

Improving sentiment reviews classification performance using support vector machine-fuzzy matching algorithm

Vivine Nurcahyawati¹, Zuriani Mustaffa²

¹Department of Information System, Faculty of Technology and Informatics, Universitas Dinamika, Surabaya, Indonesia

²Department of Computer Science, Faculty of Computing, Universiti Malaysia Pahang, Pahang, Malaysia

Article Info

Article history:

Received Sep 20, 2022

Revised Nov 11, 2022

Accepted Nov 27, 2022

Keywords:

Classification

Dimensionality

Performance

Preprocessing

Sentiment analysis

Text mining

ABSTRACT

High dimensionality in data sets is one of the challenges faced in classification, data mining, and sentiment analysis. In the data set, many dimensionalities require effort to simplify. Many of these dimensionalities have a major impact on the complexity and performance of the algorithms used for classification. Various challenges were encountered, including how to determine the optimal combination of pre-processing techniques, how to clean the dataset, and determine the best classification algorithm. This study uses a new approach based on the combination of three powerful techniques which are: tokenizing-lowercasing-stemming (for series of preprocessing), support vector machine (SVM) for supervised classification, and fuzzy matching (FM) for dimensionality reduction. The proposed model was realized using 3 different datasets, namely Amazon product review, movie review, and airline review from Twitter. This study provides better findings than the previous results. Improved performance is generated by SVM combined with FM, resulting in 96% accuracy. So that the SVM-FM combination can be said to be the best combination for sentiment analysis on the given data set.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Vivine Nurcahyawati

Department of Information System, Faculty of Technology and Informatics, Universitas Dinamika

Jl. Kedung Baruk 98 Surabaya, 60298, Indonesia

Email: vivine@dinamika.ac.id

1. INTRODUCTION

Sentiment analysis is an important task to detect sentiment polarity in the text, which is widely applied in e-commerce systems, blogs, and social media. Its main task is to group documents into various polarities. Based on automatic predictions, it assists the business owners to make an informed decisions and plan directions to grow their business [1]. The task of sentiment analysis can be thought of as a text classification problem as the process includes several operations ending in classifying whether a given text expresses a positive or negative sentiment [1]. It is estimated that 90% of the data to be analyzed, there is still unstructured or unorganized data. Unstructured business data is produced and retrieved every day in large quantities. The form is in the form of emails, testimonials, chats, social media conversations, surveys, articles, and documents. The data is difficult to conduct sentiment analysis in a timely and efficient manner.

The challenge in conducting sentiment analysis lies in several stages. Noise found in text data is still very interesting to be a research topic. Several studies have stated the advantages of pre-processing techniques. Tokenizing is recommended to get the best classification results [2]-[4]. On the study of Resyanto *et al.* [5] stemming is able to produce the highest accuracy. Stop word elimination and transform case are also recommended to be applied to reduce noise in a group of text [6]. In this study, the performance

comparison for each pre-processing technique was carried out and applied to several classification algorithms to determine the best performance.

High dimensionality in data sets is one of the challenges faced in classification, data mining, and sentiment analysis. In the data set there are many dimensionalities that require effort to simplify. Many of these dimensionalities have a major impact on the complexity and performance of the algorithms used for classification. These challenges can be addressed through the dimensionality reduction process. Fuzzy matching (FM) can be used to measure the inequality between two strings, where text matches can be found, even with misspelled and different words [3]. Measurement of text similarity can also apply Euclidean distance similarity (EDS) [7]. Principal component analysis (PCA) is a conversion technique that allows to reduce the size of a data set that includes a large number of interrelated dimensionalities, so that the current data can be expressed with a smaller number of variables [8]. However, some existing studies only solve problems on one data source. Therefore, it is necessary to know whether optimal performance is also generated when applied to other datasets.

Analyzing large amounts of data is an expensive and time-consuming process. Due to that matter, it is vital to have the best model to produce the best classification performance. In this study, support vector machine (SVM) will be used for sentiment analysis classification. Prior to that, to reduce the dimensionality of the data, three algorithms will be used namely PCA, FM, and EDS. The proposed model was realized using 3 different datasets, namely Amazon product review, movie review, and airline review from Twitter and compare against several identified algorithms which includes naive Bayes (NB), k-nearest neighbor (kNN), dan deep learning (DL). The results of this study will present a combination of pre-processing techniques, algorithms to reduce data dimensions, and the most optimal classification algorithm. The combination of several techniques and algorithms will produce the best accuracy, precision, recall, and f-measure values.

Currently, much research on sentiment analysis have been carried out. Various challenges were encountered, including how to determine the optimal combination of pre-processing techniques, how to clean the dataset, and determine the best classification algorithm. The combination of pre-processing techniques used are tokenizing, stop words elimination, transform case, and stemming [1]. The tokenizing technique helps to reduce the dimensions of the problem in a group of texts. By cutting a group of sentences into chunks of words makes the analysis process into a simpler form [1], [7], [9]–[11]. Capitalization variability in this dataset can cause problems during classification and degrade performance. The technique of converting capital letters to lowercase is the most common method of dealing with problems in text data. The lowercasing technique helps avoid different variations of the same word as determined by the case [5], [7], [9]–[15]. The existence of the stop word in a group of text is quite a lot, but this word has no meaning to be analyzed. Therefore, removing the stop word is a good way to simplify the dimensions of the problem in the text [1], [4]–[6], [9], [11]–[16]. Stemming technique [1], [5]–[7], [9]–[12] has the advantage of being able to change words into basic forms (removing affixes). This step can reduce the analysis process on unnecessary text.

Several classification algorithms show good results in classifying sentiment analysis. These algorithms include NB [17], kNN [9], [12], [17], SVM [4], [6], [9]–[16], [18], and DL [19]–[21]. Research by Shaban *et al.* [22] in his study yielded an accuracy of 98% with a classification using NB. According to Kumari *et al.* [23] uses SVM and produces 90% accuracy. According to Romadhon and Kurniawan [17] apply kNN and produce 75% accuracy. To simplify the dimensions of the analysed text, an algorithm is used FM [3], PCA [8], [24], [25], EDS [7].

SVM has been used in research on several system areas, such as: presidential election [12], health record data [18], customer satisfaction [16], fake consumer reviews [6], restaurant review [9], flood disaster news [14], Vietnamese [15], product reviews [4], [11]. The accuracy is quite high in research using the SVM algorithm. In the presidential election system, the accuracy is 76.5% [12], customer satisfaction system 84.85% [18], and product reviews with accuracy 88.13% [11]. Various datasets are used, including Twitter [10], [12], [14], [16], [18], Cornell University, trip advisor [6], restaurant review [9], Vietnamese [15], product reviews [4], application comments, financial market news [10], and product review Amazon [11]. SVM already has the advantages of support vector and a dividing line (hyperplane) so it takes a little time to classify [12]. SVM with basic pre-processing can significantly improve accuracy [18]. SVM is great for big data. Takes longer time for classification [16]. Combination of SVM with n-gram improves accuracy [16]. The number and length of extracted word segments have a major influence on the performance of the classifier. The number of word segments is divided in sufficient numbers in the form of bigrams and trigrams [6]. SVM is suitable and produces good accuracy values for various types of datasets [13]. Good performance SVM in non-linear classification [4]. SVM produces best for several types of data sets. SVM yields best for multiple pre-processing combinations. SVM is recommended as the best algorithm for sentiment analysis [10]. The main objective of this study is to present the optimal combination of techniques and algorithms and of course the best accuracy value in sentiment analysis. We introduce a modified model to achieve this goal by using a combination of SVM and FM [3].

2. METHOD

Generating a dataset that is free from noise and achieves the best accuracy values is a challenge in the sentiment analysis domain. The proposed model consists of 6 stages. The first is data collection. The dataset is taken from 3 datasets about consumer reviews. Second, an experiment was conducted to apply a combination of pre-processing techniques to the collected datasets. Third, determine training data and testing data. Fourth, apply algorithms to reduce the dimensionality of the data. The fifth step is to classify with several algorithms. The last step is to measure, compare several algorithms, and compare with benchmark models. The following is a step by step of the method that has been carried out.

2.1. Collecting dataset

This study uses 3 datasets about customer reviews from different sources, namely Amazon product review (dataset A), movie review (dataset B), and airline review from Twitter (dataset C). The dataset is a dataset from studies that become benchmarks [11], [26]. Data were taken from the Kaggle website [27] and Amazon dataset [28]. The polarity of the data consists of positive and negative. The dataset file is then made into one file and processed with a word processing application (Microsoft Excel).

2.2. Data pre-processing configurations

The data pre-processing process is a series of procedures to clean up unnecessary data before the data is used in the analysis process. A series of techniques used are basic techniques to clean data, namely tokenizing, stop words elimination, transform case, and stemming [1].

- Tokenizing: the tokenizing technique is a method of breaking text into smaller parts (which are referred to as tokens), turning the content into meaningful data while retaining the text in sentences [1], [11]. Tokens are earned by separating text by spaces, punctuation, or line breaks [9].
- Stop words elimination: stop words are words that are ignored in processing and are usually stored in stop lists. Stop lists are lists of common words that have function but no meaning. The main characteristic in choosing stop words is usually words that have a high frequency of occurrence, for example connecting words such as "and", "or", "but" and "will" [12], [29].
- Transform case: this technique is one of the basic techniques that converts the entire text into lowercase or uppercase. If there are the same words, then the results will be combined into one. So that this technique can reduce the dimensionality of the dataset to be analyzed [9], [15].
- Stemming: stemming technique is a technique that converts words into basic words by removing the suffix attached to the word [1], [9].

2.3. Data determination

Experiments using datasets are determined by the composition of the training data: testing data is 80:10. Data in the three datasets are relatively the same in number, but dataset B has the greatest number of words. This happens because dataset B is movie audience reviews. The composition of the dataset settings is as shown in Table 1.

Table 1. Compositon of dataset

Dataset	Data training			Data testing				
	#Pos	#Neg	#Total	#Words	#Pos	#Neg	#Total	#Words
Dataset A	4,064	3,952	8,016	660,192	1,016	988	2,004	165,048
Dataset B	3,824	4,176	8,000	1,899,664	956	1,044	2,000	474,916
Dataset C	3,216	4,960	8,176	138,736	804	1,240	2,044	34,684

2.4. Dimensionality reduction

The challenge in pre-processing data is how to reduce noise and dimensionality of the data, so that it can speed up the analysis process or improve analysis performance. Performance improvement can be done by applying pre-processing techniques and reducing the dimensions of the data to be processed [7], [9], [11], [13]. Several algorithms are applied to see how they perform against the classification results of sentiment analysis, namely term frequency-inverse document frequency (TF-IDF), PCA, FM, and EDS.

2.4.1. The term frequency-inverse document frequency

The TF-IDF is a process of dimensionality reduction technique with the process of assigning a value to each word in the training data. To find out how important a word is in representing a sentence, a calculation will be given. The value of the TF-IDF depends on the frequency with which words appear in the document. TF is considered to have a proportion of importance according to the total occurrence in the text

or document. IDF is a token weighting method that functions to monitor the occurrence of tokens in a text set. In extraction with TF-IDF, calculate the value (w) of each document against keywords [11].

2.4.2. Principal component analysis

Algorithm for reducing dimensions or variables by changing a set of correlated dimensions into uncorrelated dimensions. This algorithm will produce a value called the principal component (PC). The PC data is a linear combination of the original values before being reduced. The PC are obtained by projecting the vector into the space defined by the eigenvectors with some calculations, namely: i) calculate the covariance matrix; ii) find the eigenvalues and eigenvectors; iii) compute reduction percentage; and iv) PC [8], [24], [25].

2.4.3. Fuzzy matching

A fuzzy way based on two nominal attributes. This means it matches examples which are not necessarily equal, but similar. Between the two chosen attribute we calculate a similarity. The operator merges the k most similar examples from both sides. The similarity method can be defined using the 'distance measure' parameter. Currently all similarity measures are Levenshtein distance based. Levenshtein distance is using the number of changes you need to do to get from one string to the other to define a distance [3].

2.4.5. Euclidean distance similarity

Euclidean distance is a way to gauge how close vectors are to one another in a vector space. Euclidean is related to the Pythagorean Theorem and is usually applied to 1, 2 and 3 dimensions. But it's also simple when applied to higher dimensions. Therefore, it's crucial that we clarify what we mean when we talk about the distance between two vectors since, as we'll see in a moment, it's not always clear [7].

2.5. Sentiment classification

Sentiment classification is a branch of text mining. Sentiment classification can be important in the process of evaluating a topic of concern. The main purpose of sentiment classification is to find out the polarity of positive, negative, and neutral sentiments. Based on research [11], [17], [30], for comparison purposes, the SVM will be compared against comparable classifier namely NB, kNN, and DL [19].

2.5.1. Naive Bayes

With the use of conditional probabilities. The NB classifier assigns class labels to instances and records in order to perform the supervised method of object categorization in the future. The likelihood of an event occurring dependent on other occurrences that have (assumed, presumed, stated, or confirmed) to occur is known as conditional probability.

2.5.2. K-nearest neighbor

The kNN algorithm is a classification algorithm that works by taking a number of K data closest (neighbors) as a reference to determine the class of new data. This algorithm classifies data based on similarity or similarity or proximity to other data [12], [17]. In general, the way the kNN algorithm works is as follows: i) determine the number of neighbors (K) that will be used for class determination considerations, ii) calculate the distance from the new data to each data point in the dataset, and iii) take several K data with the closest distance, then determine the class of the new data.

2.5.3. Deep learning

A multi-layer feed-forward artificial neural network that uses back-propagation to train with stochastic gradient descent forms the foundation of DL. DL architectures have already been employed in a variety of applications, such as computer vision, pattern recognition, and NLP. The ability to learn multi-level dimensionality representations is provided by DL architectures. The architectures look for learning models based on numerous layers of hierarchically nonlinear information processing [20].

2.6. Measurement and evaluation

High retrieval performance is maintained while an effective preparation method accurately reflects the document in terms of both space and time. In this study, three metrics were employed namely accuracy, precision, and recall. These metrics serve to determine a sentiment classifier performance.

3. RESULTS AND DISCUSSION

The datasets used in this study are data on Amazon product reviews (dataset A), movie reviews (dataset B), and airline reviews from Twitter (dataset C). The reviews written by consumers have various

forms. In datasets A and B, the sentences written are quite long compared to dataset C. Dataset C (Twitter) contains a short review because there are character restrictions in writing reviews on Twitter.

To be able to see the performance of each pre-processing technique, measurements were made on several combinations of using techniques. The combination is: i) comparing the results of dataset classification without techniques using all pre-processing techniques; ii) comparing the performance of individual pre-processing techniques; and iii) comparing the performance of the combination of several pre-processing techniques. In this section, we conduct several experiments to see the effect of each data pre-processing technique. The experiment was carried out on 3 datasets that had been prepared. Accuracy results were compared using 4 classifiers, NB, kNN, SVM, and DL. The first experiment was conducted to see the performance of each pre-processing technique on three datasets and several classifiers. The application of pre-processing techniques can increase the value of classification accuracy, as shown in Table 2.

Table 2. Accuracy with and without pre-processing techniques

	Dataset A				Dataset B				Dataset C			
	NB	kNN	SVM	DL	NB	kNN	SVM	DL	NB	kNN	SVM	DL
No pre-processing	50	51	51	51	48	52	52	52	39.22	39.22	61.76	38.24
All pre-processing	71	69	78	55	62	66	75	53	72.55	76.47	79.41	73.53

The SVM algorithm produces the highest accuracy value both before and after applying the pre-processing technique. The effect of pre-processing to produce the highest accuracy value is the use of the SVM algorithm, which is 79.41%. A significant increase is shown by the application of the kNN algorithm on dataset C, namely the accuracy increases by 37.25%, as shown in Figure 1.

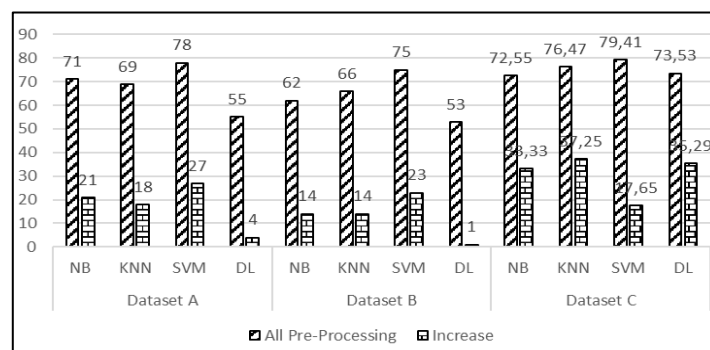


Figure 1. Accuracy of application of classification algorithm

The performance of each pre-processing technique was measured in the next experiment. The use of tokenizing technique produces a significant effect on accuracy results, as shown in Table 3. The effect of tokenizing produces the highest accuracy value is the use of the kNN algorithm, which is 77.45%. The increase in the accuracy value is 39.22% from before using the pre-processing technique.

Table 3. Tokenization technique performance

	Dataset A				Dataset B				Dataset C			
	NB	kNN	SVM	DL	NB	kNN	SVM	DL	NB	kNN	SVM	DL
No pre-processing	50	51	51	51	48	52	52	52	39.22	39.22	61.76	38.24
Tokenizing	75	64	74	59	70	63	71	53	71.57	77.45	75.49	65.69

Table 4 shows that the lowercasing, stop word elimination, and stemming techniques have no significant effect. In datasets A and B using the DL algorithm, lowercasing, stop word elimination, and stemming techniques actually make the accuracy value decrease. For dataset C using the SVM algorithm, only lowercasing and stemming techniques can increase the accuracy value. But the increase in accuracy is not very significant. Meanwhile for dataset C using the DL algorithm, lowercasing techniques, stop word elimination, and stemming can improve accuracy. The biggest increase is when the lowercasing technique is applied.

The next experiment is to do a combination of pre-processing techniques. The combination is as follow: i) tokenizing+lowercasing; ii) tokenizing+stop word elimination; and iii) tokenizing+lowercasing+stemming. Based on the experimental results, the highest accuracy value resulted from the use of the SVM algorithm on three datasets. As seen in Table 5, In dataset A, the highest accuracy is obtained by using a combination of tokenizing+lowercasing. Meanwhile, in datasets B and C, the highest accuracy is obtained with a combination of tokenizing+lowercasing+stemming.

Table 4. Lowercasing, stop word elimination, and stemming performance

	Dataset A				Dataset B				Dataset C			
	NB	kNN	SVM	DL	NB	kNN	SVM	DL	NB	kNN	SVM	DL
No pre-processing	50	51	51	51	48	52	52	52	39.22	39.22	61.76	38.24
Lowercasing	50	51	51	49	48	52	52	48	39.22	39.22	62.75	62.75
Stop word elimination	50	51	51	49	48	52	52	52	39.22	39.22	61.76	50
Stemming	50	51	51	49	48	52	52	52	39.22	39.22	62.75	39.22

Table 5. Combination of text pre-processing techniques

	Dataset A				Dataset B				Dataset C			
	NB	kNN	SVM	DL	NB	kNN	SVM	DL	NB	kNN	SVM	DL
Tokenizing+lowercasing	75	71	80	55	67	69	73	52	73.53	77.45	79.41	62.75
Tokenizing+stop word elimination	72	70	79	55	69	55	73	53	68.63	76.47	77.45	62.75
Tokenizing+lowercasing+stemming	71	70	77	54	61	66	76	53	72.55	76.47	81.37	75.49

The classification algorithm that shows the best results is SVM, while the best combination of pre-processing techniques is tokenizing, lowercasing, and stemming. The combination is then tested by applying an algorithm to reduce the dimensions of the data, namely TF-IDF, PCA, FM, and EDS. The experimental results in Figures 2(a)-(c) show that the FM algorithm gives the best results, namely increasing the accuracy value to 96% in 3 datasets. Figure 2(a) shows that FM produces the highest accuracy value in dataset A, while PCA shows the lowest result. The use of FM in dataset A showed an increase of 16% (from 80% to 96%). Likewise in Figure 2(b), in dataset B, FM shows the highest accuracy value compared to TF-IDF, PCA, and EDS. The application of FM on dataset B increased the accuracy value by 20% (from 76% to 96%). Not different from dataset A and dataset B, Figure 2(c) shows that FM produces the highest accuracy value, and the FM has succeeded in increasing accuracy by 14.63% (from 81.37% to 96%).

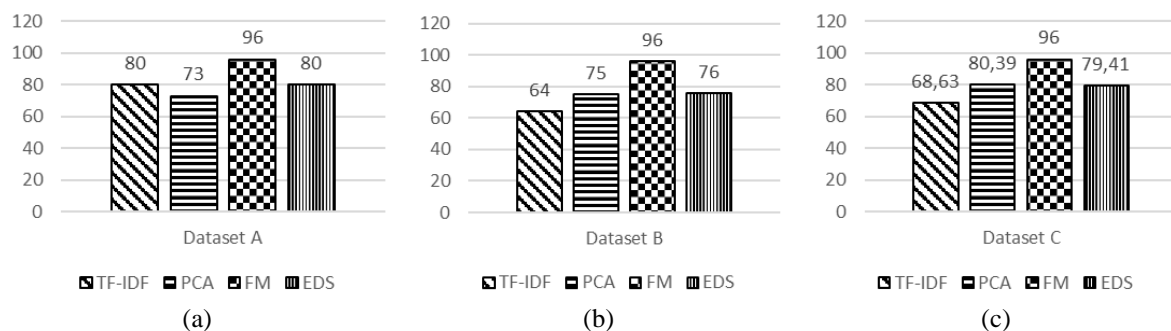


Figure 2. Performance comparison of dimensionality reduction algorithm for (a) dataset A, (b) dataset B, and (c) dataset C

At the evaluation stage, we implemented a series of experiments using pre-processing tokenizing, lowercasing, and stemming techniques. The SVM classification algorithm is applied with a combination of TF-IDF, PCA, FM, and EDS algorithms individually. The proposed model achieves improved performance results compared to the benchmark model. On benchmark models [11], the highest accuracy result obtained is 88.75%. While in the proposed model, the accuracy value increased to 96%. Figure 3 shows that the experiment on 3 datasets has a higher accuracy value than the benchmark model. Table 6 shows the complete data of accuracy, precision, recall, and f-measure values.

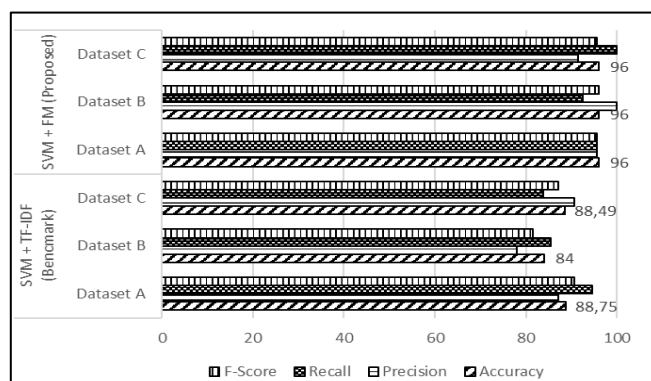


Figure 3. Performance of proposed model

Table 6. Measurement of proposed model

		Accuracy	Precision	Recall	F-Measure
SVM+TF-IDF (benchmark)	Dataset A	88.75	87.18	94.56	90.72
	Dataset B	84	78.08	85.42	81.59
	Dataset C	88.49	90.61	83.87	87.11
SVM+FM (proposed)	Dataset A	96	95.65	95.65	95.65
	Dataset B	96	100	92.45	96.08
	Dataset C	96	91.49	100	95.56

4. CONCLUSION

This paper proposes a combination of SVM with dimensionality reduction techniques for sentiment analysis classification. To improve the performance of SVM, it is combined with dimensionality reduction techniques namely TF-IDF, PCA, FM, and EDS algorithms. Prior to that, data preprocessing techniques have been applied to clean the data which includes tokenizing, stop words elimination, transform case, and stemming. Upon completing the experiment, it is demonstrated that SVM-FM produces the best result with the accuracy of 96% for all the identified datasets. Therefore, it is safe to conclude that SVM-FM can be used to produce the best sentiment analysis performance.

ACKNOWLEDGEMENTS

Thanks to Dinamika University for the motivational encouragement, self-development opportunities, and funding for this research. Thanks also to all the research teams involved.




REFERENCES

- [1] M. Birjali, M. Kasri, and A. Beni-Hssane, "A comprehensive survey on sentiment analysis: Approaches, challenges and trends," *Knowl Based Syst*, vol. 226, pp. 1–26, Aug. 2021, doi: 10.1016/j.knosys.2021.107134.
- [2] P. Chawla, S. Hazarika, and H.-W. Shen, "Token-wise sentiment decomposition for ConvNet: Visualizing a sentiment classifier," *Visual Informatics*, vol. 4, no. 2, pp. 132–141, Jun. 2020, doi: 10.1016/j.visinf.2020.04.006.
- [3] M. S. Oliveira, A. Mourthe, and M. C. Duque, "Extracting events from daily drilling reports using fuzzy string matching," *The APPEA Journal*, vol. 62, no. 2, pp. 158–161, May 2022, doi: 10.1071/AJ21118.
- [4] Y. Hong and X. Shao, "Emotional analysis of clothing product reviews based on machine learning," in *2021 3rd International Conference on Applied Machine Learning (ICAML)*, Jul. 2021, pp. 398–401. doi: 10.1109/ICAML54311.2021.00090.
- [5] F. Resyanto, Y. Sibaroni, and A. Romadhony, "Choosing the most optimum text preprocessing method for sentiment analysis: Case: iPhone Tweets," in *2019 Fourth International Conference on Informatics and Computing (ICIC)*, Oct. 2019, pp. 1–5. doi: 10.1109/ICIC47613.2019.8985943.
- [6] A. Barushka and P. Hajek, "The effect of text preprocessing strategies on detecting fake consumer reviews," in *Proceedings of the 2019 3rd International Conference on E-Business and Internet*, Nov. 2019, pp. 13–17. doi: 10.1145/3383902.3383908.
- [7] R. Setiabudi, N. M. S. Iswari, and A. Rusli, "Enhancing text classification performance by preprocessing misspelled words in Indonesian language," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 19, no. 4, pp. 1234–1241, Aug. 2021, doi: 10.12928/telkomnika.v19i4.20369.
- [8] E. O. Omuya, G. O. Okeyo, and M. W. Kimwele, "Feature selection for classification using principal component analysis and information gain," *Expert Syst Appl*, vol. 174, pp. 1–12, Jul. 2021, doi: 10.1016/j.eswa.2021.114765.
- [9] M. İŞIK and H. DAĞ, "The impact of text preprocessing on the prediction of review ratings," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 28, no. 3, pp. 1405–1421, May 2020, doi: 10.3906/elk-1907-46.
- [10] D. N. de Oliveira and L. H. de C. Merschmann, "Joint evaluation of preprocessing tasks with classifiers for sentiment analysis in Brazilian Portuguese language," *Multimed Tools Appl*, vol. 80, no. 10, pp. 15391–15412, Apr. 2021, doi: 10.1007/s11042-020-10323-8.
- [11] M. Arief and M. B. M. Deris, "Text preprocessing impact for sentiment classification in product review," in *2021 Sixth International Conference on Informatics and Computing (ICIC)*, Nov. 2021, pp. 1–7. doi: 10.1109/ICIC54025.2021.9632884.




- [12] F. Firmansyah *et al.*, “Comparing sentiment analysis of Indonesian presidential election 2019 with support vector machine and k-nearest neighbor algorithm,” in *2020 6th International Conference on Computing Engineering and Design (ICCED)*, Oct. 2020, pp. 1–6. doi: 10.1109/ICCED51276.2020.9415767.
- [13] Y. HaCohen-Kerner, D. Miller, and Y. Yigal, “The influence of preprocessing on text classification using a bag-of-words representation,” *PLoS One*, vol. 15, no. 5, pp. 1–22, May 2020, doi: 10.1371/journal.pone.0232525.
- [14] M. K. Delimayanti, R. Sari, M. Laya, M. R. Faisal, Pahrul, and R. F. Naryanto, “The effect of pre-processing on the classification of Twitter’s flood disaster messages using support vector machine algorithm,” in *2020 3rd International Conference on Applied Engineering (ICAE)*, Oct. 2020, pp. 1–6. doi: 10.1109/ICAE50557.2020.9350387.
- [15] H.-T. Duong and T.-A. Nguyen-Thi, “A review: Preprocessing techniques and data augmentation for sentiment analysis,” *Comput Soc Netw*, vol. 8, no. 1, pp. 1–16, Dec. 2021, doi: 10.1186/s40649-020-00080-x.
- [16] J. Mahilraj*, G. Tigistu, and S. Tumsa, “Text preprocessing method on Twitter sentiment analysis using machine learning,” *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 12, pp. 233–240, Sep. 2020, doi: 10.35940/ijitee.K7771.0991120.
- [17] M. R. Romadhon and F. Kurniawan, “A comparison of naive Bayes methods, logistic regression and KNN for predicting healing of Covid-19 patients in Indonesia,” in *2021 3rd East Indonesia Conference on Computer and Information Technology (EIConCIT)*, Apr. 2021, pp. 41–44. doi: 10.1109/EIConCIT50028.2021.9431845.
- [18] C. S. P. Kumar and L. D. D. Babu, “Novel text preprocessing framework for sentiment analysis,” in *Smart intelligent computing and applications*, 2019, pp. 309–317. doi: 10.1007/978-981-13-1927-3_33.
- [19] W. Zhao *et al.*, “Weakly-supervised deep embedding for product review sentiment analysis,” *IEEE Trans Knowl Data Eng*, vol. 30, no. 1, pp. 185–197, Jan. 2018, doi: 10.1109/TKDE.2017.2756658.
- [20] A. Onan, “Sentiment analysis on product reviews based on weighted word embeddings and deep neural networks,” *Concurr Comput*, vol. 33, no. 23, pp. 1–12, Dec. 2021, doi: 10.1002/cpe.5909.
- [21] M. Aydoğan and A. Karci, “Improving the accuracy using pre-trained word embeddings on deep neural networks for Turkish text classification,” *Physica A: Statistical Mechanics and its Applications*, vol. 541, pp. 1–19, Mar. 2020, doi: 10.1016/j.physa.2019.123288.
- [22] W. M. Shaban, A. H. Rabie, A. I. Saleh, and M. A. Abo-Elsoud, “Accurate detection of COVID-19 patients based on distance biased Naïve Bayes (DBNB) classification strategy,” *Pattern Recognit*, vol. 119, pp. 1–15, Nov. 2021, doi: 10.1016/j.patcog.2021.108110.
- [23] U. Kumari, A. K. Sharma, and D. Soni, “Sentiment analysis of smart phone product review using SVM classification technique,” in *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, Aug. 2017, pp. 1469–1474. doi: 10.1109/ICECDS.2017.8389689.
- [24] I. Jolliffe, “A 50-year personal journey through time with principal component analysis,” *J Multivar Anal*, vol. 188, pp. 1–9, Mar. 2022, doi: 10.1016/j.jmva.2021.104820.
- [25] P. J. Atkins and M. Cummins, “Improved scalability and risk factor proxying with a two-step principal component analysis for multi-curve modelling,” *Eur J Oper Res*, vol. 304, no. 3, pp. 1331–1348, Feb. 2023, doi: 10.1016/j.ejor.2022.04.044.
- [26] E. Sutoyo, A. P. Rifai, A. Risnumawan, and M. Saputra, “A comparison of text weighting schemes on sentiment analysis of government policies: a case study of replacement of national examinations,” *Multimed Tools Appl*, vol. 81, no. 5, pp. 6413–6431, Feb. 2022, doi: 10.1007/s11042-022-11900-9.
- [27] Kaggle Inc, “Kaggle_ Your Machine Learning and Data Science Community,” *Kaggle*, 2022. <https://www.kaggle.com/> (accessed Dec. 12, 2022).
- [28] AWS, “Registry of Open Data on AWS,” AWS, 2022. <https://registry.opendata.aws/> (accessed Dec. 12, 2022).
- [29] V. Gurusamy and S. Kannan, “Preprocessing techniques for text mining,” *International Journal of Computer Science & Communication Networks*, vol. 5, no. 1, pp. 7–16, 2014.
- [30] W. Etaifi and G. Naymat, “The impact of applying different preprocessing steps on review spam detection,” *Procedia Comput Sci*, vol. 113, pp. 273–279, 2017, doi: 10.1016/j.procs.2017.08.368.

BIOGRAPHIES OF AUTHORS



Vivine Nurcahyawati    is a Ph.D candidate at the Faculty of Computing, Universiti of Malaysia Pahang, her interest is in Data Mining and Natural Language Processing. She has 7 years of experience as a System Analyst and Database Administrator and 17 years in teaching at the Department of Information System, Universitas Dinamika, Surabaya, Indonesia. Her research areas include data mining, natural language processing, and software engineering. She can be contacted at email: vivine@dinamika.ac.id.



Zuriani Mustafa    is Senior Lecturer in Faculty of Computing, Universiti Malaysia Pahang, Malaysia. Her research interest includes computational intelligence (CI) algorithm, specifically in swarm intelligence (SI) and machine learning techniques. Her research area focuses on hybrid algorithms which involves optimization and machine learning techniques with particular attention for time series predictive analysis. She has authored and co-authored various scientific articles in the field of interest. She can be contacted at email: zuriani@ump.edu.my.