Journal of Theoretical and Applied Information Technology
30th September 2018. Vol.96. No 18

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

HYBRID FEATURE SELECTION BASED ON MUTUAL INFORMATION AND AUC FOR PARKINSON'S DISEASE CLASSIFICATION

[1]ZAINAB N. SULTANI, [2]SUHAD A. YOUSIF
[1,2] Department of Computer Science, Al-Nahrain University, Iraq
E-mail: [1]zna@sc.nahrainuniv.edu.iq

ABSTRACT

Machine learning classifiers are used to distinguish healthy individuals from patients with Parkinson's disease through the use of a dataset of voice measurements based on patient speech recordings. Feature selection based on information theory is used in many data mining and machine learning applications. Mutual information is used on the Parkinson disease dataset to select a subset of relevant features that contribute the most in the decision making process. In conjunction with Mutual Information, the area under curve (AUC) is applied for feature selection, and features are eliminated by majority voting. In this paper, five classifiers are used to classify Parkinson's disease: Multilayer Feedforward Artificial Neural Network, k-Nearest Neighbor (kNN), Support Vector Machines, Naïve Bayes, and k-Means. The dataset is preprocessed prior to the classification, and the classifiers are trained using the k-fold cross validation evaluation model. The performance of the classifiers is evaluated based on the accuracy and the area under curve before and after the feature selection. The results are promising, particularly for the kNN classifier; k-Means presents the worst performance.

Keywords: *Machine Learning, Feature Selection, Mutual Information, Area Under Curve, Parkinson's Disease*

1. INTRODUCTION

This Parkinson's disease (PD) is a neurodegenerative disorder and a highly debilitating disease that affects 1%–1.5% of individuals aged 60 years old and above. PD also affects younger individuals aged 30–50 years. PD is considered the second leading neurological health disease in the elderly [1]. Therefore, biomarkers of PD must be urgently detected at the early stages to minimize damage. Machine Learning (ML) algorithms are used in many fields; such as Intrusion Detection Classification [2, 3], Face Recognition, Image Classification, and Fraud Detection. For PD classification, speech signal features are used as natural candidates for distinguishing PD [4]. Different ML classifiers have been used in PD classification studies. In the study by [5], an Artificial Neural Network (ANN) trained with backpropagation was used with k-means clustering techniques to classify PD, and they obtained 80% accuracy, 83.3% sensitivity, and 63.6% specificity. For the k-Nearest Neighbor (kNN) and Support Vector Machine (SVM) presented in the study by [1], the researchers collected the datasets and performed several cross-validation techniques. These classifiers achieved an accuracy rate of 78.57% for the test dataset when k was equal to 5 and 82.14% for the SVM. Khan in [6] classified PD by using three classifiers: kNN, Random Forest, and Ada-Boost. K-fold cross validation was used with a k value of 10. The accuracy of the kNN classifier with a k value of 3 and with Manhattan as the distance measure was 90.25%. Frid et al. in [7] designed the dataset of PD and healthy patients, using the SVM classifier they obtained an average accuracy rate of 81.8%. In the study by Ozkan [8] multiple ML algorithms were evaluated before and after the feature transformation and PCA dimensionality reduction, and the best accuracy results for the original dataset using 5-fold cross validation was 87.28% for SVM, 75.34% for NB, and 83.95% for kNN. PD disease affects the patient's brain and body and the early detection can reduce the disease impact, therefore to enhance the detection rate, the classification accuracy should be large as possible.

This paper aimed to classify individuals with PD by using different ML classifiers where feature selection methods (MI, AUC and Accuracy) are used to reduce the number of irrelevant features

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

and evaluate each classifier performance through the use of different measurements. The proposed block diagram is shown in figure 1.

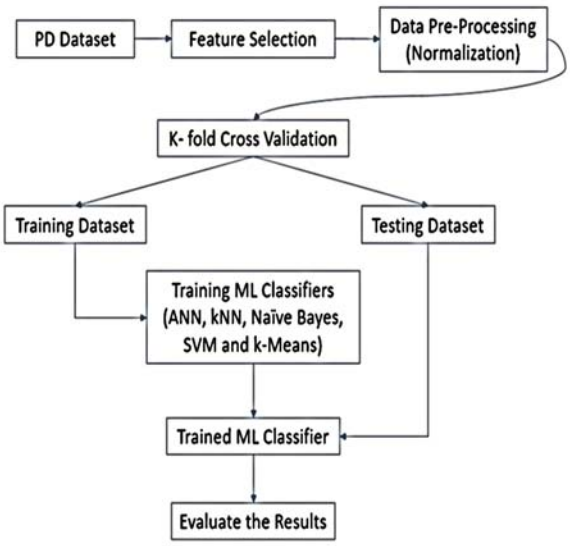


Figure 1: Classification Proposed Diagram

This paper is organized as follows. Section 2 provides a brief introduction on the dataset. Section 3 presents the methods used in this paper, including feature selection, preprocessing, and the classifiers. The evaluation metrics are given in Section 4. The experimental results are discussed in Section 5. The drawn conclusion is explained in Section 6.

2. DATASET

The PD dataset used in this paper was collected by Max Little of the University of Oxford [9] from the UCI repository. The dataset contains a table with biomedical voice measurements for 31 individuals, of whom 23 have PD and 9 are healthy. Each attribute (column) holds a specific voice measurement (Jitter, Shimmer) with 195 voice recording (rows). The data of the 22 attributes are provided in table 1, and one target column was collected to distinguish healthy individuals from those with PD by using the "status" column as the target, where 0 indicates healthy and 1 denotes PD.

3. METHODS

In ML applications, several factors affect the performance and success of ML in a specific task or problem. Data preprocessing is required to enhance the feature values range and identify the irrelevant or redundant features to ease the training process. Prior to preprocessing, feature selection is needed to reduce the number of irrelevant features. The cross validation and ML algorithms applied in this paper are explained in the following subsections.

Table 1: PD Features Dataset Description

Attribute (Feature)	Description
MDVP:Jitter(%)	Frequency Variation
MDVP:Jitter(Abs)	
MDVP:RAP	
MDVP:PPQ	
Jitter:DDP	
MDVP: Fo (Hz)	Average vocal fundamental frequency
MDVP: Fhi (Hz)	Maximum vocal fundamental frequency
MDVP:Flo(Hz)	Minimum vocal fundamental frequency
MDVP:Shimmer	Amplitude Variation
Shimmer:DDA	
MDVP:Shimmer(dB)	
Shimmer:APQ3	
Shimmer:APQ5	
MDVP:APQ	
NHR	Noise to Harmonic Ratio
HNR	Harmonic to Noise Ratio
RPDE	nonlinear dynamical complexity measures
D2	
DFA	Signal fractal scaling exponent
Spread 1	Three nonlinear measures of fundamental frequency variation
Spread2	
PPE	

3.1 Feature Selection

Feature selection techniques are used in several areas relating to expert and intelligent systems, such as ML, anomaly detection, data mining, and bioinformatics [10].

One of the feature selection methods is the filter feature selection. Filter feature selection methods are heuristic, but computationally are much cheaper ways of selecting an optimal (best) feature subset. Simple score S(i) is computed which measures how informative each feature x_i is about the Target labels y. After calculating the scores, pick the k features with the largest scores S(i).

The score S(i) is calculated using the absolute value of the correlation between feature x_i and target y, as measured on the training data. The score S(i) result reflects how the features are correlated with the class labels. The most common is

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

to choose S(i) to be the mutual information MI(xi, y) between xi and y.

As a feature selection method, Mutual Information (MI) measures the information contributed by the feature to the class decision (Eq. (1)). MI is equal or approximately zero when the term's distribution in the class is the same as that in the collection. MI reaches its maximum value if the term is a perfect indicator of the class type [11].

$$MI(u,c) = \sum_{e_t=\{1,0\}} \sum_{e_c=\{1,0\}} P(U=e_t, C=e_c) * \log_2 \frac{P(U=e_t, C=e_c)}{P(U=e_t)P(C=e_c)} \quad (1)$$

In this paper, the MI values were calculated by using Eq. (2), which is equivalent to Eq. 1.

$$MI(u,c) = \frac{N_{11}}{N}\log_2 \frac{NN_{11}}{N_1.N_1} + \frac{N_{01}}{N}\log_2 \frac{NN_{01}}{N_0.N_1} + \frac{N_{10}}{N}\log_2 \frac{NN_{10}}{N_1.N_0} + \frac{N_{00}}{N}\log_2 \frac{NN_{00}}{N_0.N_0} \quad (2)$$

where N is the training dataset record, $N_{11,00}$ is the number of records that are correctly classified as either PD or Normal, N_{01} is the number of records that are Normal (0) but are predicted as PD (1) by the classifier, and N_{10} is the number of training records with PD disease (1) that are considered Normal (0) by the classifier. N_0 is the number of records that either their target or their classifier output is Normal (0), and N_1 is the number of records that either their target or their classifier output is PD.

By using Eq. (2), the MI values were calculated for the 22 features and recorded in figure 2.

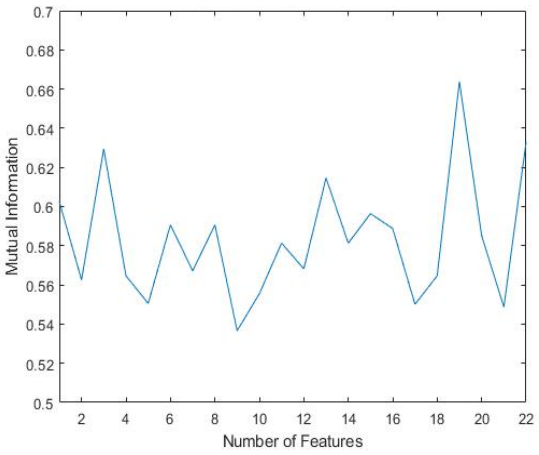


Figure 2: Mutual Information for the 22 features

As shown in Figure 2, the most relevant features holds the maximum MI values. These features, such as 19 and 22, depend on the class target. For feature selection, features with MI that is less than the mean MI subtracted from one standard deviation (Eq. (3)) are eliminated. The mean MI is equal to 0.583, and the standard deviation is equal to 0.0314. Four features (5, 9, 17, and 21) had MI less than or equal to 0.5516, which lies in the minimum range of MI values.

$$MI(f_i) < \mu(MI) - \sigma(MI) \quad (3)$$

where $MI(f_i)$ is the MI value for feature i; $\mu(MI), \sigma(MI)$ are the mean and standard deviation of the calculated MI values for all 22 features.

The area under the receiver operating characteristics (ROC) curve or simply the AUC has been used in medical application diagnosis since the 1970s. The AUC has been proposed as an alternative single number measure for evaluating the predictive ability of ML algorithms [12]. An area of 1 or 100% indicates a perfect test, whereas an area of 0.5 or 50% represents a worthless (bad) test. The AUC, which simply summarizes the performance of the ROC curve, is an important benchmark for performance comparisons of classification algorithms [13]. To select the least significant attributes, the AUC was calculated for each of the 22 features, as shown in figure 3. The features within the range of the AUC mean minus one standard deviation were 5, 9, and 21.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

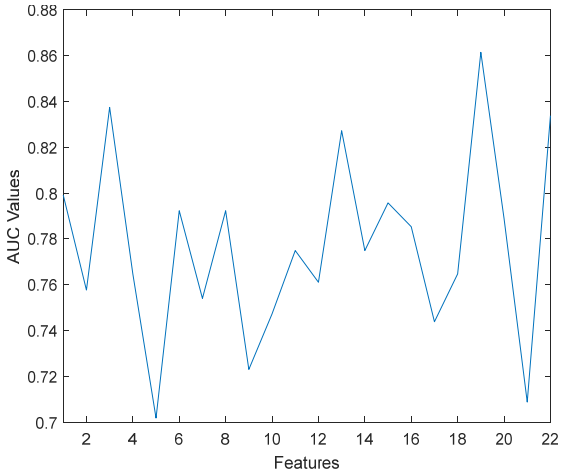


Figure 3: AUC values for the 22 features

For the final selection of the features that will be removed, the Accuracy (ACC) was calculated for each feature by using kNN as the classifier test, where k is equal to one. The features that had the minimum accuracy were 5, 6, 7, 9, and 10, as shown in figure 4.

By using majority voting among the MI, AUC, and ACC values, features 5, 7, 9, and 10 were eliminated. Features 5 and 9 were common among all three metrics, and 5, 7, and 10 held the minimum accuracy.

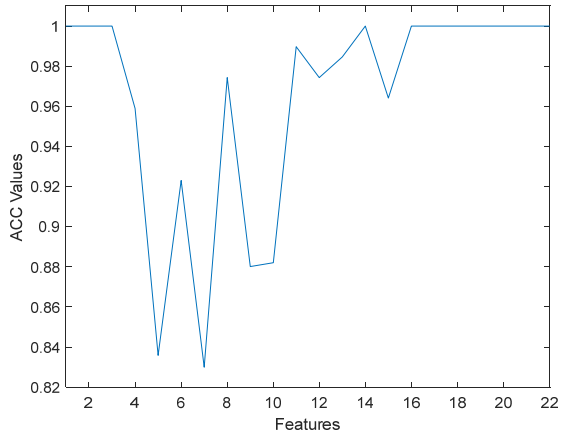


Figure 4: ACC values for the 22 features

3.2 Preprocessing And Cross Validaton

Data preprocessing plays a critical role prior to classification. One of the first steps of data preprocessing is data normalization. This step is required when the data have parameters of different units and scales. For normalization scales, all numeric variables in each column should be in the range of [0,1] (Eq. (4)) [14]:

$$f_{new} = \frac{f - f_{min}}{f_{max} - f_{min}} \quad (4)$$

where f is the feature column.

Using the same set of data for the training and validation of an algorithm yields an overoptimistic result [9]. Cross Validation is based on the principle that testing the algorithm on a new set of data yields a better estimate of its performance [10]. Most real applications have a limited amount of data. Because of this the dataset is split into the training sample and the validation sample. The training sample is used to train the algorithm and the validation sample is used as "new data" to evaluate the performance of the algorithm.

To provide a better generalization, a k-fold cross validation method was used, where the dataset was divided into training and testing datasets k times. k-fold cross validation algorithm is as follows [15]:

1. Randomly split dataset D into disjoint subsets (k) where in each k subset there are m training examples.
Consider these subsets are denoted by $D_1, \ldots, D_k$.
2. Train the model M on
$D_1 \cup \cdots \cup D_{j-1} \cup D_{j+1} \cup \cdots D_k$
(i.e., train on all subsets except Dj) to get some hypothesis hj
3. Test the hypothesis hj on Dj, to get ˆεDj (hj).

The generalization error of M is then calculated as the average of the ˆεDj (hj)'s (averaged over j).

A typical (good) choice for the number of folds to use here would be k = 10. The advantage of this method is that the size of the test dataset can be chosen differently for each trial, and the average is obtained.

In this study, since the dataset is not large enough in training examples; the k value was set to 5 fold cross validation, in which the dataset was divided into five groups with approximately 40 records in each group. One group was used as the testing dataset, and the other four groups were used for training.

3.3 Classification

Different ML algorithms were used to train PK datasets. ANN, kNN, Naïve Bayes (NB),

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

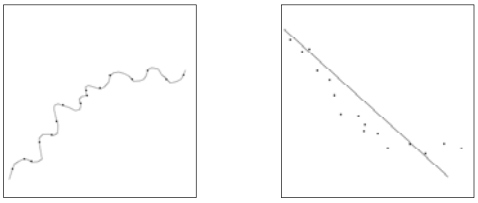
Support Vector Machine (SVM), and k-Means are explained in the following subsections.

3.3.1 Backpropagation Artificial Neural Network (Bpann)

Multilayer feedforward ANN is a biologically inspired model that attempts to build computer models that operate like a human brain. ANN acquires knowledge through learning. In this study, a backpropagation learning algorithm was used to train the multilayer neural network [15].

The Backpropagation neural network is a feedforward multilayered network and is most used among other neural networks paradigm. It is supervised training where the target class is known. Backpropagation algorithm approximates the non-linear relationship between the input and the output by adjusting the internal weight values [16], see figure 6.

Generalization is one of the main objective of learning algorithm. Under-training and over-training data problems can occur if the neural network is not trained properly. Under-training occurs when the artificial neural network is not complex enough (few hidden nodes are used) to detect a pattern. While over-fitting occurs when the neural network structure is too complex (overused hidden nodes are used), the effect of the two problems appears in figure 5 [17].



(a) Over-fitting data (b) Under fitting data

Figure 5: Over fitting vs Under Fitting

Different computational paradigms were designed using the training dataset to provide maximum generalization accuracy on the test (unseen) records. One hidden layer was used as a start with 10 hidden neurons. Then, the number of hidden neurons was incremented by 2 up to 30, and the training process was repeated for each trial. This method was employed because using large values of hidden neurons may lead to an over-training (over-fitting) problem, where the neural would fail to generalize or classify new records correctly. The designed ANN architecture consisted of 18 neurons in the input layer and two neurons in the output layer, as shown in table 2. The best number of hidden neurons was determined through experiments, as shown in the results section.

```
Assign all network inputs and output

Initialize all weights with small random numbers, typically between
-1 and 1

repeat

for every pattern in the training set

Present the pattern to the network

//      Propagated the input forward through the network:
        for each layer in the network
          for every node in the layer
            1. Calculate the weight sum of the inputs to the node
            2. Add the threshold to the sum
            3. Calculate the activation for the node
          end
        end

//      Propagate the errors backward through the network
        for every node in the output layer
          calculate the error signal
        end

for all hidden layers
          for every node in the layer
            1. Calculate the node's signal error
            2. Update each node's weight in the network
          end
        end

//      Calculate Global Error
        Calculate the Error Function

end

while ((maximum  number of iterations < than specified) AND
       (Error Function is > than specified))
```

Figure 6: Backpropagation Algorithm

Table 2: Target Column Transformation

Label/ Target Column	Output1 Neuron	Output2 Neuron
PD	1	0
Normal	0	1

3.3.2 K-Nearest Neighbor (KNN)

kNN is a simple algorithm that stores all available cases and classifies new cases on the basis of a similarity measure. KNN is considered a lazy algorithm, where the training data is not used to perform generalization. In other words, in kNN explicit training phase doesn't exist. KNN uses all the training data during the testing phase. KNN uses feature similarity; where it calculates how closely the testing data features resemble the training set determines how the testing data can be classified, see figure 7.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

KNN can be used for classification and regression. In classification, the output is a discrete value (class membership). However in regression, the output is a continuous value. This value is calculated by averaging the values of its k nearest neighbors [18].

k-Nearest Neighbor
Classify (**X**, **Y**, *x*) // **X**: training data, **Y**: class labels of **X**, *x*: unknown sample
for $i = 1$ **to** m **do**
  Compute distance $d(\mathbf{X}_i, x)$
end for
Compute set I containing indices for the k smallest distances $d(\mathbf{X}_i, x)$.
return majority label for $\{\mathbf{Y}_i$ where $i \in I\}$

Figure 7: kNN classification algorithm

The k parameter refers to the number of neighbors used to make the classification for the testing record on the basis of the distance (similarity). An object is classified using majority vote of its neighbors, with the object being assigned to the most common class among its k nearest neighbors. Similarity is measured in terms of the measured Euclidean or Manhattan distance [19]. See figure 8 for an example of k-NN classification.

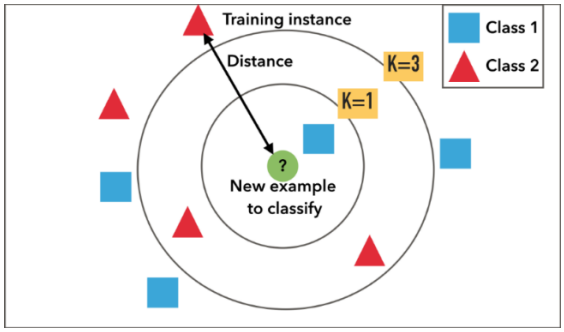


Figure 8: kNN classification example

In this study, Euclidean distance measure is used, while the k value is chosen iteratively starting from 1 up to 7.

3.3.3 Naïve Bayes (NB)

The NB classifier is based on Bayes Theorem and can be realized simply (Eq. (5)). The word "naïve" refers to the characteristic of the classifier to assume that the attributes (features) for the given class are conditionally independent of one another. A prediction can be obtained by selecting the most likely (higher probability) class [20].

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)} \qquad (5)$$

Where

P(x) is the prior probability of x
P(y) is the prior probability of y
P(x|y) is the posterior probability of x given y
p(y|x) is the probability that record x belongs to class y.

The Bayesian algorithm is a supervised learning method as well as a statistical method used for classification. An underlying probabilistic model is assumed by determining the outcomes' probabilities. Bayesian algorithm can be applied to diagnostic and predictive problems. Explicit probabilities for hypothesis are calculated and it is robust to noise in the training data [20].

The naïve Bayes classifier is a simple form of Bayesian classifiers which assumes all the features are independent of each other. Despite this assumption, the naïve Bayes classifier's accuracy is comparable to other sophisticated classifiers.

3.3.4 K-Means Clustering

Clustering is the classification of objects into different groups, or more precisely, the partitioning of a data set into subsets (clusters), so that the data in each subset (ideally) share some common trait - often according to some defined distance measure.

The K-means ML algorithm is a well-known data clustering algorithm. This unsupervised learning algorithm requires the number of data clusters to be pre-specified, and the data are partitioned into clusters by using the distance measure [21,22]. In this paper, k was set to 2 (PD or Normal), and the Euclidean distance was used as the distance measure.

Given k, the k-means algorithm is implemented in four steps:

1. Partition objects into k nonempty subsets
2. Compute seed points as the centroids of the clusters of the current partition (the centroid is the center, i.e., mean point, of the cluster)
3. Assign each object to the cluster with the nearest seed point
4. Go back to Step 2, stop when no more new assignment

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3.3.5 Support Vector Machine (SVM)

In SVM, the training dataset was used to create a model using their labels. A hyperplane that had the maximum margin (separation) was used for the binary classification [23].

Support Vector Machine (SVM) is an algorithmic technique for pattern classification that has grown in popularity in recent times, and has been used in many fields including bioinformatics.

Although the SVM can be applied to various optimization problems such as regression, the classic problem is that of data classification. The basic idea is shown in figure 9. The data points are identified as being positive or negative, and the problem is to find a hyper-plane that separates the data points by a maximal margin [24].

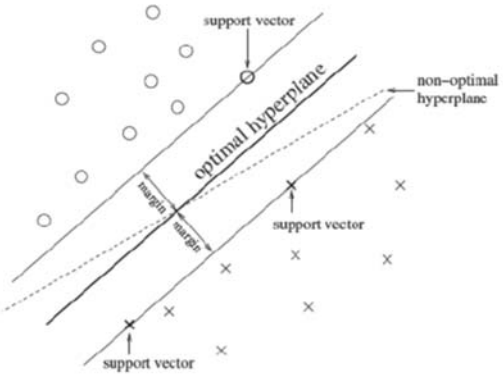


Figure 9: SVM classification

In this paper, a linear SVM was used.

4. EVALUATION METRICS

Classifier performance can be measured using several measurements, such as accuracy, specificity, and sensitivity [25]. ACC (Eq. (6)) is the measurement of correctly classified records:

$$accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (6)$$

where True positive (TP) is the number of records correctly identified as patient, **false positive (FP) is** the number of records incorrectly identified as patient, **true negative (TN) is** the number of records correctly identified as healthy, and **false negative (FN) is** the number of records incorrectly identified as healthy. Specificity and sensitivity (Eq. (7 & 8)) are statistical measures of false negatives and false positives. Sensitivity (recall) refers to a classifier's ability to find all true (positive) samples in the dataset.

$$sensitivity = \frac{TP}{TP+FN} \quad (7)$$

$$specificity = \frac{TN}{TN+FP} \quad (8)$$

The ROC curve is a graphical plot that can describe the performance of a binary classifier algorithm. It is created by plotting the fraction of true positive rate (sensitivity) versus the fraction of false positive rate (1-specifcity) at different threshold settings [21].

5. RESULTS AND DISCUSSION

The classification system based on different classifiers was evaluated before and after feature selection. A 5-fold cross validation evaluation method was used, in which the original dataset had 195 records divided into five subsets with approximately 40 records, for the testing dataset. After feature selection and normalization, the dataset was ready to be classified. The following tables and figures present the results for the classifiers described in the previous section.

kNN was used first, with k=1, 3, 5, and 7 and the Euclidean distance as the distance metric. The evaluation results for the kNN classifier are shown in table 3 before and after feature selection, in which the classification was repeated for 10 times. The best average accuracy for the kNN was 95.95% for using feature selection and 95.79% for the original features. Figures 10–13 present the AUC values for 1-nn, 3-nn, 5-nn, and 7-nn, respectively, before and after feature selection. As clearly shown, feature reduction improved the AUC values.

Table 3: Average Accuracy for 1-nn, 3-nn, 5nn and 7-nn

	Original Features		Feature Selection	
K	ACC (AVG)	ACC (BEST)	ACC (AVG)	ACC (BEST)
1-nn	95.79%	96.298%	**95.95%**	97%
3-nn	93.24%	94.4%	93.85%	94.9%
5-nn	91.95%	93.33%	93.92%	96%
7-nn	92.4%	100%	93.1%	97.4%

Journal of Theoretical and Applied Information Technology
30th September 2018. Vol.96. No 18

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

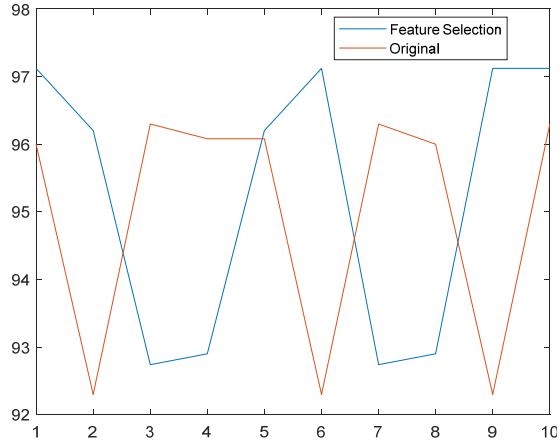


Figure 10: 1-nn AUC values for the 10 iterations

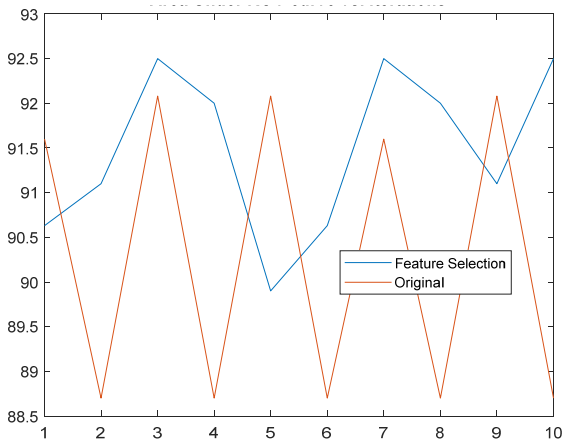


Figure 11: 3-nn AUC values for the 10 iterations

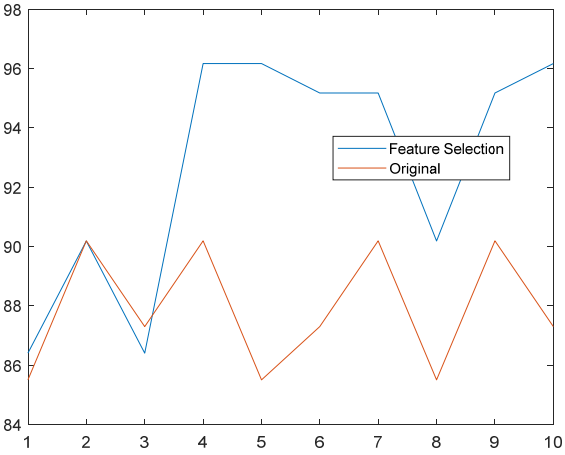


Figure 12: 5-nn AUC values for the 10 iterations

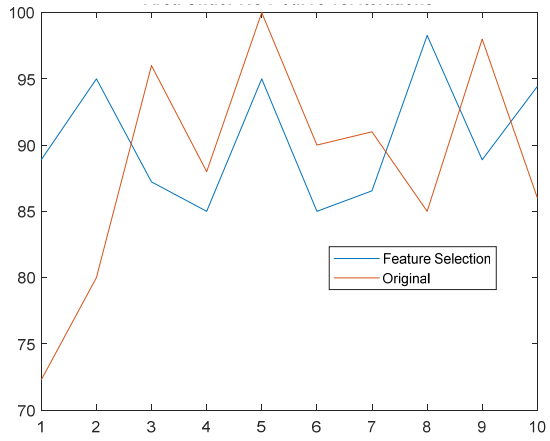


Figure 13: 7-nn AUC values for the 10 iterations

Feed-forward Neural Network trained using a resilient backpropagation was used as our second classifier. The classifier training parameters are shown in table 4.

Table 4: ANN Training Parameters

Training Algorithm	Resilient Backpropagation
Transfer Functions	Sigmoid
No of Hidden Layers	1
No of Hidden Neurons	Iteratively from 10 to 29
Dataset	Mapped from 0 to 1

ANN was trained iteratively for different numbers of hidden neurons starting from 10 to 27. The Best and Average Accuracy before and after feature selection for each hidden neuron are shown in table 5.

The results for using ANN were promising. For example, a testing accuracy of 100% was repeated for several times. However, the best average accuracy was 92.9%, as shown in table 5, when 18 hidden neurons in the hidden layer were used.

Table 5: Average Accuracy for ANN

	Original		Feature Selection	
H	ACC (AVG)	ACC (BEST)	ACC (AVG)	ACC (BEST)
10	89.3%	100%	90.09%	94.8%
12	91.35%	100%	90.52%	94.8%
14	91.91%	97.5%	91.75%	97.5%
16	92.54%	97.5%	90.59%	100%
18	**93.03%**	100%	**92.9%**	100%
20	91.23%	97.5%	91.93%	94.8%
22	90.66%	97.5%	92.28%	100%
25	92.74%	100%	91.23%	97.43%
27	91.72%	97.5%	91.9%	100%
29	92.14%	100%	89.4%	95%

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Figure 14 presents the AUC values for the H (number of hidden neurons) before and after feature selection. The best AUC value was for H=12, whereas the second best AUC value was H=18, which had the best average accuracy among all other H values.

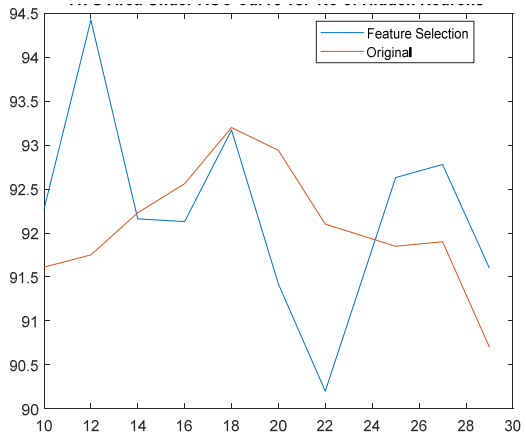


Figure 14: ANN classifier AUC values for different number of hidden neurons

The performances of the NB, SVM and K-Means are presented in table 6. Feature selection improved the performance of NB and SVM but not K-Means. In the NB classifier, the average ACC was 78.4%, which was unacceptable. This result was due to the "Naïve" assumption of the independent features, which was incorrect in this case because some dependencies existed among the features.

Table 6: Average Accuracy for NB, SVM and K-Means

	Original		Feature Selection	
Classifiers	ACC (AVG)	ACC (BEST)	ACC (AVG)	ACC (BEST)
NB	77.18%	87.2%	78.4%	89.7%
SVM	87.23%	92.31%	89.07%	94.9%
K Means	65.04%	67.18%	64.95%	67.2%

In Table 7, the proposed classification methods presented in this paper are compared to the related works presented in the literature survey.

Table 7: Results of the proposed and the related classification systems

Ref.	Accuracy
[5]	ANN: 80%
[1]	kNN: 78.57% SVM: 82.14%
[6]	kNN: 90.25%
[7]	SVM: 81.8%
[8]	SVM : 87.28% NB : 75.34% kNN : 83.95%
Proposed System	ANN: 92.9% kNN:95.95% SVM:89.07% NB: 78.4%

As shown in Table 7, the proposed machine learning algorithms obtained better results than the related systems. This is due to the use of the proposed feature selection method where three technique were used to enhance the selection of the best features subset.

6. CONCLUSION

In this research, a feature selection was used to select the best subset of relevant features. A MI, AUC and accuracy were used to reduce the number of irrelevant features. This technique improved the accuracy as shown in table 7. In this paper a PD dataset was classified using different ML classifiers with a 5-fold cross validation evaluation model. The classifiers were trained for 10 iterations, and the accuracy was averaged from 10 trials. kNN and ANN obtained the best accuracy of 100%. However, average accuracy is the more accurate indicator. kNN achieved an average accuracy of 95.95% for 1-nn, whereas ANN attained 92.9%. Even though kNN is considered lazy because it compares the testing dataset with the training dataset, it obtained the best results especially when k was 1. However, it decreased once k exceeded 5. Thus, the testing dataset was highly sensitive to the value of k and the dataset values because the normal and PD were not very far from each other (distance).

The ANN classifier obtained good results by attaining an accuracy of 100% for four times out of 10 iterations. Nonetheless, feature selection did not always improve the accuracy. ANN was highly sensitive for the initial division of the training sets (which was randomly chosen for each iteration for each k fold), which is why sometimes a 100% is obtained. For ANN, the best number of hidden neurons was 18 because it obtained the best average accuracy and has the second best AUC value. By contrast, although 12 attained the best AUC value, its average accuracy was 90.52%, which was lower than the best accuracy of 94.8%.

The NB classifier performance was not good, because the NB assumed that the features were conditionally independent for a given class, and this assumption is inaccurate for the PD dataset because a number of some features are correlated. The K-means results were the worst among all algorithms, because this approach constructed linear decision boundaries that did not divide the data accurately. The SVM performance can be improved using non-linear kernel because the data were not linearly

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

separable, as clearly shown by the poor results of the K-means algorithm.

One of the main limitations of the existing tools is the collection of PD dataset is limited to analyzing only voices and do not account for other measurements such as genetic and environmental factors. PD disease is a multifactorial disorder disease and collecting many factors will enhance the early detection.

REFRENCES:

[1] B. E. Sakar, M. E. Isenkul, C. O. Sakar, A. Sertbas, F. Gurgen, S. Delil, H. Apaydin, and O. Kursun, "Collection and analysis of a Parkinson speech dataset with multiple types of sound recordings", *IEEE J Biomed Health Inform*, 17(4), 2013, pp. 828-834.

[2] R. S. Naoum, and Z.N. Sultani, "Hybrid System of Learning Vector Quantization and Enhanced Resilient Backropagation Artificial Neural Network for Intrusion Classification", *International Journal of Research and Reviews in Applied Sciences (IJRRAS),* 14 (2), 2013.

[3] R. S. Naoum, and Z.N. Sultani, "Learning Vector Quantization and k-Nearest Neighbor for Intrusion Classification", *World of Computer Science and Information Technology Journal* (WCSIT), 2(3), 2012, pp. 105-109.

[4] S. Sindhu, and D. Sindhu, "Data Mining and Gene Expression Analysis in Bioinformatics", *International Journal of Computer Science and Mobile Computing*, 6(5), 2017, pp. 77-82.

[5] R. F. Olanrewaju, N. S. Sahari, A. A. Musa, and, N. Hakiem, "Application of neural networks in early detection and diagnosis of Parkinson's disease", *Paper presented at the 2014 International Conference on Cyber and IT Service Management (CITSM),* 2014.

[6] S. U. Khan, "Classification of Parkinson's Disease Using Data Mining Techniques". *J Parkinsons Dis Alzheimer Dis*, 2(2), 2015.

[7] A. Frid, H. Hazan, D. Hilu, L. Manevitz, L. O. Ramig, and S. Sapir, "Computational Diagnosis of Parkinson's Disease Directly from Natural Speech Using Machine Learning Techniques". *Paper presented at the 2014 IEEE International Conference on Software Science, Technology and Engineering*, 2014.

[8] H. Ozkan, "A Comparison of Classification Methods for Telediagnosis of Parkinson's Disease", *Entropy,* 18(4), 2016.

[9] M. A. Little, P. E. McSharry, S. J. Roberts, D. A. E. Costello, and I. M. Moroz, "Exploiting Nonlinear Recurrence and Fractal Scaling Properties for Voice Disorder Detection", *BioMedical Engineering OnLine,* 6, 2007.

[10] M. Bennasar, Y. Hicks, and R. Setchi, "Feature selection using Joint Mutual Information Maximisation", *Expert Systems with Applications, 42*(22), 2015, pp. 8520-8532.

[11] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. USA: Cambridge University, 2008.

[12] J. Huang, and C. X. Ling, "Using AUC and accuracy in evaluating learning algorithms", *IEEE Transactions on Knowledge and Data Engineering, 17*(3), 2005.

[13] P.Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms", *Pattern Recognition, 30*(7), 1997, pp. 1145-1159.

[14] S. Saitta, Standardization vs. normalization. Retrieved from http://www.dataminingblog.com/standardization-vs-normalization/, 2007.

[15] Ng. Andrew, "CS 229 Lecture notes Andrew Ng Part VI Regularization and model selection." , 2012

[16] E. Turban, *Business Intelligence : A Managerial Approach*. India: Pearson, 2013.

[17] R. Barry, "Artificial Neural Network Prediction of Wavelet Sub-bands for Audio Compression", Thesis for Bachelor of Science, Dept. of Electrical & Computer Engineering, 2000. http://www.ilos.net/~rbarry/Thesis.pdf

[18] A. Bronshtein, "A quick introduction to k-nearest neighbors algorithm", Retrieved from https://medium.com/@adi.bronshtein/a-quick-introduction-to-k-nearest-neighbors-algorithm-62214cea29c7 , 2017.

[19] A. Kataria, and M. D. Singh, "A Review of Data Classification Using K-Nearest Neighbour Algorithm", *International Journal of Emerging Technology and Advanced Engineering, 3*(6), 2013.

[20] S. Russell, and P. Nowig, *Artificial Intelligence: A Modern Approach*. New Jersey. USA: Prentice Hall, 2002.

[21] D. T. Pham, S. S. Dimov, and C. D. Nguyen (2016). Selection of K in K-means clustering. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 219*(1), 2016, pp.103-119.

[22] R. O. Duda, P. E. Hart, and D. G. Stor, *Pattern Classification*. USA: John Wiley & Sons, 2000.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[23] DATA MINING Concepts and Techniques,Jiawei Han, Micheline Kamber Morgan Kaufman Publishers, 2003

[24] Asadi Ghanbari, Abdolreza & Mohsen Pedram, Mir & Ahmadi, Ali & Navidi, Hamidreza & Broumandnia, Ali. (2012). Brain Computer Interface with Wavelets and Genetic Algorithms. 10.5772/37289.

[25] AK. Akobeng, "Understanding diagnostic tests 3: Receiver operating characteristic curves". *Acta Paediatrica, 96*(5), 2007.

.