

Research Article

A Novel Approach to Ensemble Classifiers: FsBoost-Based Subspace Method

Adeeb Noor ¹, Muhammed Kürşad Uçar,² Kemal Polat ³, Abdullah Assiri,⁴
and Redhwan Nour⁵

¹Department of Information Technology, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 80221, Saudi Arabia

²Sakarya University, Faculty of Engineering, Electrical-Electronics Engineering, Sakarya 54187, Turkey

³Department of Electrical and Electronics Engineering, Faculty of Engineering, Bolu Abant İzzet Baysal University, Bolu 14280, Turkey

⁴Department of Clinical Pharmacy, College of Pharmacy, King Khalid University, Abha 62529, Saudi Arabia

⁵Department of Computer Science, Taibah University, Medina 42353, Saudi Arabia

Correspondence should be addressed to Kemal Polat; kpolat@ibu.edu.tr

Received 10 April 2020; Revised 5 June 2020; Accepted 18 June 2020; Published 15 July 2020

Academic Editor: Elio Masciari

Copyright © 2020 Adeeb Noor et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this article, an algorithm is proposed for creating an ensemble classifier. The name of the algorithm is the F-score subspace method (FsBoost). According to this method, the features are selected with the F-score and classified with different or the same classifiers. In the next step, the ensemble classifier is created. Two versions that are named FsBoost.V1 and FsBoost.V2 have been developed based on classification by the same or different classifiers. According to the results obtained, the results are consistent with the literature. Besides, a higher accuracy rate is obtained compared with many algorithms in the literature. The algorithm is fast because it has a few steps. It is thought that the algorithm will be successful due to these advantages.

1. Introduction

An ensemble classifier is a method in which multiple classifications are used together to improve classification performance [1, 2]. For example, when three classifiers are used to classify an object, the classifier works like this: if the first classifier is classified as a cat, the second classifier is classified as a dog, and the third classifier is classified as a cat, the ensemble classifier generates the result by taking the average of these decisions. There are many ways to create an ensemble classifier. Some of the most commonly used ones are (1) adaptive resampling and combining (boosting) [3], (1.1) AdaBoost (adaptive boosting) [4], (2) bagging (bootstrap aggregating) [5], and (3) random subspace [6].

The boosting method can create powerful classifiers by combining and training weak classifiers [3]. The most commonly used boosting method is AdaBoost [4]. The AdaBoost method tries to improve performance by focusing

on misclassified instances [4]. In the bagging method, classifiers trained with different training sets randomly selected (random sampling method) from the dataset are combined [5]. Outputs of classifiers are combined with majority voting or weighted voting [5]. In the random subspace method, feature subsets are generated by randomly selecting from N samples [6]. Each subset has an M element and an m feature [6]. In other words, a subset of features is created, not a subset of instances [6]. In this way, the training process is accelerated. These subclasses and classifiers are trained to form ensemble classifiers. The outputs of the classifiers are combined with majority voting or weighted voting.

These algorithms have a disadvantage. As the education levels for AdaBoost increase, the number of samples decreases, and training becomes more difficult. There is a need for more samples for training. It is quite slow because the training stages are too much [7, 8]. The bagging method

involves complex calculations [7]. Both methods require many iterations [1]. So, the success rate is usually lower than the random forest method [1]. These models cannot explain the dataset by modeling it as decision trees [7]. When these disadvantages are taken into consideration, these methods are still in need of improvement. In this study, a new ensemble algorithm based on the F-score feature selection algorithm has been developed to reduce the processing load of existing ensemble algorithms and to increase the accuracy rate.

Feature selection algorithms are often used in the machine learning field to improve the performance of systems [9–11]. In the field of machine learning, datasets are used in a variety of sizes and types [12–15]. Large size data will cause the classifier to lengthen the training duration. Feature selection algorithms have been developed to solve this problem [9, 16, 17]. They do this by clearing irrelevant data when holding relevant data [9]. Thus, data size, process load, and training time decrease while classification accuracy increases [9, 18]. Many feature selection algorithms have been developed in the literature [1, 16]. However, in this study, the F-score feature selection algorithm is used because it can work fast, and its performance is good [16]. Feature selection algorithms can be used in many places such as health areas [19–22].

In this study, two different methods have been developed, namely, FsBoost.V1 and FsBoost.V2, based on the F-score feature selection algorithm that can enhance training performance for ensemble classifiers. The FsBoost.V1 method is like the random subspace method. However, the features are chosen concerning the data label, not random. Selected datasets are classified with a single classifier, and then ensemble classifier 1 is created. This process is repeated for three or more different classifiers. Eventually, ensemble classifiers for three different classifiers are merged. In this way, it is ensured that unnecessary data are removed from the training process. The operation can be interrupted first in the ensemble classifier. In the FsBoost.V2 method, all data are classified with different classifiers. In the second step, the subfeature space is created by the F-score feature selection algorithm and reclassified. The ensemble classifier is created because of classification. This process was repeated a second time. Eventually, ensemble classifiers for three different classifiers are merged. The use of a single classifier reduces the cost. Only relevant features are retrieved by using the F-score feature selection algorithm. This process accelerates the training process. Complexity is less than other algorithms.

2. Materials and Methods

The operation was performed according to the flow in Figure 1. Firstly, the records to be used in the study were collected. Then, features were selected with the F-score feature selection algorithm. Finally, the data are classified with different classifiers, and their performances are calculated. When these operations are performed, ensemble classifiers are created, and their performances are calculated at different levels and formations.

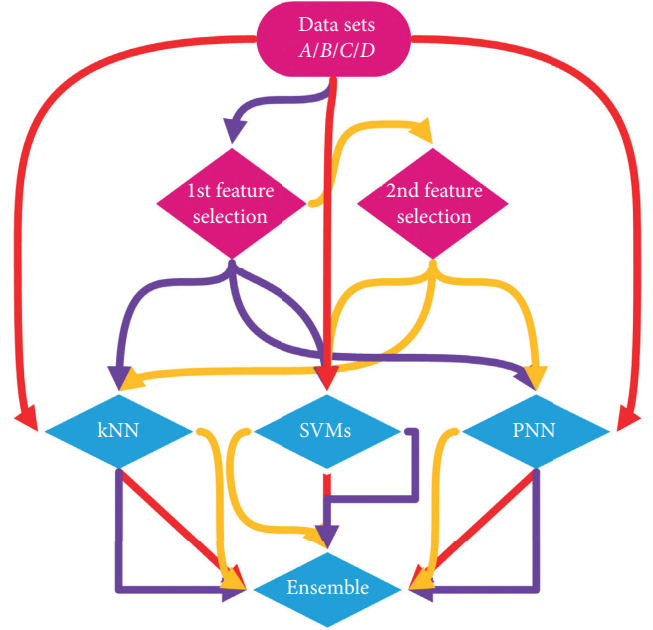


FIGURE 1: Process flow diagram.

2.1. Collection of Data. The data used in the study were downloaded from the Machine Learning Repository website of the University of California, Irvine (UCI) [23, 24]. The data consist of 4 groups (A/B/C/D) belonging to epilepsy patients (Table 1). Records include EEG records of individuals. Each record is 23.6 seconds. 2300 EEG recordings were taken during the epileptic seizure. The other 2300 records (nonepilepsy) were recorded while in a healthy condition. However, the records belong to epileptic patients. The epilepsy data in each set are the same. However, nonepilepsy records are different. The database contains 178 features for each EEG recording.

2.2. F-Score Feature Selection Algorithm. The F-score is one of the feature selection algorithms that helps distinguish classes from each other [25]. To select the feature, an F-score value (F_i) is calculated for each feature (equation (1)). The F-score threshold value (F_E) is determined by taking the average of all F-score values. For the i th feature, if $F_i > F_E$, i th feature is selected. This step is repeated for each feature.

$$F(i) = \frac{(\bar{x}_i^{(+)} - \bar{x}_i)^2 + (\bar{x}_i^{(-)} - \bar{x}_i)^2}{(1/n_+ - 1) \sum_{k=1}^{n_+} (x_{k,i}^{(+)} - \bar{x}_i^{(+)})^2 + (1/n_- - 1) \sum_{k=1}^{n_-} (x_{k,i}^{(-)} - \bar{x}_i^{(-)})^2} \quad (1)$$

The variables in equation (1) are (1) feature vector $x_{k,i}$, and $k = 1, 2, \dots, m$. (2) n_+ , n_- , respectively, are the positive (+) and negative (-) total number of elements in the class, and $m = n_+ + n_-$. (3) i is the feature number. (4) \bar{x}_i , $\bar{x}_i^{(-)}$, and $\bar{x}_i^{(+)}$ are the average value, mean value in the negative class, and mean value in the positive class of the i th property, respectively. (5) $x_{k,i}^{(+)}$ represents the k th positive example of the i th feature. $x_{k,i}^{(-)}$ represents the k th negative example of the i th feature.

TABLE 1: Dataset distributions.

Dataset	Group		Total	Number of features
	Epilepsy	Nonepilepsy		
A	1150	1150	2300	178
B	1150	1150	2300	178
C	1150	1150	2300	178
D	1150	1150	2300	178

In the study, A, B, C, and D dataset features were selected with the F-score (Table 2). Feature selection has been applied twice.

2.3. Ensemble Classifier. The ensemble classifier is a system created by combining different classifiers to produce safer and more stable estimates [26]. The system is built with N classifiers. N can be single or double. While classifying according to the feature vector, for each feature vector 1 , each classifier generates an output value. The output values produced are counted. Then, the output of the ensemble classifier is determined by the number of votes. If the number of classifiers is even, the average of the decision values of the classifiers is rounded off, and the decision of the ensemble classifier is determined. This process applies to all feature vectors. The ensemble classifier was prepared in MATLAB using three different classifiers: kNN, PNN, and SVMs [27].

The kNN is one of the machine learning classification methods with advisory learning [28, 29]. Under the structure of the training dataset, classification is done according to nearest k of the new classifier. In this study, $k = 5$ was selected, and ten distance calculation formulas were used. These include Spearman, Seclidean, Minkowski, Mahalanobis, Jaccard, Hamming, Euclidean, Cosine, Correlation, and Cityblock.

PNN is a statistical classification algorithm based on kernel and Bayesian [30]. The method is developed based on feedforward networks [30]. The classifier takes care of all class elements when processing [31]. The radial-based kernel function calculates the distance between class samples. The user in the PNN classifier can manipulate the spread parameter. As the spread parameter approaches zero, the network begins to behave like the nearest neighbor classifier [32]. This value when farther away from zero, the classifier classifies, considering several vectors that separate data from each other [32]. In the study, PNN networks were designed with a total of 500 different values ranging from 0.01 to 5 steps of the spread parameter, with 0.01 step range. At the end of the study, the best performing network parameters and performance criteria were calculated.

SVMs are among the best machine learning algorithms [33]. They can be used in the regression analysis as well as classification [33]. SVMs try to separate datasets from each other with a linear and nonlinear line. The purpose of the SVM algorithm is to be able to distinguish between the data with the minimum error [34]. Gaussian or radial basis function (RBF) kernel (rbf) was used in the study. The BoxConstraint box limit is set between 1 and 100 so that the best performance can be achieved.

TABLE 2: Selected features with the F-score for datasets.

Dataset	A	B	C	D
Number of features	178	178	178	178
1st feature selection	60	68	62	54
2nd feature selection	28	25	23	25

2.4. Ensemble Classifier Powered by the F-Score. In this study, two different ensemble classifiers, namely, Classifier-FsBoost.V1 and Feature-FsBoost.V2, were developed.

2.4.1. Classifier-Based Ensemble Classifier: FsBoost.V1. The implementation steps of this method are shown in detail in Figure 2. Accordingly to this, firstly, a dataset (A) is classified in a classifier (kNN). In the second step, the first feature selection is performed and again classified in the same classifier (kNN). In the third step, the first and second feature selection are performed and again classified in the same classifier (kNN). Thus, it is classified in three different steps, but only in a classifier (kNN). These three results are combined to form the kNN ensemble. The same process is repeated in PNN and SVMs. Eventually, the kNN ensemble, the PNN ensemble, and the SVM ensemble are combined into a single ensemble classifier.

2.4.2. Feature-Based Ensemble Classifier: FsBoost.V2. The steps for this method are shown in Figure 3. Accordingly to this, firstly, a dataset (A) is classified by each classifier (kNN, PNN, and SVMs). These three classifiers are combined to obtain ensemble classifier 1. In the second step, the first feature selection is performed, and the process in the first step is repeated. In the third step, the first and second property selection steps are performed together, and then the first process is repeated. Ensemble 1, 2, and 3 classifiers are combined to create the ensemble classifier.

2.5. Performance Evaluation Criteria and Distribution of Data for Classification. Different performance evaluation criteria were used to test the accuracy rates of the proposed systems. These are accuracy rates, sensitivity, specificity, kappa value, receiver operating characteristic (ROC), area under a ROC (AUC), and k (10-fold) cross-validation accuracy.

While classifying the datasets, they were divided into two groups: training (50%) and test (50%) (Table 3).

3. Results

The work aims to develop a new algorithm to improve the ensemble classifier performance. We have developed an algorithm (FsBoost) that is similar to the random subspace method but with less workload, faster running, and better performance. F-score feature selection algorithm based on this method has two versions (FsBoost.V1 vs. FsBoost.V2). The ensemble classifier is created with a single classifier in FsBoost.V1 (Şekil 2, Level 1) and at least three different classifiers in FsBoost.V2 (Şekil 3, Level 1). The developed

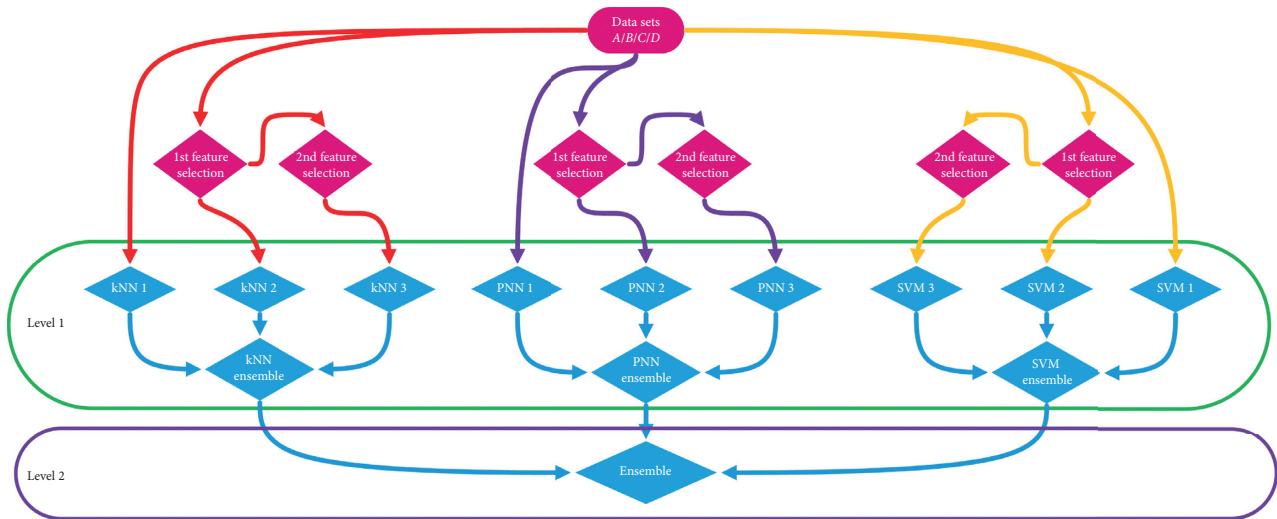


FIGURE 2: FsBoost.V1—classifier-based ensemble classifier.

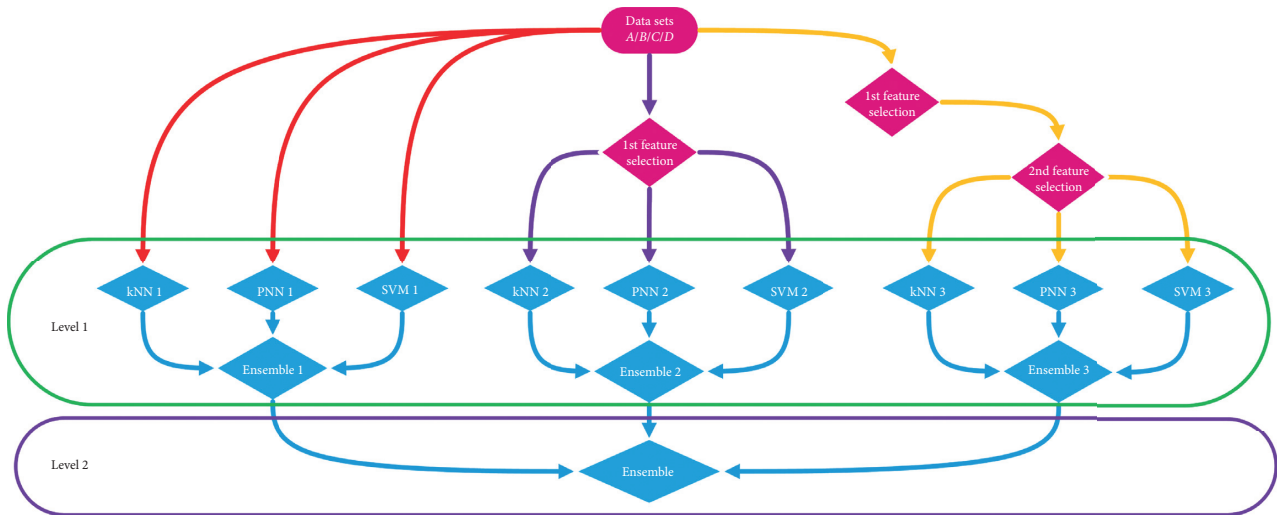


FIGURE 3: FsBoost.V2—feature-based ensemble classifier.

TABLE 3: Training and testing data distribution.

Class	For A dataset			For B dataset			For C dataset			For D dataset		
	Training (50%)	Test (50%)	Total	Training (50%)	Test (50%)	Total	Training (50%)	Test (50%)	Total	Training (50%)	Test (50%)	Total
Epilepsy	1150	1150	2300	1150	1150	2300	1150	1150	2300	1150	1150	2300
Nonepilepsy	1150	1150	2300	1150	1150	2300	1150	1150	2300	1150	1150	2300
Total	2300	2300	4600	2300	2300	4600	2300	2300	4600	2300	2300	4600

algorithms were tested with four two-class datasets (A, B, C, and D) (Table 3).

According to the FsBoost algorithm, the dataset features were selected twice using the F-score feature selection algorithm. For example, according to FsBoost.V1, the dataset (A) is classified with the same classifier after each property selection (Figure 2, Level 1—kNN1, kNN2, and kNN3) (Table 4). kNN ensemble was formed by

combining classifiers of three kNNs (Figure 2, Level 1) (Table 5). This process was repeated with three different classifiers to create PNN ensemble and SVM ensemble (Figure 2, Level 1) (Table 5). Then, the kNN ensemble, PNN ensemble, and SVM ensemble were combined to form the final ensemble classifier (Figure 2, Level 2) (Table 5). This process is repeated for each dataset (Tables 4–8).

TABLE 4: Level 1 classification results for the *A* dataset.

k-nearest neighbor classification algorithm									
<i>k</i> = 2			<i>k</i> = 2				<i>k</i> = 2		
NP	Euclidean			DF			Euclidean		
FS	0			1			2		
NF	68			25			15		
Class	Sen	Spe	Acc	Sen	Spe	Acc	Sen	Spe	Acc
E	0.84	1.00	92.17	0.87	0.89	93.26	0.89	1.00	94.57
N-E	1.00	0.84		1.00	1.00		1.00	0.89	
AUC	0.92			0.87			0.94		
Kappa	0.84			0.93			0.88		
F-measure	0.92			0.93			0.93		
10-fold (%)	88.85			91.22			92.54		
Probabilistic neural networks									
NP	Spread			Spread			Spread		
FS	0			1			2		
NF	68			25			15		
Class	Sen	Spe	Acc	Sen	Spe	Acc	Sen	Spe	Acc
E	0.87	1.00	93.26	0.94	0.75	93.09	0.75	1.00	87.70
N-E	1.00	0.87		0.92	1.00		1.00	0.75	
AUC	0.93			0.86			0.94		
Kappa	0.87			0.93			0.89		
F-measure	0.93			0.93			0.94		
10-fold (%)	0.11			0.11			0.21		
Support vector machines									
NP	3			BoxConstraint			2		
FS	0			1			2		
NF	68			25			15		
Class	Sen	Spe	Acc	Sen	Spe	Acc	Sen	Spe	Acc
E	0.99	1.00	99.65	0.99	0.98	99.39	0.98	0.99	98.83
N-E	1.00	0.99		1.00	0.99		0.99	0.98	
AUC	1.00			0.99			0.99		
Kappa	0.99			0.99			0.99		
F-measure	1.00			0.99			0.99		
10-fold (%)	99.76			99.54			99.02		

DF: distance function, Sen: sensitivity, Spe: specificity, Acc: accuracy (%), NP: network parameters, FS: feature selection, NF: number of features, EC: ensemble classifier, E: epilepsy, and N-E: nonepilepsy.

In FsBoost.V2, the dataset (*A*) is classified with different classifiers after each feature selection (Figure 3, Level 1—kNN1, PNN1, and SVM1) (Table 4). These three classifiers were combined to create ensemble 1 (Figure 3, Level 1—ensemble 1) (Table 9). Then, ensemble 1, ensemble 2, and ensemble 3 were combined to form the final ensemble classifier (Figure 3, Level 2) (Table 9). This process is repeated for each dataset (Tables 4–7 and 9). Finally, the FsBoost ensemble algorithm is also compared with the ensemble algorithms available in the literature (Table 10).

Accuracy rates for FsBoost.V1 and FsBoost.V2 are higher than those for single classifiers (Table 10). The FsBoost algorithm is well ranked compared to other boosting algorithms in the literature (Table 10). FsBoost.V1—Level 1—SVM ensemble method is the best method when compared with the literature (Table 10, Rank).

Three different datasets were used to reconfirm the results obtained. The distribution of datasets is shown in Table 11.

In order to compare the FsBoost algorithm with boosting algorithms, three different datasets were reanalyzed. The results obtained from the analysis are summarized in Table 12. According to the results, the algorithm with the average best performance is the FsBoost.V1 Level 2 ensemble algorithm.

4. Discussion and Conclusion

FsBoost is one of the best algorithms developed until now [4–7]. This method has very few steps. In this way, it provides results faster. A high accuracy rate is a distinct advantage. Algorithms with high accuracy and fast results are preferred in medical data classification. In this regard, FsBoost may be preferred.

FsBoost contains fewer calculations and steps than the algorithms in the literature [4–7]. The accuracy rate is very good compared with other algorithms (Table 10) [4].

TABLE 5: Results for the FsBoost.V1 ensemble algorithm (Levels 1 and 2).

Level Classifier	Level 1									Level 2		
	kNN ensemble			PNN ensemble			SVM ensemble			Ensemble		
Class	Sen	Spe	Acc	Sen	Spe	Acc	Sen	Spe	Acc	Sen	Spe	Acc
<i>For A dataset</i>												
E	0.88	1.00	93.87	0.89	1.00	94.26	0.99	1.00	99.48	0.93	1.00	96.43
N-E	1.00	0.88		1.00	0.89		1.00	0.99		0.99	1.00	
AUC		0.94			0.94			0.99		0.99		
Kappa		0.88			0.89			0.99		0.99		
F-measure		0.93			0.94			0.99		0.99		
<i>For B dataset</i>												
E	0.85	1.00	92.48	0.76	1.00	87.83	0.96	0.99	97.48	0.87	1.00	93.52
N-E	1.00	0.85		1.00	0.76		1.00	0.99		0.96	1.00	
AUC		0.92			0.88			0.97		0.97		
Kappa		0.85			0.76			0.95		0.95		
F-measure		0.92			0.86			0.97		0.97		
<i>For C dataset</i>												
E	0.84	0.99	91.91	0.80	0.99	89.17	0.97	0.98	97.43	0.88	0.99	93.78
N-E	0.99	0.84		0.99	0.80		0.99	0.98		0.97	0.99	
AUC		0.92			0.89			0.97		0.97		
Kappa		0.84			0.78			0.95		0.95		
F-measure		0.91			0.88			0.97		0.97		
<i>For D dataset</i>												
E	0.83	0.97	89.87	0.64	0.96	80.13	0.95	0.93	94.04	0.86	0.96	91.30
N-E	0.97	0.83		0.96	0.64		0.96	0.93		0.95	0.96	
AUC		0.90			0.80			0.94		0.94		
Kappa		0.80			0.60			0.88		0.88		
F-measure		0.89			0.77			0.94		0.94		

Sen: sensitivity, Spe: specificity, Acc: accuracy (%), E: epilepsy, and N-E: nonepilepsy.

TABLE 6: Level 1 classification results for the C dataset.

<i>k</i> -nearest neighbor classification algorithm									
NP	<i>k</i> = 2			<i>k</i> = 2			<i>k</i> = 2		
	Euclidean			DF Euclidean			Minkowski		
FS		0			1			2	
NF		68			25			15	
Class	Sen	Spe	Acc	Sen	Spe	Acc	Sen	Spe	Acc
E	0.80	1.00	90.00	0.83	0.88	90.96	0.88	0.96	92.04
N-E	1.00	0.80		0.99	0.96		0.96	0.88	
AUC		0.90			0.82			0.92	
Kappa		0.80			0.90			0.84	
F-measure		0.89			0.90.96			0.91	
10-fold (%)		85.39		87.54		90.41			
Probabilistic neural networks									
NP		Spread			Spread			Spread	
FS		0.11			0.21			0.21	
NF		68			25			15	
Class	Sen	Spe	Acc	Sen	Spe	Acc	Sen	Spe	Acc
E	0.80	0.98	89.04	0.79	0.78	88.91	0.78	0.88	82.96
N-E	0.98	0.80		0.99	0.88		0.88	0.78	
AUC		0.89			0.78			0.89	
Kappa		0.78			0.88			0.78	
F-measure		0.88			88.91			0.88	
10-fold (%)		0.11		0.21		0.21			

TABLE 6: Continued.

Support vector machines									
				BoxConstraint					
NP									
NP	4			88			4		
FS	0			1			2		
NF	68			25			15		
Class	Sen	Spe	Acc	Sen	Spe	Acc	Sen	Spe	Acc
E	0.98	0.99	98.13	0.97	0.95	97.17	0.95	0.96	95.57
N-E	0.99	0.98		0.97	0.96		0.96	0.95	
AUC	0.98			0.94			0.97		
Kappa	0.96			0.97			0.95		
F-measure	0.98			0.97			0.97		
10-fold (%)	98.48			97.37			95.41		

DF: distance function, Sen: sensitivity, Spe: specificity, Acc: accuracy (%), NP: network parameters, FS: feature selection, NF: number of features, EC: ensemble classifier, E: epilepsy, and N-E: nonepilepsy.

TABLE 7: Level 1 classification results for the *D* dataset.

<i>k</i> -nearest neighbor classification algorithm									
<i>k</i> = 2				<i>k</i> = 1			<i>k</i> = 5		
Euclidean				Minkowski			Euclidean		
NP				DF					
FS	0			1			2		
NF	68			25			15		
Class	Sen	Spe	Acc	Sen	Spe	Acc	Sen	Spe	Acc
E	0.79	0.97	88.04	0.82	0.84	89.17	0.84	0.92	88.13
N-E	0.97	0.79		0.97	0.92		0.92	0.84	
AUC	0.88			0.78			0.90		
Kappa	0.76			0.89			0.80		
F-measure	0.87			0.89			0.89		
10-fold (%)	84.24			86.30			87.17		
Probabilistic neural networks									
Spread				Spread			Spread		
NP	0.41			0.41			0.41		
FS	0			1			2		
NF	68			25			15		
Class	Sen	Spe	Acc	Sen	Spe	Acc	Sen	Spe	Acc
E	0.68	0.97	82.43	0.64	0.53	79.52	0.53	0.95	74.09
N-E	0.97	0.68		0.95	0.95		0.95	0.53	
AUC	0.82			0.59			0.80		
Kappa	0.65			0.76			0.60		
F-measure	0.80			0.77			0.77		
10-fold (%)	0.41			0.41			0.41		
Support vector machines									
				BoxConstraint					
NP	1			21			2		
FS	0			1			2		
NF	68			25			15		
Class	Sen	Spe	Acc	Sen	Spe	Acc	Sen	Spe	Acc
E	0.97	0.93	94.61	0.93	0.93	93.78	0.93	0.92	92.30
N-E	0.93	0.97		0.94	0.92		0.92	0.93	
AUC	0.95			0.88			0.94		
Kappa	0.89			0.94			0.88		
F-measure	0.95			0.93			0.94		
10-fold (%)	95.43			94.54			92.33		

DF: distance function, Sen: sensitivity, Spe: specificity, Acc: accuracy (%), NP: network parameters, FS: feature selection, NF: number of features, EC: ensemble classifier, E: epilepsy, and N-E: nonepilepsy.

Considering these advantages, FsBoost may be a commonly used algorithm soon.

FsBoost algorithms are also suitable for use in biomedical signal processing, deep learning, and communication [35–37].

FsBoost can be used with three or more classifiers. Besides, FsBoost.V1 is a version of FsBoost that can be used with a single classifier. Achieving high performance with a single classifier is a distinct advantage of FsBoost.V1. The F-score feature selection algorithm creates this advantage.

TABLE 8: Level 1 classification results for the B dataset.

<i>k</i> -nearest neighbor classification algorithm									
	<i>k</i> = 2			<i>k</i> = 2			<i>k</i> = 4		
NP	Euclidean			DF			Euclidean		
FS	0			1			2		
NF	68			25			15		
Class	Sen	Spe	Acc	Sen	Spe	Acc	Sen	Spe	Acc
E	0.84	1.00	91.96	0.85	0.83	92.30	0.83	0.98	90.52
N-E	1.00	0.84		0.99	0.98		0.98	0.83	
AUC	0.92			0.85			0.92		
Kappa	0.84			0.92			0.85		
F-measure	0.91			92.30			0.92		
10-fold (%)	88.13			89.72			90.50		
Probabilistic neural networks									
	Spread			Spread			Spread		
NP	0.11			0.21			0.31		
FS	0			1			2		
NF	68			25			15		
Class	Sen	Spe	Acc	Sen	Spe	Acc	Sen	Spe	Acc
E	0.84	0.98	91.00	0.79	0.58	89.17	0.58	1.00	78.65
N-E	0.98	0.84		0.99	1.00		1.00	0.58	
AUC	0.91			0.78			0.88		
Kappa	0.82			0.88			0.76		
F-measure	0.90			89.17			0.86		
10-fold (%)	0.11			0.21			0.31		
Support vector machines									
	BoxConstraint			BoxConstraint			BoxConstraint		
NP	4			15			3		
FS	0			1			2		
NF	68			25			15		
Class	Sen	Spe	Acc	Sen	Spe	Acc	Sen	Spe	Acc
E	0.97	1.00	98.48	0.96	0.92	97.22	0.92	0.97	94.48
N-E	1.00	0.97		0.99	0.97		0.97	0.92	
AUC	0.98			0.94			0.97		
Kappa	0.97			0.97			0.95		
F-measure	0.98			97.22			0.97		
10-fold (%)	99.22			97.35			94.07		

DF: distance function, Sen: sensitivity, Spe: specificity, Acc: accuracy (%), NP: network parameters, FS: feature selection, NF: number of features, EC: ensemble classifier, E: epilepsy, and N-E: nonepilepsy.

TABLE 9: Results for the FsBoost.V2 ensemble algorithm (Levels 1 and 2).

Level	Level 1						Level 2					
Classifier	Ensemble 1			Ensemble 2			Ensemble 3			Ensemble		
<i>For A dataset</i>												
FS	0			1			2			0/1/2		
NF	68			25			15			68/25/15		
Class	Sen	Spe	Acc	Sen	Spe	Acc	Sen	Spe	Acc	Sen	Spe	Acc
E	0.92	1.00	95.91	0.96	1.00	98.00	0.91	1.00	95.52	0.94	1.00	97.22
N-E	1.00	0.92		1.00	0.96		1.00	0.91		1.00	0.94	
AUC	0.96			0.98			0.96			0.97		
Kappa	0.92			0.96			0.91			0.94		
F-measure	0.96			0.98			0.95			0.97		
<i>For B dataset</i>												
FS	0			1			2			0/1/2		
NF	68			25			15			68/25/15		
Class	Sen	Spe	Acc	Sen	Spe	Acc	Sen	Spe	Acc	Sen	Spe	Acc
E	0.90	1.00	94.83	0.89	0.99	94.17	0.83	0.99	91.09	0.88	1.00	94.04
N-E	1.00	0.90		0.99	0.89		0.99	0.83		1.00	0.88	

TABLE 9: Continued.

Level Classifier	Level 1						Level 2					
	Ensemble 1			Ensemble 2			Ensemble 3			Ensemble		
AUC	0.95			0.94			0.91			0.94		
Kappa	0.90			0.88			0.82			0.88		
F-measure	0.95			0.94			0.90			0.94		
<i>For C dataset</i>												
FS	0			1			2			0/1/2		
NF	68			25			15			68/25/15		
Class	Sen	Spe	Acc	Sen	Spe	Acc	Sen	Spe	Acc	Sen	Spe	Acc
E	0.87	1.00	93.30	0.88	0.99	93.39	0.90	0.96	93.35	0.90	0.99	94.52
N-E	1.00	0.87		0.99	0.88		0.96	0.90		0.99	0.90	
AUC	0.93			0.93			0.93			0.95		
Kappa	0.87			0.87			0.87			0.89		
F-measure	0.93			0.93			0.93			0.94		
<i>For D dataset</i>												
FS	0			1			2			0/1/2		
NF	68			25			15			68/25/15		
Class	Sen	Spe	Acc	Sen	Spe	Acc	Sen	Spe	Acc	Sen	Spe	Acc
E	0.84	0.97	90.39	0.85	0.97	91.09	0.86	0.94	89.57	0.87	0.97	91.65
N-E	0.97	0.84		0.97	0.85		0.94	0.86		0.97	0.87	
AUC	0.90			0.91			0.90			0.92		
Kappa	0.81			0.82			0.79			0.83		
F-measure	0.90			0.91			0.89			0.91		

Sen: sensitivity, Spe: specificity, Acc: accuracy (%), FS: feature selection, NF: number of features, EC: ensemble classifier, E: epilepsy, and N-E: nonepilepsy.

TABLE 10: Comparing the FsBoost algorithm with single classifiers and boosting algorithms.

Methods	Datasets									
	A		B		C		D			
	Rank	Acc	Rank	Acc	Rank	Acc	Rank	Acc	Rank	Acc
AdaBoostM1	3	99.04	2	97.22	4	96.48	3	94.35		
Bag	2	99.43	4	96.87	2	96.83	1	94.87		
GentleBoost	4	98.96	3	97.04	3	96.78	2	94.65		
LogitBoost	5	98.96	5	96.35	5	96.35	4	94.22		
LPBoost	6	98.57	6	96.35	7	94.74	7	92.83		
RobustBoost	12	95.74	16	87.78	17	86.35	15	87.17		
RUSBoost	17	90.83	17	84.48	16	86.43	16	84.39		
Subspace	15	94.20	10	94.15	13	92.02	11	90.54		
TotalBoost	7	98.35	7	95.78	6	95.87	6	93.30		
<i>FsBoost.V1</i>										
Level 1—kNN ensemble	16	93.87	13	92.48	14	91.91	13	89.87		
Level 1—PNN ensemble	14	94.26	15	87.83	15	89.17	17	80.13		
Level 1—SVM ensemble	1	99.48	1	97.48	1	97.43	5	94.04		
Level 2—ensemble	9	97.22	11	94.04	8	94.52	8	91.65		
<i>FsBoost.V2</i>										
Level 1—ensemble 1	11	95.91	8	94.83	12	93.30	12	90.39		
Level 1—ensemble 2	8	98.00	9	94.17	10	93.39	10	91.09		
Level 1—ensemble 3	13	95.52	14	91.09	11	93.35	14	89.57		
Level 2—ensemble	10	96.43	12	93.52	9	93.78	9	91.30		
kNN		92.17		91.96		90.00		88.04		
PNN		93.26		91.00		89.04		82.43		
SVMs		99.65		98.48		98.13		94.61		

Acc: accuracy (%).

By combining different features, the same data can be interpreted differently. If the classifiers are strong, FsBoost increases in performance. Therefore, it is recommended that

the algorithm is used with robust classifiers. Ensemble classifiers often bring out a strong classifier by combining weak classifiers. This is the weakness of FsBoost.

TABLE 11: Comparing the FsBoost algorithm with single classifiers and boosting algorithms for other datasets.

Datasets	NF	Training (50%)		Total	Test (50%)		Total
		Class 1	Class 2		Class 1	Class 2	
Basehock	89	497	500	997	497	499	996
Madelon	95	680	620	1300	620	680	1300
PCMAC	87	491	481	972	491	480	971

NF: number of features.

TABLE 12: Comparing the FsBoost algorithm with single classifiers and boosting algorithms for other datasets.

Method	Datasets							
	Basehock		Madelon		PCMAC		Mean	
	Rank	Acc	Rank	Acc	Rank	Acc	Acc	Rank
AdaBoostM1	5	59.34	17	52.92	2	63.75	58.67	5
Bag	17	50.90	8	55.00	17	52.73	52.88	17
GentleBoost	13	58.23	15	53.54	5	62.82	58.20	9
LogitBoost	12	58.43	14	53.69	3	63.54	58.56	6
LPBoost	16	56.22	7	55.15	16	52.83	54.74	16
RobustBoost	4	59.54	13	53.77	1	63.95	59.09	2
RUSBoost	15	56.33	6	56.00	12	56.02	56.12	14
Subspace	11	58.55	16	53.31	15	54.04	55.30	15
TotalBoost	6	59.04	12	54.38	4	62.82	58.75	4
FsBoost.V1								
Level 1 kNN ensemble	10	58.73	3	56.85	14	54.27	56.62	12
Level 1 PNN ensemble	2	59.74	2	56.92	11	58.81	58.49	8
Level 1 SVM ensemble	7	58.84	11	54.54	10	60.14	57.84	11
Level 2 ensemble	1	60.14	4	56.62	9	60.56	59.10	1
FsBoost.V2								
Level 1—ensemble 1	9	58.84	10	54.62	6	62.20	58.55	7
Level 1—ensemble 2	8	58.84	9	54.69	7	60.87	58.13	10
Level 1—ensemble 3	14	57.33	1	57.38	13	54.89	56.54	13
Level 2 ensemble	3	59.74	5	56.54	8	60.76	59.01	3
kNN		59.04		54.23		54.58		
PNN		57.23		53.00		58.39		
SVMs		56.22		54.69		61.89		

Acc: accuracy (%).

As a result, we can say that FsBoost is an alternative method to create an ensemble classifier. A high-performance ensemble classifier can be created with a powerful classifier and the F-score feature selection algorithm.

Data Availability

The datasets in our paper could be downloaded from the UCI Machine Learning Repository (<https://archive.ics.uci.edu/ml/datasets/index.html>). The authors can send all the datasets based on the readers' requests.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This project was funded by the Deanship of Science Research (DSR) at King Abdulaziz University, Jeddah, Saudi Arabia, under grant no. RG-2-611-40. The authors, therefore,

acknowledge with thanks to DSR for the technical and financial support.

References

- [1] L. Rokach, "Ensemble-based classifiers," *Artificial Intelligence Review*, vol. 33, no. 1-2, pp. 1-39, 2010.
- [2] M. K. Uçar, "Classification performance-based feature selection algorithm for machine learning: P-score," *Innovation and Research in BioMedical Engineering*, 2020.
- [3] Y. Freund and E. Robert, "Schapire. Experiments with a new boosting algorithm," in *Proceedings of the ICML '96: 13th International Conference on Machine Learning*, pp. 148-156, Bari Italy, July 1996.
- [4] R. Rojas, *AdaBoost and the Super Bowl of Classifiers A Tutorial Introduction to Adaptive Boosting*, pp. 1-6, 2009.
- [5] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123-140, 1996.
- [6] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 832-844, 1998.

- [7] B. Peter, "Bagging, boosting and ensemble learning," in *Handbook of Computational Statistics: Concepts and Methods*, J. E. Gentle, W. Karl Härdle, and Y. Mori, Eds., pp. 1–38, Springer-Verlag Berlin Heidelberg, Heidelberg, Germany, 2012.
- [8] M. Toğaçar, B. Ergen, and Z. Cömert, "A deep feature learning model for pneumonia detection applying a combination of mRMR feature selection and machine learning models," *Innovation and Research in BioMedical Engineering*, 2019.
- [9] D. Guan, W. Yuan, Y.-K. Lee, K. Najeebullah, and M. K. Rasel, "A review of ensemble learning based feature selection," *IETE Technical Review*, vol. 31, no. 3, pp. 190–198, 2014.
- [10] N. Daldal, K. Polat, and Y. Guo, "Classification of multi-carrier digital modulation signals using NCM clustering based feature-weighting method," *Computers in Industry*, vol. 109, p. 45, 2019.
- [11] M. K. Uçar, M. Nour, H. Sindi, and K. Polat, "The effect of training and testing process on machine learning in biomedical datasets," *Mathematical Problems in Engineering*, vol. 2020, Article ID 2836236, 17 pages, 2020.
- [12] K. Polat, S. Şahan, H. Kodaz, and S. Güneş, "Breast cancer and liver disorders classification using artificial immune recognition system (AIRS) with performance evaluation by fuzzy resource allocation mechanism," *Expert Systems with Applications*, vol. 32, no. 1, pp. 172–183, 2007.
- [13] S. AlMuhaideb and M. E. B. Menai, "An individualized preprocessing for medical data classification," *Procedia Computer Science*, vol. 82, pp. 35–42, 2016.
- [14] N. Daldal, M. Nour, and K. Polat, "A novel demodulation structure for quadrature modulation signals using the segmentary neural network modelling," *Applied Acoustics*, vol. 164, Article ID 107251, 2020.
- [15] N. Daldal, "A novel demodulation method for quadrature type modulations using a hybrid signal processing method," *Physica A: Statistical Mechanics and Its Applications*, vol. 540, Article ID 122836, 2020.
- [16] K. Polat and S. Güneş, "A new feature selection method on classification of medical datasets: kernel F -score feature selection," *Expert Systems with Applications*, vol. 36, no. 7, pp. 10367–10373, 2009.
- [17] K. Polat and K. Onur Koc, "Detection of skin diseases from dermoscopy image using the combination of convolutional neural network and one-versus-all," *Journal of Artificial Intelligence and Systems*, vol. 2, no. 1, pp. 80–97, 2020.
- [18] J. Cai, J. Luo, S. Wang, and S. Yang, "Feature selection in machine learning: a new perspective," *Neurocomputing*, vol. 300, pp. 70–79, 2018.
- [19] A. Noor, "The utilization of E-health in the kingdom of Saudi Arabia," *International Research Journal of Engineering and Technology (IRJET)*, vol. 6, no. 9, 2019.
- [20] A. Noor, L. Wang, B. Ahmed et al., "D4: deep drug-drug interaction discovery and demystification," *Bioinformatics*, 2020.
- [21] A. Noor, "Discovering gaps in Saudi education for digital health transformation," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 10, pp. 105–109, 2019.
- [22] A. Noor, A. Assiri, S. Ayyaz, C. Clark, and M. Dumontier, "Drug-drug interaction discovery and demystification using Semantic Web technologies," *Journal of the American Medical Informatics Association*, vol. 24, no. 3, pp. 556–564, 2017.
- [23] R. G. Andrzejak, K. Lehnertz, F. Mormann et al., "Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: dependence on recording region and brain state," *Physical Review E—Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, vol. 64, no. 6, 2001.
- [24] R. G. Elger CE Andrzejak, K. Lehnertz, C. Rieke, F. Mormann, and P. David, *UCI Machine Learning Repository: Epileptic Seizure Recognition Data Set*, University of California, Oakland, CA, USA, 2001.
- [25] M. K. Uçar, M. R. Bozkurt, C. Bilgin, and K. Polat, "Automatic detection of respiratory arrests in OSA patients using PPG and machine learning techniques," *Neural Computing and Applications*, vol. 28, no. 10, pp. 2931–2945, 2017.
- [26] L. Rokach, A. Schclar, and E. Itach, "Ensemble methods for multi-label classification," *Expert Systems with Applications*, vol. 41, no. 16, pp. 7507–7523, 2014.
- [27] A. S. D. P. Wallisch, M. E. Lusignan, M. D. Benayoun, T. I. Baker, and N. G. Hatsopoulos, "MATLAB for neuroscientists: an introduction to scientific computing in MATLAB," *Journal of Undergraduate Neuroscience Education*, vol. 13, no. 1, 2014.
- [28] S. Şahan, K. Polat, H. Kodaz, and S. Güneş, "A new hybrid method based on fuzzy-artificial immune system and k -nn algorithm for breast cancer diagnosis," *Computers in Biology and Medicine*, vol. 37, no. 3, pp. 415–423, 2007.
- [29] M. Khan, Q. Ding, and W. Perrizo, " k -nearest neighbor classification on spatial data streams using P -trees," in *Advances in Knowledge Discovery and Data Mining*, Chapter Lecture No, pp. 517–528. Springer Berlin Heidelberg, Heidelberg, Germany, 2002.
- [30] A. Khamis, S. Hussain, A. Mohamed, and E. Bizkevelci, "Islanding detection in a distributed generation integrated power system using phase space technique and probabilistic neural network," *Neurocomputing*, vol. 148, pp. 587–599, 2015.
- [31] E. Parzen, "On estimation of a probability density function and mode," *The Annals of Mathematical Statistics*, vol. 33, no. 3, pp. 1065–1076, 1962.
- [32] P. D. Wasserman, *Advanced Methods in Neural Computing*, Van Nostrand Reinhold, New York, NY USA, 1993.
- [33] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [34] V. N. Mandhala, V. Sujatha, and B. Renuka Devi, "Scene classification using support vector machines," in *Proceedings of the 2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies*, pp. 1807–1810, Ramanathapuram, India, May 2014.
- [35] M. Arican, K. Polat, and K. Polat, "Binary particle swarm optimization (BPSO) based channel selection in the EEG signals and its application to speller systems," *Journal of Artificial Intelligence and Systems*, vol. 2, no. 1, pp. 27–37, 2020.
- [36] A. Ozdemir and K. Polat, "Deep learning applications for hyperspectral imaging: a systematic review," *Journal of the Institute of Electronics and Computer*, vol. 2, no. 1, pp. 39–56, 2020.
- [37] N. Daldal, Z. Cömert, and K. Polat, "Automatic determination of digital modulation types with different noises using convolutional neural network based on time–frequency information," *Applied Soft Computing Journal*, vol. 86, Article ID 105834, 2020.