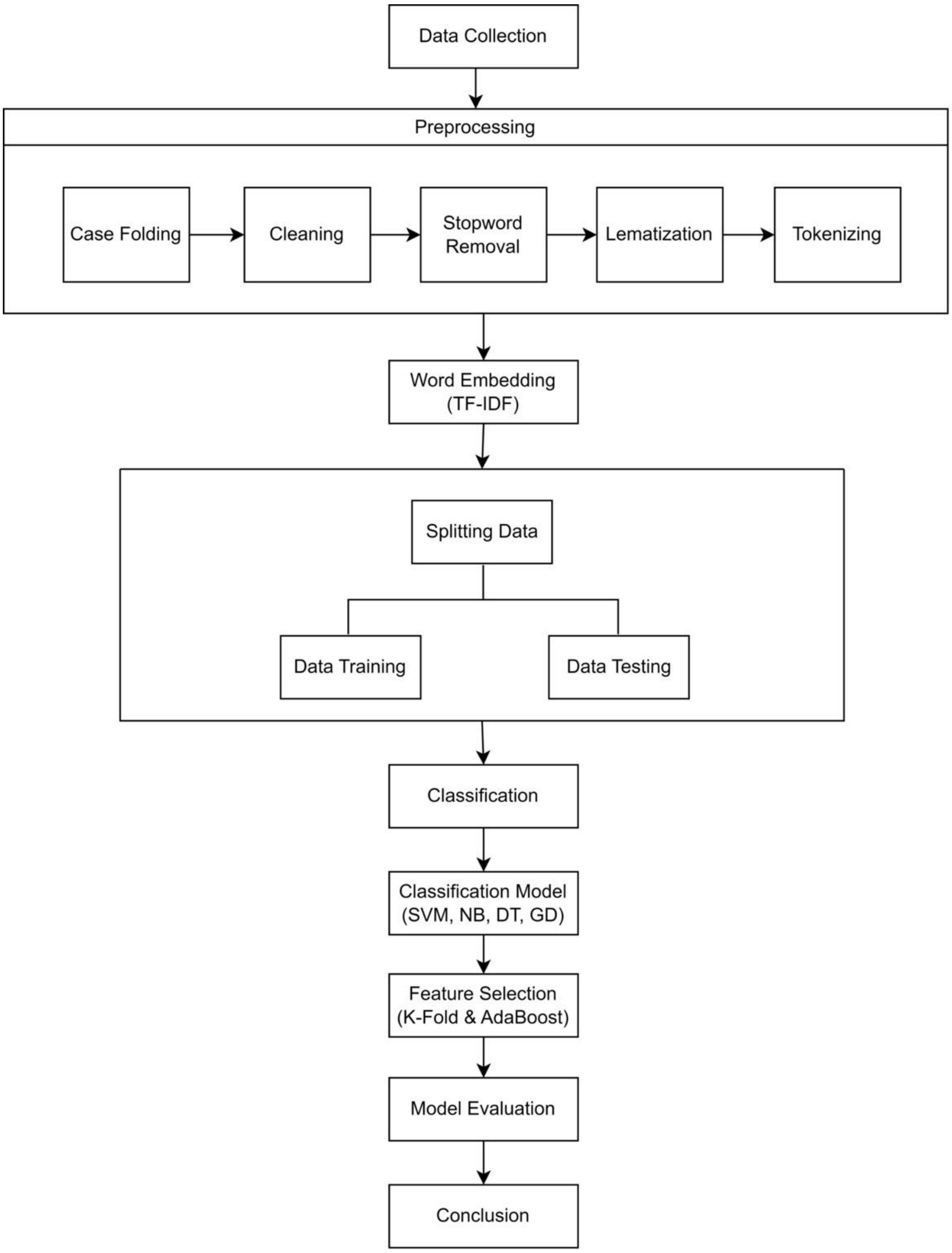

Figure 1. Research Methods

This research uses steps such as, data collection through Kaggle which uses 20,000 tweets from the Twitter platform and then applies sentiment analysis steps. The data was classified into negative, positive, or neutral. Pre-processing involved converting letters to lowercase, data cleaning, removal of irrelevant words, lemmatization, and tokenization. Words are analyzed using the TF-IDF method. Then the data is divided into training and testing, and four classification methods, namely SVM, Naïve Bayes, Decision Tree, and Gradient Boosting, are evaluated. There is also feature selection using AdaBoost and K-Fold methods. Model evaluation is done based on accuracy. This research aims to find the best method in classifying tweet sentiment.

2.1. Data Collection

The research employs a collection of tweet information for analysis taken from the Twitter platform and obtained through the Kaggle data source. The amount of data used was 20,000 tweets. The keywords used to retrieve tweet data are phrases containing the phrase "ChatGPT". Labeling of each data is done automatically and directly from the public Kaggle dataset, there are 3 labels given, namely negative, positive and neutral classes. Some of the libraries used in testing use the Python programming language through the Jupyter IDE, which includes pandas, numpy, nltk, matplotlib, and word cloud.

2.2. Preprocessing

The dataset collected is still in the form of unstructured data, so it is necessary to do preprocessing so that the data becomes more structured. The preprocessing process carried out in this study involves several stages, namely changing letters to lowercase letters (case folding), cleaning data (cleaning), filtering words (filtering) or stopword removal, restoring basic words (lemmatization), and separating words into tokens (tokenizing).

1. The Case folding stage involves the process where capital letters are converted into lowercase letters as a whole [10].
2. Cleaning stage is the stage of eliminating words that are not needed [10]. For example, numbers (1), word separators namely commas (,), periods (.), exclamation marks (!), and other punctuation marks. Tokenization is the act of dividing a provided text into separate words or phrases [11].
3. Next is the filtering stage, where the selection of words that are considered important from the tokenization results is done [13]. Words that do not provide information such as the, is, a, and, are, etc., will be removed.
4. Lemmatization will be utilized on the refined reviews, while stemming involves converting words to their root form to enhance model efficiency [14]. Example: helping, helped, helps will become help.
5. Tokens are unique words that will be used as identification for sentiment clustering. Before tokenizing, the words in the tweet are separated using space characters. The tweet data is converted to lowercase before processing. Links, usernames, and symbols are removed, and spelling or writing errors are corrected to make the text more accurate.

2.3. Word Embedding

Word Embedding is a process that converts words into numerical representations (word vectors) [9]. The research applies the TF-IDF method on word embedding in documents. This technique converts text data into vectors, considering the proper word order. Each word in the corpus receives a numerical value through TF-IDF, indicating its significance in the collection [15]. These numerical values derived from TF-IDF are then utilized as input for a supervised learning classifier [16] to enable interpretation within the context of machine learning methods. To perform TF-IDF calculation using the Python Sklearn library, we can use TfidfVectorizer. Here is the word weighting formula.

$$\mathrm{W}_{ij} = tf_{ij} \times \log\left(\frac{n}{df}\right) \tag{1}$$

Where Wij is the weight that determines the importance of a word or term, tfij is the term frequency which is the number of times the term appears, and ndf is the total document frequency of the term in the entire document collection.

2.4. Splitting Data

The data was split into training data and test data, consisting of three experiments conducted for data separation, namely:

1. Training data consists of 70% and testing data of 30%.
2. The training data consists of 80% and the testing data of 20%.
3. Training data consists of 90% and testing data 10%.

2.4.1. Data Training

The utilization of training data for the purpose of training this research system is to enable the SVM, Naïve Bayes, Decision Tree, and Gradient Boosting classification methods to acquire the ability to categorize comments into negative, neutral, or positive. The training data plays a crucial role in familiarizing these classification methods with the patterns and characteristics of different types of comments, enabling them to learn and develop the necessary skills to accurately classify new comments.

2.4.2. Data Testing

After training the classification model, the subsequent phase involves assessing the effectiveness of the approach using a test dataset. The objective of this testing process is to evaluate the performance of SVM, Naïve Bayes, Decision Tree, and Gradient Boosting classification methods. This is accomplished by inputting new data into the model, which will then correctly categorize the new information as either negative, neutral, or positive.

2.5. Classification

This research attempts to apply four different classification methods to analyze sentiment on tweet data. The classification methods used are Support Vector Machine (SVM), Naïve Bayes, Decision Tree, and Gradient Boosting, which aims to compare and evaluate the performance of each method in classifying tweet sentiment.

2.5.1. Support Vector Machine

The SVM (Support Vector Machine) method relies on support vectors to separate classes of data that have different characteristics [17]. SVM tries to find the hyperplane that best separates tweet data with negative, neutral, and positive sentiments. SVM has a good ability to handle complex data and has advantages in handling cases with high feature dimensions. In this case, the general formula for linear SVM can be written as follows:

$$f(x) = \mathrm{sign}(w.x + b) \tag{2}$$

Where f(x) is the prediction function, w is the hyperplane normal vector, x is the input feature vector, and b is the bias or intercept.

2.5.2. Naïve Bayes

The primary characteristic of Naïve Bayes is its ability to generate robust hypotheses by considering specific conditions or events [18]. This approach computes the conditional probability for each sentiment category by analyzing the occurrence of words in the training dataset. It then classifies new tweet data by assigning it to the category with the highest probability. Here is the equation:

$$P(C|X) = \left(\frac{P(X|C) * P(C)}{P(X)}\right) \tag{3}$$

The equation (3) represents the conditional probability P(C|X), which is calculated as the product of the probability of event X occurring given that C has happened (P(X|C)) and the joint probability of C and X (P(C)P(X)). In other words, it determines the likelihood of event C happening given that event X has occurred.

2.5.3. Decision Tree

Decision Tree is a sequential model that combines a series of basic tests to classify data that uses numerical features in the dataset to divide the data into smaller subsets [19]. The decision tree is built by making decisions at each node based on features that divide the data into smaller subsets. The decision at each node is based on criteria such as entropy or information gain. In the Decision Tree algorithm, there are several formulas used to build and predict using decision trees. Here are some formulas related to Decision Tree:

1. Gini Index, used to measure impurity at nodes in the decision tree.

$$\text{Gini Index} = 1 - \Sigma(p\_i)\text{^}2 \tag{4}$$

2. Information Gain, used to measure the decrease in impurity resulting from dividing data based on a feature at each node. The Information Gain formula is:

$$\text{Information Gain} = \text{Impurity before sharing} - \text{Impurity after sharing} \tag{5}$$

3. Entropy, used to measure the impurity of the nodes in the decision tree. The Entropy formula is:

$$\text{Entropy} = -\Sigma(p\_i * \log2(p\_i)) \tag{6}$$

2.5.4. Gradient Boosting

The Gradient Boosting method is a classification method that combines several weak learner models into one stronger model [20]. This method iterates to improve model performance by focusing on the most difficult to predict samples. Gradient Boosting usually uses Decision Tree as a weak learner. Here are the formulas associated with Gradient Boosting in the context of classification:

1. Initial phase, initialize the initial model prediction (F_0) as the average target value on the training data.

$$F\_0 = \text{argmin } \Sigma(y - F\_0)\text{^}2 \quad (7)$$

2. Iteration (t = 1, 2, ..., T), calculate the residual (r) as the difference between the actual target (y) and the current model prediction (F_t-1).

$$r = y - F\_t\text{-}1 \quad (8)$$

Build a weak learner (usually a Decision Tree) that will predict the residual r. Each weak learner is generated with respect to samples that are still difficult to predict. Update the model prediction by adding the weak learner prediction (h_t) multiplied by the learning rate (α).

$$F\_t = F\_t\text{-}1 + \alpha * h\_t \quad (9)$$

The equation (9) represents the iterative process in which the model prediction (F_t) is updated by adding a weighted contribution from a new weak learner (h_t). This process is repeated until a specified number of iterations (T) is reached. The learning rate (α) determines the extent of each weak learner's impact on the overall model prediction.

2.6. Feature Selection

In feature selection, the most important and informative set of features is selected from the various features available in the dataset. The purpose of feature selection is to identify and retain those features that contribute most to predicting or classifying the target or variable of interest, while reducing complexity and noise that may be present in the dataset.

The feature selection used in the test is AdaBoost and K-Fold. This test was conducted to evaluate the effectiveness of AdaBoost and K-Fold feature selection in improving the accuracy of SVM, Naïve Bayes, Decision Tree, and Gradient Boosting algorithms.

2.7. Model Evaluation

Model testing is required to assess the effectiveness of four methods: SVM, Naïve Bayes, Decision Tree, and Gradient Boosting. In research evaluated using the accuracy value. The accuracy value is an evaluation matrix used to measure how well a classification or prediction model can provide correct results or match existing data. Accuracy describes the percentage of success of the model in predicting the right class or label.

To assess the effectiveness of Support Vector Machine (SVM), Naïve Bayes, Decision Tree, and Gradient Boosting algorithms utilizing AdaBoost and K-Fold, a confusion matrix table is employed to present the classification outcomes. Furthermore, during the model evaluation accuracy is computed to evaluate the classification methods for each class. These calculations are performed using the Python 3 programming language.

It measures the extent to which the model can correctly classify the data as a whole. In other words, accuracy describes the percentage of success the model has in predicting the correct class. The accuracy scores have a scale of zero to one, where higher values indicate better model performance.

$$\text{Accuracy} = \frac{(Number\ of\ True\ Predictions)}{(Total\ Number\ of\ Data)} \quad (10)$$

2.8. Conclusion

Based on the above steps, this research is expected to find the classification method that gives the best results in classifying tweet sentiment. The evaluation results will provide information on the relative performance of each classification method in terms of accuracy and their ability to classify sentiment into negative, neutral, or positive classes.

3. RESULTS AND ANALYSIS

The raw data taken is 20,000 user tweets data against ChatGPT. The raw data will be preprocessed to reduce deviations in the data to be analyzed. The preprocessing stage in this study includes removing nulls, removing URLs and symbols, converting the entire text to lowercase, removing stopwords, converting words into their base words, and separating sentences into words.

Table 1. Preprocessing text results

Tweets	Cleaning	Stopword Removal	Lemmatization	Tokenizing
ChatGPT: Optimizing Language Models for Dialog...	chatgpt optimizing language models for dialogu...	chatgpt optimizing language models dialogue rk...	chatgpt optimize language model dialogue rkrygyyn	[chatgpt, optimize, language, model, dialogue,...
Try talking with ChatGPT, our new AI system wh...	try talking with chatgpt our new ai system whi...	try talking chatgpt new ai system optimized di...	try talk chatgpt new ai system optimize dialog...	[try, talk, chatgpt, new, ai, system, optimize...
ChatGPT: Optimizing Language Models for Dialog...	chatgpt optimizing language models for dialogu...	chatgpt optimizing language models dialogue gl...	chatgpt optimize language model dialogue glebm...	[chatgpt, optimize, language, model, dialogue,...

The next stage is to assign values to words in the text using TF-IDF which can help in determining the most relevant or meaningful words in determining the sentiment of a text.

Table 2. Result of word weighting with TF-IDF

Term Frequency	Inverse Document Frequency	TF-IDF
0	539	0,3849
0	842	0,19245
0	1262	0,19245
0	1269	0,19245
0	2688	0,19245
…..	…..	…..
0	20405	0,19245
0	21451	0,19245
0	24611	0,19245
0	25765	0,19245
0	31425	0,19245

Next, implement the data into the Support Vector Machine, Naïve Bayes, Decision Tree, and Gradient Boosting models based on feature selection and data splitting ratios of 70:30, 80:20, and 90:10.

Table 3. Results of model comparison based on accuracy

Splitting Data	Without K-Fold Cross & Adaboost Validation				With K-Fold Cross Validation				Accuracy Improvement with AdaBoost			
	SVM	NB	DT	GB	SVM	NB	DT	GB	SVM	NB	DT	GB
70.30.00	0,78	0,66	0,69	0,68	0,75	0,64	0,69	0,69	0,47	0,47	0,53	0,76
80.20.00	0,78	0,66	0,69	0,68	0,76	0,65	0,69	0,69	0,47	0,47	0,54	0,76
90.10.00	0,8	0,67	0,71	0,69	0,77	0,66	0,7	0,69	0,47	0,47	0,54	0,78

The table above shows that the SVM algorithm without using feature selection with 90:10 data splitting has the highest accuracy rate which reaches 80%. While the Gradient Boosting algorithm using K-fold feature selection experienced a performance increase of 1% and using Adaboost feature selection of 8-9%. The following graph shows the results of the accuracy comparison.

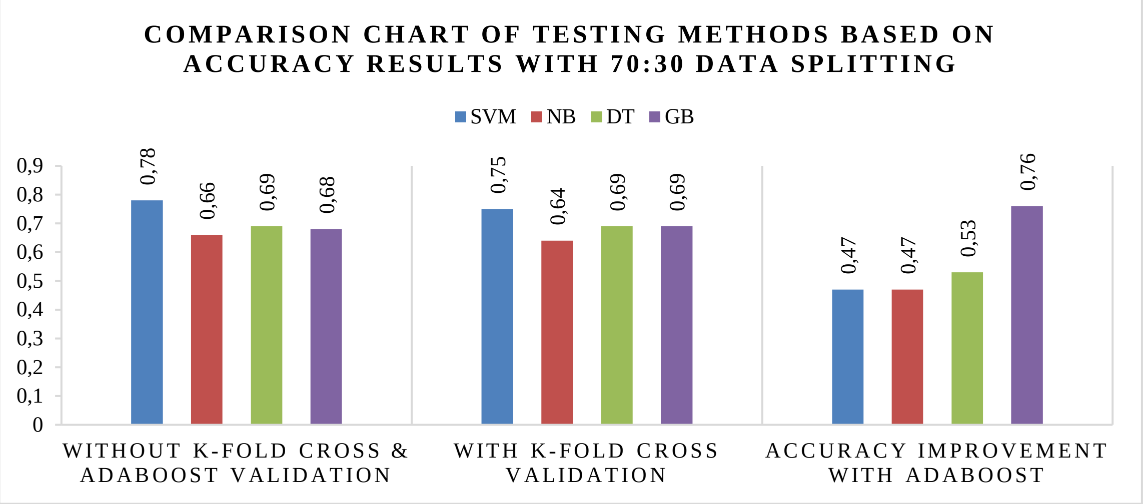

Figure 2. Comparison Chart of Testing Methods Based on Accuracy Results with 70:30 data splitting

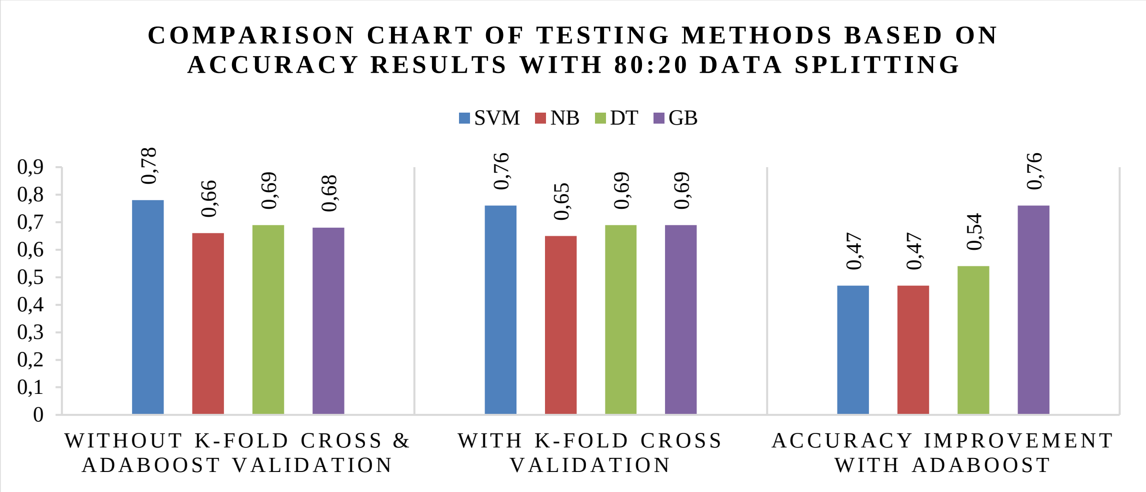

Figure 3. Comparison Chart of Testing Methods Based on Accuracy Results with 80:20 data splitting

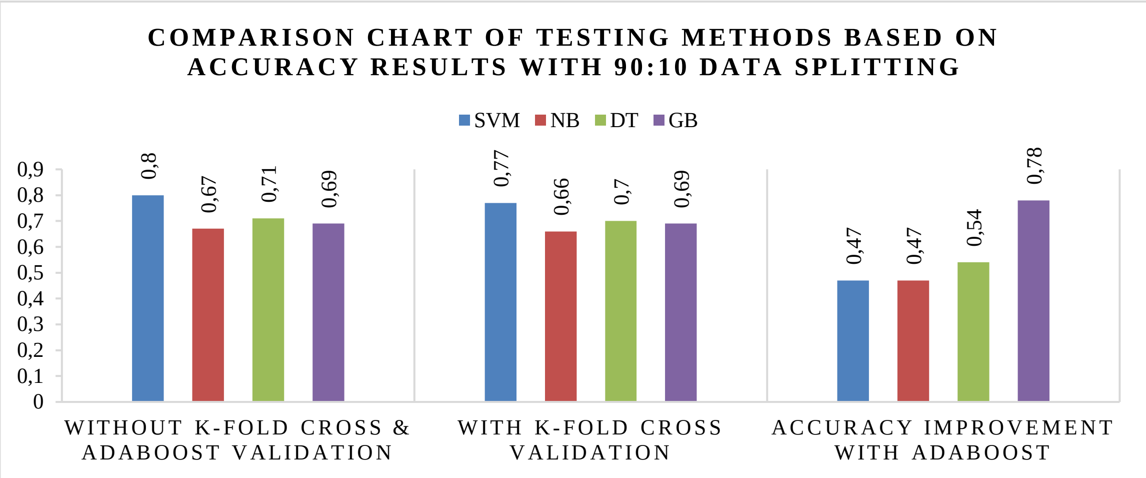

Figure 4. Comparison Chart of Testing Methods Based on Accuracy Results with 90:10 data splitting

Figure 2, 3, and 4 above shows the accuracy comparison results of the Support Vector Machine, Naive Bayes, Decision Tree, and Gradient Boosting methods on datasets both using the K-Fold and AdaBoost selection features and those that do not. In the test, which achieved the highest accuracy was in the 90:10 data division, namely Support Vector Machine without using the selection feature with the accuracy achieved was 80%, but in the Gradient Boosting algorithm using the AdaBoost feature it rose to 78%.

The dataset contains negative, positive, and neutral sentiments. Each sentiment is visualized using Chart bar and Clouding Words for the frequency of words that appear frequently.

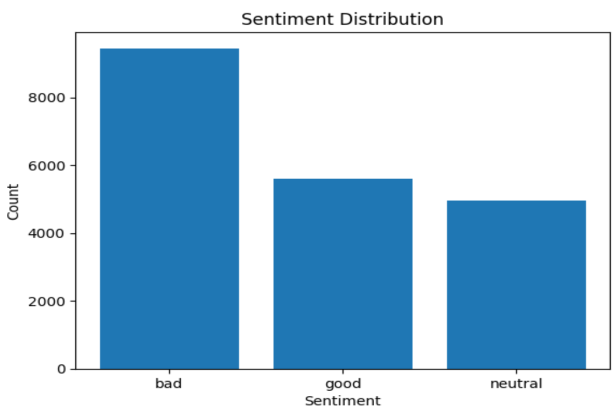

Figure 5. Visualization of ChatGPT sentiment chartbar

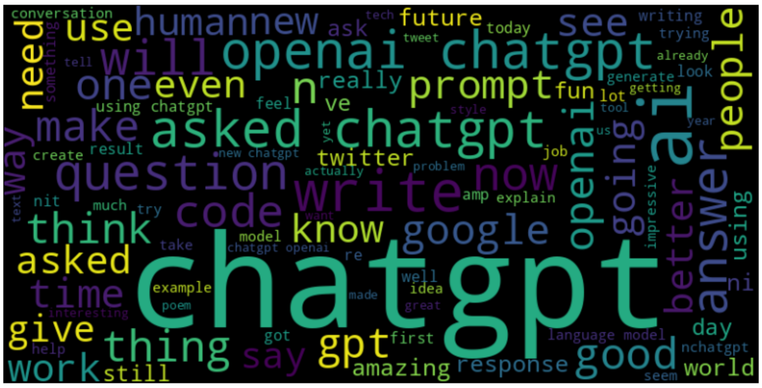

Figure 6. Clouding words for negative sentiment

In the dataset, there are 9448 sentiments labeled as negative, 5590 labeled as positive, and 4962 labeled as neutral. In Figure 6 is a visualization of the number of sentiments using a bar chart. From Figure 5 and Figure 6, it can be concluded that most twitter users give more negative responses to ChatGPT with the frequency of words that often appear in chatgpt.

4. CONCLUSION

Based on the information preparing, testing, and examination, it can be concluded that the ChatGPT dialect demonstrated has gotten a negative reaction by Twitter users. This can be seen from testing 20000 data there are 9448 data is negative responses. From the results of testing the 4 algorithms, namely, SVM, Naïve Bayes, Decision Tree, and Gradient Boosting, it shows that the SVM algorithm with 90:10 data splitting is the best algorithm for sentiment classification of ChatGPT analysis.

When utilizing Adaboost feature selection, the Gradient Boosting method model studies indicate an improvement in accuracy value in each splitting of data. This shows that Adaboost is a good feature selection for Gradient Boosting modeling. In addition, the best data splitting of ChatGPT sentiment classification is 90:10, because it has the best accuracy value of most of the models used.

Testing the Gradient Boosting algorithm using K-Fold feature selection shows an increase in accuracy. Although the increase is not as large as Adaboost, it shows that by selecting the optimal subset of features, the Gradient Boosting+K-Fold model can be more efficient and able to produce more accurate results than the original Gradient Boosting model.

Overall, this research shows that the SVM algorithm is the best technique with the highest accuracy value to classify ChatGPT sentiment. In addition, feature selection can improve the performance of Gradient Boosting algorithm in the dataset.

Based on the conclusions of the research that has been done, there are suggestions for further research that can be done, one of which uses Deep Learning Method, you can consider using deep learning methods such as neural networks to classify sentiment. Neural networks can tackle complex problems and offer better results in some cases.

REFERENCES

[1] Parimala, M., Swarna Priya, R. M., Praveen Kumar Reddy, M., Lal Chowdhary, C., Kumar Poluru, R., & Khan, S. (2021). Spatiotemporal-based sentiment analysis on tweets for risk assessment of event using deep learning approach. *Software: Practice and Experience*, *51*(3), 550-570.

[2] George, A. S., & George, A. H. (2023). A review of ChatGPT AI's impact on several business sectors. *Partners Universal International Innovation Journal*, *1*(1), 9-23.

[3] Juniarsih, S., Ripanti, E. F., & Pratama, E. E. (2020). Implementasi Naive Bayes Classifier pada Opinion Mining Berdasarkan Tweets Masyarakat Terkait Kinerja Presiden dalam Aspek Ekonomi. *JUSTIN (Jurnal Sistem dan Teknologi Informasi)*, *8*(3), 239-249.

[4] D. Marutho, Muljono, S. Rustad and Purwanto, "Sentiment Analysis Optimization Using Vader Lexicon on Machine Learning Approach," 2022 International Seminar on Intelligent Technology and Its Applications (ISITIA), Surabaya, Indonesia, 2022, pp. 98-103, doi: 10.1109/ISITIA56226.2022.9855341.

[5] Leelawat, N., Jariyapongpaiboon, S., Promjun, A., Boonyarak, S., Saengtabtim, K., Laosunthara, A., ... & Tang, J. (2022). Twitter data sentiment analysis of tourism in Thailand during the COVID-19 pandemic using machine learning. Heliyon, 8(10).

[6] Khanday, A. M. U. D., Rabani, S. T., Khan, Q. R., & Malik, S. H. (2022). Detecting twitter hate speech in COVID-19 era using machine learning and ensemble learning techniques. International Journal of Information Management Data Insights, 2(2), 100120.

[7] Lund, B. D., & Wang, T. (2023). Chatting about ChatGPT: how may AI and GPT impact academia and libraries?. Library Hi Tech News, 40(3), 26-29.

[8] Zhong, Q., Ding, L., Liu, J., Du, B., & Tao, D. (2023). Can chatgpt understand too? a comparative study on chatgpt and fine-tuned bert. *arXiv preprint arXiv:2302.10198*.

[9] Rahmaddeni, & Akbar, F. (2021). Public Opinion on Covid-19 Vaccination in Indonesia: A Sentiment Analysis on Twitter. *International Journal of Advanced Intelligence and Data Mining*, 6(1), 8-17.

[10] Sidik, F., Suhada, I., Anwar, A. H., & Hasan, F. N. (2022). Analisis Sentimen Terhadap Pembelajaran Daring Dengan Algoritma Naive Bayes Classifier. Jurnal Linguistik Komputasional, 5(1), 34-43.

[11] A. S. Alammary, "Arabic Questions Classification Using Modified TF-IDF," in IEEE Access, vol. 9, pp. 95109-95122, 2021, doi: 10.1109/ACCESS.2021.3094115.

[12] M. S. Neethu and R. Rajasree, “Sentiment analysis in twitter using machine learning techniques,” in 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), 2013, pp. 1–5.

[13] Khaira, U., Johanda, R., Utomo, P. E. P., & Suratno, T. (2020). Sentiment analysis of cyberbullying on twitter using SentiStrength. Indones. J. Artif. Intell. Data Min, 3(1), 21.

[14] S. Al-Saqqa, A. Awajan, and S. Ghoul, ‘‘Stemming effects on sentiment analysis using large arabic multi-domain resources,’’ in Proc. 6th Int. Conf. Social Netw. Anal., Manage. Secur. (SNAMS), Oct. 2019, pp. 211–216.

[15] S. Amin et al., "Recurrent Neural Networks With TF-IDF Embedding Technique for Detection and Classification in Tweets of Dengue Disease," in IEEE Access, vol. 8, pp. 131522-131533, 2020, doi: 10.1109/ACCESS.2020.3009058.

[16] S. Nirmal and T. Verma, ‘‘E-Mail spam detection and classification using SVM and feature Extraction,’’ Int. J. Advance Res., Ideas Innov. Technol., vol. 3, no. 3, pp. 1491–1495, 2017.

[17] M. O. Pratama et al., “The sentiment analysis of Indonesia commuter line using machine learning based on twitter data,” J. Phys. Conf. Ser., vol. 1193, no. 1, pp. 1–6, 2019, doi: 10.1088/1742-6596/1193/1/012029.

[18] Y. Findawati, I. R. I. Astutik, A. S. Fitroni, I. Indrawati, and N. Yuniasih, “Comparative analysis of Naïve Bayes, K Nearest Neighbor and C.45 method in weather forecast,” J. Phys. Conf. Ser., vol. 1402, p. 066046, Dec. 2019, doi: 10.1088/1742-6596/1402/6/066046.

[19] I. S. Damanik, A. P. Windarto, A. Wanto, S. R. Andani, and W. Saputra, “Decision Tree Optimization in C4. 5 Algorithm Using Genetic Algorithm,” in Journal of Physics: Conference Series, 2019, vol. 1255, no. 1, p. 012012.

[20] Yang, H., Luo, Y., Ren, X., Wu, M., He, X., Peng, B., et al. (2021). Risk Prediction of Diabetes: Big Data Mining with Fusion of Multifarious Physical Examination Indicators. Inf. Fusion 75, 140–149. doi:10.1016/j.inffus.2021.02.015.

[21] Setiawan, D., Karuniawati, E. A. D., & Janty, S. I. (2023). Peran Chat Gpt (Generative Pre-Training Transformer) Dalam Implementasi Ditinjau Dari Dataset. Innovative: Journal Of Social Science Research, 3(3), 9527-9539.

[22] Bilal, M., Israr, H., Shahid, M., & Khan, A. (2016). Sentiment classification of Roman-Urdu opinions using Naïve Bayesian, Decision Tree and KNN classification techniques. Journal of King Saud University-Computer and Information Sciences, 28(3), 330-344.

[23] Ligthart, A., Catal, C., & Tekinerdogan, B. (2021). Analyzing the effectiveness of semi-supervised learning approaches for opinion spam classification. Applied Soft Computing, 101, 107023.

[24] Saura, J. R., Palacios-Marqués, D., & Ribeiro-Soriano, D. (2021). Using data mining techniques to explore security issues in smart living environments in Twitter. Computer Communications, 179, 285-295.

[25] Qorib, M., Oladunni, T., Denis, M., Ososanya, E., & Cotae, P. (2023). Covid-19 vaccine hesitancy: Text mining, sentiment analysis and machine learning on COVID-19 vaccination Twitter dataset. Expert Systems with Applications, 212, 118715.

[26] Khandokar, I. A., Islam, A. M., Islam, S., & Shatabda, S. (2023). A gradient boosting classifier for purchase intention prediction of online shoppers. Heliyon, 9(4).

BIBLIOGRAPHY OF AUTHORS

Salsabila Rabbani, a student of STMIK Amik Riau, majoring in Informatics Engineering who is interested in data science. The focus of research is machine learning, deep learning, and data mining.

Dea Safitri, a student at STMIK Amik Riau, majoring in Informatics Engineering with a concentration in artificial intelligence & Data Science. The focus of research is data mining, deep learning, machine learning, and natural language processing

Farida Try Puspa Siregar, a third-year student of Informatics Engineering at STMIK Amik Riau. I am interested in the development of methods and techniques related to data analysis, machine learning, natural language processing, deep learning, and data mining.

Rahmaddeni was born in Padang and is currently a Lecturer in the Informatics Department at STMIK Amik Riau. Riau. He actively teaches in the field of Machine Learning and Data Mining. He holds a bachelor's degree in Informatics Engineering STMIK Amik Riau and S2 Informatics Engineering Universitas Putra Indonesia "YPTK" Padang. His research focus is Machine Learning and Data Mining.

Lusiana, born in Pekanbaru on October 20, 1970. Lecturer at STMIK Amik Riau in the Informatics Engineering study program. Research focuses on data science, machine learning, deep learning, data mining, natural language processing and software engineering. S1 at STMIK YPTK Padang (UPI YPTK Padang) with GPA = 3.58 Predicate "Best Graduate". S2 at STTI - Benarif Indonesia - Jakarta with GPA 3.80; Predicate "Cumlaude". S3 at UPI YPTK Padang with GPA 3.91 Predicate "Cumlaude" and "Best Graduate".