https://doi.org/10.17576/apjitm-2021-1001-04
Asia-Pacific Journal of Information Technology and Multimedia
Jurnal Teknologi Maklumat dan Multimedia Asia-Pasifik
Vol. 10 No. 1, June 2021: 43 - 51
e-ISSN: 2289-2192

Received: 3 January 2021
Accepted: 15 April 2021
Published: 1 June 2021

OPTIMIZATION OF K-NEAREST NEIGHBOUR TO CATEGORIZE INDONESIAN'S NEWS ARTICLES

AFDHALUL IHSAN
EDNAWATI RAINARLI

ABSTRACT

Text classification is the process of grouping documents based on similarity in categories. Some of the obstacles in doing text classification are many words appeared in the text, and some words come up with infrequent frequency (sparse words). The way to solve this problem is to conduct the feature selection process. There are several filter-based feature selection methods; some are Chi-Square, Information Gain, Genetic Algorithm, and Particle Swarm Optimization (PSO). Aghdam's research shows that PSO is the best among those methods. This study examined PSO to optimize the k-Nearest Neighbour (k-NN) algorithm's performance in categorizing news articles. k-NN is an algorithm that is simple and easy to implement. If we use the appropriate features, then the k-NN will be a reliable algorithm. PSO algorithm is used to select keywords (term features), and it is continued with classifying the documents using k-NN. The testing process consists of three stages. The stages are tuning the parameter of k-NN, the parameter of PSO, and measuring the testing performance. The parameter tuning process aims to determine the number of neighbours used in k-NN and optimize the PSO particles. Otherwise, the performance testing compares the performance of k-NN with and without using PSO. The optimal number of neighbours is 9, with the number of particles is 50. The testing showed that using the k-NN with PSO and a 50% reduction in terms. The results 20 per cent better accuracy than k-NN without PSO. Although the PSO's process did not always find the optimal conditions, the k-NN method can produce better accuracy. In this way, the k-NN method can work better in grouping news articles, especially in Indonesian language news articles.

Keywords**:** feature selection, k-nearest neighbour, metaheuristic, optimization, text classification.

INTRODUCTION

The growth of internet users made the transition of mass media to digital platforms increase rapidly. Articles that appear on the news portal web have various news categories. So, we need automatic news grouping based on categories to search for news efficiently. News categorization is one of the applications of document classification studies. Research on document classification is improving along with the massive increase in digital documents. Methods widely used in document classification are Support Vector Machine (SVM) (Afia and Amiri, 2016; Wongso *et al.*, 2017; Tudu *et al.*, 2018; Yovellia Londo *et al.*, 2019; Djajadinata *et al.*, 2020; Rabbimov and Kobilov, 2020), k-Nearest Neighbor (k-NN) (Alhutaish and Omar, 2015; Afia and Amiri, 2016; Rahman and Akter, 2019; Chen *et al.*, 2020; Djajadinata *et al.*, 2020), Multinominal Naïve Bayes (MNB) (Afia and Amiri, 2016; Wongso *et al.*, 2017; Rahman and Akter, 2019; Yovellia Londo *et al.*, 2019; Djajadinata *et al.*, 2020; Rabbimov and Kobilov, 2020), and Decision Tree (DT) (Afia and Amiri, 2016; Tudu *et al.*, 2018; Rahman and Akter, 2019; Djajadinata *et al.*, 2020; Rabbimov and Kobilov, 2020). Among these methods, k-NN is the easiest method to implement. This method does not require time for training and has a good performance as SVM and MNB (Afia and Amiri, 2016).

k-NN is a simple algorithm, which uses distance-based measures for classification. The classifier determines the testing data class by looking at the k nearest neighbours' training data. The algorithm reported the majority class of neighbours as the label of testing data. Even k-NN requires no training time, but it takes time to classify test documents. The testing process involves the computation of the distances of all the training vectors from the test vector. Some researchers make some improvements like developing feature weighting (Alhutaish and Omar, 2015), purpose new term to finding relevant features (Afia and Amiri, 2016), or make a normalization and dimension reduction (Chen *et al.*, 2020). All solutions try to enhance the accuracy of k-NN. Dimensional reduction in k-NN has a close relationship with problems in text classification. The main problem of text classification is how to find a representative keyword to categorize the news. The fact that most occurrences of words are sparse. Additionally, words occurrence with high frequency implies that they are not suitable keywords in determining the categories. The appearance of a lot of the rare words in the document also cannot be used as keywords. Therefore, the feature selection method aims to select the relevant keywords features for each category and indirectly reduce features' dimension.

Aghdam and Heidari (2015) compared several word feature selection methods such as Information Gain (IG), Chi-Square, Genetic Algorithm (GA), Particle Swarm Optimization (PSO). The testing results show that PSO and GA performance are the best among the other selection methods. However, finding the optimal solution from the GA method takes a longer time than the PSO method. Aghdam and Heidari's research (2015) used a feature selection method with a filtering approach. In this research, we used PSO as a feature selection method. The difference with the previous research is we utilize the k-NN accuracy value to become its objective function. Our study aims to evaluate k-NN PSO and compare the accuracy to k-NN only. We also want to implement the possibility of k-NN PSO being used to classify articles in Indonesian.

RESEARCH METHOD

We divide the system into two parts: training and testing. The training aims to find a list of word features using PSO. The testing uses the list of training results word features to determine news articles' category from the testing data. The data set consists of 250 news articles taken from online news portals in Indonesian: Kompas, Liputan 6, Detik.com. The system classifies news articles into five categories. They are health, sport, technology, automotive, and travelling.

(Figure 1) shows an overview of our proposed system. There are two process lines. The first is the classification of news articles using k-NN, and the second is the classification of news using k-NN PSO. The first flow used the pre-processed word list; from the training data; as a word list in the testing data. After the pre-processing of testing data, k-NN directly classified the weighted results of testing data. We calculate the performance of classification using a confusion matrix. The initial process of k-NN PSO is the same as the first line. After going through pre-processing, we continue with weighting words of the training data. The system will use the weighting words results to classify news articles from the training data using k-NN. The performance of k-NN will be the fitness value on PSO. The PSO algorithm selects the optimal candidate keyword words, namely those that minimize the fitness value. The system uses the selection words to weigh the testing data. The final output is the accuracy value of the confusion matrix.

PRE-PROCESSING

There are four processes in the pre-processing stage: case folding, filtering, tokenizing, and stop word removal. Case folding is a process for homogenizing the characters in the article. Uniform characters can use lower case or upper case. In this study, we convert all text into lower case. The next process is filtering. Filtering aims to remove noise in news articles. The noise can be in the form of punctuation marks and numbers. In (Table 1) shows the details of the non-letter characters filtered. The rules in filtering are as follows:

1. We replace the delimiter characters with spaces.
2. The system will delete each numeric character.
3. The final process is to remove excess space due to changing the delimiter character to spacing. This third process will affect the tokenizing process.

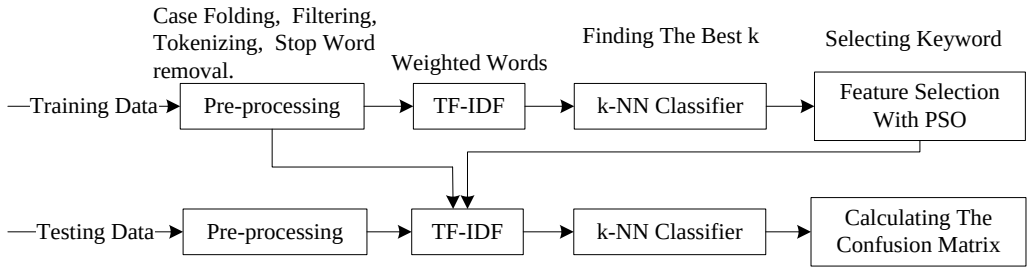

FIGURE 1. Overview of the news document classification system

The next process is tokenizing. Tokenizing separates the articles into tokens. Tokens can be in the form of fragments of words, words, sentences, or paragraphs. In this study, tokenizing is carried out based on words or word fragments. The system split tokens using a space delimiter. After we collect words and word fragments, they are listed as a bag of words. The stop word list deletes unnecessary words in the bag of words. We use a stop word list taken from the Sastrawi's library (Andy, 2015).

TABLE 1. The Character Removed on Filtering

Character								
1	2	3	4	5	6	7	8	9
!	@	#	$	%	^	&	*	(
)	_	+	=	{	}	[	]	:
;	"	<	,	>	.	?	/	\
-	0		`	‘	“	~		

TERM FREQUENCY – INVERSE DOCUMENT FREQUENCY (TF-IDF)

The selected word lists are then weighted using TF-IDF (Aizawa, 2003). The TF-IDF method combines two concepts in calculating the weight of a term. The first is to calculate the frequency of occurrences of words in one document, and the second to calculate the inverse of the document containing the word term. We can use equations (1), (2), and (3) to calculate the terms weight (Kulaib, 2020),

$$W_t = Tf_{t,d} \times Idf_t, \quad (1)$$

$$Tf_{t,d} = \frac{n_{t,d}}{\sum_{i=1}^{m} n_{i,d}}, \quad (2)$$

$$Idf_{t,d} = \log \frac{n_d}{n_{d,t}}, \quad (3)$$

with W_t = the weight of *t*-th term,
$Tf_{t,d}$ = the term frequency of *t*-th term in *d*-th document,
$Idf_{t,d}$ = the inverse document frequency of *t*-th term in *d*-th document,
$n_{t,d}$ = the number of *t*-th term that appear in *d*-th document,
$\sum_{i=1}^{m} n_{i,d}$ = the number of all term that appear in *d*-th document,
m = the number of terms that appear in *d*-th document,
n_d = the number of documents,
$n_{d,t}$ = the number of documents that contain *t*-th term.

K-NEAREST NEIGHBOUR CLASSIFIER

The k-Nearest Neighbor (k-NN) classifier is an algorithm that classifies datum based on *k* nearest neighbors. The number of the closest neighbors are more than one. We use the Euclidean distance equation (4) to measure the distance between two data. Even the k-NN does not require time for training, but the method requires memory a lot. The algorithm use the memory to remember the distances between each datum with others (Cunningham and Delany, 2007).

$$D(\vec{x}, \vec{y}) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_r - y_r)^2} \quad (4)$$

with $D(\vec{x}, \vec{y})$ = the distance between two documents $\vec{x}$ and $\vec{y}$,
r = the dimension of $\vec{x}$ and $\vec{y}$ (number of term).

The following is the algorithm from the k-NN (Harrison, 2018):

1. Initialize the number of neighbors (*k*).
2. For each classified datum, calculate the distance of datum with the training data.
3. For each classified datum, sort the proximity based on the distance.
4. For each classified datum, take the *k* neighbors, then the class of the classified data is the highest class voting from the *k* neighbors.

FEATURE SELECTION WITH PARTICLE SWARM OPTIMIZATION

Feature selection using PSO will take as many as *m* features from the existing wordlist features. This feature selection is the wrapper-based model. In the wrapper-based, feature selection requires a classification method to evaluate the features (El Aboudi and Benhlima, 2016). The fitness value for particle evaluation is the accuracy of k-NN. PSO is an optimization algorithm that is a group of Swarm-based algorithms. The collective behaviour of social animals inspires Swarm-based algorithms works. The PSO algorithm defines a set of candidate solutions as a collection of moving particles in searching for the optimal solution. During movement, each particle can remember the value of its best function. The algorithm seeks the optimal solution by updating the particle's position based on the best experience and its surroundings (Xue, Zhang and Browne, 2014; Marini and Walczak, 2015).

In this study, the position of the *i*-th particle ($\vec{x}_i$) is the candidate term that becomes the keyword, namely $\vec{x}_i = (x_{i,1}, x_{i,2}, x_{i,3}, \dots, x_{i,m})$, *m* is the number of keywords to be searched for, and *i* is the number of particles in the group (swarm). Each particle $\vec{x}_i$ has velocity $\vec{v}_i = (v_{i,1}, v_{i,2}, v_{i,3}, \dots, v_{i,m})$. The particle dimension *m* is determined based on the number of the desired features (terms). We determine the dimensions of the particles by using the simulation of the test section. The algorithm saves the best position of the previous particle as personal best (pbest) and saves the best position obtained by the swarm as global best (gbest). PSO looks for the optimal solution by updating the position and velocity of each particle. In (Figure 2), the selection stages of the k-NN PSO are as follows (Xue, Zhang and Browne, 2014):

1. Initialize the PSO parameters. The PSO parameters are the number of particles in the swarm (n), the initial velocity at 0 iterations ($\vec{v}_i^0$), the acceleration constants (C_1, C_2), the number of neighbors from k-NN (k), the initial pbest value, the initial value gbest, and the maximum iteration (i_{max}).
2. Generate the initial particle in the 0th iteration ($\vec{x}_i^0), i = 1,2, \dots, n$. The value of each particle element is a sequence of terms taken randomly as many as *m*.
3. Evaluate the fitness value of each particle using the accuracy value. Equation (5) is a fitness function. We get the fitness function from the k-NN classification results using training data.

$$f(x_i^j) = \frac{number\ of\ documents\ classified\ correctly}{number\ of\ document}, \tag{5}$$

with $f(\vec{x}_i^j)$ is the fitness value of the particle $\vec{x}_i$ in the *j*-th iteration.

4. Update pbest and gbest. The algorithm updates the pbest by comparing the current pbest value with the previous pbest value. Pbest is the best position of each particle in each iteration, while gbest is the best position of pbest in each iteration.
5. Update the velocity of each particle component using equation (6),

$$v_{i,d}^{j+1} = v_{i,d}^j + C_1 \cdot r_{1,i}\left(pbest_d - x_{i,d}^j\right) + C_2 \cdot r_{2,i}\left(gbest_d - x_{i,d}^j\right), \tag{6}$$

with $v_{i,d}^{j+1}$ is the velocity of the *i*-th particle with the *d*-th particle element in the *j*+1-th iteration, $r_{1,i}$ and $r_{2,i}$ are random numbers with the uniform distribution between 0 to 1, $pbest_d$ and $gbest_d$ are the pbest values and the gbest values of the *d*-th particle element.

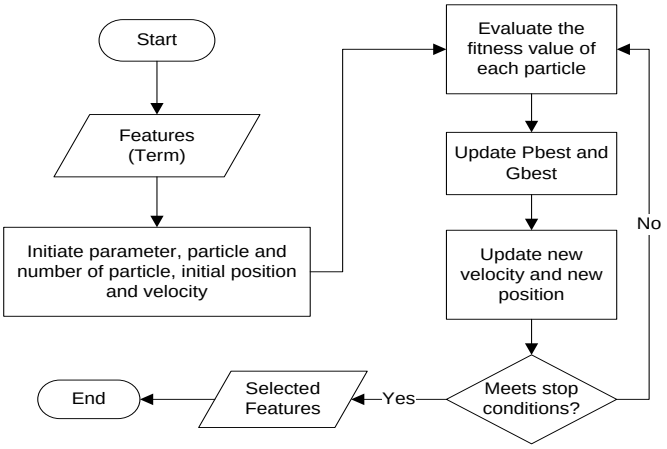

FIGURE 2. Flow Chart of k-NN and PSO

6. Update the particle position using the velocity from equation (6). The algorithm calculates the particle position using equation (7),

$$x_{i,d}^{j+1} = x_{i,d}^{j} + v_{i,d}^{j+1}. \quad (7)$$

7. Check the stop condition using two stop conditions, namely if it meets the maximum interaction or the fitness value = 1.

RESULT AND DISCUSSION

We carry out several test scenarios to measure news article classification performance using k-NN PSO and k-NN. The test scenarios are as follows:

1. Implement k-NN without PSO to classify news articles based on the categories. This scenario wants to determine the best *k* value. We choose *k* value from 3 to 10.
2. Test the feature selection method using 75% and 50% of all features and determining the number of generated particles. The number of particles used is 15, 30, and 50.
3. Evaluate the performances of k-NN PSO with *k* values selected from 3 to 10.

We split the data set into 80% training and 20% testing. There is no specific explanation regarding the basis for selecting the proportion of training data and testing data. However, Brownlee explained that we could choose the ratio of training and testing based on its computational and represented data (Brownlee, 2020). In this study, we used the accuracy value to measure the performance of k-NN with the addition of PSO as a feature selection process. Kadry and Ismael (2020) stated that the feature selection process carried out at k-NN aims to produce better accuracy values. The test results in (Table 2) is used to determine the number of *k*. The news articles classify using k-NN only. The results from (Table 2) show that for values of $k = 3, 9, 10$, the highest accuracy is 0.6. We chose $k = 9$ as the optimal number because Band (2020) said that a low *k* value will result in unstable decision boundaries.

TABLE 2. The Accuracy Value for Each *k* of k-NN

k	3	4	5	6	7	8	9	10
Accuracy	0.60	0.44	0.52	0.56	0.56	0.56	0.60	0.60

The second test uses 75% and 50% of the word feature for each particle 15, 30, 50. (Table 3) describes the accuracy value for each value. (Table 3) shows the relationship between the number of particles and the number of features with the fitness value. All fitness values in (Table 3) do not reach one. These values mean that using PSO does not obtain an optimal solution. The result is in line with what was stated by Voratas Kachitvichyanukul (2012), who explained that using the PSO does not guarantee that the PSO will find the optimal solution. (Table 3) also shows that the more particles generated, the higher fitness value achieved. The result is consistent with what Aghdam and Heidari (2015) stated: the maximum particle in the PSO to produce good accuracy is 50 with maximal iteration is 100. The addition of excessive particles will have an impact on the length of computation time. This result also applies to the number of word features taken. Although there are no specific rules for selecting the number of terms from the results, based on Aghdam and Heidari (2015), a reduction of 50% of all features becomes the maximum value in the feature selection process. Reducing redundant features will result in deleted feature keywords, which will affect system performance. Based

on (Table 3), we will use a combination of particles 30 and 50 using 75% features keyword and 50% features keyword.

TABLE 3. Comparison of Fitness Value Between the Number of Particles and the Number of Features

Number of Particles	Fitness Value	
	Using 75% of Features	Using 50% of Features
15	0.64	0.68
30	0.76	0.72
50	0.76	0.76

The final test compares the k-NN PSO accuracy with the number of k-NN using four test conditions. The graph of (Figure 3) shows the accuracy value obtained from the four test conditions. We get the best accuracy when k = 9 using 50 particles and 50% features as keywords or using 50 particles and 75% feature keywords. This result strengthens the results obtained by Aghdam and Heidari (2015), who state that the use of 50 particles in PSO produces the best accuracy. For determining the number of feature keywords used, a smaller percentage will make the computation faster. Therefore, the best results were obtained from the test results when using 50 particles and using 50% feature keywords.

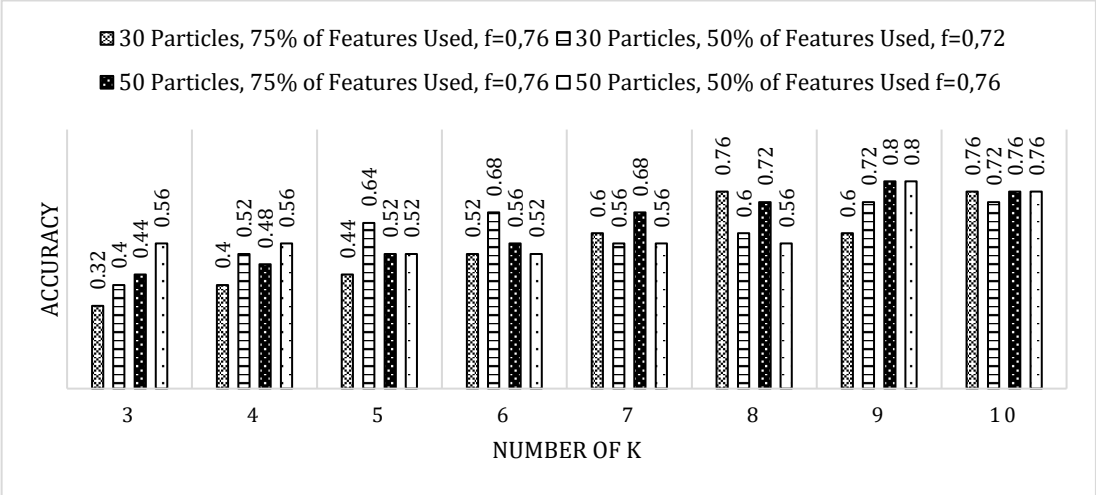

FIGURE 3. Result of k-NN PSO with Four Testing Scenarios and Variation of k

(Figure 4) shows a comparison of the accuracy value between k-NN and k-NN PSO. We can see that for the sum of k = 3, classification with PSO yields better accuracy values than k-NN PSO. The accuracy of k-NN and k-NN PSO achieve the same values for k = 5 and k = 6. Both k-NN and k-NN PSO produced the highest accuracy values when k = 9 were 0.6 and 0.8, respectively. We find that the use of k-NN PSO, in this case, can increase the accuracy by 0.2. We still obtain increasing the accuracy even though using PSO the algorithm does not achieve the optimal solution (the fitness value does not reach one).

CONCLUSION

This study has used PSO to perform a wrapper-based feature selection. In the iterative process, PSO used the k-NN accuracy value as a fitness function. This value measured the achievement of the optimal conditions from PSO algorithm. The test results show that k-NN PSO can produce an accuracy of 20% better than using k-NN only. The experiment still gave a good performance even we did not get the optimal fitness value of the k-NN. In the end, we have

evidenced that the k-NN PSO can classify news documents, particularly articles in Indonesian. For further research, we can modify the fitness function on the PSO to determine the optimal features. We can also explore finding the best number of k of k-NN adaptively.

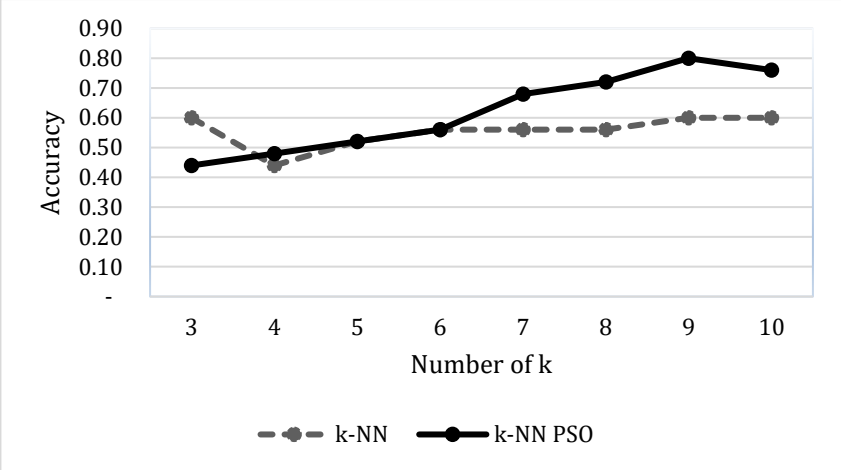

FIGURE 4. Comparison of the accuracy of k-NN with k-NN PSO

REFERENCES

Afia, A. B. & Amiri, H. 2016. Text classification using scores based k-NN approach and term to category relevance weighting scheme. *International Journal of Signal and Imaging Systems Engineering*, 9(4–5):283–290.

Aghdam, M. H. and Heidari, S. 2015. Feature selection using particle swarm optimization in text categorization. *Journal of Artificial Intelligence and Soft Computing Research*, 5(4):231–238.

Aizawa, A. 2003. An information-theoretic perspective of tf–idf measures. *Information Processing & Management*, 39(1):45–65.

Alhutaish, R. and Omar, N. 2015. Arabic text classification using K-nearest neighbour algorithm. *International Arab Journal of Information Technology*, 12(2):190–195.

Andy, L. 2015. *Sastrawi*. https://github.com/sastrawi/sastrawi [January 10th, 2019].

Band, A. 2020. *How to find the optimal value of K in KNN?* https://towardsdatascience.com/how-to-find-the-optimal-value-of-k-in-knn-35d936e554eb [November 10th 2020].

Brownlee, J. 2020. *Train-Test Split for Evaluating Machine Learning Algorithms*. https:// machinelearningmastery.com/train-test-split-for-evaluating-machine-learning-algorithms/ [April 2nd , 2021].

Chen, Z., Zhou, L.J., Li, X. Da, Zhang, J.N. & Huo, W.J. 2020. The Lao text classification method based on k-NN. *Procedia Computer Science*, 166:523–528.

Cunningham, P. & Delany, S.J. 2007. *K -Nearest Neighbour Classifiers*.

Djajadinata, K., Faisol, H., Shidik, G.F., Muljono & Fanani, A.Z. 2020. Evaluation of feature extraction for Indonesian news classification. *Proceedings - 2020 International Seminar on Application for Technology of Information and Communication.* Semarang: IEEE, 585–591.

El Aboudi, N. & Benhlima, L. 2016. Review on wrapper feature selection approaches. in *Proceedings - 2016 International Conference on Engineering and MIS*. Agadir: IEEE, 1–5.

Harrison, O. 2018. Machine learning basics with the k-Nearest Neighbors algorithm. *https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761* [April 2nd, 2019].

Kachitvichyanukul, V. 2012. Comparison of three evolutionary algorithms: GA, PSO, and DE. *Industrial Engineering & Management Systems*, 11(3): 215–223.

Kadry, R. and Ismael, O. 2020. A New Hybrid KNN Classification Approach based on Particle Swarm Optimization. *International Journal of Advanced Computer Science and Applications*, 11(11): 291–296.

Kulaib, B. 2020. TF-IDF steps by hand. *https://medium.com/baraakulaib/tf-idf-steps-by-hand-260fe6f4474b* [February 1st, 2021].

Marini, F. & Walczak, B. 2015. Particle swarm optimization (PSO). A tutorial. *Chemometrics and Intelligent Laboratory Systems*, 149:153–165.

Rabbimov, I.M. & Kobilov, S.S. 2020. Multi-class text classification of Uzbek news articles using machine learning. *Journal of Physics: Conference Series*, 1546(1):1–11.

Rahman, M.A. & Akter, Y.A. 2019. Topic classification from text using decision tree, k-NN, and multinomial naïve bayes. *1st International Conference on Advances in Science, Engineering and Robotics Technology*. Bangladesh: IEEE, 1–4.

Tudu, R., Saha, S., Pritam, P.N. & Palit, R. 2018. Performance analysis of supervised machine learning approaches for Bengali text categorization. *5th Asia-Pacific World Congress on Computer Science and Engineering*, Nadi: IEEE, 221–226.

Wongso, R., Luwinda, F.A., Trisnajaya, B.C., Rusli, O. & Rudy. 2017. News article text classification in Indonesian language. *Procedia Computer Science*, 116:137–143.

Xue, B., Zhang, M. & Browne, W.N. 2014. *Particle swarm optimisation for feature selection in classification*. PhD thesis, Victoria University of Wellington.

Yovellia Londo, G.L., Kartawijaya, D.H., Ivariyani, H.T., Yohanes Sigit, P.W.P., Muhammad Rafi, A.P. & Ariyandi, D. 2019. A study of text classification for Indonesian news article. *Proceeding - 2019 International Conference of Artificial Intelligence and Information Technology*. Yogyakarta: IEEE, 205–208.

Afdhalul Ihsan
Ednawati Rainarli
Faculty of Engineering and Computer Science,
Universitas Komputer Indonesia.
afdhalulihsan@email.unikom.ac.id, ednawati.rainarli@email.unikom.ac.id.