OPEN ACCESS
VOL. 6, NO. 1, PP.011-020, NOVEMBER 2023

ISSN : 2622-8106 (ONLINE)
DOI: 10. 20895/INISTA.V6I1.1073

Journal of Informatics, Information System, Software Engineering and Applications

Ensemble Machine Learning to Detect Sarcasm in English on Twitter Social Media

Mochamad Alfan Rosid*[1], Muhammad Arginanta Kafi Sambada[2], Suhendro Busono[3], Fajar Muharram[4]

Department of Informatics, Universitas Muhammadiyah Sidoarjo
Jl. Raya Gelam No.250, Pagerwaja, Gelam, Kec. Candi, Kabupaten Sidoarjo, Jawa Timur 61271

[1] alfanrosid@umsida.ac.id
[2] argkaf@umsida.ac.id
[3] suhendro@umsida.ac.id
[4] fajar@umsida.ac.id

Received on 23-05-2023, revised on 01-09-2023, accepted on 13-10-2023

Abstract

Detecting sarcasm in English tweets on social media platforms like Twitter is complex due to its subtle and ambiguous nature. This study uses ensemble machine learning techniques, including Logistic Regression, Naive Bayes, Decision Tree, and Support Vector Machine (SVM), to effectively identify sarcasm. A dataset containing sarcastic and non-sarcastic English tweets was collected and pre-processed. Features representing lexical, syntactic, and semantic information were extracted to train and evaluate the ensemble models. The Support Vector Machine method demonstrated the highest performance among the techniques, achieving an accuracy of 80% and an F1-score of 80% for sarcasm detection. This highlights the efficacy of Support Vector Machines in capturing complex patterns and differentiating between sarcastic and non-sarcastic tweets. By leveraging the strengths of multiple machine learning algorithms, the ensemble approach enhances the overall performance of the sarcasm detection system. It provides a more robust and accurate detection of sarcasm, improving understanding of user sentiments and opinions in online conversations. This research contributes to sentiment analysis and natural language processing, offering valuable insights into sarcasm detection in social media. The findings have practical implications for interpreting user-generated content on platforms like Twitter, enabling a better understanding of user sentiments and facilitating more meaningful interactions. This Model can be used to detect the presence of sarcasm in texts with a success rate of 80% in dimension 20.

Keywords: Ensemble machine learning, Sarcasm detection, Sentiment analysis, Twitter

Corresponding Author:
*Mochamad Alfan Rosid
Department of Informatics, Universitas Muhammadiyah Sidoarjo, Jl. Raya Gelam No.250, Pagerwaja, Gelam, Kec. Candi, Kabupaten Sidoarjo, Jawa Timur 61271
Email: alfanrosid@umsida.ac.id

I. INTRODUCTION

TEXT classification is grouping documents based on their content and textual characteristics. This process can be done by dividing the text into specific classes or categories. According to the KBBI (Indonesian Dictionary), sarcasm is using harsh word meanings to hurt others' feelings or boast about oneself. Sarcasm alters the meaning of sentences or texts in a way that contradicts the text's content and can confuse the meaning of the text. Detecting the presence of sarcasm requires attention to words' semantic meaning and selecting an appropriate algorithm for irony-related problems. Therefore, the study of sarcasm is a hot topic for sarcasm researchers[1].

Sentiment analysis is a part of text mining research that aims to classify text documents, including tweets, based on their positive or negative sentiment. Unfortunately, accurately determining the emotions conveyed in a tweet is often challenging, especially if it contains sarcasm. Irony is a specific form of irony that occurs when someone conveys implied information and often means the opposite of what is said. Sarcasm is difficult to analyze automatically, even for humans. Sentiment analysis detects polarity based on the value of each word, while sarcasm detection also considers the intonation or facial expressions when the person is speaking[2]. Unfortunately, information about intonation or facial expressions is still not available. Therefore, detecting irony remains a challenging issue in sentiment analysis, including sentiment analysis of tweets.

In this study, an analysis was conducted on a collection of information related to similar research. Similar research will be examined for its strengths and weaknesses by looking at previous results. Research on sarcasm detection has been increasingly studied in recent years. The research [5] utilized interjection features and unigrams as the main features for sarcasm sentence detection, comparing two classification methods: Naive Bayes and Support Vector Machine with a polynomial kernel. The best accuracy result was achieved by the Naive Bayes method, with an accuracy of over 91%. Subsequent research [6] employed a BERT-based approach. Firstly, the AraBERTv02 model was refined for sarcasm detection. Then, a Sentence-BERT model trained with contrastive learning was used to extract representative tweet embeddings. Finally, inspired by how the human brain understands the surface and implicit meanings of sarcastic tweets. The obtained result was 2.36% lower than the first place, affirming the capability of the combined Model used in sarcasm detection.

Further research on sarcasm detection in feature-based approaches using supervised machine learning models [7] focuses on existing approaches for automatic sarcasm detection, primarily relying on lexical and linguistic cues. This study proposes implementing a robust and efficient system for sarcasm detection to enhance sentiment analysis accuracy. The results demonstrate that Decision Trees (91.84%) and Random Forests (91.90%) outperform other supervised machine learning algorithms in terms of accuracy for selecting appropriate features.

Further research [8] investigates negative sentiment tweets with the presence of hyperbole for sarcasm detection. Six thousand six hundred previously processed negative sentiment tweets with hashtags such as #Chinesevirus, #Kungflu, #COVID19, #Hantavirus, and #Coronavirus were collected for sarcasm detection. The experiments and analysis conducted in this study conclude that hyperbole exists in an unbiased dataset, which also helps improve sarcasm detection. The proposed method using elongated words achieves an accuracy and F-score of 78.74% and 71%, respectively. Based on research conducted by Aisyah et al. [3], 21% of comments on the Instagram accounts of politicians in Indonesia use sarcasm. Research comparing four classification methods for sarcasm detection, with the highest accuracy of 83% obtained using the Random Forest algorithm [4].

In the research discussed above, machine learning and deep learning techniques were utilized, but there hasn't been any study that utilizes ensemble machine learning for sarcasm detection. Hence, there is an opportunity to develop sarcasm detection using ensemble machine learning methods. This study aims to classify English text from the Twitter platform to uncover public opinions embedded in sarcasm. The research begins with data collection, pre-processing, feature extraction, and classification using Logistic Regression, Naive Bayes, Decision Tree, and Support Vector Machine, followed by final Meta Learning prediction. The objective is to identify the ensemble machine learning method that performs the best and can serve as a reference for developing a sarcasm detection model.

II. RESEARCH METHOD

The research method used in this scientific article involves the use of Ensemble Machine Learning, which includes several stages starting from pre-processing, feature extraction, and classification methods such as Logistic Regression (LR), Support Vector Machine (SVM), Decision Tree (DT), Naive Bayes, and the use of a meta learner for the final prediction as shown in Figure 1. The first step in this method is data pre-processing, which involves cleaning and transforming raw text into a suitable format for analysis. Relevant features are then extracted from the pre-processed data using techniques such as TF-IDF. These features serve as numerical representations of the text in the classification stage.

The classification methods used include Logistic Regression, Support Vector Machine, Decision Tree, and Naive Bayes, each trained using the extracted features. These classification models serve as the base models in the ensemble system. A meta-learner combines the results from the base models and obtains the final prediction. The meta-learner employs techniques such as majority voting, weighted voting, or stacking to combine the results from the base models. By utilizing the predictions generated by the meta-learner, the system can provide the final prediction of whether the text contains sarcasm.

This research method also involves evaluation and performance analysis, using appropriate evaluation metrics such as accuracy, precision, recall, or F1-score to assess the performance of the ensemble machine learning system. Additionally, the implemented system is tested on new unseen data to evaluate its effectiveness in real-world situations. This research's ensemble machine learning method includes pre-processing, feature extraction, classification using LR, SVM, DT, and Naive Bayes, and a meta-learner for the final prediction. These methodological steps aim to improve the accuracy and effectiveness of sarcasm detection in the analyzed text data.

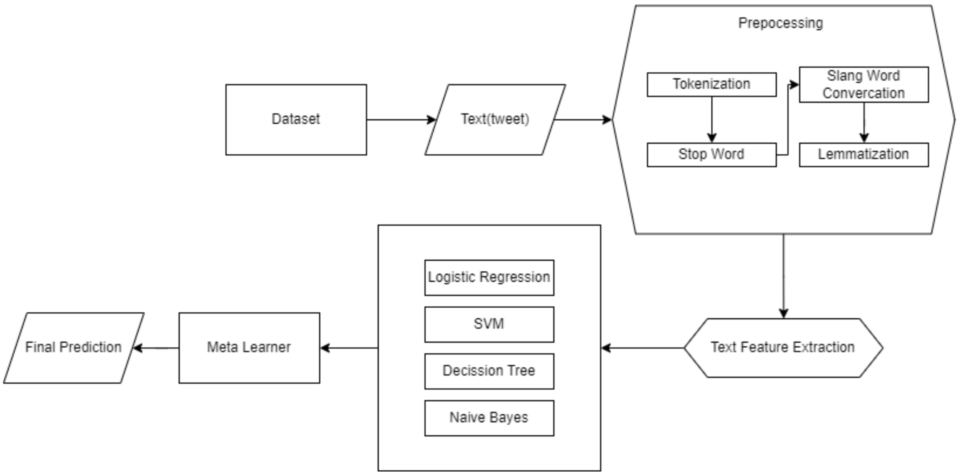

Figure 1. System Architecture for Ensemble Machine Learning

A. Dataset

The dataset was taken from the website Kaggle[1]. The obtained data consists of the main news headlines. The dataset of main news headlines has minimal grammatical errors, unique sentences, and very clean data. The data was processed using the Natural Language Toolkit (NLTK) for tokenization, lemmatization, and pre-processing. The dataset of news headlines consists of 26,709 news headlines. This dataset for detecting sarcasm in headlines has been compiled from two distinct news websites. To create a satirical commentary on current affairs, the Onion has accumulated all the headlines from the News in Brief and News in Photos categories, which are inherently sarcastic. In contrast, the headlines from HuffPost are authentic and devoid of sarcastic undertones. Table 1 is an example of the dataset used.

Table 1. Example of the dataset

No.	Headline	Sarcastic
1.	former versace store clerk sues over secret 'black code' for minority shoppers	0
2.	the 'roseanne' revival catches up to our thorny political mood, for better and worse	0
3.	mom starting to fear son's web series closest thing she will have to grandchild	0
4.	boehner just wants wife to listen, not come up with alternative debt-reduction ideas	1
5.	j.k. rowling wishes snape happy birthday in the most magical way	0
6.	advancing the world's women	0
7.	the fascinating case for eating lab-grown meat	0
8.	this ceo will send your kids to school, if you work for his company	0
9.	top snake handler leaves sinking huckabee campaign	1
10.	friday's morning email: inside trump's presser for the ages	0

B. Text Pre-processing

The text pre-processing process is applied to the data used in sentiment analysis, where we extract the sentiment information from the data, whether negative or positive, based on the author's sentiment. To facilitate data management, we manually analyze the sentiment by reading the meaning of the sentences in the sentiment. This allows us to assess whether the sentiment is negative or positive [5]. In this research, the pre-processing process includes the following steps:

[1] https://www.kaggle.com/datasets/rmisra/news-headlines-dataset-for-sarcasm-detection

1. Tokenizing
   Tokenization is splitting a document into smaller parts, known as tokens. At the same time, tokenization also removes certain specific characters that are considered punctuation marks.
2. Stopword Removal
   Stopword removal is the process of eliminating words that do not contribute significantly to the content of a document. Words that fall into the category of stopwords are removed because they have little impact on text mining processes, such as words like "and", "I", "you", "with", "she", "he", and so on.
3. Slangword
   Slang words are informal modifications of Indonesian words that are not considered standard or found in dictionaries. These words are translated to their closest counterparts using a dictionary created by observing the patterns of occurrence of standard words.
4. Lemmatization
   Lemmatization is the process of transforming the words in a text into their base or root form, which can have various inflected forms. Its goal is to reduce the variation in word forms so that text analysis can be performed more efficiently and accurately.

Next, Feature Extraction is performed, which involves transforming data into new features projected onto a lower-dimensional space [6]. Finally, the machine learning classification step is conducted in sentiment analysis using Logistic Regression, Naive Bayes, Decision Tree, and Support Vector Machine methods.

C. Support Vector Machine (SVM)

Support Vector Machine (SVM) is a machine learning system that acts as a classifier. It uses linear functions as hypotheses in a high-dimensional feature space [7]. SVM is trained using learning algorithms based on optimization theory and incorporates learning bias from statistical theory. Support Vector Machine (SVM) is a method used for prediction in classification and regression tasks. SVM is based on the principle that linear problems can be separated. Still, it can also handle non-linear problems by introducing the concept of kernels into a high-dimensional feature space. In this space, a separating line (hyperplane) is sought that maximizes the distance between the data layers [9].

D. Decision Tree

Decision Tree is a commonly used method in data mining and machine learning for classification and prediction. This method builds a predictive model by dividing the data into smaller subsets, determining the best features that efficiently separate the data. Decision Tree uses supervised machine learning, a learning process where new data is classified based on existing training samples [10]. The resulting Model from the Decision Tree method takes the form of a tree, with nodes representing decisions and branches representing the outcomes of those decisions.

At each stage, the Decision Tree algorithm uses various metrics to select the best feature, which measures how well the feature separates the data. Some common metrics used in Decision Trees include the Gini Index, Information Gain, and Chi-Square. In practice, Decision Tree can be used for various problems, such as classification and regression. In classification, the Decision Tree model predicts the target class by making decisions based on the selected features at each node, while in regression, the Decision Tree model predicts the target value based on the average target value in the data subset at each leaf. In developing Decision Tree models, several techniques such as Pruning, Boosting, and Random Forest can be used to improve model performance and prevent overfitting.

E. Logistic Regression

Logistic Regression is a statistical method used to analyze and model the relationship between a dependent variable and one or more independent variables. The dependent variable in logistic regression is binary, meaning it can only have two values (e.g., 0 or 1, true or false, yes or no). The Logistic Regression method estimates the probability of the dependent variable having a certain value (e.g., 1) given a set of independent variables. This is done using the logistic function (the sigmoid function), which maps real-valued input to a value between 0 and 1. Logistic Regression analysis is based on a function called the Logistic Regression function, which is written in Equation 1:

$$x = \frac{exp(Q_0 + Q_1x_{1i} + Q_1x_{1i} + \cdots + Q_nx_{ni})}{1exp\left(Q_0 + Q_1x_{1i} + Q_1x_{1i} + \cdots + Q_px_{pi}\right)} \tag{1}$$

The value (x) represents the probability of a successful event with the independent variable x being p. In this equation, the value (x) is a non-linear function, so a logit transformation is needed to obtain a linear function, allowing us to observe the relationship between the independent and dependent variables [8]. Logistic Regression is a statistical method that models the relationship between a binary dependent variable and one or more independent variables. It estimates the probability of the dependent variable having a certain value using the logistic function, and the coefficients of the independent variables are estimated using maximum likelihood estimation. In addition to performing binary classification, logistic regression can also perform multinomial regression, multiclass logistic regression, or maximum entropy classification [9].

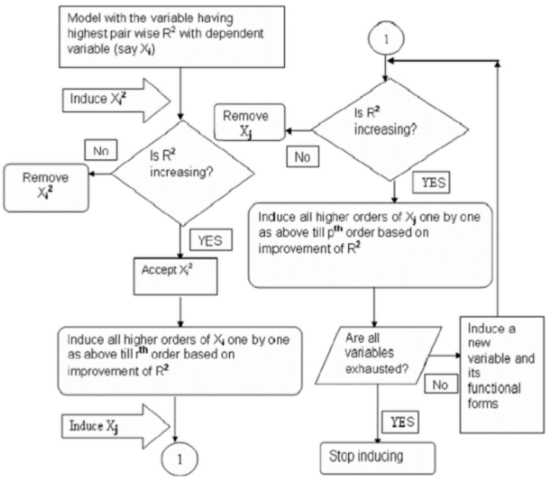

Figure 2. Diagram instructions for Logistic Regression [10]

The learning process of Logistic Regression requires training data to build a model that can be used for classification. In previous studies, Logistic Regression has been compared with Naive Bayes and SVM algorithms in binary classification. Some studies have also shown that Logistic Regression yields good results in text classification.

F. Naïve Bayes

Naive Bayes is a classification method based on Bayes' theorem. This method is popular and commonly used in Natural Language Processing (NLP) and text recognition. In NLP, this method can be used for text classification tasks such as spam or non-spam email, document, or sentiment classification [11]. The basic formula of Bayes' theorem is as follows in Equation 2:

$$P(Y|X) = \frac{P(Y|X)P(Y)}{P(X)} \tag{2}$$

Eq. 2 represents Y as a specific class X as data of an unknown class. At the same time, $P(Y|X)$ is the probability of the hypothesis given the condition, and $P(Y)$ and $P(X|Y)$ are the prior probabilities of the class based on the hypothesis condition, while $P(X)$ is the probability of X [12].

G. Ensemble Learning

Ensemble learning is a data learning technique that combines multiple algorithms or models to obtain more accurate results [13]. Several ensemble learning methods that can be used include voting, bagging, boosting, and stacking. In this study, the authors used Voting. The voting classification method is available in the scikit-learn framework. This method produces an output combined using the average voting probability rule. Then, the class with the highest average score is selected as the final result [14].

H. Confusion Matrix

Confusion Matrix is a table commonly used to describe the performance of a classification model on a testing dataset with known actual values. The multiclass confusion matrix applied in this study is shown in Figure 3:

		Predicted Class	
		Positive	Negative
Actual Class	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Figure 3. Confusion Matrix

In Figure 3, there are predicted labels influenced by the dataset. TP stands for True Positive, the situation where the predicted and actual values are both correct. In a multiclass matrix, the issue is that only TP is specified, as FN (False Negative) is determined from all rows of each label, and FP (False Positive) is determined from the sum of each label. Label and TN (True Negative) refer to situations where there is no predicted value and the actual value is incorrect. The performance detecting abusive sentences is measured using several parameters such as precision, recall, and F1 Score. The equations for these parameters are given in Equations (1-3) [15]. Precision is the ratio of correctly classified data to all correctly classified data.

$$Precision = \frac{TP}{(TP + FP)} \tag{3}$$

Recall is the ratio of correctly classified data to the total number of data in that class. The formulas for each calculation are as follows.

$$Recall = \frac{TP}{(TP + FN)} \tag{4}$$

The F1 Score is the weighted average between the precision and recall values.

$$F1score = 2 \times \frac{Precision \times \ Recall}{Precision + Recall} \tag{5}$$

III. RESULTS AND DISCUSSION

The dataset used in this study consists of 26,709 tweet data collected from the Twitter platform. Four trained annotators have annotated these tweets to detect the presence of sarcasm. Each tweet is labeled as "sarcastic" or "non-sarcastic" based on the consensus of three annotators. An example of the data can be seen in Table 1. The dataset consists of 14,985 sarcastic headlines and 11,724 non-sarcastic headlines. The sentiment label used is binary, with one representing positive data and 0 representing negative data. The distribution of the dataset can be seen in Table 2 and Figure 1.

Table 2. Data Distribution

Class	Number of Data.
Sarcasm	14985
Non-Sarcasm	11724
Total	26709

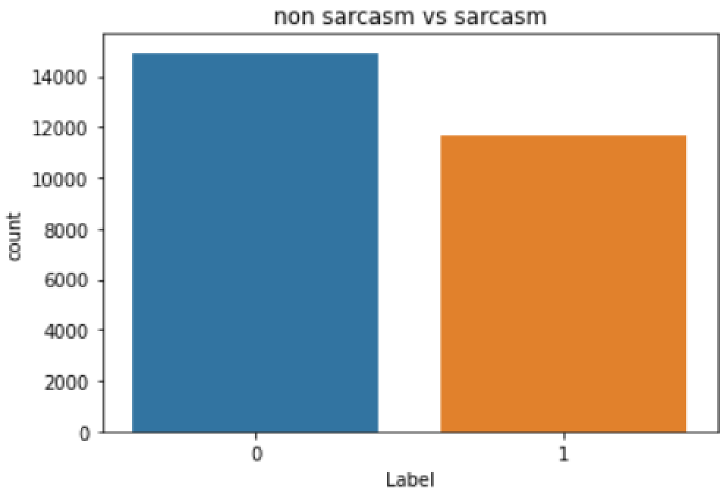

Figure 4. Dataset Distribution Diagram

During the pre-processing of the dataset, punctuation removal, tokenizing, stopword removal, and lemmatization are performed. In the stopword and punctuation removal stage, all words in the data will be converted to lowercase. Meaningless words such as "and", "then", "also", "a", "an", and others will be removed, along with the punctuation marks. The stopword and punctuation removal process results can be seen in Table 3.

Table 3. Stopword Removal and Punctuation Removal Process

Before	After
Former versace store clerk sues over secret 'black code' for minority shoppers	Former Versace store employee files lawsuit over secret black code for minority shoppers.
the 'roseanne' revival catches up to our thorny political mood, for better and worse	the roseanne revival catches up to our thorny political mood for better and worse
mom starting to fear son's web series closest thing she will have to grandchild	mom starting to fear son s web series closest thing she will have to grandchild
boehner just wants wife to listen, not come up with alternative debt-reduction ideas	boehner just wants wife to listen not come up with alternative debt reduction ideas
j.k. rowling wishes snape happy birthday in the most magical way	j k rowling wishes snape happy birthday in the most magical way

After removing meaningless words and punctuation marks, the dataset is then divided into tokens, and the results are shown in Table 4.

Table 4. Tokenizing Process

Before	After
courtroom sketch artist has clear manga influences	'courtroom', 'sketch', 'artist', 'has', 'clear', 'manga', 'influences'
top snake handler leaves sinking huckabee campaign	'top', 'snake', 'handler', 'leaves', 'sinking', 'huckabee', 'campaign'
friday s morning email inside trump s presser for the ages	'friday', 's', 'morning', 'email', 'inside', 'trump', 's', 'presser', 'for', 'the', 'ages'

airline passengers tackle man who rushes cockpit in bomb threat	'airline', 'passengers', 'tackle', 'man', 'who', 'rushes', 'cockpit', 'in', 'bomb', 'threat'
facebook reportedly working on healthcare features and apps	'facebook', 'reportedly', 'working', 'on', 'healthcare', 'features', 'and', 'apps'

This research aims to model sarcasm detection using Logistic Regression, Support Vector Machine, Decision Tree, and Naive Bayes methods with an ensemble approach. Models based on Logistic Regression, Support Vector Machine, Decision Tree, and Naive Bayes are created using the same methodology to ensure fair comparison and classification in the ensemble approach. The testing process is performed to obtain the best accuracy, using accuracy, precision, recall, and F1-score as evaluation metrics. The dataset is divided into training and testing data, with 21,262 training data and 5,316 testing data. Experiments are conducted by varying the dimensions to 20, 50, and 80. The results of the experiments for dimension 20 can be seen in Table 4. The results of the experiments for dimension 50 can be seen in Table 6. The results of the experiments for dimension 80 can be seen in Table 7.

Table 5. Experiment Results with 20-dimension Model

No	Method	Accuracy	Precision	Recall	F1
1	Logistic Regression	0,78	0,78	0,78	0,78
2	Naïve Bayes	0,76	0,83	0,76	0,77
3	Support Vector Machine	0,80	0,80	0,80	0,80
4	Decision Tree	0,63	0,89	0,63	0,70
5	Ensemble	0,75	0,80	0,75	0,76

Table 5 indicates each classification method's performance in terms of accuracy, precision, recall, and F1-score. The Support Vector Machine method apparently achieved the highest accuracy, precision, recall, and F1-score among the individual methods, while the ensemble approach achieved a slightly lower performance. The Decision Tree method had a lower accuracy and recall than the other methods.

Table 6. Experiment Results with Model 50

No	Method	Accuracy	Precision	Recall	F1
1	Logistic Regression	0,77	0,77	0,77	0,77
2	Naïve Bayes	0,74	0,84	0,74	0,76
3	Support Vector Machine	0,78	0,79	0,78	0,79
4	Decision Tree	0,63	0,89	0,63	0,70
5	Ensemble	0,74	0,80	0,74	0,75

The results in Table 6 show that the Support Vector Machine method performs better with higher accuracy, precision, recall, and F1 score than other methods. Logistic Regression, Naïve Bayes, and Ensemble also demonstrate good results with comparable performance. However, despite having high precision, the Decision Tree method has lower accuracy and recall than the other methods. Therefore, the Support Vector Machine method may be a better choice for classifying the data in this context.

Table 7. Experiment Results with Model 80

No	Method	Accuracy	Precision	Recall	F1
1	Logistic Regression	0,74	0,74	0,74	0,74
2	Naïve Bayes	0,72	0,82	0,72	0,74
3	Support Vector Machine	0,75	0,75	0,75	0,75
4	Decision Tree	0,63	0,88	0,63	0,70
5	Ensemble	0,71	0,77	0,71	0,73

The experimental results in Table 7 show the performance of different methods in terms of accuracy, precision, recall, and F1 score. The Logistic Regression method achieved an accuracy of 0.74, with precision, recall, and F1 score also at 0.74. Naïve Bayes performed slightly lower with an accuracy of 0.72 but had a higher precision of 0.82. The Support Vector Machine method showed balanced performance across all metrics, with accuracy, precision, recall, and an F1 score of 0.75. The Decision Tree method had the lowest accuracy of 0.63 but had a high precision of 0.88. The Ensemble method achieved an accuracy of 0.71, with a precision, recall, and F1 score of 0.77, 0.71, and 0.73, respectively. Overall, the results suggest that the Support Vector Machine method performs well in terms of overall accuracy and balanced precision and recall.

Among the various experiments conducted, the best results were obtained in dimension 20 using the Support Vector Machine method, with an accuracy of 80% and an F1 score of 80%. This was followed by dimension 50, which achieved an accuracy of 78% and an F1 score of 79%. In dimension 80, the method achieved an accuracy of 75% and an F1 score of 75%. From these results, the number of dimensions affects the accuracy rate. The more the number of dimensions in this study decreases, the accuracy rate of the SVM method. These results can be observed in Table 8.

Table 8. Comparison of Dimensions 20, 50, and 80 in Support Vector Machine Model

Method	Dimension	Accuracy	Precision	Recall	F1
Support Vector Machine	20	0,80	0,80	0,80	0,80
	50	0,78	0,79	0,78	0,79
	80	0,75	0,75	0,75	0,75

IV. CONCLUSION

Based on the results of this research, it can be concluded that the sarcasm detection model developed using the Support Vector Machine (SVM) algorithm has achieved good accuracy. This Model can be used to detect the presence of sarcasm in texts with a success rate of 80% in dimension 20. Although the developed sarcasm detection model is already quite good, there are still challenges in developing a more accurate model that can detect various types of sarcasm. Therefore, it is recommended to further this research by trying other advanced sarcasm detection algorithms, conducting experiments with larger datasets, and considering contextual factors in sarcasm detection. Additionally, natural language processing technology should be considered to improve the accuracy of the sarcasm detection model.

ACKNOWLEDGMENT

The authors would like to thank Universitas Muhammadiyah Sidoarjo for providing research facilities and funding, enabling this research to be conducted successfully.

REFERENCES

[1] Y. V. Aritonang, D. P. Napitupulu, M. H. Sinaga, and J. Amalia, "Pengaruh Hyperparameter pada Fasttext terhadap Performa Model Deteksi Sarkasme Berbasis Bi-LSTM," *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 9, no. 3, pp. 2612–2625, 2022, doi: 10.35957/jatisi.v9i3.1331.

[2] Y. Yunitasari, A. Musdholifah, and A. K. Sari, "Sarcasm Detection For Sentiment Analysis in Indonesian Tweets," *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 13, no. 1, p. 53, 2019, doi: 10.22146/ijccs.41136.

[3] A. Muhaddisi, B. N. Prastowo, D. Utami, and K. Putri, "Sentiment Analysis With Sarcasm Detection On Politician ' s Instagram," vol. 15, no. 4, pp. 349–358, 2021.

[4] V. Govindan and V. Balakrishnan, "A machine learning approach in analyzing the effect of hyperboles using negative sentiment tweets for sarcasm detection," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 8, pp. 5110–5120, 2022, doi: 10.1016/j.jksuci.2022.01.008.

[5] F. Ugm and F. Ugm, "Analisis Sentimen Twitter untuk Teks Berbahasa Indonesia dengan Maximum Entropy dan Support Vector Machine," vol. 8, no. 1, pp. 91–100, 2014.

[6] A. F. Hidayatullah *et al.*, "Analisis sentimen dan klasifikasi kategori terhadap tokoh publik pada twitter," vol. 2014, no. semnasIF, pp. 115–122, 2014.

[7] P. Arsi and R. Waluyo, "Analisis Sentimen Wacana Pemindahan Ibu Kota Indonesia Menggunakan

Algoritma Support Vector Machine (SVM)," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 8, no. 1, p. 147, 2021, doi: 10.25126/jtiik.0813944.

[8] A. Syahadati, N. C. Lengkong, O. Safitri, S. Machsus, Y. R. Putra, and R. Nooraeni, "ANALISIS SENTIMEN PENERAPAN PSBB DI DKI JAKARTA DAN DAMPAKNYA TERHADAP PERGERAKAN IHSG," vol. 15, no. 1, pp. 20–25, 2021.

[9] M. Shandy, T. Putra, and Y. Azhar, "Perbandingan Model Logistic Regression dan Artificial Neural Network pada Prediksi Pembatalan Hotel," vol. 6, no. 1, pp. 29–37, 2021.

[10] R. Rahmanda and D. S. Informasi, "Rancang bangun aplikasi berbasis microservice untuk klasifikasi sentimen. studi kasus: pt. yesboss group indonesia (kata.ai)," 2018.

[11] A. Setiawan, L. W. Santoso, R. Adipranata, U. K. Petra, and J. Siwalankerto, "Klasifikasi Artikel Berita Bahasa Indonesia Dengan Naive Bayes Classifier," pp. 3–8.

[12] U. Verawardina, F. Edi, and R. Watrianthos, "Analisis Sentimen Pembelajaran Daring Pada Twitter di Masa Pandemi COVID-19 Menggunakan Metode Naïve Bayes," vol. 5, pp. 157–163, 2021, doi: 10.30865/mib.v5i1.2604.

[13] A. Subekti, "Analisis Sentiment pada Ulasan Film Dengan Optimasi Ensemble Learning," vol. 7, no. 1, pp. 5–8, 2020.

[14] M. Ma, A. Prayogo, P. Subarkah, and F. Nida, "Sentiment analysis of customer satisfaction levels on smartphone products using Ensemble Learning," vol. 14, no. 3, pp. 339–347, 2022.

[15] J. Nasional, S. Informasi, M. Kamil, T. Endra, and E. Tju, "Naïve Bayes dan Confusion Matrix untuk Efisiensi Analisa Intrusion Detection System Alert," vol. 02, pp. 81–88, 2022.