Depth Limitation and Splitting Criteria Optimization on Random Forest for Efficient Human Activity Classification

Syarif Hidayat[1], Ahmad Ashari[2] *, Agfianto Eko Putra[3]
Department of Informatics, Faculty of Industrial Technology, Universitas Islam Indonesia, Yogyakarta, Indonesia
Department of Computer Science and Electronics, Faculty of Mathematics and Natural Sciences
Universitas Gadjah Mada, Indonesia, Yogyakarta, Indonesia

Abstract—**Random Forest (RF) is known as one of the best classifiers in many fields. They are parallelizable, fast to train and to predict, robust to outlier, handle unbalanced data, have low bias, and moderate variance. Apart from these advantages, there are still opportunities to increase RF efficiency. The absence of recommendations regarding the number of trees involved in RF ensembles could make the number of trees very large. This can increase the computational complexity of RF. Recommendations for not pruning the decision tree further aggravates the condition. This research attempts to build an efficient RF ensemble while maintaining its accuracy, especially in problem activity. Data collection is performed using an accelerometer sensor on a smartphone device. The data used in this research are collected from five peoples who perform 11 different activities. Each activity is carried out five times to enrich the data. This study uses two steps to improve the efficiency of the classification of the activity: 1) Optimal splitting criteria for activity classification, 2) Measured pruning to limit the tree depth in RF ensemble. The first method in this study can be applied to determine the splitting criteria that are most suitable for the classification problem of activities using Random Forest. In this case, the decision model built using the Gini Index can produce the highest accuracy. The second method proposed in this research successfully builds less complex pruned-tree without reducing its classification accuracy. The research results showed that the method applied to the Random Forest in this study was able to produce a decision model that was simple but yet accurate to classify activity.**

Keywords—Activity; accuracy; classification; fall; optimization; random forest

I. Introduction

Nowadays, there were researches in the field of activity classification and fall detection due to the development of mobile [1] and wearable device [2]. It promised an important role in improving human life quality. Among its application are healthcare, security, work safety [3], [4]. There were several techniques that could be utilized in the activity classification and fall detection system. Khojasteh et al. use a rule-based system to decrease computational cost [5]. Fall detection could also be solved using threshold-based algorithms [6]–[8]. Meanwhile, others trying to make use of machine learning algorithms to increase the accuracy of fall detection [9]–[11]. Aziz et al. compared the accuracy of the two methods in an experiment involving ten participants. The outcomes demonstrate that the general performance of falls detection of the five machine learning was superior to the performance of five threshold-based methods. Likewise, the testing of the five machine learning demonstrates Support Vector Machine (SVM) as the best performer notably when sensitivity and specificity measure combined [12]. Indeed, current states of the art in Machine Learning are Random Forest and Support Vector Machine [14]. However, Support Vector Machine is more suitable in the case of two class problem [13]. Furthermore, Support Vector Machine tends to work best in a situation where data are reasonably clean with a few outliers. Random Forest generally outperforms Support Vector Machine in many class cases with many outliers to be expected. Research on fall detection which becomes part of human activity classification research needs to identify several activities. Moreover, the accelerometer data generated from human activity could be very noisy [15]. Hence, Random Forest will likely to be more suitable in this case as it needs to classify several classes.

Random Forest is essentially a group classifier that comprises of several decision trees. Those trees then vote to get the final prediction result. It is one of the best-recognized ensemble methods. It could solve the classification task as well as regression task [16].

As Random Forest consisted of several decision trees, it shares the same traits which are bias, variance and overfitting as if in decision trees. The decision model will produce high accuracy if tested on training data. This is also known as the low bias term. However, when the resulting model is tested on testing data that has never been seen before, the accuracy is low. This is referred to as high variance. Supposedly, a good model must be able to produce high accuracy in both training and testing data. Random Forest will likely result in better model stability as it capable of suppressing variance while maintaining bias.

Random Forest will have best performance if the decision tree produces high accuracy (low bias) from the start. Splitting criterion is one of the most influencing factors in decision tree accuracy [18]. Therefore, this study will investigate the most fitting splitting criteria to produce a classification model with highest accuracy.

*Corresponding Authors Email : ashari@ugm.ac.id

Random Forest will build unpruned trees [19]. Unpruned trees will produce a model with high accuracy (low bias). However, this will make the generalization ability low (high variance). This research aim for classification model that produce high accuracy (low bias) while still having high generalization ability. As the trees accuracy seems to be not increased after a certain depth, this research proposes optimal depth limitation as a mean of pruning in Random Forest decision tree.

Therefore this research is answering the question of which splitting criterion will make the most accurate trees (resulting in low bias model) and the optimal depth of the tree to produce highest generalization model.

II. Theoretical Framework

A. Random Forest

Fall activity would be classified using the legacy random forest where a number of decision trees are constructed as sampling data using bootstrap and some randomly selected features. Classification by group of trees in Random Forest work by voting a class after each tree in the group make a classification. Random Forest will choose the class which is supported by most of the trees. Fundamentally, data classification techniques using Random Forest works as follows:

1) Assume that the number of the original training data record is A.

2) Perform bootstrapping on original data by sampling A into a which are chosen randomly with replacement such as a<A.

3) Perform the bootstrapping for n time to create training data from n trees.

4) Given some feature/predictor is B, select of b variable at random such as b<B for each sub-sample created before.

5) Build decision trees for each sub-sample data by splitting a node using the best split on the n predictor.

6) Grow tree as large as possible **with no pruning**.

7) Make a classification by voting the classification result of n trees. The majority class will be selected as the Random Forest classification result.

The decision trees vote to classify activity. This research tried to understand the effect of decision tree depth and the splitting criterion by performing classification on trees with the depth of 5, 10, 15, and 20.

B. Splitting Criteria

Decision Tree used several measures for selecting best split based on impurity measures. Decision tree tried to split the nodes on all available predictor and choose the best splitter which able to produce the most similar sub-nodes. This step resulted in sub-nodes with higher homogeneity compared to original nodes. There are three most commonly used measures in decision tree splitting:

1) Information gain (IG): Information Gain provides an overview of how much information the feature provides in determining the class. When a feature highly determines a class, the value of information gain will be maximal. On the other hand, a feature that does not correspond to class determination will likely give no information [20]. IG provides an overview of the relation of the predictor to a class by measuring the reduction in entropy value. Entropy gives an overview of class impurity of several data records. Entropy is a measure of impurity in an arbitrary collection of examples. When the node is less impure, the information to describe it is lesser. On the contrary, the more impure node will likely give more information. Entropy function is expressed by (1).

$$H = \sum_{i=1}^{k} p_k \log p_k \qquad (1)$$

Information Gain is described by (2).

$$\Delta H = H - \frac{m_L}{m} H_L - \frac{m_R}{m} H_R \qquad (2)$$

2) Gain ratio: Information gain has a relatively high bias on high branching features. Gain ration modifies information gain so that bias could be reduced when applied on high branching features. Feature selection is taking into account the number and size of branches [20]. Information gain was normalized by intrinsic information of a split. Intrinsic information could be described as the number of information needed to decide on a node to classify a record. It gives an overview of how much information could be acquired whenever dataset split into i partitions. Intrinsic information could be described by (3).

$$D = \sum_{j=1}^{i} \frac{|Dj|}{D} \cdot log_2 \left(\frac{|Dj|}{D}\right) \qquad (3)$$

Intrinsic information with higher value resulted make the size of sub-sample that is generated during splitting relatively to be the same. On the other hand, less intrinsic information resulted in few sub-sample that contain most of the data record. Gain ratio will select feature that generates a maximum gain ratio. Gain ratio could be described by (4).

$$Gain\ Ratio(F) = \frac{Gain\ (F)}{Intrinsic\ Info(F)} \qquad (4)$$

3) Gini index: Gini index shows the number of randomly picked data that is incorrectly labeled. It reaches its maximum value on heterogeneous data [21]. Consequently, it gets a minimum value on similar data. Gini Index could be described by (5).

$$Gini = 1 - \sum_{i=1}^{k} (p_k)^2 \qquad (5)$$

III. Materials

The research data was obtained through accelerometer sensor readings from 5 respondents. Each respondent performed 11 different activities. Each activity was repeated five times to increase the variety of data. There are seven attribute information on the data recorded. Table I contains an example of one data.

Accelerometer data type in Table I could be explained as follows:

TABLE. I. DATA DESCRIPTION

Type	Value
Sequence Name	A01
Timestamp	633790226053172000
Date	27.05.2009 14:03:25:317
X	4.373908043
Y	1.887959599
Z	0.769018948
Activity	walking

a) Sequence Name: It contains person and repetition code. Sensor readings of five respondents are marked with the letters A to E. Repetition activities marked with the number 01 to 05. For example, C05 shows the results of the readings from the third respondent in the fifth repetition.

b) Timestamp: Each data is given a timestamp to mark the time of the accelerometer sensor readings.

c) Date

d) X shows accelerometer sensor reading on axis X

e) Y shows accelerometer sensor reading on axis Y

f) Z shows accelerometer sensor reading on axis Z

g) Activity: There are 11 activities which are 1) walking, 2) falling, 3) sitting, 4) sitting down, 5) lying, 6) lying down, 7)sitting on the ground, 8) on all fours, 9) standing up from sitting on the ground ,10) standing up from sitting, and 11) standing up from lying.

IV. METHODOLOGY

This research follows methodology including a) Data pre-processing, b) Feature extraction, c) Splitting-Criteria Optimization, d) Tree-Depth Optimization, and e) Validation.

A. Data Pre-Processing

The data used in the study needs to be pre-processed so that it can be in accordance with the context of this study. The length of the data for each activity is different. Thus, the information regarding the minimum data of accelerometer record needed to be able to recognize an activity is necessary. This information will be used as a base reference for what is called the data window. Activities with the minimum record will only be represented by one data while activities with more data lengths than data window will be divided into several data. In this research, fall activities have the minimum data representation. The minimum data for fall activities is 17 data. Thus, all other activities data would be windowed by this number.

B. Feature Extraction

There are a number of accelerometer feature. It is important to extract the right features in order to be able to classify activity efficiently. Pannurat et al summarize 36 Accelerometer Feature to detect activities including fall [22]. This research extracts 21 features from the data which are:

1) Mean (μ)*:* This feature is informative in classifying static activities such as sitting and lying. This feature is extracted for each axis. Equation (6) used to obtain this feature.

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i \qquad (6)$$

where x = accelerometer data on each axis, *i* = the data index, and N = number of data samples.

2) Standard deviation (σ*):* This feature is useful for classifying dynamic activities such as walking and running. Equation (7) used to obtain this feature.

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \qquad (7)$$

3) Variance (σ^2*):* This feature is calculated to measure the spread between accelerometer data in each axis. Equation (8) used to obtain this feature.

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2 \qquad (8)$$

4) Standard deviation magnitude ($|\sigma|$*):* This feature measure the spread between the combination of accelerometer data on all axis. Equation (9) used to obtain this feature.

$$|\sigma| = \sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_z^2} \qquad (9)$$

5) Sum vector magnitude ($|a|$*):* This feature is useful for detecting abnormal activities such as falling. However, this feature alone is not enough to detect falls because jumping activity also results in sudden changes. Equation (10) used to obtain this feature.

$$|a| = \sqrt{a_x^2 + a_y^2 + a_z^2} \qquad (10)$$

where a_x, a_y. and a_z denote the accelerometer value on the x, y, and z axis

6) Standard deviation of sum vector magnitude ($\sigma_{|a|}$*):* This feature measures the spread between the sum vector magnitudes values previously calculated. Equation (11) used to obtain this feature.

$$\sigma_{|a|} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(|a|_i - \mu_{|a|}\right)^2} \qquad (11)$$

7) Sum vector magnitude on horizontal plane ($|a|_h$*):* Instead of counting sum vectors from the entire axis, this feature only observes the sum vector in the horizontal plane between axis x and y. Equation (12) used to obtain this feature.

$$|a|_h = \sqrt{a_x^2 + a_y^2} \qquad (12)$$

8) Sum vector magnitude on vertical plane ($|a|_v$*):* This feature only observes the sum vector in the vertical plane between axis x and z. Equation (13) used to obtain this feature.

$$|a|_v = \sqrt{a_x^2 + a_z^2} \qquad (13)$$

9) Standard deviation of sum vector magnitude on horizontal plane ($\sigma_{|a|_h}$*):* This feature analyzes the distribution of the data from the previous feature vector sum magnitude. Equation (14) used to obtain this feature.

$$\sigma_{|a|_h} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(|a|_{h_i} - \mu_{|a|_h}\right)^2} \qquad (14)$$

10)Energy on axis X (E_x): This feature represents the change of acceleration at each measurement in a single window on the axis X. Equation (15) used to obtain this feature.

$$\mathrm{E}_x = \sqrt{\frac{1}{w-1}\sum_{i=1}^{w-1}|x_{i+1} - x_i|} \qquad (15)$$

where w is the window size.

11)Energy on axis Y (E_y): This feature represents the change of acceleration at each measurement in a single window on the axis Y. Equation (16) used to obtain this feature.

$$\mathrm{E}_y = \sqrt{\frac{1}{w-1}\sum_{i=1}^{w-1}|y_{i+1} - y_i|} \qquad (16)$$

12)Energy on axis Z (E_z): This feature represents the change of acceleration at each measurement in a single window on the axis Z. Equation (17) used to obtain this feature.

$$\mathrm{E}_z = \sqrt{\frac{1}{w-1}\sum_{i=1}^{w-1}|z_{i+1} - z_i|} \qquad (17)$$

13)Energy XY (E_{xy}): This feature represents the change of acceleration at each measurement in a single window on the plane X and Y. Equation (18) used to obtain this feature.

$$\mathrm{E}_{xy} = \sqrt{\frac{1}{w-1}\sum_{i=1}^{w-1}\left|\sqrt{{x_i}^2 + {y_i}^2}_{i+1} - \sqrt{{x_i}^2 + {y_i}^2}_{i}\right|} \qquad (18)$$

14)Energy YZ (E_{yz}): This feature represents the change of acceleration at each measurement in a single window on the plane Y and Z. Equation (19) used to obtain this feature.

$$\mathrm{E}_{yz} = \sqrt{\frac{1}{w-1}\sum_{i=1}^{w-1}\left|\sqrt{{y_i}^2 + {z_i}^2}_{i+1} - \sqrt{{y_i}^2 + {z_i}^2}_{i}\right|} \qquad (19)$$

15)Energy XZ (E_{xz}): This feature represents the change of acceleration at each measurement in a single window on the plane X and Z. Equation (20) used to obtain this feature.

$$\mathrm{E}_{xz} = \sqrt{\frac{1}{w-1}\sum_{i=1}^{w-1}\left|\sqrt{{x_i}^2 + {z_i}^2}_{i+1} - \sqrt{{x_i}^2 + {z_i}^2}_{i}\right|} \qquad (20)$$

C. Splitting Criteria Optimization

This method is used to determine whether splitting criteria have a significant impact on classification performance. If the impact is significant, this study will provide recommendations for splitting the most appropriate criteria to solve the problem of classification of human activity. Splitting Criteria Optimization stages can be illustrated in Fig. 1.

The stages of Splitting Criteria Optimization in Fig. 1 can be explained as follows:

1) Build the Random Forest ensemble n times where n is the number of splitting criteria that you want to investigate. This study evaluated three of the most widely used splitting criteria, namely Information Gain, Gain Ratio, and Gini Index.

2) Generate decision trees in each forest without pruning. The number of decision trees that are generated is as many attributes as there are. Because this study uses 20 features / attributes, each forest will have 20 decision trees.

3) Measure the average accuracy of the decision tree in each Random Forest using Out-Of-Bag data (data that is not used in the training process).

4) Compare the results of the accuracy between the three ensembles with the same number of trees.

5) Determine splitting criteria that is able to build Random Forest ensemble with the best average accuracy value.

D. Tree-Depth Optimization

One of the main contributions in this study was the limitation of tree depth with measured pruning techniques. The original Random Forest algorithm does not do pruning. Thus, the decision tree structure becomes very deep and large. Furthermore, it will make the variance even higher. High variance means that the classification model will be very accurate if tested using training data yet will be inaccurate if tested using testing data that is never seen before. This is known as overfitting which is often found in classification algorithms (Kuhn and Johnson, 2013). The pruning technique could makes the decision tree in Random Forest ensemble concise. It resulted in a group of small size trees. Thus, the complexity of the decision tree becomes smaller. It will give considerable classification strength even though the data conditions are diverse and have not been recognized before. However, this technique needs to be applied carefully as if it performed improperly will reduce the accuracy of the classification model.

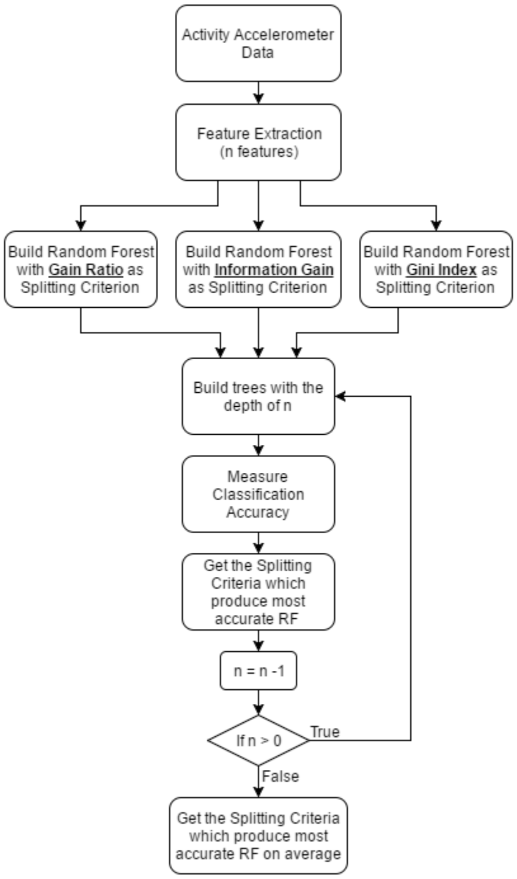
Activity Accelerometer Data
Feature Extraction (n features)
Build Random Forest with Gain Ratio as Splitting Criterion
Build Random Forest with Information Gain as Splitting Criterion
Build Random Forest with Gini Index as Splitting Criterion
Build trees with the depth of n
Measure Classification Accuracy
Get the Splitting Criteria which produce most accurate RF
n = n -1
If n > 0
True
False
Get the Splitting Criteria which produce most accurate RF on average

Fig. 1. Splitting Criteria Optimization Method.

This study proposes a measured pruning technique by evaluating the accuracy as a result of tree depth reduction. The steps to get the optimal tree depth are as follows:

1) Build Random Forest ensemble without pruning to get maximum accuracy.

2) Measure the classification accuracy of the Random Forest classification model.

3) Reduce the depth of the tree in Random Forest classification model.

4) Repeat the accuracy measurement for the new Random Forest classification model.

5) If the accuracy does not changes significantly, go to step 3.

6) If the accuracy changes significantly, stop reducing the depth of the tree as the optimum tree depth has been obtained.

The pseudocode used in this study to reach the optimal depth in the Random Forest ensemble is as follows:

```
k = 21
for j=1 to 5
    build trees() with depth of k;
RF_vote();
Acc_old = RF_Accuracy();

Acc_new = 100;
While Acc_new >= Acc_old
    k = k -1
    for j=1 to 5
        build trees() with depth of k;
    RF_vote();
    Acc_new = RF_Accuracy();
return k;
```

In this study the initial depth of tree (k) is 21 because in the worst case scenario, the decision tree will use the entire feature (twenty one features) as leaf nodes to determine the class of a data. Variable i indicates the number of decision trees in Random Forest ensemble. This study only shows results of performance measurement in the ensemble Random Forest classification consisting of 5, 10, 15, and 20 decision tree as the results are significant to each other.

E. Validation

The performance of the activity classification system needs to be measured correctly to determine the quality of the system being built. The measuring index that is generally used to determine the performance of the classification system is the value of specificity, sensitivity, and accuracy (Han, Kamber et al., 2011). However, this study only uses accuracy as a measure of performance which is calculated using the following formula:

$$Accuracy = \frac{TP+TN}{P+N} \tag{21}$$

where TP = True Positive, TN= True Negative, P = Positive, N = Negative.

This research makes use of 10-Fold Cross-Validation to obtain accuracy value in activity classification as the performance value calculated by the K-fold cross-validation method is less dependent on the data distribution characteristics in the training set and test set. Therefore the resulting performance value can be considered more.

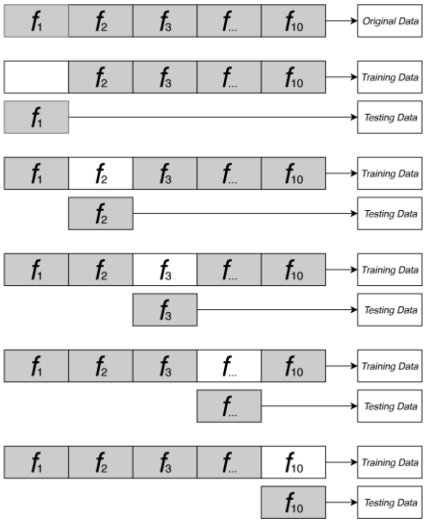

Fig. 2. Ten-Fold Cross-Validation.

The 10-Fold Cross Validation is superior to the usual split method. The number of folds selected, which is ten, also proved to have produced a variance against a relatively small performance. Taking into account the computational complexity required, this method is better than the more expensive methods, such as leave-one-out cross-validation [24]. The fold cross validation method with k = 10 becomes the standard for predicting the performance of algorithms in machine learning. The dataset benchmark test shows that k = 10 represents the number of folds appropriate to obtain the best accuracy estimation [25] (Fig. 2).

V. Experiment Result

The resulting experiment shows Random Forest Accuracy with the number of trees in each ensemble varies from 5, 10, 15, and 20 trees. On each ensemble, classification accuracy was measured and analyzed towards tree depth to find out the optimum tree depth for activity classification.

A. Random Forest with 5 Trees

The first experiment tested the accuracy of the Random Forest algorithm when using 5 trees. The results of the research show that initially, Random Forest was able to achieve accuracy up to 87.86% when classifying human activities.

However, the accuracy starts to decrease significantly when the tree depths is 7. Here, Random Forest that uses Information Gain as the splitting criterion is slightly better than one that uses Gini Index and Gain Ratio as the splitting criterion (Fig. 3).

Fig. 3 also shows that Gain Ratio and Gini Index have better ability to retain accuracy on the occasion of tree depth reduction. On the event that tree depth is reduced from 7 to 3, both splitting criteria have better result compared to Information Gain. Classification result in trees with Information gain shows significant accuracy drop to only 67%.

B. Random Forest with 10 Trees

The second experiment measures the classification accuracy of the Random Forest with 10 trees. Increasing the number of trees involved in Random Forest improved human activities classification accuracy up to 89.29%. The accuracy starts to decrease significantly when the tree depths is 6. Gini Index exhibits better performance compared to Gini Index or Information Gain as Random Forest splitting criterion (Fig. 4). We also noticed that additional trees affect the tree depth as the classification accuracy started to decrease at the depth of 6 instead of 7 in the first experiment.

C. Random Forest with 15 Trees

Further increment on the number of Random Forest trees shows that Random Forest with 15 trees could achieve accuracy up to 91.43% when classifying human activities. The accuracy starts to decrease significantly when the tree depths is 5. This finding assures previous assumption that additional trees could reduce the tree depth as the more trees lead to shallower tree depth (Fig. 5).

Gini Index and Gain Ratio exhibit better performance as Random Forest splitting criterion compared to Information Gain. However, Gini Index exhibits better performance on shallow tree depth as it gives better accuracy even at the depth of 3.

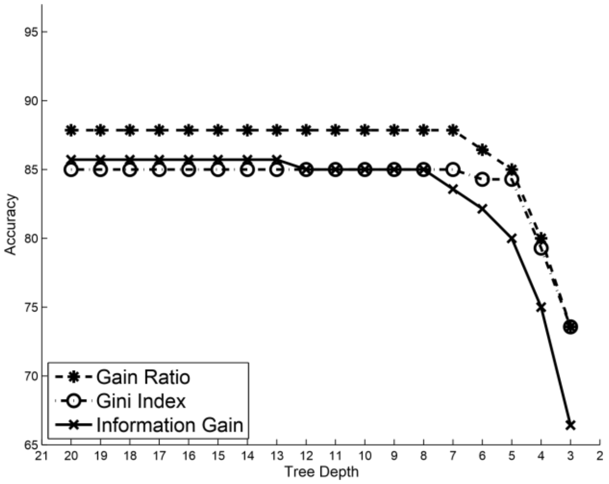

Fig. 3. Splitting Criterion Impact on Random Forest with 5 Trees Activity Classification Accuracy.

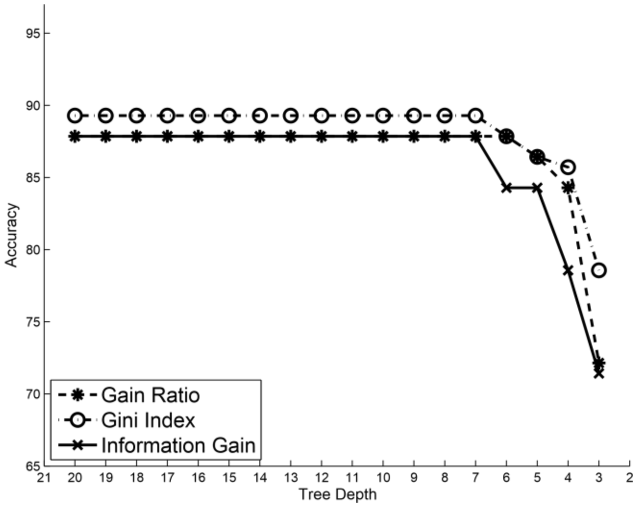

Fig. 4. Splitting Criterion Impact on Random Forest with 10 Trees Activity Classification Accuracy.

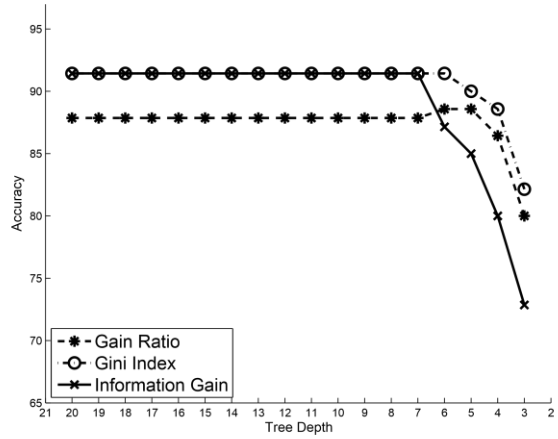

Fig. 5. Splitting Criterion Impact on Random Forest with 15 Trees Activity Classification Accuracy.

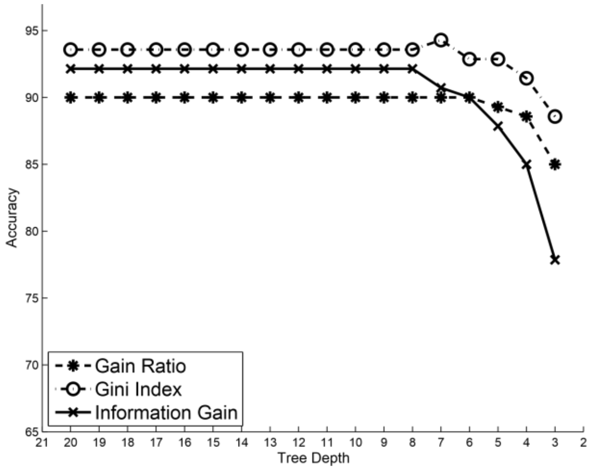

Fig. 6. Splitting Criterion Impact on Random Forest with 20 Trees Activity Classification Accuracy.

D. Random Forest with 20 Trees

The last experiment, Random Forest with 20 trees is able to achieve accuracy up to 93.57% when classifying human activities including falls. The accuracy starts to decrease significantly when the tree depths is 5 (Fig. 6). This result suggests that additional trees on Random Forest Ensemble no longer reduce tree depth.

In this experiment setup, Gini Index clearly gives the best result in contrast to Information Gain or Gain Ratio. This experiment results even further emphasize the previous hypothesis that Gini Index has better performance compared to other splitting criteria as it could retain accuracy in the minimum tree depth.

VI. Discussion

The experiment result indicates that Gini Index generally gives better performance for Activity Classification. When the number of trees in the RF ensemble is small, the three splitting criteria look like they give relatively the same accuracy, but when the number of trees increases, the Gini Index has a significant impact. The only time Gini Index has lower performance compared to other splitting criteria was in the experiment with 5 trees. Above those number, Gini Index clearly outperforms other splitting criteria. Based on this, it can be concluded that the most optimum splitting criteria for activity classification is Gini Index.

Furthermore, RF decision trees that are built using the Gini Index show the ability to produce better performance in conditions where decision trees depth are less. Fig. 3 shows that that Gini Index has the lowest performance at the beginning (deepest trees). At the end, as the depth of the trees becoming less, it was able to outperform other splitting criteria at the depth of 3. Thus, it can be inferred that Gini Index has the ability to maintain classification accuracy.

The minimum depth to have good accuracy is 5. It could be achieved with 10 trees. By choosing the right splitting criterion, classification accuracy drop could be reduced. As could be seen in Fig. 3, Gini Index splitting criteria relatively could retain accuracy better than other splitting criteria.

There is a trade-off between the number of trees and the depth of the tree. The more trees, the lower the depth of trees needed to achieve the best accuracy. Likewise the opposite, the smaller number of trees, the deeper is the tree in order to achieve the best accuracy. However, this thing only happens before a certain point. The depth of the tree involved in the ensemble could be suppressed until the number of trees is 15. Afterward, additional trees only improved accuracy only. It could not take advantage of lowering the depth of the trees. In another word, additional accuracy after that point would add significant complexity. As an illustration, 10 trees with the depth of 6 will have 10*(2^6) = 640 logic gate while 15 trees with the depth of 5 will only have 15*(2^5) = 480 logic gate. This proves that 15 trees with the depth of 5 are less complex than 10 trees with the depth of 6. Therefore other than Gini Index as splitting criteria and 5 as the most optimum depth in RF ensemble, it could be conclude that the most optimum number of trees in Random Forest ensemble is 15.

VII. CONCLUSIONS

This research proposed several methods to optimize Random Forest algorithm performance as Human Activity Classifier. Those are the selection of the most optimal splitting criterion and measured pruning to limit the tree depth in RF ensemble in order to find the minimum depth of the tree to get optimum accuracy.

The results of this study indicate that splitting criteria greatly influence the accuracy of the decision models produced by Random Forest. The first method in this study found that Gini Index is the most suitable splitting criteria to construct decision tree models used solve activity classification. Gini Index exhibits the ability to retain classification accuracy on the shallow tree depth. Furthermore, trees that was build using Gini Index has the minimum accuracy reduction upon reduction of the tree depth.

The measured pruning method applied in this research find that the minimum tree depth for activity classifier is 5. Additional depth no longer increases the accuracy yet significantly increases computational complexity. Limiting the depth of the decision tree will reduce the complexity of the algorithm, thereby increasing the efficiency of the decision model.

VIII. FUTURE WORK

This research was aimed to be preliminary research on efficient fall detection using the accelerometer. There is a trend in the increasing number of cores in the processor. The Random Forest can benefit from this trend by distributing decision trees evenly on each core. Therefore, the use of RF activity classification can be done quicker as it performed in parallel.

This study provides methods to optimize decision tree models constructed with Random Forest algorithm by utilizing the most splitting criteria for certain problem and limiting the trees depth. Other than those two things, the number of trees in the Random Forest ensemble also influences the complexity of the decision model that is built. However, there are no exact numbers to determine the number of trees in the RF ensemble. Therefore, there is still an opportunity to maximize the Random Forest algorithm by compressing the number of RF trees. The next research could address this issue to extend the efficiency of Random Forest.

ACKNOWLEDGMENT

This research was supported by The Ministry of Research, Technology and Higher Education (Kemenristekdikti) of the Republic of Indonesia (Grant No. 109/SP2H/LT/DRPM/2018).

REFERENCES

[1] Hakim, M. S. Huq, S. Shanta, dan B. S. K. K. Ibrahim, “Smartphone Based Data Mining for Fall Detection: Analysis and Design,” Procedia Computer Science, vol. 105, hlm. 46–51, Jan 2017.

[2] P. Kumari, L. Mathew, dan P. Syal, “Increasing trend of wearables and multimodal interface for human activity monitoring: A review,” Biosensors and Bioelectronics, vol. 90, hlm. 298–307, Apr 2017.

[3] P. Jatesiktat dan W. T. Ang, “An elderly fall detection using a wrist-worn accelerometer and barometer,” dalam 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2017, hlm. 125–130.

[4] M. Daher, A. Diab, M. E. B. E. Najjar, M. A. Khalil, dan F. Charpillet, “Elder Tracking and Fall Detection System Using Smart Tiles,” IEEE Sensors Journal, vol. 17, no. 2, hlm. 469–479, Jan 2017.

[5] S. Khojasteh, J. Villar, C. Chira, V. González, dan E. de la Cal, “Improving Fall Detection Using an On-Wrist Wearable Accelerometer,” Sensors, vol. 18, no. 5, hlm. 1350, Apr 2018.

[6] A. K. Bourke, J. V. O’brien, dan G. M. Lyons, “Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm,” Gait & posture, vol. 26, no. 2, hlm. 194–199, 2007.

[7] M. Kangas, A. Konttila, I. Winblad, dan T. Jamsa, “Determination of simple thresholds for accelerometry-based parameters for fall detection,” dalam 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2007. EMBS 2007, 2007, hlm. 1367–1370.

[8] J. Jacob dkk., “A fall detection study on the sensors placement location and a rule-based multi-thresholds algorithm using both accelerometer and gyroscopes,” dalam 2011 IEEE International Conference on Fuzzy Systems (FUZZ), 2011, hlm. 666–671.

[9] T. H. Nguyen, T. P. Pham, C. Q. Ngo, dan T. T. Nguyen, “A SVM Algorithm for Investigation of Tri-Accelerometer Based Falling Data,” American Journal of Signal Processing, vol. 6, no. 2, hlm. 56–65, 2016.

[10] B. T. Nukala dkk., “An Efficient and Robust Fall Detection System Using Wireless Gait Analysis Sensor with Artificial Neural Network (ANN) and Support Vector Machine (SVM) Algorithms,” Open Journal of Applied Biosensor, vol. 3, no. 04, hlm. 29, 2015.

[11] F. A. J. Parera dan C. Angulo, “Accelerometer Signals Analisys Using Svm And Decision Tree In Daily Activity Identification,” Gerontechnology, vol. 7, no. 2, hlm. 184, 2008.

[12] O. Aziz, M. Musngi, E. J. Park, G. Mori, dan S. N. Robinovitch, “A comparison of accuracy of fall detection algorithms (threshold-based vs. machine learning) using waist-mounted tri-axial accelerometer signals from a comprehensive set of falls and non-fall trials,” Med Biol Eng Comput, hlm. 1–11, Apr 2016.

[13] M. Kounelakis, M. Zervakis, dan X. Kotsiakis, “Chapter 13 - The Impact of Microarray Technology in Brain Cancer,” dalam Outcome Prediction in Cancer, A. F. G. Taktak dan A. C. Fisher, Ed. Amsterdam: Elsevier, 2007, hlm. 339–388.

[14] T. Nef dkk., “Evaluation of Three State-of-the-Art Classifiers for Recognition of Activities of Daily Living from Smart Home Ambient Data,” Sensors (Basel), vol. 15, no. 5, hlm. 11725–11740, Mei 2012.

[15] Q. Li, “Noise Reduction of Accelerometer Signal with Singular Value Decomposition and Savitzky-Golay Filter,” Journal of Information and Computational Science, vol. 10, no. 15, hlm. 4783–4793, Okt 2013.

[16] V. Svetnik, A. Liaw, C. Tong, J. C. Culberson, R. P. Sheridan, dan B. P. Feuston, “Random Forest: A Classification and Regression Tool for Compound Classification and QSAR Modeling,” J. Chem. Inf. Comput. Sci., vol. 43, no. 6, hlm. 1947–1958, Nov 2003.

[17] A. Cutler, D. R. Cutler, dan J. R. Stevens, “Random Forests,” dalam Ensemble Machine Learning, C. Zhang dan Y. Ma, Ed. Boston, MA: Springer US, 2012, hlm. 157–175.

[18] M. Jaworski, L. Rutkowski, dan M. Pawlak, “Hybrid Splitting Criterion in Decision Trees for Data Stream Mining,” dalam Artificial Intelligence and Soft Computing, vol. 9693, L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. A. Zadeh, dan J. M. Zurada, Ed. Cham: Springer International Publishing, 2016, hlm. 60–72.

[19] Leo Breiman, “Random forests,” Machine learning, vol. 45, no. 1, hlm. 5–32, 2001.

[20] J. R. Quinlan, “Induction of decision trees,” Machine learning, vol. 1, no. 1, hlm. 81–106, 1986.

[21] L. Breiman, Classification and regression trees. Routledge, 2017.

[22] N. Pannurat, S. Thiemjarus, dan E. Nantajeewarawat, “Automatic Fall Monitoring: A Review,” Sensors, vol. 14, no. 7, hlm. 12900–12936, Jul 2014.

[23] B. H. Menze dkk., “A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data,” BMC Bioinformatics, vol. 10, no. 1, hlm. 213, 2009.

[24] R. Kohavi, “A study of cross-validation and bootstrap for accuracy estimation and model selection,” dalam Ijcai, 1995, vol. 14, hlm. 1137–1145.

[25] R. Bouman dan J. van Dongen, Pentaho solutions: business intelligence and data warehousing with Pentaho and MySQL, 1. Aufl. Indianapolis, Ind: Wiley, 2009.