Jurnal CoreIT, Vol.10, No.1, June 2024
ISSN 2460-738X (Print)
ISSN 2599-3321 (Online)

IMPROVING PERFORMANCE OF RANDOM FOREST ALGORITHM USING ABC FEATURE SELECTION FOR SOFTWARE DEFECT PREDICTION

Zaina Fadia Laila Hidayati[1*], Friska Abadi[2], Muliadi[3], Mohammad Reza Faisal[4], Dwi Kartini[5]
[1]Dept. of Computer Science, Universitas Lambung Mangkurat, Indonesia
[1]1911016220025@mhs.ulm.ac.id*, [2]friska.abadi@ulm.ac.id, [3]muliadi@ulm.ac.id, [4]reza.faisal@ulm.ac.id, [5]dwikartini@ulm.ac.id

Abstract. Defects that may arise in software during the development process can affect the quality of the software. The classification method is used to predict software defects to minimize defects. However, the dataset used in the classification process may contain less relevant or have too many features. This can be overcome by selecting features in the dataset. In this research, the Random Forest algorithm is applied for the classification process, and the Artificial Bee Colony (ABC) algorithm is used as a feature selection method. The research aims to determine the accuracy of Random Forest with ABC feature selection. From the results of research conducted on 3 Relink datasets, without feature selection, an average accuracy of 73% was obtained. After implementing ABC feature selection, the average accuracy increased to 82%.

Keywords: Artificial Bee Colony, Feature Selection, Random Forest, Software Defect Prediction

Received March 2024 / **Revised** August 2024 / **Accepted** August 2024

INTRODUCTION

The quality of a software cannot be doubted if no defects are found during the inspection and testing phases [1]. If the quality of the software is low, it can lead to significant losses and inflated maintenance costs [2]. Certainly, such losses are undesirable for any party involved in the software development process. As an effort to minimize the occurrence of errors or defects in software, software defect prediction is necessary [3]. Analyzing a software module that has the potential for defects using machine learning classifiers is the process of software defect prediction [4].

Classification techniques can predict the presence of defects in software [5]. One of the classification algorithms that has been applied for software defect prediction is the Random Forest algorithm. Random Forest is a classification algorithm that can outperform several other algorithms such as Decision Tree and Neural Network in detecting faults or defects in modules [6]. Moreover, it can also influence classification results positively even when there is missing data [7].

In the classification process, not all features can be used as some may be irrelevant to the classification outcome [8]. When irrelevant features and excessive feature numbers are addressed, this can improve classification performance [6]. According to [9], the number of features in software is usually high-dimensional, and poor performance of a software defect prediction model is often caused by high-dimensional data. Meanwhile, according to [10], with the advancement of technology, high-dimensional data increases complexity and negatively affects algorithm performance, often referred to as the "curse of dimensionality". Issues related to data dimensionality can affect the accuracy of classification algorithm performance in software defect prediction. One way to address this issue is by performing feature selection.

Feature selection is an essential step in data preprocessing for machine learning. It can be employed in software defect prediction processes when dealing with high-dimensional datasets and noise [11]. By utilizing feature selection, the dimensionality of the dataset can be reduced, thereby identifying subsets of features and data with precision. Classifiers trained on feature spaces that have undergone feature selection tend to be stronger and more reproducible compared to those trained on feature spaces without prior feature selection [12].

DOI: 10.24014/coreit.v10i1.29283

The Artificial Bee Colony (ABC) algorithm is a population-based algorithm that mimics the foraging behavior of bee colonies. ABC has been widely applied in various research fields, including data mining, with roles in clustering, feature selection, and rule discovery [13]. In a study by [14], ABC feature selection was used to enhance the performance of the CART classification algorithm, and it was found that ABC significantly influenced the accuracy of the CART algorithm compared to other feature selection techniques such as Particle Swarm Optimization (PSO) and Ant Colony Optimization. Beyond that, the ABC algorithm also has the ability to influence the performance of the Backpropagation algorithm and produce significant accuracy improvements [15].

In this research, a prediction of software defects was conducted using the Random Forest algorithm with feature selection using the ABC algorithm on the Relink dataset. Additionally, a comparison of performance was made between the Random Forest classification without feature selection and the Random Forest classification using feature selection.

METHODS

The research flow in this study is shown in the Figure 1.

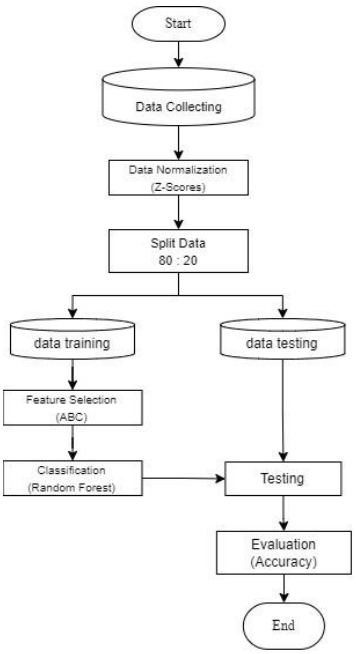

Figure 1. Research Flow

Data Collecting

The dataset used in this research is the Relink dataset, which consists of the Apache, Safe, and Zxing datasets [16]. This dataset can be downloaded from the following link: https://github.com/bharlow058/AEEEM-and-otherSDP-datasets/tree/master/dataset/Relink. This dataset has the same number of features but different numbers of records. There are two classes in the dataset: Buggy and Clean. Buggy indicates defective data, while Clean indicates non-defective data. The difference in the number of data is shown in Table 1.

Table 1. The Number of data in Relink dataset.

Dataset	Number of *Record*	Class	
		Buggy	Clean
Apache	194	98	96
Safe	56	22	34
Zxing	399	118	281

Data Normalization

Before the data is divided into training and testing data, it is normalized first. The normalization stage in predicting defects is considered important because often the existing features differ in order and magnitude [17]. One data normalization technique that can be used is z-score normalization. Z-score normalization is a standardization technique where the values of its attributes are normalized based on their mean and standard deviation. The formula for z-score is as follows.

$$Z = \frac{X - mean}{Std} \qquad (1)$$

X is the observed value (original data), Mean is the mean value, and *Std* is the standard deviation value.

Split Data

Splitting data in data science is used to divide data into two or more parts. Training data is used to train the model, while testing data is for proper performance assessment [18]. The data is divided based on a certain ratio to determine how much is used for training data and likewise for testing data. In this study, the separation ratio used is 80:20. 80% of the data will be used to train the model, and the remaining 20% will be used to test the model afterward. The separation is done in a stratified manner, so the dataset remains in the same class proportions. Table 2 shows the results of the dataset split performed.

Table 2. Split data results

Dataset	Amount of training data	Amount of Testing data
Apache	155	39
Safe	44	12
Zxing	319	80

Feature Selection

Artificial Bee Colony (ABC) is a population-based algorithm that operates similarly to a colony of bees searching for food sources. There are three groups of bees in the ABC structure, namely onlooker bees, employed bees, and scout bees. The way it works is as follows: first, employed bees will head towards the food source. Once they have found the food source, employed bees will perform a waggle dance to provide information to onlooker bees waiting at the hive. The information conveyed by bees in this process includes three important aspects: direction, distance, and quality of the nectar. The general structure of the ABC algorithm [19] is as follows.

1. Initializing food sources.
2. Employee bees head towards the food sources and determine the amount of food.
3. Upon receiving signals from employee bees, onlooker bees head towards the food sources to determine the amount of nectar.
4. Scout bees then search for new food sources.
5. Remember the best food source obtained.

Repeat Steps 1-5 until the specified number of iterations.

The ABC algorithm has been widely applied in various research fields, including data mining, with roles in clustering, feature selection, and rule discovery [13]. Compared to other metaheuristic algorithms, the ABC algorithm has many advantages. Firstly, its working structure is simpler. Secondly, it has fewer control parameters compared to other algorithms. Thirdly, the ABC algorithm is easy to implement [20]. The steps for feature selection using ABC are as follows.

a. Initializing Food Source. Food sources are determined randomly using the following equation.

$$X_{i,j} = X_{min.j} + rand(0,1)(X_{max.j} - X_{min.j}) \qquad (2)$$

Explanation:
X : initial position
$X_{\max}$: upperbound
$X_{\min}$: lowerbound

Each food source will be converted into a bit vector, where each vector consists only of the numbers 1 and 0 [21].

b. Calculate the initial fitness values using the Random Forest algorithm. These fitness values are considered as the evaluation results of the initial population.
c. Employed bees then begin exploration by selecting food sources. Exploration is done using the following equation.

$$V_{ij} = X_{ij} + \Phi ij\ (X_{ij} - X_{kj}) \quad (3)$$

Explanation:

X	: *Food Sources*
Φij	: random value betweeen -1 and 1
i dan *j*	: random variable
k	: value within the range N (solution) and not equal to *i*

d. Calculate the fitness of the employed bee by determining its accuracy, update the fitness if the fitness of the employed bee is better than the initial fitness. Then determine the probability value using the following formula:

$$Pi = \frac{fitness_i}{\sum_{n=1}^{F} fitness_i} \quad (4)$$

e. Onlooker bee is distributed and then determine the neighboring solution. Then go back to the previous step.
f. The scout bee performs random selection to update the solution from the abandoned solution.
g. The final solution is formed.

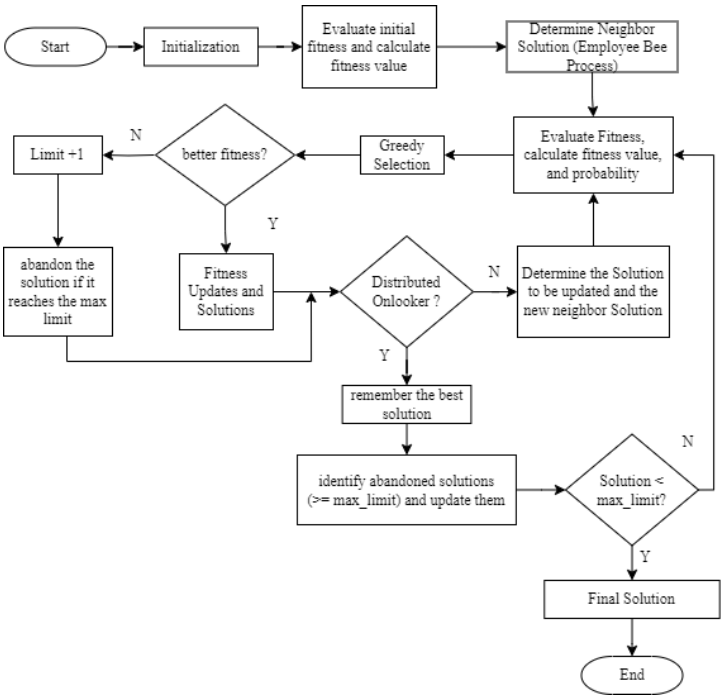

Figure 2. Feature Selection Process ABC

Classification

In this stage, classification will be performed using Random Forest. The data, which consists of selected features after previous feature selection, will be trained on the Random Forest classification. Then, testing will be conducted using testing data adjusted to the selected features.

Random Forest

Random forest is an algorithm widely used for classification and regression. It is part of the ensemble methods that combine many decision trees to improve prediction accuracy. Random Forest is a development of the Decision Tree algorithm and contains many decision trees. In the classification process,

these decision trees are filled with each sample of the input data [6]. Leo Breiman first introduced this algorithm in 2001 with several advantages. Among them are better classification performance, lower error rates, and the ability to handle large amounts of data [22]. The general structure of random forest is shown in Figure 3 below.

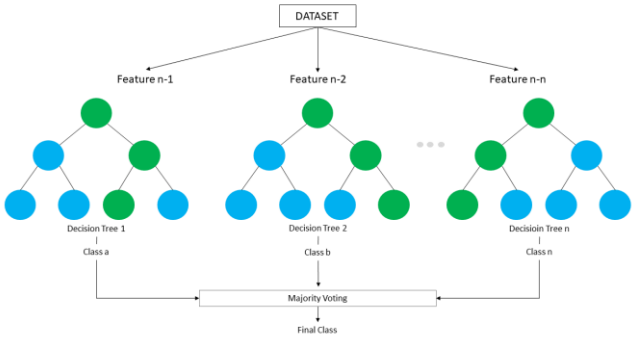

Figure 3. General structure of a random forest

Evaluation

Evaluation in the classification of the random forest algorithm uses accuracy values. Accuracy is a benchmark for assessing how precise the classification is, predicting the class of data based on previously trained data [23]. One of the most popular methods of machine learning evaluation. Accuracy is commonly used for balanced data.

RESULT AND DISCUSSION

Classification results of the Random Forest algorithm with Artificial Bee Colony feature selection for software defect prediction will be described as follows.

In this study, the parameters used are N (number of solutions) which is twenty-six solutions, T (number of iterations) namely 10, 50, and 100, Max_limit (maximum limit) is 5, Upperbound = 1 and Lowerbound = 0, and threshold = 0.5. The value of N is 26 because the number of solutions is equal to the number of features in the dataset [24]. Feature selection is performed based on 3 different numbers of iterations, namely ten, fifty, and one hundred. With the use of 3 datasets and 3 different iterations (in Table 3 and Table 4), feature selection is performed nine times.

Table 3. Fitness values feature selection

Dataset	Fitness Values		
	iteration 10	iteration 50	iteration 100
Apache	0,261219	0,225079	0,189827
Safe	0,18320	0,197281	0,182430
Zxing	0,253868	0,271291	0.269151

Table 4. Number of selected features

Dataset	Number of selected features		
	iteration 10	iteration 50	iteration 100
Apache	10	10	13
Safe	14	14	12
Zxing	13	10	17

In the selected number of features, the feature selection results indicate a change in the number of features in each different iteration. More iterations do not guarantee that the selected features will decrease or

increase. Meanwhile, the fitness value, the better the fitness value obtained, can affect the accuracy results obtained. After feature selection is performed, all resulted in a selected number of features not exceeding seventeen features out of the 26 features in the dataset, as seen in Table 4. In these feature data, the number of iterations does not affect the quantity of selected features. However, when viewed based on the frequency of selected features overall, it is as follows (Figure 4).

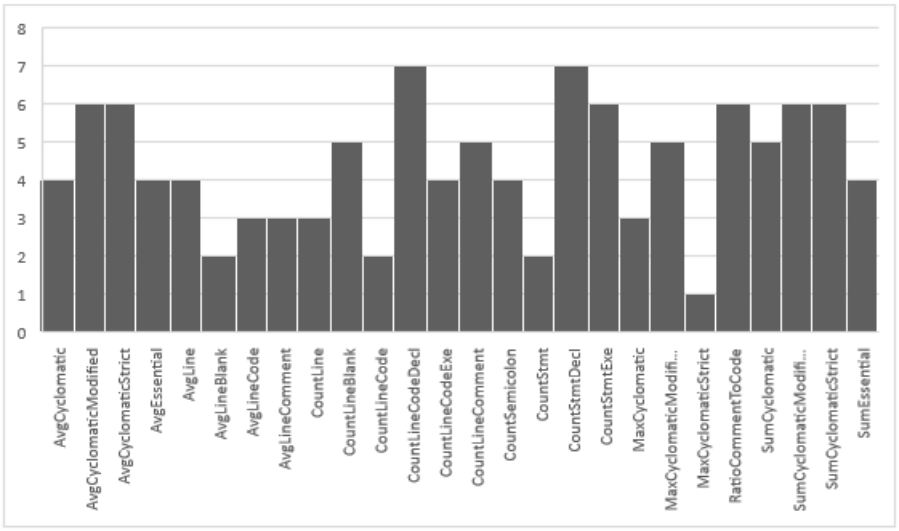

Figure 4. Selected Features

From the total of nine feature selection processes from each dataset and number of iterations, the graph above shows which features frequently appear and which ones rarely appear. The most frequently selected features are 'CountLineCodeDecl' and 'CountStmtDecl', which appear seven times out of nine feature selection processes. Meanwhile, the rarely selected feature is 'MaxCyclomaticStrict', which is only selected once, specifically in the Apache dataset with 100 iterations.

The accuracy results are compared between two different conditions that have been specified: the accuracy results of random forest classification without feature selection performed beforehand, and random forest classification with feature selection already conducted.

Table 5. Accuracy Results

Dataset	Accuracy Results Without Feature Selection	Accuracy Results with Feature Selection		
		iteration 10	iteration 50	iteration 100
Apache	74,3%	64,1%	69,2%	79,5%
Safe	83,3%	75%	83,3%	91,6%
Zxing	62,5%	71,25%	68,8%	75%
Average Accuracy	**73,37%**	**70,12%**	**73,7%**	**82,03%**

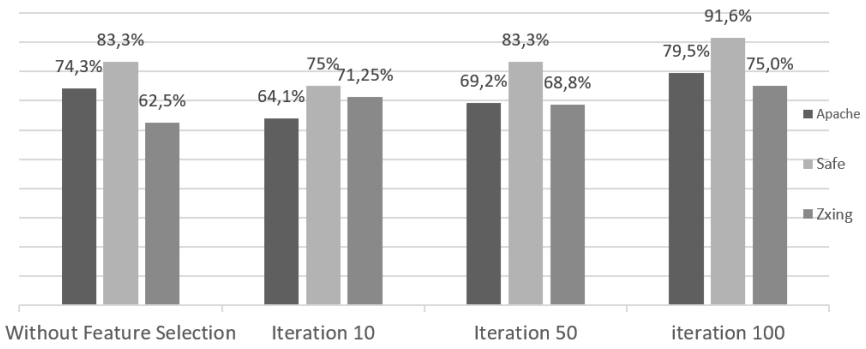

Figure 5. Accuracy results

Table 5 and Figure 5 show the overall results of the classification process. Feature selection was conducted in 3 different numbers of iterations, namely 10, 50, and 100. In terms of Accuracy evaluation results, it indicates an increase in accuracy with an increase in the number of iterations, except for the Zxing dataset which decreased when the number of iterations was 50, at 68.8%. The highest accuracy results for each dataset were achieved with 100 iterations. Accuracy of 79.5% for the Apache dataset, 91.6% for the Safe dataset, and 75% for the zxing dataset. Meanwhile, the overall average accuracy for each dataset in every iteration, the results with 100 iterations had the highest accuracy value at 82.03%.

CONCLUSION

The study found that the Artificial Bee Colony (ABC) algorithm can improve the performance of the Random Forest algorithm in software prediction tasks. The dataset Relink was used for testing, and the results showed an average accuracy of 82.03% after applying feature selection with the ABC algorithm. This is a significant improvement compared to the average accuracy of 73.37% before feature selection. The ABC algorithm was able to increase the performance of the Random Forest algorithm in predicting software defects.

REFERENCES

[1] N. Ichsan, H. Fatah, E. Ermawati, I. Indriyanti, and T. Wahyuni, "Integrasi Distribution Based Balance dan Teknik Ensemble Bagging Naive Bayes Untuk Prediksi Cacat Software," *Media J. Inform.*, vol. 14, no. 2, p. 79, 2022, doi: 10.35194/mji.v14i2.2623.

[2] E. Irawan and R. S. Wahono, "Penggunaan Random Under Sampling untuk Penanganan Ketidakseimbangan Kelas pada Prediksi Cacat Software Berbasis Neural Network," *J. Softw. Eng.*, vol. 1, no. 2, pp. 92–100, 2015.

[3] Sugiono, A. Taufik, and R. Faizal Amir, "Penerapan Penerapan Teknik Pso Over Sampling Dan Adaboost J48 Untuk Memprediksi Cacat Software," *J. Responsif Ris. Sains dan Inform.*, vol. 2, no. 2, pp. 198–203, 2020, doi: 10.51977/jti.v2i2.249.

[4] H. Aljamaan and A. Alazba, "Software defect prediction using tree-based ensembles," *PROMISE 2020 - Proc. 16th ACM Int. Conf. Predict. Model. Data Anal. Softw. Eng. Co-located with ESEC/FSE 2020*, pp. 1–10, 2020, doi: 10.1145/3416508.3417114.

[5] R. S. Wahono, "A Systematic Literature Review of Software Defect Prediction: Research Trends, Datasets, Methods and Frameworks," *J. Softw. Eng.*, vol. 1, no. 1, pp. 1–16, 2015.

[6] I. H. Laradji, M. Alshayeb, and L. Ghouti, "Software defect prediction using ensemble learning on selected features," *Inf. Softw. Technol.*, vol. 58, pp. 388–402, 2015, doi: 10.1016/j.infsof.2014.07.005.

[7] R. Supriyadi, W. Gata, N. Maulidah, and A. Fauzi, "Penerapan Algoritma Random Forest Untuk Menentukan Kualitas Anggur Merah," *E-Bisnis J. Ilm. Ekon. dan Bisnis*, vol. 13, no. 2, pp. 67–75, 2020, doi: 10.51903/e-bisnis.v13i2.247.

[8] A. Kustiyo, H. Firqiani, and E. Giri, "Seleksi Fitur Menggunakan Fast Correlation Based Filter pada Algoritma Voting Feature Intervals 5," *J. Ilm. Ilmu Komput.*, vol. 6, no. 2, p. 245184, 2008.

[9] A. O. Balogun *et al.*, "Impact of feature selection methods on the predictive performance of software defect prediction models: An extensive empirical study," *Symmetry (Basel).*, vol. 12, no. 7, 2020, doi: 10.3390/sym12071147.

[10] X. F. Song, Y. Zhang, Y. N. Guo, X. Y. Sun, and Y. L. Wang, "Variable-Size Cooperative Coevolutionary Particle Swarm Optimization for Feature Selection on High-Dimensional Data," *IEEE Trans. Evol. Comput.*, vol. 24, no. 5, pp. 882–895, 2020, doi: 10.1109/TEVC.2020.2968743.

[11] R. S. Wahono and N. Suryana, "Combining particle swarm optimization based feature selection and bagging technique for software defect prediction," *Int. J. Softw. Eng. its Appl.*, vol. 7, no. 5, pp. 153–166, 2013, doi: 10.14257/ijseia.2013.7.5.16.

[12] U. M. Khaire and R. Dhanalakshmi, "Stability of feature selection algorithm: A review," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 4, pp. 1060–1073, 2022, doi: 10.1016/j.jksuci.2019.06.012.

[13] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: Artificial bee colony (ABC) algorithm and applications," *Artif. Intell. Rev.*, vol. 42, no. 1, pp. 21–57, 2014, doi: 10.1007/s10462-012-9328-0.

[14] I. Irawan, "Peningkatan Performa Algoritma CART dengan Seleksi Fitur Menggunakan ABC untuk Penilaian Kredit," *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 8, no. 1, pp. 199–208, 2021, doi: 10.35957/jatisi.v8i1.553.

[15] P. D. Putra, S. Sukemi, and D. P. Rini, "Peningkatan Akurasi Klasifikasi Backpropagation Menggunakan Artificial Bee Colony dan K-NN Pada Penyakit Jantung," *J. Media Inform. Budidarma*, vol. 5, no. 1, p. 208, 2021, doi: 10.30865/mib.v5i1.2634.

[16] R. Wu, H. Zhang, S. Kim, and S. C. Cheung, "ReLink: Recovering links between bugs and changes," *SIGSOFT/FSE 2011 - Proc. 19th ACM SIGSOFT Symp. Found. Softw. Eng.*, pp. 15–25, 2011, doi: 10.1145/2025113.2025120.

[17] T. Zhou, X. Sun, X. Xia, B. Li, and X. Chen, "Improving defect prediction with deep forest," *Inf. Softw. Technol.*, vol. 114, no. June, pp. 204–216, 2019, doi: 10.1016/j.infsof.2019.07.003.

[18] E. Sutoyo and M. A. Fadlurrahman, "Penerapan SMOTE untuk Mengatasi Imbalance Class dalam Klasifikasi Television Advertisement Performance Rating Menggunakan Artificial Neural Network," *J. Edukasi dan Penelit. Inform.*, vol. 6, no. 3, p. 379, 2020, doi: 10.26418/jp.v6i3.42896.

[19] D. Karaboga and B. Basturk, "On the performance of artificial bee colony (ABC) algorithm," *Appl. Soft Comput. J.*, vol. 8, no. 1, pp. 687–697, 2008, doi: 10.1016/j.asoc.2007.05.007.

[20] H. Rao *et al.*, "Feature selection based on artificial bee colony and gradient boosting decision tree," *Appl. Soft Comput. J.*, vol. 74, pp. 634–642, 2019, doi: 10.1016/j.asoc.2018.10.036.

[21] M. Schiezaro and H. Pedrini, "Data feature selection based on Artificial Bee Colony algorithm," *EURASIP J. Image Video Process.*, vol. 2013, no. 1, pp. 1–8, 2013, doi: 10.1186/1687-5281-2013-47.

[22] L. Breiman, *Random Forest*, vol. 45. Kluwer Academic Publishers, 2001. doi: 10.1007/978-3-030-62008-0_35.

[23] W. Musu, A. Ibrahim, and Heriadi, "Pengaruh Komposisi Data Training dan Testing terhadap Akurasi Algoritma C4 . 5," *Pros. Semin. Ilm. Sist. Inf. Dan Teknol. Inf.*, vol. X, no. 1, pp. 186–195, 2021.

[24] L. Zahedi, F. Ghareh Mohammadi, and M. H. Amini, "A2BCF: An Automated ABC-Based Feature Selection Algorithm for Classification Models in an Education Application," *Appl. Sci.*, vol. 12, no. 7, 2022, doi: 10.3390/app12073553.