Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)
Vol. 7, No. 3, December 2021, pp. 433-440
ISSN: 2338-3070, DOI: 10.26555/jiteki.v7i3.22080

Sentiment Analysis on Work from Home Policy Using Naïve Bayes Method and Particle Swarm Optimization

Rista Azizah Arilya, Yufis Azhar, Didih Rizki Chandranegara
Department of Informatics University Muhammadiyah Malang, Indonesia

ARTICLE INFO

Article history:

Received October 28, 2021
Revised November 14, 2021
Accepted December 10, 2021

Keywords:

Naive Bayes;
Particle Swarm Optimization;
Sentiment Analysis;
Work From Home

ABSTRACT

At the beginning of 2020, the world was shocked by the coronavirus, which spread rapidly in various countries, one of which was Indonesia. So that the government implemented the Work from Home policy to suppress the spread of Covid-19. This has resulted in many people writing their opinions on the Twitter social media platform and reaping many pros and cons of the community from all aspects. The data source used in this study came from tweets with keywords related to work from home. Several previous studies in this field have not implemented feature selection for sentiment analysis, although the method used is not optimal. So that the contribution in this study is to classify public opinion into positive and negative using sentiment analysis and implement PSO for feature selection and Naïve Bayes for classifiers in building sentiment analysis models. The results showed that the best accuracy was 81% in the classification using Naive Bayes and 86% in the classification using naive Bayes based on PSO through a comparison of 90% training data and 10% test data. With the addition of an accuracy of 5%, it can be concluded that the use of the Particle Swarm Optimization algorithm as a feature selection can help the classification process so that the results obtained are more effective than before.

Corresponding Author:

Yufis Azhar, Informatics Department, University of Muhammadiyah Malang, Raya Tlogomas No. 246 Malang, Indonesia.
Email: yufis@umm.ac.id

1. INTRODUCTION

The spread of the Covid-19 pandemic that has taken place in Indonesia has changed the order of people's lives and gave birth to a new order that was never thought of before so that people are required to adapt a new lifestyle that is regulated by health protocols that have been adjusted. One of the living arrangements in question is the application of Work from home. This policy is one of the efforts to strengthen the implementation of physical restrictions to prevent the spread of virus transmission cases. This policy causes activities such as seminars, meetings, teaching and learning processes to be carried out online. With the implementation of work from home, it encourages people to express their opinions through social media platforms, one of which is Twitter. Twitter is known for its practical and simple use [1]. Twitter is a very popular communication tool used today as an important source of information because of the rapid spread of information [2]. This social media utilizes a microblogging service that allows its users to read and send messages to each other. Messages sent by users are known as tweets [3]. The opinions expressed have many pros and cons. Therefore sentiment analysis can be used to solve this problem. Sentiment analysis deals with differences in consumer or expert opinion about a product, service, or agency through different media and is a combination of data mining and text mining [4]. Sentiment analysis aims to identify the ways in which sentiment is expressed in text form to determine whether the expression is positive or negative towards a subject [5][6]. By using sentiment analysis, it can be seen that the output of positive or negative sentiment from a tweet will be visualized in the form of a cloud.

Several previous studies related to sentiment analysis using Naive Bayes have been carried out in various fields, which can be explained as follows. In this study, it discusses sentiment analysis on candidates the 2019 Presidential Election of the Republic of Indonesia, and comparisons were made using the Naive Bayes method, SVM, and the K-Nearest Neighbor method, which were tested using RapidMiner with a Naive Bayes accuracy value of 75.58%, 63.99% SVM accuracy value and K-NN accuracy value of 73.34% [7]. Further research related to online comments from news portal readers about sentiments on the attitudes and actions of state officials in carrying out their duties, the data used consists of 200 data classified into positive and negative classes. The results of this study produce a system that can classify sentiment automatically with an accuracy of 67% for Naive Bayes and 76% for Naive Bayes based on PSO. With a fairly high level of accuracy, this model can provide a solution in classifying public opinion on state officials' news to be more accurate and optimal [8]. Another study compared three Machine Learning methods such as Naïve Bayes, Vector machine, and Maximum entropy by classifying the data into positive, negative, and neutral. The results showed that machine learning methods such as Naïve Bayes have the highest accuracy with an accuracy of 86% [9]. The next study is related to sentiment analysis on the use of E-money by comparing C4.5 Decision Tree Classifier and Naive Bayes. The sample data used is sourced from Twitter posts by mentioning links to e-money service providers in Indonesia, namely Ovo, Dana, Gopay, with a total of 205 data. The results showed that Naive Bayes gave better results than C4.5 Decision Tree in sentiment analysis with 80% accuracy for C4.5 decision tree and 84% for Naive Bayes [10].

Based on the literature review from previous studies, several studies have not implemented feature selection, even though this method can reduce the number of features that are considered as noise while at the same time increasing the accuracy of the model. Several previous studies did not normalize and clean the preprocessing data, even though this process is very important to remove noise in inconsistent data and normalize any non-standard words into standard words that can affect the results of the model evaluation. In addition, there have been many studies that discuss sentiment analysis by taking data from Twitter, but no one has conducted research related to public sentiment on the application of work from home, so it can be concluded that this research is still very minimally done until now.

Based on this background, the contribution in this study is to classify public opinion into positive and negative using sentiment analysis and implement PSO for feature selection and Naïve Bayes for classifiers in building sentiment analysis models, the selection of this method is carried out by considering the accuracy of the classification results in previous studies so that this research is expected to be able to produce an even better performance on sentiment analysis text classification. The basis of this research is to determine the accuracy results of combining the two methods carried out by considering the accuracy of the classification results in previous studies so that this research is expected to be able to produce better performance than previous studies.

2. METHOD

At this stage, a review is carried out as a step in achieving the desired output. This study uses the Naive Bayes method, which will compare the results with the addition of the Particle Swarm Optimization algorithm as a feature selection. Therefore, a flowchart will be made to explain the series of processes that will be carried out. Fig. 1 is a flowchart of the system design that has been made.

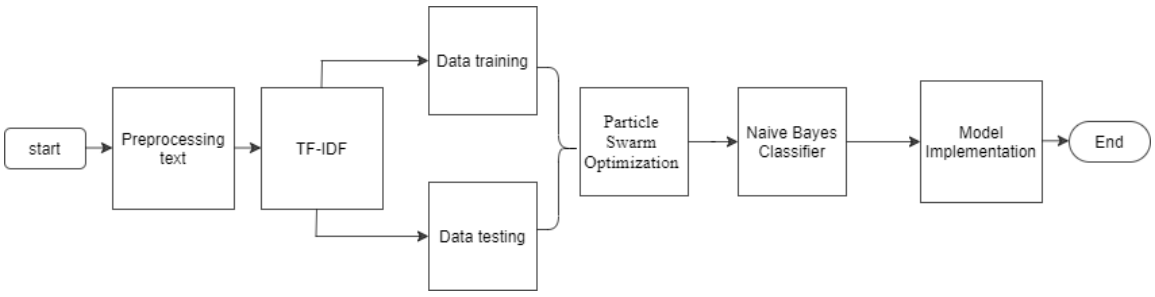

Fig. 1. System design

2.1. Data Collection

The data used in this study is 1.200 data of Indonesian language tweets with a time span from March 2020 to December 2020 and stored in .csv form as input data. The data will be used as training data and test data, where each data consists of two classes, namely the positive class and the negative class. Data collection is carried out by crawling data through a tweet data crawler from the Twitter API using RapidMiner and manually

collected through tweets from Twitter users containing keywords related to the implementation of work from home. The use of Rapid miner tools can make it easier to collect data because this tool is open source which is devoted to the use of data mining. The development of this application supports all stages of the machine learning process, such as data collection and visualization of results.

2.2. Manual Labeling

The process of making datasets in this study requires a mechanism so that the data collected has the correct class label. Data labels are purchased manually. The manual labeling process is an analysis of determining whether tweet sentiment is positive or negative. Because it uses a lot of data, the determination of class labels is carried out by three analysts by labeling the original tweets that have not been obtained. Manual labeling produces accurate data because humans can distinguish a sentiment by reading and understanding each content of the tweet. The data is divided equally (balanced) in each class because unbalanced data will complicate the method on the generalization function so that it can produce poor performance. The distribution of balanced data resulted in 600 positive sentiment data and 600 negative sentiment data. The same sentiment labels are combined into a .csv file extension. Data from the manual labeling process are shown in Table 1.

Table 1. Example Dataset

Text	Label
#Work from Home sometimes makes us more creative and of course increases our body weight, just enjoy it	Positive
WFH certainly has a different feel to working in an office. When the house is filled with so many things that are not very clear, the house feels cramped, full, and crowded	Negative

2.3. Preprocessing Text

Preprocessing is the data preparation stage that aims to simplify the data processing process that works by ignoring unwanted items from the dataset [11]. Preprocessing focuses on data cleaning, such as removing noise in the data, overcoming bad data structures, and missing information. This stage is a process where data is prepared into data that is ready to be analyzed and into a machine-understandable data format [12]. The preprocessing stages in this study include Case folding, tokenization, normalization, Stop Forward Removal, Stemming and Data Cleaning. The display of the data after pre-processing the text is shown in Table 2.

Table 2. Preprocessing text

Preprocessing	Before	After
Case folding	Work from Home sometimes makes us more creative and of course we gain weight, just enjoy it	work from home sometimes makes us more creative and of course we gain weight just enjoy it
Tokenization	Work from Home sometimes makes us more creative and of course we gain weight, just enjoy it	work, from, home, sometimes, make, us, more, creative, and, of course, we, gain, weight, just, enjoy, it
Normalization	Work from Home sometimes makes us more creative and of course we gain weight, just enjoy it	work, from, home, sometimes, make, us, more, creative, and, of course, we, gain, weight, just, enjoy, it
Stop Forward Removal	Work from Home sometimes makes us more creative and of course we gain weight, just enjoy it	work, from, home, sometimes, creative, of course, gain, weight, enjoy
Stemming	Work from Home sometimes makes us more creative and of course we gain weight, just enjoy it	work from home sometimes creative of course gain weight enjoy
Cleaning	Work from Home sometimes makes us more creative and of course we gain weight, just enjoy it	Work from Home sometimes makes us more creative and of course we gain weight. Enjoy it

2.4. Word Weighting (Term Weighting)

TF-IDF word weighting is a way to weigh a term with a document [13][14]. This method combines two-weight calculation concepts. That is the frequency of a term in one document and the frequency of occurrence of the term throughout the document.

2.4.1. Term Frequency (TF)

Term Frequency is a weighting concept by finding how often (frequency) a term appears in a document [15][16]. This concept is usually divided by the total length of words in a document [17]. The more words that appear, the higher the TF value. This process will display the TF results.

$$tf_{ij} = \frac{f_d(i)}{max\ f_d\ (j)} \quad (1)$$

where $f_d(i)$ is Frequency of appearance of term i in document j and $max\ f_d\ (j)$ is a total term on document j.

2.4.2. Inverse Document Frequency (IDF)

Inverse Document Frequency is the frequency of occurrence of the term in the entire text document [18]. In contrast to TF, the IDF value will be greater if the number of words in the document decreases. In this process, the IDF value will be displayed.

$$idf(t,d) = log\left(\frac{N}{df(t)+1}\right) \quad (2)$$

Where N is the Total number of documents, $df(t)+1$ is the Number of documents that contain the term t and the value 1 to avoid dividing by the value 0 if it is not found in the corpus.

After getting the data from the Pre-Processing results, the TF-IDF stage is carried out to detect the words that appear most often and get a value so that each word from the entire document has a weight. The results of TF-IDF can be seen in Table 3.

Table. 3 TF-IDF Results

Term	TF	TF-IDF
mood	0.038461538461538464	0.197852564838761
kalau	0.038461538461538464	0.10612859598037541
ken	0.038461538461538464	0.153809017708376
marah	0.038461538461538464	0.21079380474392276
terus	0.038461538461538464	0.11214190638001749

Table 3 is the output of the weight value of each word obtained from the multiplication of the TF value and the IDF value. The smaller the TF IDF weight/value, the more often the word appears in the document. On the other hand, the higher the TF IDF value, the less often the word appears in the document or dataset.

2.5. Particle Swarm Optimization (PSO)

In this study, the Particle Swarm Optimization algorithm is used as a feature selection. The use of PSO feature selection helps the classification process and optimizes the workings of naive Bayes. One of the advantages of using this feature selection compared to other methods is that it is easy to implement because it only uses a few parameters [19][20]. Particle Swarm Optimization (PSO) starts with a population consisting of a number of particles that are randomly generated by searching using a population (swarm) of individuals (particles) to iteratively update the position and flight speed of each particle to produce a new solution that is more efficient, good. Particle Swarm Optimization will stop when the optimum solution has been found, or certain conditions have been reached. There are four steps to optimize Naïve Bayes using PSO, namely, initialize the population (swarm), calculate the accuracy value based on the selected characteristics, choose the best classification accuracy and update the position and velocity [21]-[23]. The following is an explanation of the Particle Swarm Optimization algorithm process flow:

1. Initialize the particles randomly with the initial velocity of all particles 0.
2. Evaluate the fitness value based on the lowest value of each particle.
3. Compare the fitness values and determine the Local Best and Global Best particles.
4. Updating the speed (Vi, m) and position (Xi, d) of each article randomly using the lower limit (X_{xmin}) and upper limit (X_{max}).
5. Check whether the last solution has converged or not. If the position of all the particles goes to the same value, it can be called convergent. If not, then it is necessary to iterate again until all particle positions go to the same value.

6. The process will stop automatically if it has reached the stopping criteria value.
7. Data output after the Particle Swarm Optimization process.

PSO by using the fitness value to evaluate each feature. In each iteration, the fitness value of the attribute is calculated using the fitness function. The PSO iteration will stop after reaching the target fitness value or reaching the maximum iteration. The results obtained from this stage are in the form of data whose weight has been optimized for each term. All terms will be directly processed at the classification stage using the Naïve Bayes Classifier method.

2.6 Naive Bayes Classifier Method

The Naive Bayesian classifier is a simple classification that uses Bayes' theorem with strong independent assumptions [24]-[26]. This method is widely used because it only requires a small amount of training data to estimate the parameters (mean and variance of the variable) required for classification. This algorithm utilizes the probability of each feature per category to get a prediction [27][28]. Naive Bayes classifies using two processes that divide the data into training data and testing data. The classification process in Naive Bayes on data is done by representing each data into "X1, X2, X3, ..., Xn" [29]. The set of categories is represented by K. Naive Bayes looks for the highest probability value when classifying. The following is the equation of Bayes' theorem [30]:

$$P(X|H) = p\frac{(H|X)p(H)}{P(X)} \quad (3)$$

Where X is Data whose class is unknown, H is Hypothesis data, X is a certain class, P(H|X) is the probability of hypothesis H based on condition X (posterior probability), P(H) is the probability of hypothesis H (prior probability), P(X|H) is Probability of X according to the conditions in the hypothesis H, P(X) is Probability of X.

The following is the flow of the Naive Bayes calculation.

1. Getting data from preprocessing and TF-IDF
2. Probability calculations are carried out from the training data where $V_j V_1 = positif$ and $V_2 = negatif$
3. The probability model is stored for processing at the data testing stage.
4. Determines the highest probability value to place the test data in the most appropriate category.

3. RESULTS AND DISCUSSION

In this study, the evaluation process was performed to test the performance of using the Naïve Bayes method based on the PSO to have better performance. Testing is done by evaluating using the Confusion Matrix. In this study, accuracy, precision, recall, and F-Measure is used as evaluation parameters. A confusion matrix is a tool that can be used to analyze how well the classification has been generated and can recognize tuples from different classes. This study applies a classification method with two scenarios. The first scenario uses Naive Bayes and the second scenario uses Naive Bayes based on Particle Swarm Optimization. This scenario process uses 3 data split tests with a ratio of 90:10, 80:20, and 70:30.

3.1. First Scenario Using Naïve Bayes

In the first scenario, testing is carried out using the Naive Bayes classification. The scenario was carried out with three tests with a ratio of 90%-10%, 80%-20%, and 70%-30%. The experiment using these three tests aims to find out which test is the best in the classification process. In the first test, the prediction results from the classification process on the test data resulted in a True positive of 47, False positive of 10, False-negative of 13, True negative of 50. In the second test, it resulted in a True positive of 98, False positive of 33, False-negative of 22, True negative of 87. In the third test, it resulted in a True positive of 149, False positive of 51, False-negative of 31, True negative of 129. The results of the implementation of the first scenario can be seen in Table 4.

Table 4. Scenario 1 Results (Naive Bayes)

Ratio	% precision		Recall %		F1-score %		% accuracy
	Positive	Negative	Positive	Negative	Positive	Negative	
90%-10%	82	79	78	83	80	81	81
80%-20%	75	80	82	72	78	76	77
70%-30%	74	81	83	72	78	76	77

Table 4 shows the results of the confusion matrix on the test data, which is divided into positive and negative labels on the results of precision, recall, and f1-score using Naive Bayes classification. For the first test using 90% of train data and 10% of test data, the average precision is 80.5%, recall is 80.5%, f1-score is 80.5%, and accuracy is 81%. For the second test using 80% of the train data and 20% of the test data, the average precision of 77.5%, 77% recall, 77% f1-score, and 77% accuracy was obtained. For the third test using 70% of train data and 30% of test data, the average precision is 77.5%, recall is 77.5%, f1-score is 77%, and accuracy is 77%.

3.2. Second Scenario Using PSO-Based Naïve Bayes

In the second scenario, testing is carried out using Naïve Bayes classification based on PSO. The purpose of this test is to find out whether the implementation of using the PSO feature selection has an effect on the test. The scenario was carried out with three tests with a ratio of 90%-10%, 80%-20%, and 70%-30%. The experiment using these three tests aims to find out which test is the best in the classification process. In the first test, the prediction results were obtained from the classification process on the test data True positive of 51, False positive of 8, False-negative of 9, True negative of 52. In the second test, it resulted in a True positive of 96, a False positive of 25, a False-negative of 24, True negative of 95. In the third, it resulted in a True positive of 149, False positive of 51, False-negative of 31, True negative of 129.

Table 5 shows the results of the confusion matrix on the PSO-based Naive Bayes classification. For the first test using 90% of train data and 10% of test data, the results obtained an average precision of 85.5%, 86% recall, 86% f1-score, and 86% accuracy. For the second test using 80% of train data and 20% of test data, the average precision is 79.5%, recall is 79.5%, f1-score is 79.5%, and accuracy is 80%. For the third test using 70% of train data and 30% of test data, the average precision is 80.5%, recall is 80.5%, f1-score is 80.5%, and accuracy is 81%.

Table 5. Scenario 2 Results (Naïve Bayes PSO)

Ratio	% precision		Recall %		F1-score %		% accuracy
	Positive	Negative	Positive	Negative	Positive	Negative	
90%-10%	86	85	85	87	86	86	86
80%-20%	79	80	80	79	80	79	80
70%-30%	79	82	83	78	81	80	81

3.3. Determining the Best Scenario

Based on the two scenarios, the results have been obtained through the visualization of the confusion matrix in the two classification processes. In the first scenario using Naive Bayes classification with three tests, the first test obtained an accuracy of 81%, then in the second test, an accuracy of 77% was obtained, and the third test obtained an accuracy of 77%. In the second scenario using PSO-based Naive Bayes classification, the accuracy results are 86% for the first test, 80% for the second test, and 81% for the third test. All classification processes that use the Particle Swarm Optimization algorithm as a selection feature affect the evaluation value in the confusion matrix as seen in the comparison results of the two scenarios above that the best scenario is the PSO-based Naive Bayes classification with an accuracy value of 86% through a comparison of train data of 90% and test data by 10%. The following is a comparison of the two scenarios.

Table 6 shows the results of the comparison of the two scenarios for the first scenario using Naive Bayes classification with three tests. The first test obtained an accuracy of 81%, then the second test obtained an accuracy of 77%, and the third test obtained an accuracy of 77%. For the second scenario using PSO-based Naive Bayes classification, the accuracy results are 86% for the first test, 80% for the second test, and 81% for the third test.

Table 6. The results of the comparison of scenario 1 and scenario 2

Scenario	Ratio	Accuracy
Naive Bayes	90%-10%	81%
	80%-20%	77%
	70%-30%	77%
Naive Bayes+PSO	90%-10%	86%
	80%-20%	80%
	70%-30%	81%

4. CONCLUSION

Based on the results of the tests that have been carried out that the best scenario is obtained in the Naive Bayes classification based on Particle Swarm Optimization, which can be seen in the accuracy results obtained.

This study consisted of three tests where the best results were found in the first test through a comparison of 90% train data and 10% test data. So it can be analyzed that the use of a larger data train is able to produce a better value in the confusion matrix. From the trials that have been carried out, the best accuracy results are 81% for classification using Naive Bayes and 86% for classification using Naive Bayes based on PSO. The performance of this test is able to provide optimal results with the addition of an accuracy of 5%. The PSO algorithm performs feature selection in the three tests whose results are able to reduce from 2033 features to 1225 features in the first test, 1229 features in the second test, and 1212 features in the third test. So with this, it can be concluded that the use of particle swarm optimization feature selection can help the Naive Bayes classification process to be more effective and accurate.

REFERENCES

[1] M. A. Hassonah, R. Al-Sayyed, A. Rodan, A. M. Al-Zoubi, I. Aljarah, and H. Faris, "An efficient hybrid filter and evolutionary wrapper approach for sentiment analysis of various topics on Twitter," *Knowledge-Based Syst.*, vol. 192, p. 105353, 2020. https://doi.org/10.1016/j.knosys.2019.105353

[2] S. Bashir *et al.*, "Twitter chirps for Syrian people: Sentiment analysis of tweets related to Syria Chemical Attack," *Int. J. Disaster Risk Reduct.*, vol. 62, no. May, p. 102397, 2021. https://doi.org/10.1016/j.ijdrr.2021.102397

[3] K. Sailunaz and R. Alhajj, "Emotion and sentiment analysis from Twitter text," *J. Comput. Sci.*, vol. 36, p. 101003, 2019. https://doi.org/10.1016/j.jocs.2019.05.009

[4] G. Li, Q. S. Zheng, L. Zhang, S. Z. Guo, and L. Y. Niu, "Sentiment Infomation based Model for Chinese text Sentiment Analysis," *2020 IEEE 3rd Int. Conf. Autom. Electron. Electr. Eng. AUTEEE 2020*, pp. 366–371, 2020. https://doi.org/10.1109/AUTEEE50969.2020.9315668

[5] T. Daudert, "Exploiting textual and relationship information for fine-grained financial sentiment analysis," *Knowledge-Based Syst.*, vol. 230, p. 107389, 2021. https://doi.org/10.1016/j.knosys.2021.107389

[6] Z. Jianqiang, G. Xiaolin, and Z. Xuejun, "Deep Convolution Neural Networks for Twitter Sentiment Analysis," *IEEE Access*, vol. 6, pp. 23253–23260, 2018. https://doi.org/10.1109/ACCESS.2017.2776930

[7] M. Wongkar and A. Angdresey, "Sentiment Analysis Using Naive Bayes Algorithm Of The Data Crawler: Twitter," *Proc. 2019 4th Int. Conf. Informatics Comput. ICIC 2019*, pp. 1–5, 2019. https://doi.org/10.1109/ICIC47613.2019.8985884

[8] A. Idrus, H. Brawijaya, and Maruloh, "Sentiment Analysis of State Officials News on Online Media Based on Public Opinion Using Naive Bayes Classifier Algorithm and Particle Swarm Optimization," *2018 6th Int. Conf. Cyber IT Serv. Manag. CITSM 2018*, 2018, pp. 1–7. https://doi.org/10.1109/CITSM.2018.8674331

[9] L. Mandloi and R. Patel, "Twitter sentiments analysis using machine learning methods," *2020 Int. Conf. Emerg. Technol. INCET 2020*, pp. 1–5, 2020. https://doi.org/10.1109/INCET49848.2020.9154183

[10] W. S. J. Saputra, P. Eva Yulia, and Z. E. Sholikha, "C4.5 and naive bayes for sentiment analysis Indonesian Tweet on E-Money user during pandemic," *Proceeding - 6th Inf. Technol. Int. Semin. ITIS 2020*, pp. 172–177, 2020. https://doi.org/10.1109/ITIS50118.2020.9321081

[11] H. Zhao, Z. Liu, X. Yao, and Q. Yang, "A machine learning-based sentiment analysis of online product reviews with a novel term weighting and feature selection approach," *Inf. Process. Manag.*, vol. 58, no. 5, p. 102656, 2021. https://doi.org/10.1016/j.ipm.2021.102656

[12] A. S. Neogi, K. A. Garg, R. K. Mishra, and Y. K. Dwivedi, "Sentiment analysis and classification of Indian farmers' protest using twitter data," *Int. J. Inf. Manag. Data Insights*, vol. 1, no. 2, p. 100019, 2021. https://doi.org/10.1016/j.jjimei.2021.100019

[13] S. Fahmi, L. Purnamawati, G. F. Shidik, M. Muljono, and A. Z. Fanani, "Sentiment analysis of student review in learning management system based on sastrawi stemmer and SVM-PSO," *Proc. - 2020 Int. Semin. Appl. Technol. Inf. Commun. IT Challenges Sustain. Scalability, Secur. Age Digit. Disruption, iSemantic 2020*, pp. 643–648, 2020. https://doi.org/10.1109/iSemantic50169.2020.9234291

[14] G. A. Dalaorao, A. M. Sison, and R. P. Medina, "Integrating Collocation as TF-IDF Enhancement to Improve Classification Accuracy," *TSSA 2019 - 13th Int. Conf. Telecommun. Syst. Serv. Appl. Proc.*, pp. 282–285, 2019. https://doi.org/10.1109/TSSA48701.2019.8985458

[15] H. Aljuaid, R. Iftikhar, S. Ahmad, M. Asif, and M. Tanvir Afzal, "Important citation identification using sentiment analysis of in-text citations," *Telemat. Informatics*, vol. 56, p. 101492, 2021. https://doi.org/10.1016/j.tele.2020.101492

[16] A. Poornima and K. S. Priya, "A Comparative Sentiment Analysis of Sentence Embedding Using Machine Learning Techniques," *2020 6th Int. Conf. Adv. Comput. Commun. Syst. ICACCS 2020*, pp. 493–496, 2020. https://doi.org/10.1109/ICACCS48705.2020.9074312

[17] K. U. Manjari, S. Rousha, D. Sumanth, and J. Sirisha Devi, "Extractive Text Summarization from Web pages using Selenium and TF-IDF algorithm," *Proc. 4th Int. Conf. Trends Electron. Informatics, ICOEI 2020*, 2020, pp. 648–652. https://doi.org/10.1109/ICOEI48184.2020.9142938

[18] U. Bhattacharjee, P. K. Srijith, and M. S. Desarkar, "Term Specific TF-IDF Boosting for Detection of Rumours in Social Networks," *2019 11th Int. Conf. Commun. Syst. Networks, COMSNETS 2019*, pp. 726–731, 2019. https://doi.org/10.1109/COMSNETS.2019.8711427

[19] A. Mustopa, Hermanto, Anna, E. B. Pratama, A. Hendini, and D. Risdiansyah, "Analysis of user reviews for the

pedulilindungi application on google play using the support vector machine and naive bayes algorithm based on particle swarm optimization," *2020 5th Int. Conf. Informatics Comput. ICIC 2020*, vol. 2, 2020. https://doi.org/10.1109/ICIC50835.2020.9288655
[20] E. Souza, A. L. I. Oliveira, G. Oliveira, A. Silva, and D. Santos, "An Unsupervised Particle Swarm Optimization Approach for Opinion Clustering," *Proc. - 2016 5th Brazilian Conf. Intell. Syst. BRACIS 2016*, pp. 307–312, 2017. https://doi.org/10.1109/BRACIS.2016.063
[21] A. S. Daoud, A. Sallam, and M. E. Wheed, "Improving Arabic document clustering using K-means algorithm and Particle Swarm Optimization," *2017 Intell. Syst. Conf. IntelliSys 2017*, vol. 2018-Janua, no. September, pp. 879–885, 2018. https://doi.org/10.1109/IntelliSys.2017.8324233
[22] W. K. Jati and L. Kemas Muslim, "Optimization of Decision Tree Algorithm in Text Classification of Job Applicants Using Particle Swarm Optimization," *2020 3rd Int. Conf. Inf. Commun. Technol. ICOIACT 2020*, pp. 201–205, 2020. https://doi.org/10.1109/ICOIACT50329.2020.9332101
[23] X. Bai, X. Gao, and B. Xue, "Particle Swarm Optimization Based Two-Stage Feature Selection in Text Mining," *2018 IEEE Congr. Evol. Comput. CEC 2018 - Proc.*, pp. 1–8, 2018. https://doi.org/10.1109/CEC.2018.8477773
[24] N. K. Suchetha, A. Nikhil, and P. Hrudya, "Comparing the wrapper feature selection evaluators on twitter sentiment classification," *ICCIDS 2019 - 2nd Int. Conf. Comput. Intell. Data Sci. Proc.*, pp. 1–6, 2019. https://doi.org/10.1109/ICCIDS.2019.8862033
[25] S. Ernawati, E. R. Yulia, Frieyadie, and Samudi, "Implementation of the Naïve Bayes Algorithm with Feature Selection using Genetic Algorithm for Sentiment Review Analysis of Fashion Online Companies," *2018 6th Int. Conf. Cyber IT Serv. Manag. (CITSM), 2018*, pp. 1–5. https://doi.org/10.1109/CITSM.2018.8674286
[26] U. Pujianto, M. F. Hidayat, and H. A. Rosyid, "Text Difficulty Classification Based on Lexile Levels Using K-Means Clustering and Multinomial Naive Bayes," *Proc. - 2019 Int. Semin. Appl. Technol. Inf. Commun. Ind. 4.0 Retrosp. Prospect. Challenges, iSemantic 2019*, pp. 163–170, 2019. https://doi.org/10.1109/ISEMANTIC.2019.8884317
[27] P. Sudhir and V. D. Suresh, "Comparative Study of Various Approaches, Applications and Classifiers for Sentiment Analysis," *Glob. Transitions Proc.*, 2021. https://doi.org/10.1016/j.gltp.2021.08.004
[28] S. Zahoor and R. Rohilla, "Twitter Sentiment Analysis Using Machine Learning Algorithms: A Case Study," *Proc. - 2020 Int. Conf. Adv. Comput. Commun. Mater. ICACCM 2020*, pp. 194–199, 2020. https://doi.org/10.1109/ICACCM50413.2020.9213011
[29] P. Liu, H. Yu, T. Xu, and C. Lan, "Research on archives text classification based on Naive Bayes," *Proc. 2017 IEEE 2nd Inf. Technol. Networking, Electron. Autom. Control Conf. ITNEC 2017*, vol. 2018-January, pp. 187–190, 2018. https://doi.org/10.1109/ITNEC.2017.8284934
[30] L. Xiao, G. Wang, and Y. Liu, "Patent Text Classification Based on Naive Bayesian Method," *Proc. - 2018 11th Int. Symp. Comput. Intell. Des. Isc. 2018*, vol. 1, pp. 57–60, 2018. https://doi.org/10.1109/ISCID.2018.00020

BIOGRAPHY OF AUTHORS

Rista Azizah Arilya born on May 16, 1999 in Kendari (Southeast Sulawesi, Indonesia). He is an undergraduate student at the University of Muhammadiyah Malang, majoring in informatics. His research interests include Data Science. E-mail: ristaazizaharilya@gmail.com

Yufis Azhar is a teaching staff in the Department of Informatics, University of Muhammadiyah Malang (UMM). He completed his undergraduate studies at the same study program and university in 2009. In 2011, he continued his master's education at the Informatics Engineering study program at the Institut Teknologi Sepuluh Nopember and completed his studies in 2013. Currently, he is actively researching in the fields of data science and artificial intelligence. E-mail: yufis@umm.ac.id

Didih Rizki Chandranegara born on October 2, 1992 in Palangka Raya (Central Kalimantan, Indonesia). He is a lecturer in the Informatics study program at the University of Muhammadiyah Malang. His research interest is Data Science. Email: didihrizki@umm.ac.id