International Journal of Information Technology Science (INTENS)
Vol. 01, No. 03, September 2026, pp. 29-38

Sentiment Analysis of Jakarta's Air Pollution Management Using the Naïve Bayes and SVM Algorithms Based on PSO

Dittha Winyana Putra[1], Sutedi[2]
[1]Faculty of Computer Science, Master of Informatics Engineering, Darmajaya Institute of Informatics and Business, Bandar Lampung, Indonesia
[2] Faculty of Computer Science, Master of Informatics Engineering, Darmajaya Institute of Informatics and Business, Bandar Lampung, Indonesia
[1]dittha.2121210001@mail.darmajaya.ac.id, [2]sutedi@darmajaya.ac.id

Article Info

Article history:

Received September 02, 2026
Revised September 03, 2026
Accepted September 07, 2026

Keywords:

Sentiment Analysis
Jakarta's Air Pollution
Naïve Bayes
SVM
PSO Optimization

ABSTRACT

Air pollution is an increasingly pressing environmental issue in Jakarta, the capital of Indonesia, as well as in other major cities worldwide. Machine learning employing Naïve Bayes Classifier and Support Vector Machine (SVM) algorithms, optimized by Particle Swarm Optimization (PSO), can be applied to analyze public sentiment regarding the handling of air pollution in Jakarta. For this study, researchers collected a public dataset from Twitter comprising 797 tweets. This data encompasses various opinions and assumptions related to Jakarta's air pollution mitigation, which was subsequently pre-processed to ensure data validity. The Naïve Bayes Classifier and SVM algorithms—both established data classification methods—were implemented to construct the sentiment model. These algorithms were then optimized using PSO to enhance their classification accuracy. Based on the training and implementation results of the PSO-optimized models, public sentiment toward the handling of air pollution in Jakarta can be evaluated through the recall values: true negative at 92.12%, true positive at 32.35%, and true neutral at 20.42%. This is further reinforced by the precision values, which indicate predicted negative at 65.09%, predicted positive at 63.22%, and predicted neutral at 56.86%. Based on the accuracy evaluation of the PSO-optimized SVM algorithm (Table 7), this study concludes that current measures to handle air pollution in Jakarta are perceived as ineffective and have minimal impact on mitigating the city's overall air pollution.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Dittha Winyana Putra

1. INTRODUCTION

Air pollution is an increasingly pressing environmental issue in Jakarta, the capital city of Indonesia, and other major cities around the world. High levels of air pollution can have serious impacts on human health, the environment and the quality of life of urban residents. Jakarta, as one of the cities with significant levels of air pollution, has experienced an alarming increase in air pollution caused by factors such as motor vehicles, industry and the burning of waste.

Data on air pollution management in Jakarta, from April to June 2025, became a much-discussed topic in the real world, including on various social media platforms. This data on air pollution sparked a range of reactions, both for and against, amongst the public. People freely expressed their opinions, whether in support of or in opposition to the government's decisions regarding the management of air pollution in Jakarta. This

has prompted a sentiment analysis of public comments on the issue of air pollution management in Jakarta, in order to understand the trends in public responses how the public perceives the issue and how they react to it. One way in which the public expresses their views, feelings and opinions is via social media platforms, particularly Twitter. Twitter is frequently used by the public to discuss environmental issues, including air pollution. Therefore, sentiment analysis of comments (tweets) can be a highly effective tool for understanding how the people of Jakarta are responding to efforts to tackle air pollution.

Through the social media platform Twitter, users can carry out a variety of activities. They can communicate with other individuals or groups, share details of their daily lives, promote businesses, engage in debates, and express their views on specific topics. Posts, in the form of status updates known as tweets, are limited to 280 characters. The tweets created by these users can serve as a source for analyzing public sentiment towards the issues being discussed. This is because these messages contain sentiment that can be used as a basis for evaluation and consideration in decision-making.

Responses posted on Twitter can be regarded as sentiment, which can be grouped into three categories: positive sentiment for responses that support an event, negative sentiment for responses that oppose or reject an event, and neutral sentiment for responses that neither support nor reject it. In this study, analyzing the reactions of the Indonesian public is a particularly interesting area of investigation. Sentiment analysis of Twitter data can be carried out using classification algorithms. In this context, machine learning algorithms, such as the Naïve Bayes Classifier and Support Vector Machine (SVM), prove to be extremely useful tools. They are capable of classifying text from tweets into different sentiment categories, such as positive, negative or neutral. By applying these algorithms to tweets containing public views on the management of air pollution in Jakarta, we can identify the prevailing sentiment in the tweets, as well as feedback that may be valuable to the government and environmental organizations for further improvements in efforts to tackle air pollution.

Classification algorithms commonly used for sentiment analysis of reviews include Naïve Bayes (NB), Support Vector Machine (SVM) and K-Nearest Neighbor (k-NN) [1]. As for some related studies conducted on sentiment analysis, one such study was carried out by [2] in which the use of the Support Vector Machine and the Naïve Bayes Classifier (NBC) resulted in the Naïve Bayes Classifier method achieving a classification accuracy of 78.3%. Meanwhile, the accuracy obtained from the classification results using the Support Vector Machine method was 81.6%.

Previous research has shown that the Naïve Bayes Classifier algorithm also achieves a high accuracy rate, as demonstrated in study [3] on the relocation of the capital; this study also involved various stages, including case folding, tokenisation, stop-word removal and stemming. The results obtained from a dataset of 1,007 data points were an accuracy of 94.33%, a precision of 0.87 and a recall of 0.99. The harmonic mean of recall and precision also known as the F1-score is 0.92, indicating that this system performs well in detecting sentiment.

Another study from [2] which utilized the SVM algorithm, was also found to demonstrate good accuracy in a sentiment analysis study regarding the relocation of the capital; this study also involved various stages, including case folding, tokenisation, stop-word removal and stemming. Based on the results of the testing carried out on 1,236 tweets (404 positive and 832 negative) expressing sentiment regarding the relocation of the capital city from the social media platform Twitter, using SVM, the initial hypothesis was confirmed by the final results, namely that SVM outperformed the previous methods (BM25 + KNN and Naive Bayes). With accuracy = 96.68%, precision = 95.82%, recall = 94.04% and AUC = 0.979.

Based on the issues outlined above, research will be conducted into sentiment analysis of public opinions and comments regarding the management of air pollution in Jakarta, using the Naïve Bayes Classifier and Support Vector Machine (SVM) algorithms, based on Particle Swarm Optimization (PSO), in the sentiment analysis process.

2. METHOD

This study used several sections in the following research flowchart:

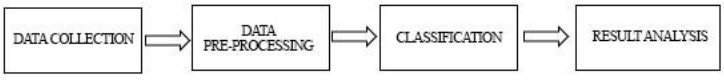


Figure 1. Research Methodology Stages

As explained above, this sentiment analysis research will be carried out in several steps, comprising data collection, data pre-processing, stop-word removal, stemming, TF/IDF weighting, followed by classification using the Naïve Bayes Classifier and Support Vector Machine methods, followed by the evaluation of the results by determining the accuracy, precision, recall and F-score from the confusion matrices

obtained from the classification results; and finally, model testing will be carried out using the cross-validation method to determine which method achieves higher accuracy.

The research method employed involved classification, which was carried out using two types of classification processes: the first was sentiment classification using the Naïve Bayes Classifier method, and the second was sentiment classification using the Support Vector Machine method.

2.1 Data Collection

The first stage is the process of collecting data from Twitter, also known as data crawling. Twitter data is extracted using the 'tweet-harvest' operator via Google Collaboratory. The tweet data is drawn from public opinions regarding the management of air pollution in Jakarta, as expressed on Twitter accounts. The parameters contained in the 'tweet-harvest' operator include filename csv, search_keyword, min_replies, min_faves, min_retweets, until, since, limit, token and language. The data collection process in this study was linked to Twitter accounts using the search_keyword "Jakarta air pollution management". The search criteria used were: `min_replies: 200`, `min_faves: 200`, `min_retweets: 200`, `lang: id`, `until: 2023-12-31`, and `since: 2024-03-01`. A total of 1,000 data points were crawled, of which 797 were Indonesian-language tweets.

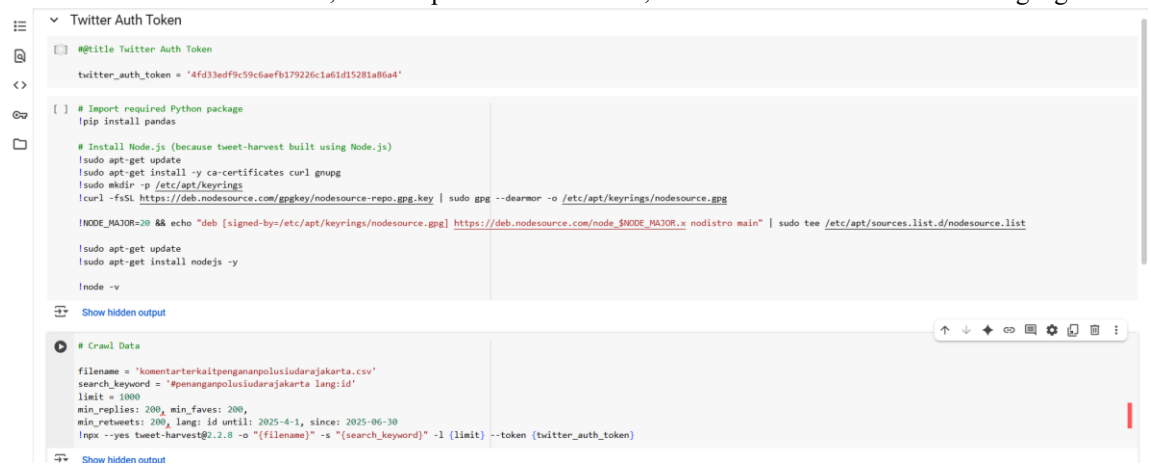


Figure 2. *Crowling* Data *Tweet* Process

2.2 Preprocessing Data

Preprocessing is one of the most important stages in text mining [1] used to transform raw data into data that is ready for use and more structured through data cleaning and standardization, as the data collected is usually still in its raw form. The pre-processing stages applied in this study are case folding, tokenisation, stop word removal, and token filtering (by length).

1. Case Folding Process. This process removes unwanted characters from the document, as they can cause noise such as emoticons and punctuation marks. It also converts characters to lower case.
2. Tokenizing Process. In this process, each word in the text is separated; this is known as tokenisation.
3. Stopword Removal at this stage, unimportant affixes such as the words "di", "dan", "karena", "oleh" and others are removed. This stage is carried out in order to achieve a higher accuracy score.
4. Filter Token (By Leght) The process of extracting key words from the tokenised output. Removing words of a certain length.

2.3 Data Classification

Once all stages of data pre-processing have been completed, the next step is to carry out sentiment classification on the prepared text data. This classification process aims to group each piece of text data into specific sentiment categories, such as positive, negative and neutral, based on the patterns and characteristics of the words it contains. In this study, two different types of classification algorithms were used to compare their performance: the Naïve Bayes Classifier (NBC) and the Support Vector Machine (SVM).

The Naïve Bayes algorithm is a simple yet effective supervised learning method, which utilizes Bayes' theorem to calculate the probability that a piece of data belongs to a particular class based on the probability distribution of its features. Naïve Bayes is known for its low computational requirements, its efficiency in

handling large datasets, and its frequent use in sentiment analysis research due to its consistent performance in classifying text [4].

Meanwhile, a Support Vector Machine (SVM) is a classification algorithm that works by finding the best hyperplane to optimally separate data across classes in a high-dimensional vector space. SVMs are known for delivering better classification results, particularly with complex text data that has a high number of features. Furthermore, SVM is frequently chosen in sentiment analysis research due to its ability to address the issue of overfitting and to achieve high accuracy across various types of text datasets [5].

These two algorithms were then compared to determine which one produced the best classification results for sentiment data relating to the issue of air pollution in Jakarta. By applying the Particle Swarm Optimization (PSO) method as a feature selection technique, it is hoped that the performance of both algorithms can be improved in terms of accuracy and the efficiency of the classification process [6].

2.4 Application of Algorithms

In this stage, a Naïve Bayes Classifier and a Support Vector Machine both optimized using Particle Swarm Optimization (PSO) were used to achieve the values for Accuracy, Recall and Precision**.**

3. RESULTS AND DISCUSSION

The dataset used in this study was obtained by crawling Twitter using RapidMiner 10.1 software. Data collection focused on tweets relating to the management of air pollution in Jakarta. Twitter data collection utilized the 'tweet-harvest' operator via Google Collaboratory; the data could not yet be classified into positive, negative, and neutral classes. Therefore, a manual classification process involving labelling was required to determine the class positive, negative, or neutral. The total number of tweets successfully collected from Twitter amounted to approximately 797. Subsequently, a pre-processing stage was carried out on this data, resulting in a reduction in the dataset size to 731 data points, which were then divided into two parts: training data and testing data, comprising 30 per cent and 70 per cent respectively. The following details the results of the dataset processing stage.

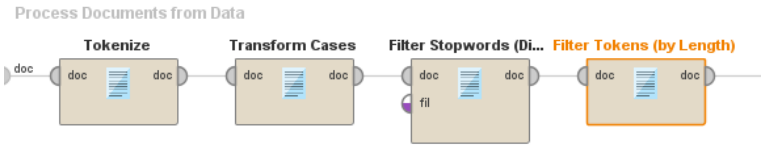


Figure 3. Data Pre-processing Model Design

If the classification results show that the positive class has a higher value than the negative class, the procedure is to assign the clause to the positive, negative or neutral class based on the values from the Support Vector Machine and the Naive Bayes Classifier computations, supported by Particle Swarm Optimization (PSO) to improve accuracy. Subsequently, if the probability for the positive class is lower than that for the negative class, the proposition will be placed in the negative class. However, if the probabilities for both the positive and negative classes are lower than that for the neutral class, the proposition will be placed in the negative class; otherwise, the proposition will be placed in the neutral class.

4. RESULTS OF THE METHOD TESTING EXPERIMENT

The following are the results of the experiments using the methods employed in this study.

4.1 Naïve Bayes Algorithm

Test results on a dataset comprising 731 data points, using the Naïve Bayes algorithm, showed an accuracy rate of 59.09% and a Kappa coefficient of 0.261. Further information regarding the test results is available in Tables 1 and 2.

Table 1. Accuracy score from testing the Naïve Bayes algorithm.

accuracy: 59.09% +/- 6.96% (micro average: 59.10%)

	true positif	true negatif	true netral	class precision
pred. positif	60	45	24	46.51%
pred. negatif	73	328	74	69.05%
pred. netral	37	46	44	34.65%
class recall	35.29%	78.28%	30.99%	

Table 2. Kappa score from testing the Naïve Bayes algorithm.

kappa: 0.261 +/- 0.118 (micro average: 0.260)

	true positif	true negatif	true netral	class precision
pred. positif	60	45	24	46.51%
pred. negatif	73	328	74	69.05%
pred. netral	37	46	44	34.65%
class recall	35.29%	78.28%	30.99%	

4.2 The Naïve Bayes algorithm based on Particle Swarm Optimization

Data testing using the Naïve Bayes algorithm enhanced with Particle Swarm Optimization yielded an accuracy of 60.60% and a Kappa value of 0.307. Further information regarding the test results is available in Tables 3 and 4 of the documents.

Table 3. Accuracy values from testing the Particle Swarm Optimization-based Naïve Bayes algorithm

accuracy: 60.60% +/- 4.27% (micro average: 60.60%)

	true positif	true negatif	true netral	class precision
pred. positif	70	43	25	50.72%
pred. negatif	63	319	63	71.69%
pred. netral	37	57	54	36.49%
class recall	41.18%	76.13%	38.03%	

Table 4. Kappa values from testing the Particle Swarm Optimization-based Naïve Bayes algorithm

kappa: 0.307 +/- 0.064 (micro average: 0.306)

	true positif	true negatif	true netral	class precision
pred. positif	70	43	25	50.72%
pred. negatif	63	319	63	71.69%
pred. netral	37	57	54	36.49%
class recall	41.18%	76.13%	38.03%	

4.3 The Support Vector Machine Algorithm

Data testing using the Support Vector Machine algorithm yielded an accuracy rate of 62.93% and a Kappa value of 0.244. Further information on the test results can be found in Tables 5 and 6 of the relevant documents.

Table 5. Accuracy score from testing the Support Vector Machine algorithm

accuracy: 62.93% +/- 5.28% (micro average: 62.93%)

	true positif	true negatif	true netral	class precision
pred. positif	48	16	17	59.26%
pred. negatif	112	387	100	64.61%
pred. netral	10	16	25	49.02%
class recall	28.24%	92.36%	17.61%	

Table 6. The Kappa coefficient from testing the Support Vector Machine algorithm

kappa: 0.244 +/- 0.106 (micro average: 0.245)

	true positif	true negatif	true netral	class precision
pred. positif	48	16	17	59.26%
pred. negatif	112	387	100	64.61%
pred. netral	10	16	25	49.02%
class recall	28.24%	92.36%	17.61%	

4.4 Support Vector Machine algorithm based on Particle Swarm Optimization

Data testing using the Support Vector Machine algorithm, based on Particle Swarm Optimization, yielded an accuracy rate of 64.29% and a Kappa coefficient of 0.276. Detailed information regarding the test results can be found in Tables 7 and 8 below:

Table 7. Accuracy values from testing the Particle Swarm Optimization-based Support Vector Machine algorithm

accuracy: 64.29% +/- 4.07% (micro average: 64.30%)

	true positif	true negatif	true netral	class precision
pred. positif	55	18	14	63.22%
pred. negatif	108	386	99	65.09%
pred. netral	7	15	29	56.86%
class recall	32.35%	92.12%	20.42%	

Table 8. The Kappa coefficient from testing the Particle Swarm Optimization-based Support Vector Machine algorithm

kappa: 0.276 +/- 0.087 (micro average: 0.277)

	true positif	true negatif	true netral	class precision
pred. positif	55	18	14	63.22%
pred. negatif	108	386	99	65.09%
pred. netral	7	15	29	56.86%
class recall	32.35%	92.12%	20.42%	

5. THE NAÏVE BAYES ALGORITHM TESTING PROCESS

Information management using the Naïve Bayes model on a dataset comprising 731 data points involves a series of structured steps. The initial stage begins by using the 'Read Excel' operator to load the dataset into the Naïve Bayes model. Next, the 'Set Role' operator is used to define the 'label' attribute as the target role; this step involves configuring the 'Change Role' operator to select 'Label'. The 'Nominal to Text' operator is used to convert nominal data types into text, incorporating all attribute values into corresponding string values. Within the 'Process Documents from Data' operator, several processes are carried out to clean the data into vectors that can be used in algorithmic computations, including 'Tokenise' to split the text into tokens, 'Transform Case' to convert the text to lowercase or uppercase, 'Tokens Filter' to filter out irrelevant tokens, and 'Stopword Filter' to remove common words that do not add any additional meaning [7].

Once the document has been converted into a computable vector, the next step is the validation process using the K-fold cross-validation method with a K-value of 10. The validation process utilises the Naïve Bayes algorithm via the 'Apply Model' function in order to apply the Naïve Bayes model to make predictions on the confusion matrix. The aim is to obtain predictions for unseen data or to transform the data by applying a pre-processing model. The confusion matrix is a performance metric for machine learning classification problems where the output consists of two or more classes. It is a table containing six different combinations of predicted and actual values. The entire process using the Naïve Bayes algorithm is illustrated in Figure 4 below:

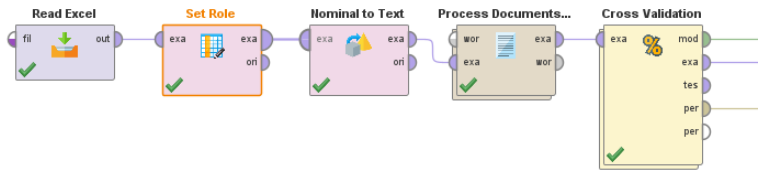


Figure 4. The testing process using the Naïve Bayes algorithm

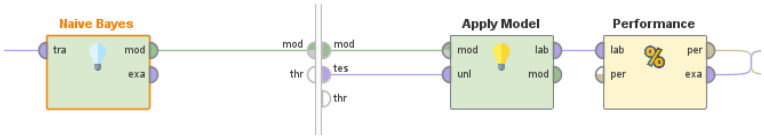


Figure 5. Continuation of the testing process using the Naïve Bayes algorithm

The initial stage in information management using the Naïve Bayes model begins by using the 'Read Excel' operator to import the dataset into the model. The next step is to use the 'Set Role' operator to define the 'label' attribute as the target role by selecting the 'Label' option in the 'Change Role' settings. The 'Nominal to Text' operator is used to convert nominal data types into text. In the 'Process Documents from Data' operator, the data is cleaned and converted into vectors that can be used for algorithm calculations through processes such as 'Tokenise', 'Transform Case', 'Filter Tokens' and 'Filter Stopwords'.

Following the attribute weighting process using Particle Swarm Optimisation, the next step is validation using K-fold cross-validation with K set to 10. This divides the dataset into 10 subsets, whereby

in each iteration one subset is used as test data and the others as training data, helping to objectively assess the model's performance. 'Apply Model' is used to apply the Naïve Bayes algorithm model for predictions on the confusion matrix, which measures classification performance with two or more classes as output, and calculates the model's accuracy, precision, recall and F1-score.

6. THE TESTING PROCESS FOR A PSO-BASED NAÏVE BAYES ALGORITHM

Particle Swarm Optimisation (PSO) is an optimisation technique that falls within the category of evolutionary computation techniques, developed by Dr Eberhart and Dr Kennedy in 1995 [8]. Particle Swarm Optimisation (PSO) is used to assign weights to the attributes in the dataset used by the author, whereby only the influential attributes are utilised in the Naïve Bayes algorithm, which has been further optimised. The implementation process of Particle Swarm Optimisation (PSO) is explained in detail in Figure 6 as follows:

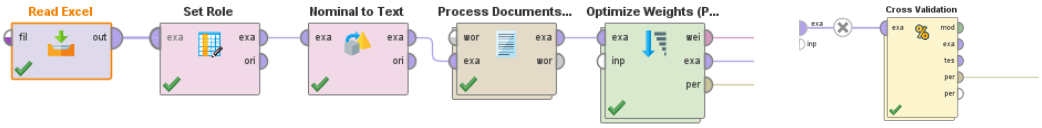


Figure 6. The testing process using a Particle Swarm Optimisation-based Naïve Bayes algorithm

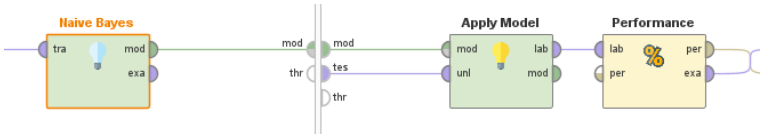


Figure 7. Continuation of the testing process using a Particle Swarm Optimisation-based Naïve

7. THE TESTING PROCESS FOR THE SUPPORT VECTOR MACHINE ALGORITHM

On the processing data using the Support Vector Machine model on a dataset of 731 records, the first step is to use the 'Read Excel' operator to import the dataset into the Support Vector Machine (SVM) model, then use the 'Set Role' operator to define the 'label' attribute as the target role by selecting the 'Label' option in the 'Change Role' settings.

The Nominal to Text operator is used to convert nominal data types into text by incorporating all attribute values into the corresponding string values. The 'Documents from Data' process involves several steps to clean the data so that it can be converted into a vector suitable for use in algorithmic calculations. These steps include 'Tokenise', 'Case Transformation', 'Filter Tokens' and 'Filter Stopwords'. Consequently, the data, which was originally in text form, will be optimally prepared for further analysis, such as classification or sentiment analysis.

Once the document has been converted into a computable vector which transforms text values into numerical values the next step is the validation process using cross-validation with a K-fold setting of K=10. In this application, two validation processes are employed. One of these is the validation process using the Support Vector Machine (SVM) algorithm.

Model application is carried out with the aim of applying the Support Vector Machine (SVM) algorithm to make predictions based on the confusion matrix, both to estimate unseen data and to transform the data through the application of pre-processing techniques [9].

A confusion matrix is a method for measuring performance in the context of machine learning classification problems, where the output may consist of two or more classes. It consists of a table showing six different combinations of predicted and actual values. A false negative (FN) is the number of positive cases in the dataset that are incorrectly predicted as negative [10]. The entire process of applying the Support Vector Machine algorithm is explained in detail in Figure 8 as follows:

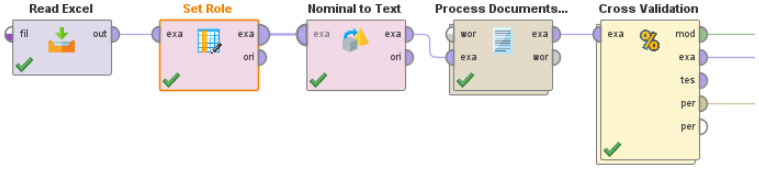


Figure 8. The testing process using the Support Vector Machine algorithm

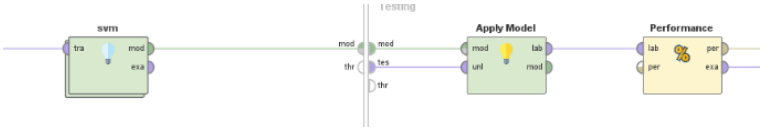


Figure 9. Continuation of the testing process using the Support Vector Machine algorithm

8. TESTING PROCESS FOR A PSO-BASED SUPPORT VECTOR MACHINE ALGORITHM

Particle Swarm Optimisation (PSO) is an optimisation method that falls within the category of evolutionary computation techniques, first developed by Dr Eberhart and Dr Kennedy in 1995 [8]. PSO is frequently used to

address the problem of attribute weighting in a dataset. Particle Swarm Optimisation (PSO) is an algorithm that operates on the principle of a population, where a collection of particles is referred to as a swarm, whilst each individual within it is called a particle [11]. The primary objective of this algorithm is to find a globally optimal solution within the search space through particle interactions by selecting relevant attributes. In the context of this paper, only those attributes deemed influential are used in the Support Vector Machine (SVM) algorithm, which has been further optimised. The process of applying Particle Swarm Optimisation (PSO) is explained in detail in Figure 10 included in this study:

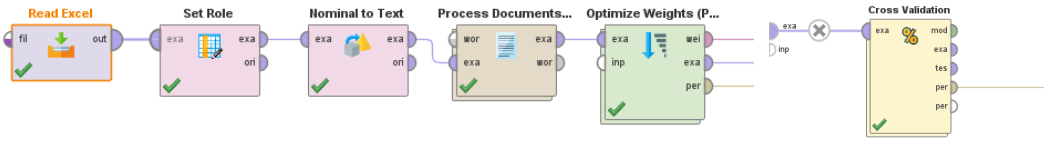


Figure 10. Testing process using a Support Vector Machine algorithm based on Particle Swarm Optimization

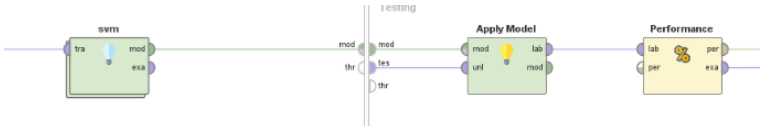


Figure 11. Continuation of the testing process using a Support Vector Machine algorithm based on Particle Swarm Optimisation

9. ANALYSIS, EVALUATION OF RESULTS AND MODEL VALIDATION

In this step, the researchers compared the results of experiments using two methods: Naïve Bayes and Support Vector Machine (SVM). Each method involved a stage utilizing Particle Swarm Optimization (PSO) for feature selection. The researchers present the results of this comparison in Table 9 below:

Table 9. Comparison Results Table

Method	Accuracy Values
Naïve Bayes	59. 09%
Support Vector Machine	62. 93%
Naïve Bayes + PSO	60. 60%
Support Vector Machine + PSO	64. 29%

10. ANALYSIS OF RESULTS

this stage, the author evaluates the results of previous research to determine the effectiveness of two classification methods in addressing the research problem; this is also in line with research conducted on the use of the Naïve Bayes algorithm, optimised using Particle Swarm Optimization (PSO) [12] as well as research [13] utilizing the Support Vector Machine (SVM) algorithm. This study compares the Naïve Bayes and Support Vector Machine (SVM) methods in conducting sentiment analysis on the management of air pollution in Jakarta, utilizing optimization via Particle Swarm Optimization (PSO).

The results show a fairly significant difference when using a classification method that incorporates Particle Swarm Optimization (PSO) for feature selection. Based on the comparison results, experiments on the dataset processed using Naïve Bayes without feature selection yielded an accuracy of 59.09 per cent. Meanwhile, for the Support Vector Machine (SVM) method without feature selection, the accuracy

obtained was 62.93 per cent. When both methods were optimised using Particle Swarm Optimization (PSO), the results showed an improvement.

In the Naïve Bayes algorithm utilising Particle Swarm Optimisation (PSO), accuracy increased to 60.60%, representing an improvement of 1.51. The Support Vector Machine (SVM) algorithm utilising Particle Swarm Optimization (PSO) showed a significant increase in accuracy to 64.29%, representing an improvement of 1.36%. Thus, the Support Vector Machine (SVM) method optimised using Particle Swarm Optimization (PSO) demonstrated better performance compared to the Naïve Bayes algorithm utilising Particle Swarm Optimization. The results of this study show that the Support Vector Machine (SVM) algorithm utilising Particle Swarm Optimization (PSO) achieved an accuracy level of 64.29 per cent; this is consistent with the findings of a study conducted [14] which demonstrated an increase in accuracy when the Support Vector Machine (SVM) algorithm was combined with the application of Particle Swarm Optimization (PSO).

The accuracy performance metric [15] can be classified into five groups, namely:

Table 10. Accuracy score classification

Accuracy Scores	*Classification*
0. 91 – 1. 00	*Excellent*
0. 81 – 0. 90	*Good*
0. 71 – 0. 80	*Fair*
0. 61 – 0. 70	*Poor*
0. 50 – 0. 60	*Failure*

As can be seen from Table 10, the Support Vector Machine (SVM) algorithm falls into the 'Poor Classification' category for sentiment analysis, with an accuracy of 62.93%. Meanwhile, the Naïve Bayes algorithm falls into the 'Poor Classification' category, with an accuracy of 59.09%. Furthermore, the Naïve Bayes algorithm based on Particle Swarm Optimization (PSO) is categorized as 'Poor Classification', with an accuracy of 60.60 per cent. Meanwhile, the Support Vector Machine (SVM) algorithm based on Particle Swarm Optimization (PSO) also falls into the 'Poor Classification' category, with an accuracy of 64.29%. This is also demonstrated by the research conducted [16] which achieved a 'Good Classification' rating in the study using the Support Vector Machine (SVM) algorithm based on Particle Swarm Optimization (PSO).

Based on the classification presented, it can be concluded that the Support Vector Machine (SVM) algorithm based on Particle Swarm Optimization (PSO) is an algorithm that can be used to perform sentiment analysis. With an accuracy of 64.29%, this algorithm demonstrates good performance in the predefined 'Poor Classification' category. These results also indicate that the integration of Particle Swarm Optimization (PSO) can improve the performance of the classification algorithm in sentiment analysis relating to the issue of air pollution in Jakarta.

11. CONCLUSION

Based on the test results described above, the following conclusions can be drawn from this study in relation to the research questions formulated in the opening chapter:

1. The use of PSO can improve the accuracy of the Naïve Bayes and Support Vector Machine algorithms. The Support Vector Machine model, enhanced using Particle Swarm Optimization, is capable of providing better solutions for classifying tweets, thereby improving the accuracy and optimality of the classification results.
2. From the results of the study, it can be concluded that the Support Vector Machine algorithm, supported by Particle Swarm Optimization (PSO), is a better choice for sentiment analysis than the Naïve Bayes algorithm, which is also supported by Particle Swarm Optimization (PSO).
3. From the research findings, it can be concluded that the effectiveness of air pollution management in Jakarta can be assessed based on the recall and true negative rates of 92.12 %, the true positive rate of 32.35 %, and the true neutral rate of 20.42%. This is further supported by the precision values: a predicted negative rate of 65.09%, a predicted positive rate of 63.22%, and a true neutral rate of 56.86%. These figures are based on Table 7, which shows the accuracy of the Support Vector Machine algorithm supported by Particle Swarm Optimization (PSO). It can therefore be concluded that the management of air pollution in Jakarta remains ineffective and has little impact on the air pollution occurring in the city.

REFERENCES

[1] S. A. Alasadi and W. S. Bhaya, "Review of Data Preprocessing Techniques," *Journal of Engineering and Applie Sciences*, vol. 12, no. 16. pp. 4102–4107, 2017.

[2] 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.

[3] P. Arsi, B. A. Kusuma, and A. Nurhakim, "Analisis Sentimen Pindah Ibu Kota Berbasis Naive Bayes Classifier," *J. Inform. Upgris*, vol. 7, no. 1, pp. 1–6, 2021, doi: 10.26877/jiu.v7i1.7636.

[4] C. Pramita, "Analisis Sentimen Media Sosial Menggunakan Algoritma Naive Bayes," *J. Dunia Data*, vol. 1, no. 5, pp. 1–18, 2024, [Online]. Available: http://www.duniabisnis.org/index.php/duniadata/article/view/91

[5] N. Z. B. Jannah and K. Kusnawi, "Comparison of Naïve Bayes and SVM in Sentiment Analysis of Product Reviews on Marketplaces," *Sinkron*, vol. 8, no. 2, pp. 727–733, 2024, doi: 10.33395/sinkron.v8i2.13559.

[6] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimization for feature selection in classification: A multi-objective approach," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1656–1671, 2013, doi: 10.1109/TSMCB.2012.2227469.

[7] M. Nahdhudin, N. Cahyo, H. Wibowo, M. R. Handayani, and K. Umam, "Klasifikasi Sentimen Masyarakat Terhadap Aplikasi Tiktok Menggunakan Algoritma Naive Bayes," vol. 1, no. 3, pp. 107–112, 2025.

[8] J. He and H. Guo, "A modified particle swarm optimization algorithm," *Proc. 29th Chinese Control Decis. Conf. CCDC 2017*, vol. 11, no. 10, pp. 6659–6663, 2017, doi: 10.1109/CCDC.2017.7978374.

[9] A. R. Purnajaya, "Evaluating Resampling Methods for Imbalanced Necrosis Classification on CT Scans," vol. 2, no. 2, 2026, doi: 10.65258/jutekom.v2.i2.56.

[10] M. S. & K. K. B. Surbhi Bhatia, "Sentiment Analysis and Mining of Opinions," *Stud. Big Data*, vol. 30, pp. 503–523, 2018, doi: 10.1007/978-3-319-60435-0_20.

[11] N. Nikentari, H. Kurniawan, N. Ritha, D. Kurniawan, U. Maritim, and R. Ali, "Particle Swarm Optimization Untuk Prediksi Pasang Surut Air Optimization of Backpropagation Artificial Neural Network With Particle Swarm Optimization To Predict Tide Level," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 5, pp. 605–612, 2019, doi: 10.25126/jtiik2018551055.

[12] A. Halim and Andri Safuwan, "Analisis Sentimen Opini Warganet Twitter Terhadap Tes Screening Genose Pendeteksi Virus Covid-19 Menggunakan Metode Naïve Bayes Berbasis Particle Swarm Optimization," *J. Inform. Teknol. dan Sains*, vol. 5, no. 1, pp. 170–178, 2023, doi: 10.51401/jinteks.v5i1.2229.

[13] A. Santoso, A. Nugroho, and A. S. Sunge, "Analisis Sentimen Tentang Mobil Listrik Dengan Metode Support Vector Machine Dan Feature Selection Particle Swarm Optimization," *J. Pract. Comput. Sci.*, vol. 2, no. 1, pp. 24–31, 2022, doi: 10.37366/jpcs.v2i1.1084.

[14] R. Darmawan, I. Indra, and A. Surahmat, "Optimalisasi Support Vector Machine (SVM) Berbasis Particle Swarm Optimization (PSO) Pada Analisis Sentimen Terhadap Official Account Ruang Guru di Twitter," *J. Kaji. Ilm.*, vol. 22, no. 2, pp. 143–152, 2022, doi: 10.31599/jki.v22i2.1130.

[15] Florin Gorunescu, *Data Mining: Concepts, Models and Techniques*, vol. 12. 2011. doi: 10.1007/978-3-642-19721-5_5.

[16] E. Indrayuni, "Analisa Sentimen Review Hotel Menggunakan Algoritma Support Vector Machine Berbasis Particle Swarm Optimization," *J. Evolusi Vol. 4 Nomor 2 - 2016*, vol. 4, no. 2, pp. 20–27, 2016.