Increasing electrical grid stability classification performance using ensemble bagging of C4.5 and classification and regression trees

Firman Aziz[1], Armin Lawi[2]

[1]Department of Computer Science, Faculty of Mathematics and Natural Sciences, Universitas Pancasakti, Makassar, Indonesia
[2]Department of Information Systems, Faculty of Mathematics and Natural Sciences, Universitas Hasanuddin, Makassar, Indonesia

Article Info		ABSTRACT
		The increasing demand for electricity every year makes the electricity infrastructure approach the maximum threshold value, thus affecting the stability of the electricity network. The decentralized smart grid control (DSGC) system has succeeded in maintaining the stability of the electricity network with various assumptions. The data mining approach on the DSGC system shows that the decision tree algorithm provides new knowledge, however, its performance is not yet optimal. This paper poses an ensemble bagging algorithm to reinforce the performance of decision trees C4.5 and classification and regression trees (CART). To evaluate the classification performance, 10-fold cross-validation was used on the grid data. The results showed that the ensemble bagging algorithm succeeded in increasing the performance of both methods in terms of accuracy by 5.6% for C4.5 and 5.3% for CART.

Corresponding Author:

Armin Lawi
Department of Computer Science, Hasanuddin University
Perintis Kemerdekaan Street Km. 10, Makassar, South Sulawesi 90245, Indonesia
Email: armin@unhas.ac.id

1. INTRODUCTION

At present, global electricity demand is increasing every year. This makes the electrical infrastructure close to the maximum threshold so that it significantly affects the stability of the electricity network. Maintaining the electricity network stability requires a balance between production and consumption of electricity. This requires an integrated power generation system that can control the system by utilizing information and communication technology reliably and efficiently [1].

Smart grid is a modern electricity network system that integrates starting from generation, transmission equipment, and consumers of all users who are connected in the system to deliver electricity efficiently, sustainably, and economically [2] covering a variety of energy operations and measurements including smart meters, smart appliances, renewable energy resources, and energy-saving resources [3], [4]. The focus of the smart grid is on technical infrastructure [5] where electronic power conditioning, production control and electricity distribution are important aspects of the smart grid [3].

The decentralized smart grid control (DSGC) system proposed in [6] has succeeded in controlling electricity prices by switching to grid frequency so that it was available to all consumers and electricity producers. Then, the DSGC system is developed by conducting simulations with various assumptions about the stability of the electricity network [7]. One of them is subjecting consumer behavior in response to price changes that affect the grid stability. The results showed that the DSGC system supports a decentralized production system by providing a decrease in line capacities and average time compared to centralized

production. Data mining methods have been investigated in [8] by gathering various assumptions and identifying issues regarding the DSGC system. After the simulation process with various input values using the Kleijnen approach [9], it was found that the application of decision trees to the data generated gave new insights and resulted in an accuracy rate of 80%. Some ensemble research conducted by [10]–[13] with several cases finding that the ensemble technique succeeded in increasing the performance of a single classification in measuring accuracy, precision, and recall.

This paper proposes the application of a new algorithm in this case by performing an ensemble that is improving the performance of decision trees using bagging techniques. We have also experimented to implement classification and regression trees (CART) and ensemble classification and regression trees (CART) algorithms to compare our proposed algorithm with the criteria of splitting, pruning, noise handling, and other features.

2. RESEARCH METHOD
2.1. Decision tree C4.5 algorithm

Decision tree algorithm is the fundamental classifier model using tree graph or hierarchical structure. The main idea of decision tree is to transform data into a rooted-tree graph as the decision rules. Some stages in making a decision tree with the C4.5 algorithm is given as follows [14]–[16]:

a. Prepare training data that has been grouped or labeled into certain classes (e.g., stable and unstable classes).
b. The root of a tree is determined by computing the highest gain value (or the lowest entropy) of each attribute. The entropy of the attribute x of classes in C is computed using (1).

$$\text{Entropy}(x) = -\sum_{c \in C} p(c|x) \cdot \log p(c|x) \tag{1}$$

c. The gain value is calculated using (2).

$$\text{Gain}(x) = \text{Entropy}(x) - \sum_i \frac{N(x_i)}{N(x)} \cdot Entropy(x_i) \tag{2}$$

d. To calculate the gain ratio, we first need to know the Split Information using (3).

$$\text{SplitInformation}(x) = -\sum_i \frac{N(x_i)}{N(x)} \cdot \log_2\left(\frac{N(x_i)}{N(x)}\right) \tag{3}$$

e. Then, we can calculate the gain ratio using (4).

$$\text{GainRatio}(C, x) = \frac{\text{Gain}(C,x)}{\text{SplitInformation}(C,x)} \tag{4}$$

f. Repeat step 2 until all records are partitioned. The partition process will be stopped if, i) all pairs of records in node n are in the same class, ii) there are no more partitionable attributes in the record, and iii) there are no records in the empty branch.

2.2. Classification and regression trees (CART)

In the decision tree technique there are several methods, one of which is classification and regression trees (CART). CART explains the relationship between response variables with several predictor variables. The use of this method depends on the shape of the response variable. When the response variable is continuous, the regression trees method is used while the categorical form is used the classification trees method [17], [18]. CART classification tree consists of three stages that require learning sample L, namely selection of the selection, determination of terminal nodes, and labeling of each terminal node.

a. The first stage is the selection of sorters. Each sorting depends only on the value derived from one independent variable. For continuous independent variables Xj with sample space of size n and there are n different sample observation values, then there will be n-1 different sorting. Whereas for Xj is the nominal category variable with L level, 2L - 1 -1 will be obtained. But if the Xj variable is an ordinal category, L-1 might be obtained as possible. The sorting method that is often used is the Gini index with the functions:

$$i(t) = \sum_{i \neq j} p(i|t)p(j|t), \tag{5}$$

where $i(t)$ is the heterogeneous function of the Gini index, $p(i | t)$ is the proportion of class i at node t, and $p(j | t)$ is the proportion of class j at node t. Goodness of split is an evaluation of sorting by sorting s at node t. Goodness of split $\emptyset(s,t)$ is defined as a decrease in heterogeneity.

$$\emptyset(s,t) = \Delta i(s,t) = i(t) - P_L i(t_L) - P_R i(t_R). \tag{6}$$

The tree development is carried out by searching for all possible sorters at node t_1 so that a s^* sorter is found which gives the highest heterogeneity reduction value, namely:

$$\Delta i(s^*, t_1) = \max_{s \in S} \Delta i(s, t_1), \tag{7}$$

where $\emptyset(s,t)$ is the goodness of split criterion, $P_L i(t_L)$ and $P_R i(t_R)$ are the proportion of observations from node t to the left node and to the right node, respectively.
b. The second step is determining the terminal node. Node t can be used as a terminal node if there is no significant decrease in heterogeneity in sorting, there is only one observation (n = 1) at each child node or there is a minimum limit of n and a limit on the number of levels or the maximum level of tree depth.
c. The third stage is labeling each terminal node based on the rule for the highest number of class members, namely:

$$p(j_0|t) = \max_j p(j|t) = \max_j \frac{N_j(t)}{N(t)}, \tag{8}$$

where $p(j | t)$ is the proportion of class j at node t, $N_j(t)$ is the number of observations of class j at node t, and $N(t)$ is the number of observations at node t. The terminal node class label t is j_0, which gives the largest estimated error in classifying node t.

The process of forming a classification tree stops when there is only one observation in each child node or there is a minimum limit of n, all observations in each child node are identical, and there is a limit on the number of levels or maximum tree depth. After the maximum tree formation, the next stage is tree pruning to prevent the formation of very large and complex classification trees, in order to obtain an appropriate tree size based on cost complexity pruning, then the magnitude of the resubstitution estimate of the T tree on the complexity parameter α is:

$$R_\alpha(T) = R(T) + \alpha|\bar{T}|, \tag{9}$$

where $R_\alpha(T)$ is the resubstitution of a T tree at complexity α, $R(T)$ is the resubstitution estimate, α is the cost-complexity parameter for adding one final node to the T tree, and $|\bar{T}|$ is the number of terminal vertices of the T tree.

The pruning cost complexity determines the subtree $T(\alpha)$ that minimizes $R_\alpha(T)$ in all part trees for each α value. The value of the complexity parameter α will slowly increase during the trimming process. Next, to look for the subtree $T(\alpha) < T_{\max}$ that can minimize $R_\alpha(T)$, i.e.:

$$R_\alpha\big(\cdots \big((T)\big) \cdots \big) = \min_{T < T_{\max}} R_\alpha(T). \tag{10}$$

After pruning the optimal classification tree is obtained which is simple in size but provides a fairly small replacement value.

2.3. Bagging

Bagging is the earliest and simplest ensemble-based algorithm, but it is very effective. It combines several sets of classifier models to strengthen the weak classification results. Bagging overcomes the instability of complex models with relatively small datasets. Pasting small vote is a bagging variant for handling large datasets by dividing them into smaller segments. A process called bites trains these segments to build independent classifiers and then combines them with a majority vote [19]. Ensemble bagging algorithm works [20]:
a. Enter the training sample order $(x_1: y_1), \ldots, (x_n: y_n)$ with the label $y \in Y = (-1, 1)$.
b. Initialize the probability of each instance in the learning set $D_1(i) = \frac{1}{n}$ and $t = 1$.
c. The iteration process where $t < B = 100$ is a member of the ensemble
   − The training is in form of n sets with replacement sampling where t in the D_t distribution

- Determine hypothesis, $h_t: X \rightarrow Y$
- Set $t = t + 1$

End the loop

d. The final hypothesis ensemble

$$C^*(x_i) = h_{final}(x_1) = argmax \sum_{t=1}^{B} I(C_t(x) = y). \tag{11}$$

2.4. Boosting

Boosting is an effective method to build an accurate classifier by combining weak classifiers [21]. One of the popular boosting methods used is adaptive boosting (AdaBoost). AdaBoost trains the basic classifier iteratively using training data with weight coefficients that depend on the performance of the classifier in the previous iteration, which gives greater weight to the misclassified data. If the classifier has been set to be trained, then all the classifiers will be combined to form a final decision on the model that shows the best performance [22].

2.5. Random forest

Random forest is a classification algorithm used for large amounts of data because the classification accuracy results depend on the number of trees [23]. The combination of tree formations is done randomly. The random forest procedure [24], [25]: i) the process of taking a random sample of size *n* with returns. This stage is the bootstrap stage; ii) using a bootstrap sample, the tree is constructed until it reaches its maximum size (without pruning). Tree construction is done by applying random feature selection to each selection process, where k explanatory variables are chosen randomly; and iii) repeat steps 1 and 2, forming a forest consisting of several trees.

2.6. Performance evaluation

The performance of the proposed classifier method was evaluated using a confusion matrix. Table 1 describes performance measures such as precision, recall, and accuracy. The measurement results are obtained using the predicted and actual values of a class [26], [27].

Table 1. Confusion matrix

	Predicted: Stable	Predicted: Unstable	Recall
Actual: Stable	True Stable (TS)	False Unstable (FU)	**TS / (TS + FU)**
Actual: Unstable	False Stable (FS)	True Unstable (TU)	**TU / (FS + TU)**
Precision	**TS / (TS +FS)**	**TU / (FU + TU)**	**Accuracy = (TS + TU) / N***

*N is the number of testing data, i.e., N = TS + FU + FS + TU

3. EXPERIMENTAL
3.1. Dataset

We use the benchmark electrical grid stability simulated dataset obtained from the UCI machine learning repository so that our results can be compared with other methods. The data label is the system stability with predictors consist of 11 predictive features and 1 composite (P1) as described in Table 2. The total data is 9,999 records with 6,379 represents stable class and 3,620 unstable. Class stability of dataset is illustrated in Figure 1.

Table 2. Description of electrical grid stability simulated data set

| Variable	Attribute	Description
Response Variable	Y	Label of the system stability. (Categorical data type: 0 = Unstable; 1 = Stable)
Predictor Variable	Tau1	Reaction time of participant (data type: real from the range [0.5, 10]s).
	Tau2	Tau1 - the value for electricity producer.
	Tau3	
	Tau4	
	P1	Nominal power consumed (negative)/produced (positive) (data type: real).
	P2	For consumers from the range [-0.5,-2]s^-2;
	P3	P1 = abs(P2 + P3 + P4)
	P4	
	G1	Coefficient (gamma) proportional to price elasticity (data type: real from the range
	G2	[0.05, 1]s^-1).
	G3	G1 - the value for electricity producer.
	G4	
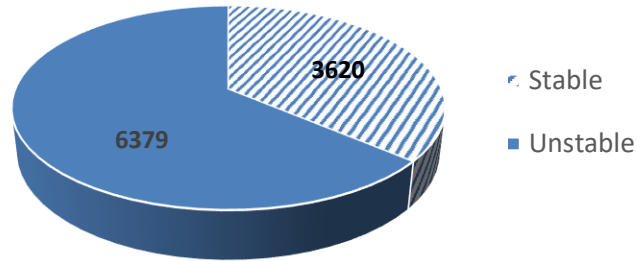
Figure 1. Stability system data set

3.2. Data partition

The total data used is 9,999. The dataset is then partitioned into 6,999 training data for building model and 3,000 testing data for performance evaluation. Stratified random strategy is used for data partition with portion of 70% training data and 30% testing data as given in Table 3.

3.3. Parameter setting

The experiment uses the default parameters of the algorithm. Determination of each of these parameters to obtain fair results on all classifiers of the decision tree. Parameter value settings are given in Table 4.

Table 3. Training and testing data composition

Class Partition	Stable	Unstable	Total
Training	4,432	2,567	**6,999**
Testing	1,947	1,053	**3,000**
Total	**6,379**	**3,620**	**9,999**

Table 4. Parameter setting of the experiment

Parameter	Value	Parameter	Value
criterion_C45	entropy	max_features	None
criterion_CART	Gini	random_state	None
Splitter	best	max_leaf_nodes	None
max_depth	None	min_impurity_decrease	0
min_samples_split	2	min_impurity_split	None
min_samples_leaf	1	class_weight	None
min_weight_fraction_leaf	0	ccp_alpha	0

4.   RESULTS

The performance of the experiment results is evaluated using confusion matrix as the basis for all metrics, i.e., accuracy, recall and precision. For the sake of simplicity, performance metrics are included in the confusion matrix to easily check their values. Tables 5 and 6 showed the performance results for C4.5 and CART decision trees, respectively, with their ensembled classifiers.

Table 5. Confusion matrices for decision tree C4.5 and its ensembled classifiers

Classifier		Stable	Unstable	Recall
C4.5	Stable	1701	246	**87.00%**
	Unstable	215	838	**80.00%**
	Precision	**89.00%**	**77.00%**	**84.63%**
Bagging C4.5	Stable	1848	99	**95.00%**
	Unstable	196	857	**81.00%**
	Precision	**90.00%**	**90.00%**	*90.16%*
Adaboost C4.5	Stable	1773	174	**91.00%**
	Unstable	250	803	**76.00%**
	Precision	**88.00%**	**82.00%**	**85.86%**
Random Forest C4.5	Stable	1845	102	**95.00%**
	Unstable	238	815	**77.00%**
	Precision	**89.00%**	**89.00%**	**88.66%**

Table 6. Confusion matrices for CART and its ensembled classifiers

Classifier		Stable	Unstable	Recall
CART	Stable	1700	247	**87.00%**
	Unstable	222	831	**79.00%**
	Precision	**88.00%**	**77.00%**	*84.36%*
Bagging CART	Stable	1850	97	**95.00%**
	Unstable	213	840	**80.00%**
	Precision	**90.00%**	**90.00%**	*89.66%*
Adaboost CART	Stable	1773	174	**91.00%**
	Unstable	250	803	**76.00%**
	Precision	**88.00%**	**82.00%**	*85.86%*
Random forest CART	Stable	1846	101	**95.00%**
	Unstable	245	808	**77.00%**
	Precision	**88.00%**	**89.00%**	*88.46%*

Figure 2 shows that the ensemble bagging method proposed to improve the performance of the Decision Trees C4.5 and CART methods gives the best performance results among other ensemble methods. The bagging ensemble succeeded in increasing the accuracy of decision trees C4.5 by 5.6% and CART by 5.3% as well as increasing recall values for the stable and unstable classes, in contrast to the adaboost and random forest ensembles which experienced a decrease in recall values for the stable class as shown in Figures 3(a) and 3(b). Figures 4(a) and 4(b) show that the bagging ensemble provides significant performance by improving the accuracy of the decision trees C4.5 and CART models in classifying stable and unstable classes which result in higher precision values among other ensemble methods.
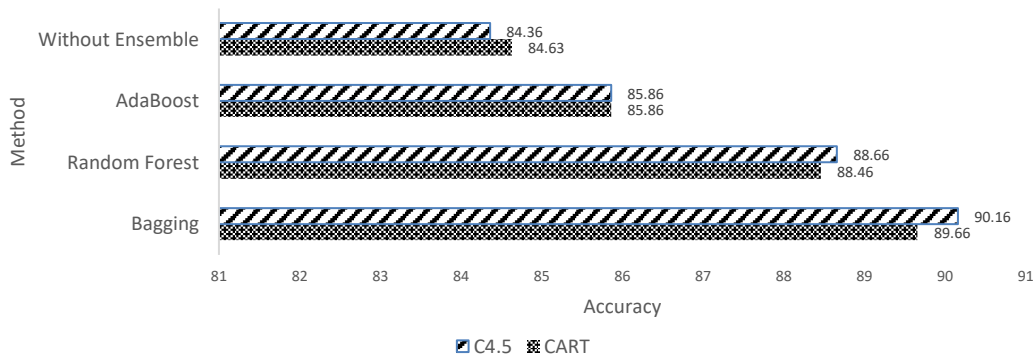

Figure 2. Accuracy comparison of decision trees C4.5 and CART with their ensembles
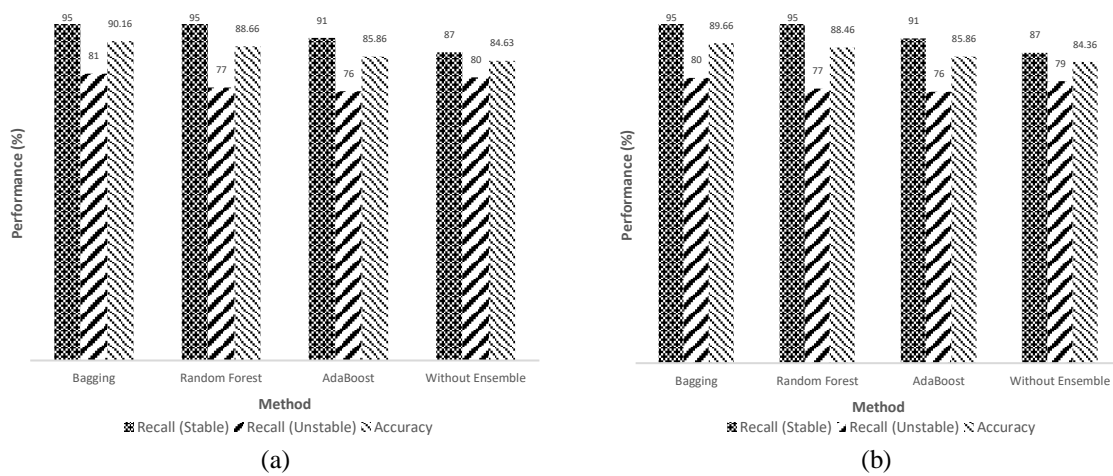

(a)   (b)

Figure 3. Comparison of recall performances for stable and unstable actual labeled data that contributes to the actual value of accuracy for both decision trees C4.5 and CART algorithms (a) recalls and accuracy of decision tree and (b) recalls and accuracy of CART
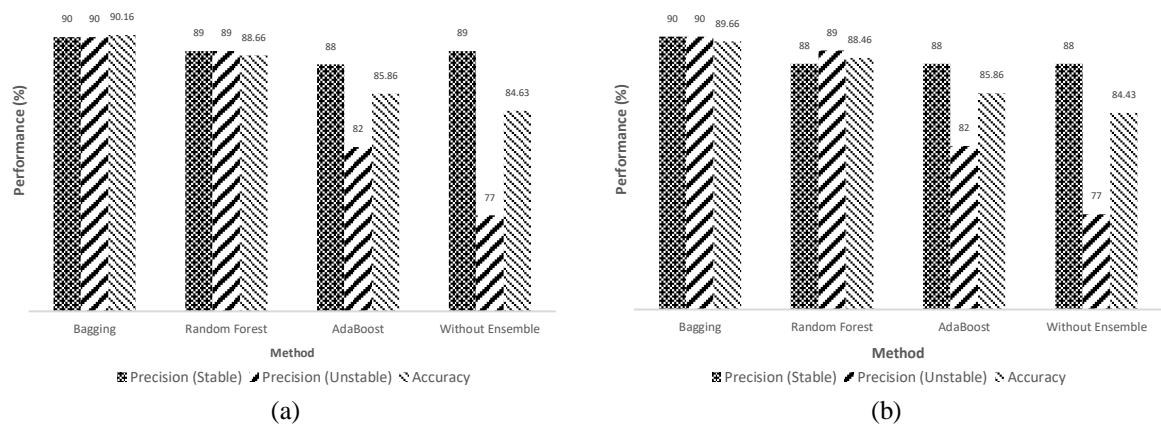
Figure 4. Comparison of precision performances for stable and unstable labeled data that contributes to the prediction value of accuracy for both decision trees C4.5 and CART algorithms (a) precisions and accuracy of decision tree and (b) precisions and accuracy of CART

5.　CONCLUSION

In this paper, we have proposed an ensemble bagging technique to reinforce the performance of the decision tree algorithms of C4.5 and CART Dataset consists of 12 features with a total of 9,999 records. The data was splitted into 70% as for training data and 30% for testing data. The experiment results showed that the proposed bagging succeeded in improving performance by correcting the misclassifications of the original decision tree classifier C4.5 with 90.16% accuracy, which increases about 5.6%. Bagging C4.5 also has better performance compared to Bagging-CART which only produces an accuracy of 89.66%. Although the experimental evaluation result of the Bagging C4.5 showed a superior performance achievement by successfully increasing the accuracy, this is only in one data partition. In the future, it is interested to investigate the performance of the Bagging C4.5 in various data partitions.

REFERENCES

[1]　J. B. Ekanayake, N. Jenkins, K. Liyanage, J. Wu, and A. Yokoyama, *Smart grid: technology and applications*. Wiley, 2012.
[2]　P. Kacejko, S. Adamek, and M. Wydra, "Optimal voltage control in distribution networks with dispersed generation," in *2010 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT Europe)*, Oct. 2010, pp. 1–4, doi: 10.1109/ISGTEUROPE.2010.5638970.
[3]　M. P. Lee, "Assessment of demand response & advanced metering," Staff Report, 2012.
[4]　M. S. Saleh, A. Althaibani, Y. Esa, Y. Mhandi, and A. A. Mohamed, "Impact of clustering microgrids on their stability and resilience during blackouts," *Proceedings - 2015 International Conference on Smart Grid and Clean Energy Technologies, ICSGCE 2015*, pp. 195–200, Apr. 2016, doi: 10.1109/ICSGCE.2015.7454295.
[5]　J. Torriti, "Demand side management for the European supergrid: occupancy variances of European single-person households," *Energy Policy*, vol. 44, pp. 199–206, May 2012, doi: 10.1016/J.ENPOL.2012.01.039.
[6]　B. Schäfer, M. Matthiae, M. Timme, and D. Witthaut, "Decentral smart grid control," *New Journal of Physics*, vol. 17, no. 1, Jan. 2015, doi: 10.1088/1367-2630/17/1/015002.
[7]　B. Schäfer, C. Grabow, S. Auer, J. Kurths, D. Witthaut, and M. Timme, "Taming instabilities in power grid networks by decentralized control," *The European Physical Journal Special Topics*, vol. 225, no. 3, pp. 569–582, May 2016, doi: 10.1140/epjst/e2015-50136-y.
[8]　V. Arzamasov, K. Bohm, and P. Jochem, "Towards concise models of grid stability," *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids, SmartGridComm 2018*, Dec. 2018, doi: 10.1109/SMARTGRIDCOMM.2018.8587498.
[9]　J. P. C. Kleijnen, "Design and analysis of simulation experiments," *Springer Proceedings in Mathematics and Statistics*, vol. 231, pp. 3–22, 2015, doi: 10.1007/978-3-319-76035-3_1.
[10]　A. Lawi, F. Aziz, and S. Syarif, "Ensemble GradientBoost for increasing classification accuracy of credit scoring," *Proceedings of the 2017 4th International Conference on Computer Applications and Information Processing Technology, CAIPT 2017*, pp. 1–4, Mar. 2018, doi: 10.1109/CAIPT.2017.8320700.
[11]　N. Hardiyanti, A. Lawi, Diaraya, and F. Aziz, "Classification of human activity based on sensor accelerometer and gyroscope using ensemble SVM method," *Proceedings - 2nd East Indonesia Conference on Computer and Information Technology: Internet of Things for Industry, EIConCIT 2018*, pp. 304–307, Nov. 2018, doi: 10.1109/EICONCIT.2018.8878627.
[12]　A. Lawi and F. Aziz, "Comparison of classification algorithms of the autism spectrum disorder diagnosis," in *2018 2nd East Indonesia Conference on Computer and Information Technology (EIConCIT)*, Nov. 2018, pp. 218–222, doi: 10.1109/EIConCIT.2018.8878593.
[13]　A. Lawi, F. Aziz, and S. L. Wungo, "Increasing accuracy of classification physical activity based on smartphone using ensemble logistic regression with boosting method," *Journal of Physics: Conference Series*, vol. 1341, no. 4, p. 042002, Oct. 2019, doi: 10.1088/1742-6596/1341/4/042002.

[14] M. I. Jordan, "A statistical approach to decision tree modeling," in *Machine Learning Proceedings 1994*, Elsevier, 1994, pp. 363–370.

[15] R. Kohavi, R. Kohavi, and R. Quinlan, "Decision tree discovery," *In Handbook of Data Mining and Knowledge Discovery*, vol. 6, pp. 267--276, 1999.

[16] S. Suthaharan, "Machine learning models and algorithms for big data classification," *Integrated Series in Information Systems*, vol. 36, p. 364, 2016.

[17] Y. Yohannes and P. Webb, *Classification and regression trees, CART: a user manual for identifying indicators of vulnerability to famine and chronic food insecurity*. Intl Food Policy Research Inst; Pap/Dsk edition, 1999.

[18] F. Ferda, "Classification and regression trees (CART) theory and applications," CASE - Center of Applied Statistics and Economics Humboldt University, 2004.

[19] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, Aug. 1996, doi: 10.1007/BF00058655.

[20] T. Report, A. Classifiers, L. Breiman, and C. Berkeley, "Bias, variance, and arcing classifier," *citeseerx*, pp. 1–22, 1996.

[21] R. E. Schapire, *Boosting: foundations and algorithms*, vol. 42, no. 1. The MIT Press; Illustrated edition, 2013.

[22] B. Kégl and R. Busa-Fekete, "Boosting products of base classifiers," in *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09*, 2009, pp. 1–8, doi: 10.1145/1553374.1553439.

[23] T. Shi and S. Horvath, "Unsupervised learning with random forest predictors," *Journal of Computational and Graphical Statistics*, vol. 15, no. 1, pp. 118–138, Mar. 2006, doi: 10.1198/106186006X94072.

[24] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–23, 2001, doi: 10.1023/A:1010950718922.

[25] A. Cutler, D. R. Cutler, and J. R. Stevens, "Random forests," *Ensemble Machine Learning*, pp. 157–175, 2012, doi: 10.1007/978-1-4419-9326-7_5.

[26] B. P. Salmon, W. Kleynhans, C. P. Schwegmann, and J. C. Olivier, "Proper comparison among methods using a confusion matrix," *International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 3057–3060, Nov. 2015, doi: 10.1109/IGARSS.2015.7326461.

[27] F. Provost and R. Kohavi, "Guest editors' introduction: on applied research in machine learning," *Machine Learning*, vol. 30, pp. 127–132, 1998, doi: 10.1023/A:1007442505281.

BIOGRAPHIES OF AUTHORS

Armin Lawi 🆔 🅶 ᶜ ℙ is an Associate Professor (Lektor Kepala) of the Department of Information Systems in the Faculty of Mathematics and Natural Sciences at Hasanuddin University. He received Bachelor's degree in Mathematics at Hasanuddin University, Master degree in Computer Science and Communication Engineering from Kyushu University, and Ph.D. in Computer Science and System Engineering from Kyushu Institute of Technology, Japan. He can be contacted by email: armin@unhas.ac.id.

Firman Aziz 🆔 🅶 ᶜ ℙ is the Head of the Study Program of Computer Science at Universitas Pancasakti Makassar. He received bachelor's degree in Informatics Technology at Universitas Islam Makassar and master's degree in electrical engineering with concentration Informatics Technology at Universitas Hasanuddin Makassar. He can be contacted by email: firman.aziz@unpacti.ac.id.