Feature selection to increase the random forest method performance on high dimensional data

Maria Irmina Prasetiyowati [a,1,*], Nur Ulfa Maulidevi [b,2], Kridanto Surendro [b,3]

[a] Institut Teknologi Bandung, Ganesha No 10, Bandung – 40132, Indonesia
[1] maria@umn.ac.id; [2] ulfa@informatika.org; [3] endro@informatika.org
* corresponding author

ARTICLE INFO

Article history
Selected paper from The 2019 2nd International Symposium on Advanced Intelligent Informatics (SAIN'19), Fukuoka-Japan, 2-4 September 2019, http://sain.ijain.org/2019/. Peer-reviewed by SAIN'19 Scientific Committee and Editorial Team of IJAIN journal.

Received March 4, 2020
Revised April 12, 2020
Accepted April 25, 2020
Available online November 30, 2020

Keywords
Random forest
Feature selection
BestFirst method
High dimensional data
CNAE-9 dataset

ABSTRACT

Random Forest is a supervised classification method based on bagging (Bootstrap aggregating) Breiman and random selection of features. The choice of features randomly assigned to the Random Forest makes it possible that the selected feature is not necessarily informative. So it is necessary to select features in the Random Forest. The purpose of choosing this feature is to select an optimal subset of features that contain valuable information in the hope of accelerating the performance of the Random Forest method. Mainly for the execution of high-dimensional datasets such as the Parkinson, CNAE-9, and Urban Land Cover dataset. The feature selection is done using the Correlation-Based Feature Selection method, using the BestFirst method. Tests were carried out 30 times using the K-Cross Fold Validation value of 10 and dividing the dataset into 70% training and 30% testing. The experiments using the Parkinson dataset obtained a time difference of 0.27 and 0.28 seconds faster than using the Random Forest method without feature selection. Likewise, the trials in the Urban Land Cover dataset had 0.04 and 0.03 seconds, while for the CNAE-9 dataset, the difference time was 2.23 and 2.81 faster than using the Random Forest method without feature selection. These experiments showed that the Random Forest processes are faster when using the first feature selection. Likewise, the accuracy value increased in the two previous experiments, while only the CNAE-9 dataset experiment gets a lower accuracy. This research's benefits is by first performing feature selection steps using the Correlation-Base Feature Selection method can increase the speed of performance and accuracy of the Random Forest method on high-dimensional data.

1. Introduction

Random Forest is an ensemble learning method [1] for classification and regression by building a number of decision trees in a forest and predicting the results by voting [2]. In addition, The algorithm built a decision tree without pruning [3], and in the classification, the approach uses a combination of "bagging" Breiman and random feature selection [4]. The Random Forest classification is done by combining the tree that conducting training on the owned data sample. The selection of features to build a Random Forest tree is made randomly at the beginning of the algorithm. Random selection of features based on impurity measures is used as a criterion to determine the best features for partition nodes. From this random selection of features, a decision tree is forming. The decision tree will be built as many times as desired. The use of the decision tree will increasingly affect the obtained accuracy. The number of trees in the forest gives the results in higher accuracy. From several decision trees that have been built, Random Forest classification is done by voting. The winners are the most votes from the decision tree formed. The performance of Random Forest depends on the diversity of the forest decision

tree and the performance of each decision tree [5]. Breiman formulates a set of trees' overall performance as average strength and average correlation between trees [4], and it shows that generalization errors from a Random Forest classifier are limited by the average correlation ratio between trees divided by the square of the strength of the average tree [5].

Seeing how the Random Forest feature selection works is done randomly and is done repeatedly, It may cause the computing process in the Random Forest is taking a long time. It is also possible that random features on the Random Forest are not informative, especially if using high dimensional data. There are several studies on feature selection [6]–[17] especially the use of feature selection in Random Forest [18][19], namely research conducted by Amaratungga [20]. This study used individual weighting features and proved to provide an increase in classification performance, but there is a possibility that features with large weights are chosen repeatedly. In contrast to Ye's research, which groups feature into two groups, groups contained strong informative features and weak formative features [5]. This study shows better performance than other algorithms such as SVM and four variants of Random Forest, Nearest Neighbor (NN), and Naïve Bayes (NB) algorithms. Two random forest feature selection studies have the aim of improving performance and accuracy. Another feature selection study was conducted by Manbari *et al.* [13]. Manbari *et al.* [13] presented a new hybrid filter-based feature selection algorithm, combining modified Clustering and Binary Ant System (BAS). The proposed model provides global and local search capabilities between and within clusters. The proposed method achieves better performance than other feature selection methods and reduces computational complexity. This study's disadvantage was greatly reducing the number of clusters and selectivity of features. Lu [21] conducted research using the embedded method, which proposed the Sparse Optimal Scoring with Adjustment (SOSA) method. Experimental results on synthetic data and three datasets show that the features selected by the SOSA method can consistently produce better or comparative classification performance compared to features chosen by traditional embedded methods. Moran and Gordon [14] also proposed a feature selection method called Curious Feature Selection (CFS) that presents the same accuracy as the simple and greedy Sequential Forward Selection algorithm. The advantages of the proposed Curious Feature Selection algorithm are overfitting, online learning, and scale. Although Manbari *et al.* [13] study performed quite well in terms of time and accuracy, the process he used is quite complicated. In comparison, Moran and Gordon [14] have a positive impact on the accuracy problems.

In this study, we focus on improving the execution time and accuracy using the Correlation-Base Feature Selection with the best first method applied to high dimensional datasets. The results are then compared to the Random Forest method's without feature selection. The speed and accuracy testing of the original Random Forest method (without a selection of features) is done by Random Forest, which has used feature selection first. The dataset used in the test is UCI's high dimensional dataset, i.e., Parkinson, CNAE-9, and Land Cover dataset. This paper is structured as follows: Introduction is outlined in Section 1; the method is explained in Section 2. Section 3 describes the results of the experiments, and section 4 is the conclusion.

2. Method

The research used UCI's datasets and analyzed them using Weka tools software version 3.9.2. The datasets used are the CNAE-9, Parkinson, and Urban Land Cover high dimension dataset that has been used by Sakar *et al.* [22] while the Urban Land Cover dataset has been used by Johnson and Xie [23] and Johnson [24]. The test was conducted in 30 repetitions, using K-Cross Fold Validation with K = 10, 70% exercise split and, 30% test. Cross Fold validation is one technique that allows all datasets to be training data as well as test data. The Weka's Cross Fold Validation default is 10, which is meaning the randomized 10 times to validate the research dataset. Each test is done by changing the value of the seed in Weka that is a Weka's function to generate random data. The 15 seeds are entered sequentially from numbers 1 to 15, while 15 other seeds are randomly entered.

Tests are carried out using the original Random Forest method and Random Forest using the dataset as a result of feature selection. Irrelevant and unused features will be discarded. This feature selection technique is used for several reasons including, 1) Simplification of the model [25]; 2) Increasing

generalization by reducing overfitting [13][26], by reducing variance [25]; 3) Improve performance in terms of speed [6], prediction accuracy or simplicity of rules [27]; 4) Reducing dimensions and eliminating noise [27][28]; and 5) Avoid the curse of dimensionality [13][15][11].

The selection of features used is attribute selection or attribute evaluator Correlation-based feature selector (CfsSubsetEval), using the BestFirst method. CfsSubsetEval is a method that evaluates the value of attribute subset by considering each features's predictive capabilities and the level of redundancy between features/attributes [29]. The BestFirst method is a search algorithm based on optimizing the best value. The results of feature selection are applied to the dataset used. After that, the dataset is analyzed by using Random Forest. More details can be seen in Fig. 1 and Fig. 2.

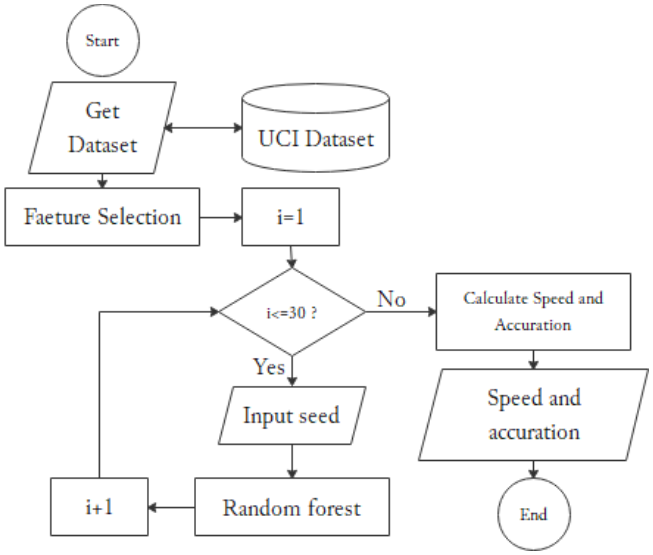

Fig. 1. Flowchart Feature Selection

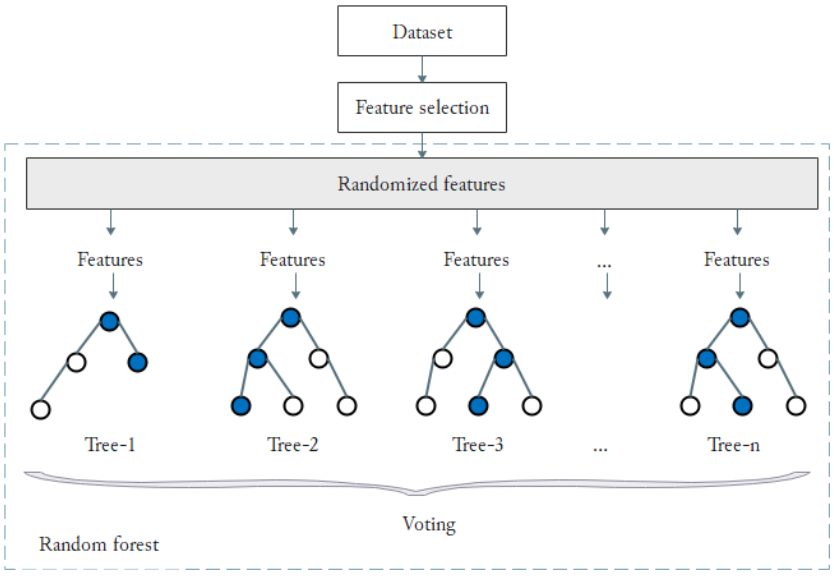

Fig. 2. Feature Selection for Random Forest

3. Results and Discussion

3.1. Experiment Results

The study was conducted by testing the high dimensional dataset 30 times using different seed values available in Weka tools software version 3.9.2. The trial uses the K-Cross Fold Validation value K = 10 and splitting the datasets into 70% training and 30% testing.

3.1.1. First experiment

In the first series of experiments, the UCI's Parkinson dataset was used that consists of 756 instances, collected from 188 patients (107 men and 81 women), aged around 33 to 87 years. This dataset is high-dimensional with 755 attributes (features), and 1 instance class attribute. The classification carried out by the Random Forest method uses 754 randomly selected features, with 30 different seed values. The first experiment using K-Cross Validation produced an average accuracy of 86.66% with an average speed of 0.48 seconds (Table 1). The percentage of 70% split the average accuracy is 85.17% with an average speed of 0.47 seconds (Table 2). The results obtained by feature selection using attribute selection or attribute evaluator Correlation-based feature selector (CfsSubsetEval) with the BestFirst method, the Random Forest with K-Cross Validation average accuracy is 88.46% and its average speed of 0.20 seconds (Table 1). The accuracy of the percentage of 70% is 86.77% with 0.20 seconds average speed (Table 2). Both the accuracy and the average speed of the Random Forest method, which previously performed feature selection, are faster and more accurate.

Table 1. Testing of Parkinson's dataset with K-Cross Validation

Random Forest						Random Forest with Feature Selection					
Seed	*Time*	*Accurate*	*Seed*	*Time*	*Accurate*	*Seed*	*Time*	*Accurate*	*Seed*	*Time*	*Accurate*
1	0.74	86.77%	33	0.47	86.11%	1	0.21	88.49%	33	0.21	88.76%
2	0.48	87.04%	40	0.47	86.77%	2	0.20	88.76%	40	0.20	88.62%
3	0.46	87.70%	57	0.46	86.64%	3	0.19	88.49%	57	0.21	88.10%
4	0.47	86.77%	70	0.46	87.17%	4	0.19	88.23%	70	0.21	89.29%
5	0.46	86.91%	73	0.48	86.38%	5	0.20	88.36%	73	0.20	89.02%
6	0.47	86.77%	80	0.47	87.04%	6	0.21	88.10%	80	0.21	88.49%
7	0.47	86.38%	94	0.48	86.77%	7	0.19	88.10%	94	0.21	88.62%
8	0.47	86.24%	100	0.48	86.24%	8	0.20	88.10%	100	0.21	88.76%
9	0.47	86.51%	153	0.47	86.24%	9	0.21	88.36%	153	0.20	88.23%
10	0.46	86.38%	251	0.47	86.77%	10	0.20	88.76%	251	0.22	88.76%
11	0.46	86.91%	300	0.46	86.11%	11	0.22	87.70%	300	0.20	87.83%
12	0.46	86.11%	457	0.59	86.64%	12	0.21	88.76%	457	0.20	88.40%
13	0.47	86.64%	505	0.46	86.77%	13	0.20	88.23%	505	0.20	88.23%
14	0.47	86.77%	603	0.46	86.51%	14	0.20	88.23%	603	0.20	88.76%
15	0.47	86.38%	700	0.47	87.43%	15	0.19	88.76%	700	0.19	88.50%
Average Time				0.48 second		Average Time				**0.20 second**	
Average Accurate				86.66%		Average Accurate				**88.46%**	

Table 2. Dataset Parkinson's Test with the percentage split 70% - 30%

Random Forest						Random Forest with Feature Selection					
Seed	*Time*	*Accurate*	*Seed*	*Time*	*Accurate*	*Seed*	*Time*	*Accurate*	*Seed*	*Time*	*Accurate*
1	0.46	84.14%	33	0.46	84.14%	1	0.20	87.67%	33	0.19	87.23%
2	0.47	85.02%	40	0.46	85.90%	2	0.22	86.78%	40	0.20	86.78%
3	0.47	85.02%	57	0.46	84.14%	3	0.21	88.55%	57	0.20	87.23%
4	0.46	85.46%	70	0.48	86.34%	4	0.20	86.34%	70	0.20	86.78%
5	0.48	84.14%	73	0.47	85.02%	5	0.20	87.23%	73	0.20	86.34%
6	0.47	85.90%	80	0.47	83.70%	6	0.20	86.34%	80	0.21	86.78%
7	0.47	85.02%	94	0.47	84.58%	7	0.19	87.23%	94	0.20	86.34%
8	0.46	85.90%	100	0.47	85.90%	8	0.19	86.78%	100	0.20	85.90%
9	0.47	85.90%	153	0.46	85.46%	9	0.20	86.78%	153	0.21	86.78%
10	0.46	84.58%	251	0.47	85.02%	10	0.19	86.34%	251	0.20	87.23%
11	0.47	85.46%	300	0.46	85.90%	11	0.19	85.46%	300	0.20	86.34%
12	0.49	85.46%	457	0.48	85.46%	12	0.20	86.78%	457	0.20	86.78%
13	0.47	85.46%	505	0.48	85.46%	13	0.19	86.34%	505	0.21	87.23%
14	0.46	85.90%	603	0.49	84.58%	14	0.20	86.34%	603	0.20	86.78%
15	0.47	84.58%	700	0.48	85.46%	15	0.19	86.78%	700	0.20	86.78%
Average Time				0.47 second		Average Time				**0.20 second**	
Average Accurate				85.17%		Average Accurate				**86.77%**	

3.1.2 The second experiment

The second series of experiments used UCI's CNAE-9 dataset that contains 1080 documents about the description of free text business from the privatized Brazilian company. This dataset is a high-dimensional dataset with 857 attributes (features), 1 instance class attribute, and 856-words frequency attributes in integer form. The Random Forest method is used for the classification of the 857 randomly selected features. The Experiments of the K-Cross Validation produced 93.72% average accuracy and 2.49 seconds average speed (Table 3), while 94.20% average accuracy and 3.08 seconds average speed were produced by the percentage of 70% split (Table 4).

Table 3. Test of the CNAE-9 dataset with K-Cross Validation

Random Forest						Random Forest with Feature Selection					
Seed	*Time*	*Accurate*	*Seed*	*Time*	*Accurate*	*Seed*	*Time*	*Accurate*	*Seed*	*Time*	*Accurate*
1	2.15	93.89%	33	2.04	93.70%	1	0.28	81.20%	33	0.26	81.02%
2	2.08	94.35%	40	2.16	93.61%	2	0.26	81.11%	40	0.26	81.11%
3	2.07	93.52%	57	2.06	93.52%	3	0.27	81.11%	57	0.31	81.11%
4	2.10	93.80%	70	3.10	93.61%	4	0.24	81.20%	70	0.27	81.20%
5	2.11	93.80%	73	3.41	93.33%	5	0.26	81.39%	73	0.25	81.30%
6	2.14	93.52%	80	3.22	93.80%	6	0.26	81.39%	80	0.27	81.11%
7	2.16	93.52%	94	3.15	94.17%	7	0.26	81.30%	94	0.26	81.11%
8	2.14	93.98%	100	2.67	93.89%	8	0.25	81.30%	100	0.24	81.20%
9	2.10	93.70%	153	2.86	93.52%	9	0.24	81.30%	153	0.27	81.30%
10	2.32	93.80%	251	3.30	93.43%	10	0.27	81.48%	251	0.26	81.20%
11	2.13	93.24%	300	2.43	93.80%	11	0.25	81.20%	300	0.26	81.30%
12	2.08	93.89%	457	3.29	93.70%	12	0.26	81.30%	457	0.27	81.11%
13	2.08	93.70%	505	3.18	93.80%	13	0.28	81.48%	505	0.25	81.20%
14	2.06	93.70%	603	2.96	93.98%	14	0.24	81.30%	603	0.24	81.30%
15	2.10	93.61%	700	3.15	93.70%	15	0.25	81.30%	700	0.27	81.11%
Average Time				2.49 second		Average Time				**0.26 second**	
Average Accurate				**93.72%**		Average Accurate				81.23%	

Table 4. Tests on the CNAE-9 dataset with split percentages of 70% - 30%

Random Forest						Random Forest with Feature Selection					
Seed	*Time*	*Accurate*	*Seed*	*Time*	*Accurate*	*Seed*	*Time*	*Accurate*	*Seed*	*Time*	*Accurate*
1	2.26	93.83%	33	2.82	94.14%	1	0.44	81.79%	33	0.26	81.79%
2	2.97	94.44%	40	2.79	94.75%	2	0.28	82.10%	40	0.26	82.10%
3	3.55	95.06%	57	3.35	94.75%	3	0.26	81.79%	57	0.26	81.48%
4	3.06	94.14%	70	3.62	94.75%	4	0.25	81.79%	70	0.26	81.79%
5	3.07	93.83%	73	3.45	94.14%	5	0.29	81.79%	73	0.25	81.79%
6	3.40	93.52%	80	3.47	94.44%	6	0.24	82.10%	80	0.27	81.79%
7	2.75	93.52%	94	3.75	93.83%	7	0.25	82.10%	94	0.27	81.48%
8	3.30	94.14%	100	3.35	93.83%	8	0.25	81.48%	100	0.26	81.79%
9	3.19	94.14%	153	2.57	93.52%	9	0.26	81.48%	153	0.26	81.79%
10	3.23	94.75%	251	3.60	94.75%	10	0.27	82.10%	251	0.27	81.48%
11	2.79	94.75%	300	2.74	94.75%	11	0.26	81.17%	300	0.27	81.79%
12	3.24	94.14%	457	3.28	93.83%	12	0.25	81.79%	457	0.26	81.79%
13	2.67	94.75%	505	2.70	95.06%	13	0.27	81.79%	505	0.26	81.48%
14	2.65	93.52%	603	3.54	93.52%	14	0.29	81.79%	603	0.26	81.79%
15	2.80	93.21%	700	2.29	94.14%	15	0.26	81.48%	700	0.26	81.79%
Average Time				3.08 second		Average Time				**0.27 second**	
Average Accurate				**94.20%**		Average Accurate				81.75%	

The Random Forest method with K-Cross Validation and feature selection obtained an average accuracy value of 81.23% and 0.26 seconds average speed (Table 3). The percentage of 70% split the average accuracy is 81.75% with an average speed of 0.27 seconds (Table 4).

3.1.3. Third Experiment

Urban Land Cover data taken from Urban training UCI data was used in the third set of experiments. The Urban Land Cover data contains 168 training data for the High-resolution urban land-cover classification. This dataset has 148 attributes (features), 1 instance class attribute, and 147-words frequency attributes in an integrated form. The classification was done by the Random Forest method that used random 857 features. K-Cross Validation produced an average accuracy of 85.08% with an average speed of 0.10 seconds (Table 5). As for the 70% split percentage, the average accuracy is 82.13%, with an average speed of 0.08 seconds (Table 6). The Random Forest method with K-Cross Validation and feature selection obtained an average accuracy value of 87.52% with an average speed of 0.06 seconds (Table 5). As for the 70% split percentage, the average accuracy is 87.27%, with an average speed of 0.05 seconds (Table 6).

Table 5. Tests on the Urban Land Cover dataset with K-Cross Validation

Random Forest						Random Forest with Feature Selection					
Seed	*Time*	*Accurate*	*Seed*	*Time*	*Accurate*	*Seed*	*Time*	*Accurate*	*Seed*	*Time*	*Accurate*
1	0.14	83.33%	33	0.10	85.12%	1	0.06	88.10%	33	0.06	89.29%
2	0.09	85.71%	40	0.09	85.12%	2	0.06	87.50%	40	0.05	88.69%
3	0.09	85.71%	57	0.10	85.12%	3	0.06	85.71%	57	0.07	88.10%
4	0.08	85.12%	70	0.09	86.31%	4	0.06	87.50%	70	0.07	87.50%
5	0.11	82.14%	73	0.10	85.71%	5	0.05	87.50%	73	0.06	86.91%
6	0.08	85.12%	80	0.09	82.74%	6	0.06	85.71%	80	0.06	88.10%
7	0.10	86.31%	94	0.09	85.12%	7	0.06	87.50%	94	0.06	88.69%
8	0.09	86.91%	100	0.09	83.93%	8	0.06	85.70%	100	0.05	88.10%
9	0.12	83.93%	153	0.09	85.71%	9	0.06	86.31%	153	0.07	87.50%
10	0.08	84.52%	251	0.10	85.12%	10	0.06	88.10%	251	0.05	87.50%
11	0.10	85.71%	300	0.10	84.52%	11	0.06	89.29%	300	0.07	86.91%
12	0.10	84.52%	457	0.11	85.12%	12	0.06	86.91%	457	0.05	86.31%
13	0.10	84.52%	505	0.10	84.52%	13	0.05	87.50%	505	0.07	85.71%
14	0.10	86.31%	603	0.10	85.12%	14	0.06	86.31%	603	0.05	88.10%
15	0.09	86.31%	700	0.10	86.91%	15	0.05	88.69%	700	0.06	89.88%
Average Time				0.10 second		Average Time				**0.06 second**	
Average Accurate				85.08%		Average Accurate				**87.52%**	

Table 6. Tests on the Urban Land Cover dataset with split percentages of 70% - 30%

Random Forest						Random Forest with Feature Selection					
Seed	*Time*	*Accurate*	*Seed*	*Time*	*Accurate*	*Seed*	*Time*	*Accurate*	*Seed*	*Time*	*Accurate*
1	0.08	82.00%	33	0.08	82.00%	1	0.05	88.00%	33	0.04	88.00%
2	0.08	80.00%	40	0.09	80.00%	2	0.06	88.00%	40	0.05	86.00%
3	0.08	84.00%	57	0.08	84.00%	3	0.05	88.00%	57	0.05	84.00%
4	0.08	80.00%	70	0.08	82.00%	4	0.05	84.00%	70	0.05	86.00%
5	0.08	84.00%	73	0.09	84.00%	5	0.04	86.00%	73	0.04	88.00%
6	0.09	82.00%	80	0.07	84.00%	6	0.04	86.00%	80	0.05	88.00%
7	0.08	84.00%	94	0.07	82.00%	7	0.05	88.00%	94	0.06	86.00%
8	0.09	82.00%	100	0.10	82.00%	8	0.04	90.00%	100	0.05	88.00%
9	0.07	82.00%	153	0.08	78.00%	9	0.04	88.00%	153	0.04	86.00%
10	0.08	82.00%	251	0.08	82.00%	10	0.05	86.00%	251	0.05	86.00%
11	0.08	80.00%	300	0.07	82.00%	11	0.05	88.00%	300	0.05	90.00%
12	0.08	80.00%	457	0.08	84.00%	12	0.05	86.00%	457	0.05	88.00%
13	0.08	80.00%	505	0.07	80.00%	13	0.05	90.00%	505	0.06	88.00%
14	0.07	84.00%	603	0.09	84.00%	14	0.04	86.00%	603	0.05	88.00%
15	0.09	84.00%	700	0.08	84.00%	15	0.06	88.00%	700	0.04	88.00%
Average Time				0.08 second		Average Time				**0.05 second**	
Average Accurate				82.13%		Average Accurate				**87.27%**	

3.2. Discussion

The original Random Forest method obtained a higher average accuracy than the Random Forest method that has been selected first (Fig. 3). However, the average speed that results from Random Forest using the feature selection first is much faster than the original Random Forest (Fig. 4).

From the previous experiments, it can be proven that making feature selection may affect the processing speed of the Random Forest method. The first experiment uses the Parkinson dataset, with K-Cross Validation, indicating that the average speed required by the first feature selection is 0.2 seconds. Whereas without feature selection, it takes 0.48 seconds. There is a difference in time needed, which is 0.28 seconds. In terms of time or speed getting faster, the accuracy increased by 1.8%, from 86.66% (without feature selection) to 88.46% (using feature selection). The same dataset is also tested using 70% split training data and 30% testing. The average speed required from 0.47 seconds to 0.20 seconds and accuracy increased from 85.17% to 86.77%. There is a decrease in the average time needed, which is equal to 0.27 seconds faster, and an increase in the average value of accuracy is 1.6%.

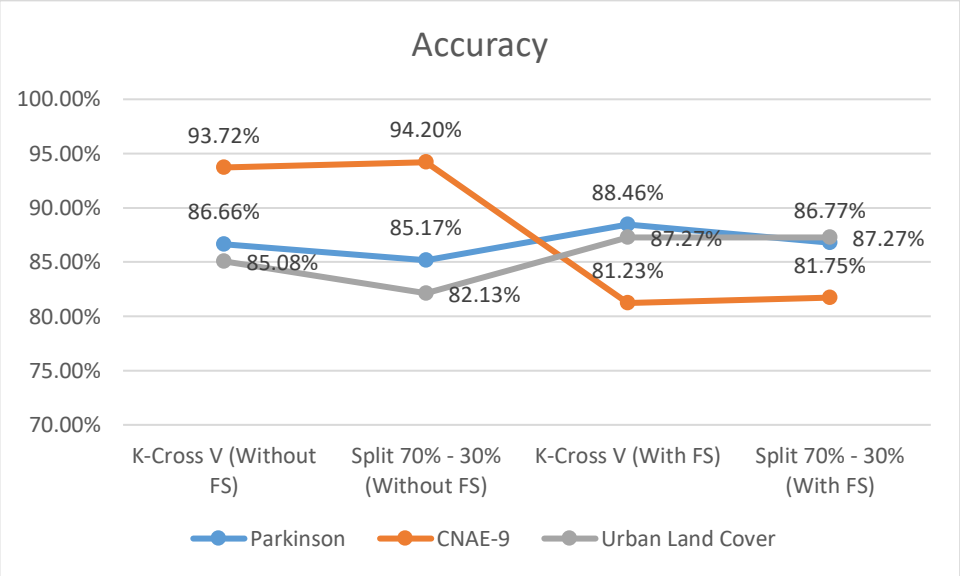

Fig. 3. Accuracy Requirement

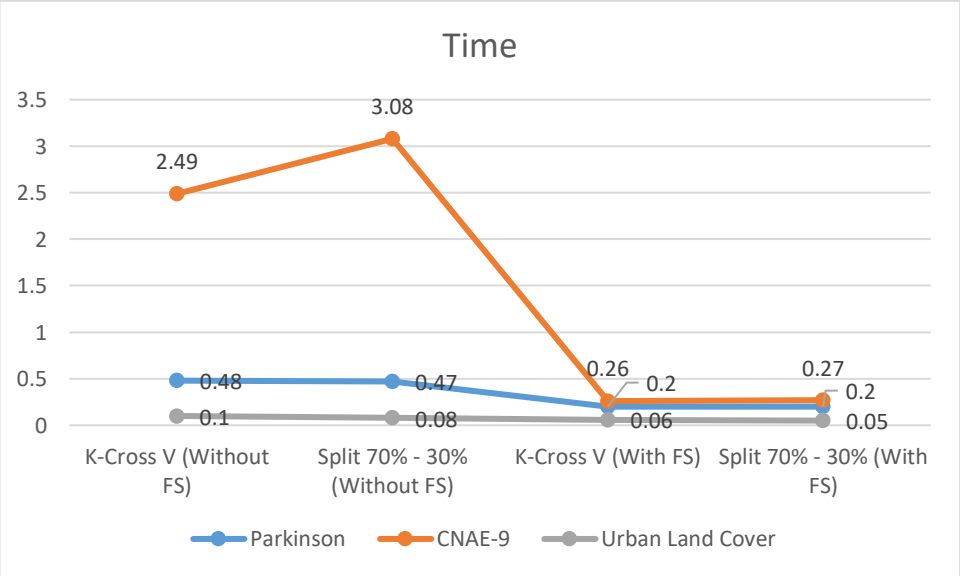

Fig. 4. Time Requirement

The second experiment was conducted using CNAE-9 data, with K-Cross Validation, it was found that there was a difference of 2.23 seconds from the average speed required in Random Forest without feature selection with Random Forest using feature selection. In the original Random Forest, the average time needed is 2.49 seconds, while the Random Forest with feature selection requires a shorter time, which is 0.26 seconds. Seen a decrease in time of 2.23 seconds faster than Random Forest without feature selection. However, for accuracy, it turns out that Random Forest without feature selection produces a much higher average accuracy than Random Forest with feature selection, which is 93.72%, 12.49% faster than Random Forest using feature selection. Likewise, the results obtained from trials using the percentage of split 70% and 30%, the average accuracy of Random Forest without feature selection is superior to 12.45% compared to Random Forest, which uses feature selection. While the Random Forest's speed that uses feature selection is 2.81 seconds faster than Random Forest without feature selection.

The third experiment was carried out using Land Cover Urban data, with K-Cross Validation, showing that the average speed required by the first feature selection was 0.06 seconds. Meanwhile, without making a feature selection, it takes 0.10 seconds. There is a time difference needed, which is 0.04 seconds. The average accuracy's total results also increased by 2.19%, from 85.08% without feature selection to 87.27% with feature selection. It means that in terms of time or speed, It is getting faster. The same dataset was tested using a 70% split percentage of exercise data and 30% test. The average speed needed from 0.08 seconds to 0.05 seconds, and accuracy increased from 82.13% to 87.27%. There is a decrease in the average time required, which is 0.03 seconds, and an increase in the average accuracy by 5.14%.

The trials were conducted on Random Forest high dimension data with feature selection using K-Cross Validation and the percentage of the split by 70% -30%, increasing execution speed. However, the average accuracy produced in the CNAE-9 dataset decreased by 12.49% and 12.45% compared to Random Forest without feature selection. It was possible to happen because there was too much irrelevant data or sparse data on the CNAE-9 dataset [30]. Therefore a method/algorithm is needed in future feature selection research. The average accuracy result was increasing, or at least the same as the Random Forest accuracy without feature selection.

4. Conclusion

This research showed that selecting the Correlation-based feature selector (CfsSubsetEval) feature with the BestFirst method can speed up the Random Forest method's classification process time and improve its accuracy. This can be proven from the prior test completed on the high dimensional Parkinson dataset, high dimensional CNAE-9 dataset and Urban Land Cover high dimension data. The average execution speed increases between 0.27 seconds to 2.81 seconds. In addition to the increasing average speed, the Random Forest method's average accuracy with feature selection also increases when tested on the Parkinson and Urban Land Cover dataset. However, when tested on CNAE-9 data, the average accuracy dropped. This might be due to a sparse problem. The further experimental development may go to seek the new method or feature selection algorithm that both increase speed and more accurate results (sensitivity and specificness) to employ on random forest method.

Acknowledgments

We would like to thank Institut Teknologi Bandung and Universitas Multimedia Nusantara (UMN) for supporting this research.

Declarations

Author contribution. The author confirms the sole responsibility for this manuscript fully as a sole author for the following: study conception and design, data collection, analysis and interpretation of results, and manuscript preparation. The author read and approved the final manuscript.

Funding statement. Not applicable. This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors
Conflict of interest. The authors declare no conflict of interest.
Additional information. No additional information is available for this paper.

References

[1] C. Hu, Y. Chen, L. Hu, and X. Peng, "A novel random forests based class incremental learning method for activity recognition," *Pattern Recognit.*, vol. 78, pp. 277–290, Jun. 2018, doi: 10.1016/j.patcog.2018.01.025.

[2] E. Scornet, G. Biau, and J.-P. Vert, "Consistency of random forests," *Ann. Stat.*, vol. 43, no. 4, pp. 1716–1741, Aug. 2015, doi: 10.1214/15-AOS1321.

[3] D. Talreja, J. Nagaraj, N. J. Varsha, and K. Mahesh, "Terrorism analytics: Learning to predict the perpetrator," in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2017, pp. 1723–1726, doi: 10.1109/ICACCI.2017.8126092.

[4] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324

[5] Y. Ye, Q. Wu, J. Zhexue Huang, M. K. Ng, and X. Li, "Stratified sampling for feature subspace selection in random forests for high dimensional data," *Pattern Recognit.*, vol. 46, no. 3, pp. 769–787, Mar. 2013, doi: 10.1016/j.patcog.2012.09.005.

[6] J. Cai, J. Luo, S. Wang, and S. Yang, "Feature selection in machine learning: A new perspective," *Neurocomputing*, vol. 300, pp. 70–79, Jul. 2018, doi: 10.1016/j.neucom.2017.11.077.

[7] P. Martín-Smith, J. Ortega, J. Asensio-Cubero, J. Q. Gan, and A. Ortiz, "A supervised filter method for multi-objective feature selection in EEG classification based on multi-resolution analysis for BCI," *Neurocomputing*, vol. 250, pp. 45–56, Aug. 2017, doi: 10.1016/j.neucom.2016.09.123.

[8] H. Zhou, Y. Zhang, Y. Zhang, and H. Liu, "Feature selection based on conditional mutual information: minimum conditional relevance and minimum conditional redundancy," *Appl. Intell.*, vol. 49, no. 3, pp. 883–896, Mar. 2019, doi: 10.1007/s10489-018-1305-0.

[9] J. Wang, J.-M. Wei, Z. Yang, and S.-Q. Wang, "Feature Selection by Maximizing Independent Classification Information," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 4, pp. 828–841, Apr. 2017, doi: 10.1109/TKDE.2017.2650906.

[10] P. Zhu, Q. Xu, Q. Hu, and C. Zhang, "Co-regularized unsupervised feature selection," *Neurocomputing*, vol. 275, pp. 2855–2863, Jan. 2018, doi: 10.1016/j.neucom.2017.11.061.

[11] A. Zakeri and A. Hokmabadi, "Efficient feature selection method using real-valued grasshopper optimization algorithm," *Expert Syst. Appl.*, vol. 119, pp. 61–72, Apr. 2019, doi: 10.1016/j.eswa.2018.10.021.

[12] K.-C. Lin, J. C. Hung, and J. Wei, "Feature selection with modified lion's algorithms and support vector machine for high-dimensional data," *Appl. Soft Comput.*, vol. 68, pp. 669–676, Jul. 2018, doi: 10.1016/j.asoc.2018.01.011.

[13] Z. Manbari, F. AkhlaghianTab, and C. Salavati, "Hybrid fast unsupervised feature selection for high-dimensional data," *Expert Syst. Appl.*, vol. 124, pp. 97–118, Jun. 2019, doi: 10.1016/j.eswa.2019.01.016.

[14] M. Moran and G. Gordon, "Curious Feature Selection," *Inf. Sci. (Ny).*, vol. 485, pp. 42–54, Jun. 2019, doi: 10.1016/j.ins.2019.02.009.

[15] P. Drotár, M. Gazda, and L. Vokorokos, "Ensemble feature selection using election methods and ranker clustering," *Inf. Sci. (Ny).*, vol. 480, pp. 365–380, Apr. 2019, doi: 10.1016/j.ins.2018.12.033.

[16] D. Panday, R. Cordeiro de Amorim, and P. Lane, "Feature weighting as a tool for unsupervised feature selection," *Inf. Process. Lett.*, vol. 129, pp. 44–52, Jan. 2018, doi: 10.1016/j.ipl.2017.09.005.

[17] H. Dong, T. Li, R. Ding, and J. Sun, "A novel hybrid genetic algorithm with granular information for feature selection and optimization," *Appl. Soft Comput.*, vol. 65, pp. 33–46, Apr. 2018, doi: 10.1016/j.asoc.2017.12.048.

[18] J. L. Speiser, M. E. Miller, J. Tooze, and E. Ip, "A comparison of random forest variable selection methods for classification prediction modeling," *Expert Syst. Appl.*, vol. 134, pp. 93–101, Nov. 2019, doi: 10.1016/j.eswa.2019.05.028.

[19] F. Degenhardt, S. Seifert, and S. Szymczak, "Evaluation of variable selection methods for random forests and omics data sets," *Brief. Bioinform.*, vol. 20, no. 2, pp. 492–503, Mar. 2019, doi: 10.1093/bib/bbx124.

[20] D. Amaratunga, J. Cabrera, and Y.-S. Lee, "Enriched random forests," *Bioinformatics*, vol. 24, no. 18, pp. 2010–2014, Sep. 2008, doi: 10.1093/bioinformatics/btn356.

[21] M. Lu, "Embedded feature selection accounting for unknown data heterogeneity," *Expert Syst. Appl.*, vol. 119, pp. 350–361, Apr. 2019, doi: 10.1016/j.eswa.2018.11.006.

[22] C. O. Sakar *et al.*, "A comparative analysis of speech signal processing algorithms for Parkinson's disease classification and the use of the tunable Q-factor wavelet transform," *Appl. Soft Comput.*, vol. 74, pp. 255–263, Jan. 2019, doi: 10.1016/j.asoc.2018.10.022.

[23] B. Johnson and Z. Xie, "Classifying a high resolution image of an urban area using super-object information," *ISPRS J. Photogramm. Remote Sens.*, vol. 83, pp. 40–49, Sep. 2013, doi: 10.1016/j.isprsjprs.2013.05.008.

[24] B. A. Johnson, "High-resolution urban land-cover classification using a competitive multi-scale object-based approach," *Remote Sens. Lett.*, vol. 4, no. 2, pp. 131–140, Feb. 2013, doi: 10.1080/2150704X.2012.705440.

[25] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*, vol. 103. New York, NY: Springer New York, 2013, doi: 10.1007/978-1-4614-7138-7

[26] M. L. Bermingham *et al.*, "Application of high-dimensional feature selection: evaluation for genomic prediction in man," *Sci. Rep.*, vol. 5, no. 1, p. 10312, Sep. 2015, doi: 10.1038/srep10312.

[27] L. Yu and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution," in *Proceedings of the 20th international conference on machine learning (ICML-03)*, 2003, pp. 856–863, Available at: Google Scholar.

[28] J. Wang, Z. Feng, N. Lu, and J. Luo, "Toward optimal feature and time segment selection by divergence method for EEG signals classification," *Comput. Biol. Med.*, vol. 97, pp. 161–170, Jun. 2018, doi: 10.1016/j.compbiomed.2018.04.022.

[29] D. Bansal, R. Chhikara, K. Khanna, and P. Gupta, "Comparative Analysis of Various Machine Learning Algorithms for Detecting Dementia," *Procedia Comput. Sci.*, vol. 132, pp. 1497–1502, 2018, doi: 10.1016/j.procs.2018.05.102.

[30] X. Li, H. Wang, B. Gu, and C. X. Ling, "The convergence of linear classifiers on large sparse data," *Neurocomputing*, vol. 273, pp. 622–633, Jan. 2018, doi: 10.1016/j.neucom.2017.08.045.