International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Mathematical Modelling of Engineering Problems
Vol. 8, No. 6, December, 2021, pp. 915-922

Journal homepage: http://iieta.org/journals/mmep

Cloud-Based Parkinson’s Disease Diagnosis Using Machine Learning

Ahmed R. Nasser*, Ali M. Mahmood

Control and System Engineering Department, University of Technology - Iraq, Baghdad 10001, Iraq

Corresponding Author Email: ahmed.r.nasser@uotechnology.edu.iq

https://doi.org/10.18280/mmep.080610

Received: 20 June 2021
Accepted: 8 October 2021

Keywords:
cloud computing, artificial intelligence, machine learning, deep learning, feature selection, Parkinson’s disease

ABSTRACT

Parkinson’s disease (PD) harms the human brain's nervous system and can affect the patient's life. However, the diagnosis of PD diagnosis in the first stages can lead to early treatment and save costs. In this paper, a cloud-based machine learning diagnosing intelligent system is proposed for the PD with respect to patient voice. The proposed system is composed of two stages. In the first stage, two machine learning approaches, Random-Forest (RF) and Long-Short-Term-Memory (LSTM) are applied to generate a model that can be used for early treatment of PD. In this stage, a feature selection method is used to choose the minimum subset of the best features, which can be utilized later to generate the classification model. In the second stage, the best diagnosis model is deployed in cloud computing. In this stage, an Android application is also designed to provide the interface to the diagnosis model. The performance evaluation of the diagnosis model is conducted based on the F-score accuracy measurement. The result shows that the LTSM model has superior accuracy with 95% of the F-score compared with the RF model. Therefore, the LSTM model is selected for implementing a cloud-based PD diagnosing application using Python and Java.

1. INTRODUCTION

Elderly people, with the age of 50 or more, may be affected seriously by neurodegenerative disorders, resulted from Parkinson’s Disease (PD) [1]. This disease affects the nervous system and leads to finally loss of control of the muscles. Several studies have revealed that the genetic factor is considered one of the main factors causing PD [2]. Tribulation dementia is one of the symptoms of PD compared to healthy people. Though, the patient could benefit from the scheduled treatment. Several symptoms occur with PD including trembling in arms, hands, head, jaw, legs, limbs' stiffness, slowness in movement, and impaired balance. Additionally, approximately 90% of PD patients suffer from speech difficulties represented by dysphonia that impaired speech production and dysarthria which means difficulties in speech articulation. Consequently, these known facts can be utilized for PD diagnosis [3]. Although there are several research studies have been conducted to diagnose PD, still there is a need for a more accurate detection system that achieves the characteristics of usability and the availability of the test in the form of a mobile application.

In this work, a cloud-based system is proposed to diagnose PD in the earlier stage. This can be achieved by using different artificial intelligence algorithms of machine learning, for instance, feature selection and classification methods over Parkinson’s dataset. Feature selection methods will be used to minimize the feature space for the input data while considering the classification accuracy to be maximum. Then trying to validate different methods for generating train and test data for applied classifiers such as LSTM and RF as well as calculating the accuracy and specificity of each classifier to choose the proper method to diagnose Parkinson’s disease with maximal success rate. Cloud computing technology is a model that allows access to an unlimited number of computing resources under favorable conditions and optionally anytime, anywhere.

The benefits of cloud computing can be summarized as follows [4]:

- Cost saving: cloud technology; eliminates investment expenses such as server racks and IT professionals working to manage infrastructure, which is a significant advantage in terms of costs.
- Speed: with the provision of many services of cloud computing, which can be provided easily without big efforts, this reduces the pressure of businesses regards capacity planning.
- Global-scale: services can be provided by cloud technology anytime, anywhere in a scalable manner.
- Productivity: in fact, data centers require a large number of “stacking and racking” operations. These are processes that include hardware tuning, patching software, which can be considered as time-consuming for the Information Technology (IT) management tasks. Cloud technology eliminates the requirement for a large number of tasks. This enables IT, teams, to utilize time for other business goals.
- Performance: clients should deal with cloud computing services due to many reasons. One of these reasons is that the data centers include the latest versions of hardware, which are updated regularly.
- Reliability: cloud technology facilitates the process of data backup and saving data in redundant locations.

The contribution of this paper is to develop a cloud-based PD diagnosis intelligent system with a mobile application using Machine Learning (ML) LSTM algorithms.

The rest of the paper is arranged as follows: in section 2 the prior research studies that have been published within the last decade are reviewed. Section 3 presents the proposed system of Parkinson's disease diagnosis. Section 4 includes the design of the cloud-based PD diagnosis system for PD diagnosis. The results of the performance evaluation are presented in section 5. Finally, section 6 illustrates the conclusion of the paper.

2. RELATED WORKS

Wroge et al. [5] proposed a supervised classification method based on deep neural networks for diagnosing PD. They used the voice data on individuals with and without PD, obtained from a clinical observational study conducted by Sage Bionetworks. Minimum Redundancy Maximum Relevance (mRMR) feature selection has been used to reduce the audio features. Patient voice data with reduced features are used to train a deep neural network for PD diagnosis. The deep learning model has achieved a peak accuracy of 85%.

Wodzinski et al. [6] have proposed a modified version of the deep learning method namely ResNet, which is a type of convolutional neural network that has been used for PD detection based on patient voice recordings. In this study, the patient audio samples are converted into an image using the spectrum of the audio to be used as an input to ResNet deep learning method. The obtained training dataset includes the voice recordings of 50 PD patients. The validation results show 91.7 % of detection accuracy.

In the work of Johri and Tripathi [7], a diagnosis model based on deep learning CNN (Convolutional Neural Networks) and Artificial Neural Network (ANN) has been developed. The diagnosis model consists of two modules, which are the VGFR spectrogram detector and the voice impairment classifier. The system has been built to diagnose the PD patient based on two decisive symptoms, which are distorted walking patterns (gait) and speech impairment. The patient data is obtained from the UCI ML repository and PhysioNet database bank. The implemented system of voice-based PD diagnosis was carried out with an accuracy of 89.15%.

Mallela et al. [8] have proposed a classification system to classify Amyotrophic Lateral Sclerosis (ALS), Healthy Controls (HC), and Parkinson's Disease (PD). The applied patient's speech data has been collected from 60 persons, such as Spontaneous speech (SPON), Diadochokinetic rate (DIDK), and Sustained phoneme production (PHON). The training of the classification model was based on deep learning techniques CNN- LSTM. The classification model based on the CNN-LSTM network has achieved 88.5% accuracy for PD classification using SPON speech data.

The distinct property of this work is represented by deploying the designed system in cloud computing to gain its advantages and made the personal PD test available on smart devices. Also, high accuracy is achieved in the design system with the application of LSTM machine learning and deep learning algorithms proposed for PD diagnosis with the mRMR as an optimization feature selection technique.

3. RESEARCH COMPONENTS

The main components of the paper are described in this section, including features selection algorithm and machine learning approaches for classification along with the details of cloud computing requirements.

3.1 Optimized features selection based on mRMR

Feature selection represents a search problem to find the optimal (m) subset features among (M) features. In other words, redundant and irrelevant features are excluded in the process of feature selection. This reduces both the complexity and computing time of the system, which improves the recognition accuracy [9, 10].

In this work, the Minimum-Redundancy-Maximum-Relevance selection (mRMR) feature selection method is used [11]. mRMR is employed to select the relevant features, where the aim is to find *m* subset of features and separate the target labels based on their common information. The details of mRMR feature selection method are described as follows.

For discrete random variables (x and y) the mutual information estimation between *x* and *y* is squared off using their individual probabilities P (x), P (y), and their joint probability P (x, y) as shown in Eq. (1) [12].

$$\mathrm{I(X,Y)} = \sum_{x\in X}\sum_{y\in Y} p(x,y)\log(\frac{p(x,y)}{p(x)p(y)}) \tag{1}$$

where, *x* denotes a characteristic from the set of carefully chosen features X, and *y* represents the class label of the prototypes Y set.

In the case of having two features, if any of these features, which are highly dependent on each other, is removed, no big change in the discriminative power for this feature. The aim of mRMR is to find the minimum set of features (*S)* that contains the (*n*) attributes {x_i}, which have the largest dependency on the target class *y* and minimum redundancy between them. The algorithm of mRMR feature selection is demonstrated in Algorithm 1, which is implemented using Python.

Eq. (2) illustrates the maximization of the dependency between the feature variable xi with the class label *y*, while using Eq. (3) shows the minimization of the dependency for ($x_i$andx_j) features. The constraint represents a filter for the mutually exclusive features.

$$MaxRv(\boldsymbol{x}_i, y) = \frac{1}{\|m\|}\sum_{x_i\in S} I(\boldsymbol{x}_i, y) \tag{2}$$

$$MinRd(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{1}{\|m\|^2}\sum_{x_i,x_j\in S} I(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{3}$$

Finally, the mRMR algorithm chooses the subset Xm, where (m < n) from the set of main features Xn, which maximized Eq. (4), and uses a process of an incremental search for finding the possible optimal feature set *S* as shown in Eq. (5).

$$max\, \varphi(Rv, Rd), \varphi = Rv - R \tag{4}$$

$$\max_{x_j\in X-S_{m-1}} \left[I(\boldsymbol{x}_i, y) - \frac{1}{m-1}\sum_{x_i\in S_{m-1}} I(\boldsymbol{x}_i, \boldsymbol{x}_j) \right] \tag{5}$$

In this work, the mRMR method is applied to select the minimum data features as possible while keeping the high accuracy after train the classifier with selected features. This can be achieved by removing the features that are irrelevant to the target classes which can lead to confuse the classification model decision and impact the accuracy. Farther more removing the redundant features leads to reduce the model complexity.

```
Algorithm 1: mRMR feature selection [10]
Input: Dataset $D(X,Y)$.
Output: Reduced Dataset S (Xm, Ym).
    1: S ← $\emptyset$ // Initialization
    2: Add Xi= argmax $j \in \varphi$ D(Xj,Y) to S
    3: For t=1: k-1 do
    4:     Add feature satisfies Eq. 5 to S
    5: End for
    6: Return S
    7: End
```

3.2 Machine learning approaches for classification

The classifier represents a machine learning tool that receipts data and places them in one of the k classes. The intended goal is to learn the classifier how to classify the test sample at high accuracy with a minimum number of samples and features [13]. In this work, two machine learning approaches have been used, which are explained as follows:

3.2.1 Long Short-Term Memory Approach

The Long Short-Term Memory (LSTM) is a type of artificial recurrent neural network approach, which can be utilized in deep learning [14]. It can keep the data for a long period. LSTM can be applied in classifying the predicting and processing of data. Figure 1 illustrates the general structure of LSTM. The typical structure of LSTM contains four layers with one input layer, two hidden layers, and one output layer. LSTM also involves different memory blocks called cells. The data is retained in the cells and the manipulations of memory are achieved in what so-called gates. The structure of LSTM consists of three kinds of gates which are the input gate, forget gate, and output gate. The forget gate is used to elects the reserved or removed information in the cell. The number of states at the previous moment has been retained to the current moment is obtained by the Foregate gate [15]. The input gate is in control of the cell state via adding the useful information, which is finally extracted from the current cell state to be presented as an output or (the input to the next cell) is achieved by the output gate [14]. The general process of LSTM is described as follows.

1) Foregate gate: The first step in LSTM is to use the Foregate gate to decide whether to keep or remove the data from the cell. The output of the Foregate gate f_t is shown in Eq. (6).

$$f_t = \sigma\,(W_f[h_{t-1}, X_t] + b_f) \tag{6}$$

where, σ is the sigmoid function, W_f and h_{t-1} represent the weight matrices of Foregate gate, b_f is the bias and X_t is the input features.

2) Input Gate: The second step is used to save the new data in the cell state by using the output of the input gate which is calculated in Eq. (7), and then updating the cell memory by the candidate values as illustrated in Eq. (8).

$$i_t = \sigma\,(W_i[h_{t-1}, X_t] + b_i) \tag{7}$$

$$C = tanh\,(W_c[h_{t-1}, X_t] + b_c) \tag{8}$$

where, W_i and W_i are wights of the input layer, b_i and b_c are the bias terms for the input gate and the $tanh$ represents the activation function.

3) Cell State: At the current moment, the cell state is calculated by the product of the forget gate and the previous moment state to be added with the product of the input gate parts as shown in Eq. (9).

$$C_t = f_t * C_{t-1} + i_t * C \tag{9}$$

where, i_t is the output of input gate, C is the cell state, C_{t-1} is the previous cell state and C_t represents the current cell state.

4) Output Gate: The final step is completed by determining the decision in the output gate shown in Eq. (12) via passing the initial output that is calculated earlier from Eq. (10) and through the sigmoid activation function illustrated in Eq. (11).

$$O_t = \sigma\,(W_o[h_{t-1}, X_t] + b_o) \tag{10}$$

$$h_t = O_t * \tanh\,(C_t) \tag{11}$$

$$Output_{class} = \sigma\,(h_t * W_{outparameter}) \tag{12}$$

where, W_o, $W_{outparameter}$ are the weights of output gate; b_o is the bias; $Output_{class}$ is the classification output, and σ is the sigmoid activation function.

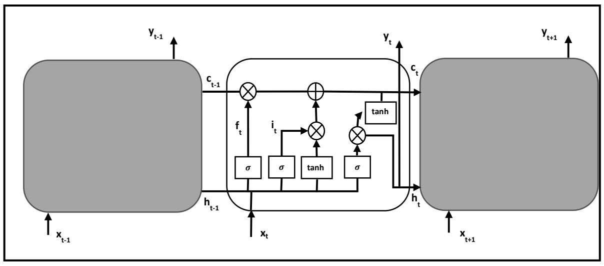


Figure 1. The structure of the LSTM network [13]

3.2.2 Random Forests Approach

Random Forest (RF) is a type of data mining algorithms that is mainly used for supervised learning by randomly generates a forest. Then the "forest" that has been established represents a group of decision trees that are mostly trained using the "bagging" method, where the bagging method is simply representing a collection of learning models for increasing the accuracy of all results in the classification phase [16].

RF consists of multiple binary decision trees as shown in Figure 2. It is used for many purposes, including regression, classification, and different tasks via constructing numbers of decision trees at the time of training. The main characteristics of RF are:

- RF can handle naturally (multiclass) classification and regression.
- It is fast in training and prediction.
- RF can depend on one or two parameters in tuning.
- It has a built-in estimate of generalization error.
- It can be applied to high-dimensional problems.
- Parallel implantation of RF is applicable.
- It can provide measurement for the variable importance.
- Supporting differential class weighting.
- RF is unsupervised learning and can provide a visualization with outlier detection.

The operation process of the RF is explained briefly as follows:

Assume having the real value input of predictor variables X = $(X_1 \dots X_p)^T$with p values. The first step is to calculate the loss function shown by Eq. (13).

$$E_{XY} = (L(Y, f(X))) \qquad (13)$$

where, $L(Y, f(X))$ is the loss function for calculating the prediction function $f(X)$ and Y is the actual value. The typical loss function can be represented by root mean square error. However, the entropy information gain of the Random Forest classifier is used as a loss function in normal circumstances [17]. The loss function L(Y, f (X)) is used for measuring the closeness between the prediction and the actual value as shown in Eq. (14).

$$L(Y, f(X)) = I(Y \neq f(X)) = \begin{cases} 0\ if\ Y = f(X) \\ 1\ otherwise \end{cases} \qquad (14)$$

For the classification problem, the possible values of Y is represented by ℝ , minimizing EXY (L(Y, f (X))) for zero-one loss gives Bayes rule shown in Eq. (15).

$$f(X) = argmax_{y\in\mathbb{R}}\, P(Y = y|X = x) \qquad (15)$$

Ensemble predictor f which is a collection of base learners represented by hJ(x) calculates the predicted final class by voting as shown in Eq. (16).

$$f(X) = argmax_{y\in\mathbb{R}} \sum_{j=1}^{J} I(y = h_j(x)) \qquad (16)$$

where, Y is the random variable that represents the target, $f(X)$ is the prediction function for predicting Y, E_{XY} is expected value of the loss, ℝ is the set of possible values of Y and $h_j(x)$ the jth base learner;

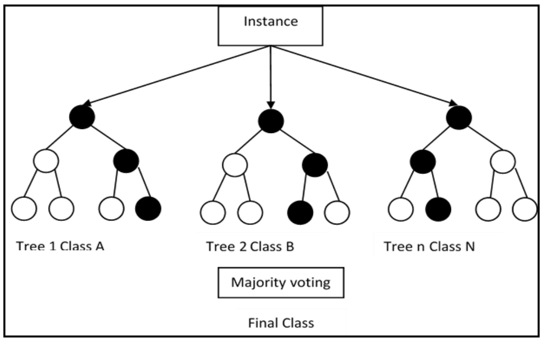


Figure 2. Random forests [14]

3.2.3 Google Application Engine

Google App Engine (GAE) is a part of the services provided by the Google cloud platform. In 2008, Google was declared the App Engine platform to develop and host different web applications in Google data centers, which represents the first cloud computing service from the company. The service was added to the Google platform with multiple cloud services, making it available publicly in November 2011. Google App Engine is essentially an entire file system (not just a locked folder), where the users have more power than a Standard engine, for example, it has read/write permission. GAE is considered a platform as a service with serverless computing environments. The users in the GAE standard can have a fixed array of libraries already set up and the user cannot distribute 3rd party libraries. In a flexible environment, the user can load the library your application requires, including custom build environments (such as Python 3) [18].

3.2.4 Representative State Transfer

Representative State Transfer (REST) is mainly used as a web service design architecture. At present, the most common types of service communication can be divided into two as REST and Simple Object Access Protocol (SOAP). REST is a more preferred design architecture than SOAP since it provides an advantage over internet usage thanks to its (due to its) less bandwidth consumption. SOAP is a client/server logic protocol that uses the Remote Procedure Call (RPC) model for service calls. In simplified form, when the user creates a web service with SOAP, the service becomes an XML document and then it can call the methods using the definitions in this XML document. The exchange between server and client is always over XML when using SOAP, which means high bandwidth usage. The key difference between REST and SOAP is that the SOAP operates on a standard-based which is XML exchange, whereas the REST does not have such a standard. Sends and retrieves messages in REST, uses XML or JavaScript Object Notation (JSON) format. Messages are forwarded to the server as HTTP requests. Companies that offer cloud-based Application Programming Interface (API) such as Google, Amazon, Twitter, and Microsoft prefer to use the REST architecture due to its simplicity and lightweight structure. When a web service uses REST architecture, it is called RESTful API [19].

4. THE PROPOSED CLOUD-BASED PD DIAGNOSIS SYSTEM

The application of the PD diagnosis system is achieved in the cloud environment. The design and the components of the cloud-based PD diagnostic are shown in Figure 3. The system involves three main parts, which are the cloud-based ML PD diagnosis model, Restful API, and Android Mobile Application.

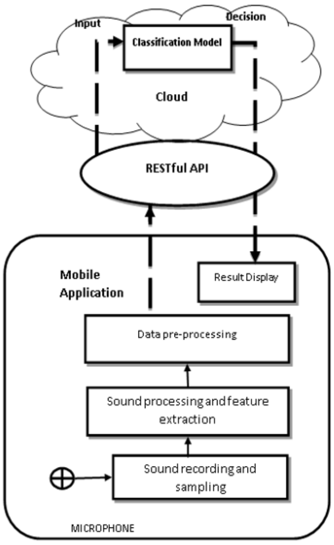


Figure 3. Overall cloud-based Parkinson's disease diagnosis diagram

4.1 Cloud-based ML PD diagnosis model generation

The proposed machine learning method for building the Parkinson’s disease diagnosis model consists of different components including the dataset for Parkinson’s disease diagnostic, the pre-processing stage perpetration the dataset for training the machine learning classification model, Feature selection stage for minimizing the dimensionality for the training data by choosing the best features which provide the highest performance in classification tasks, Machine learning algorithm stage involves training different machine learning algorithms using the training data from the previous stage and compute the model classification performance to choose the best performing method for building the classification model that can be used for Parkinson’s disease diagnostic.

The Parkinson’s dataset (PD) dataset used in this research is obtained by Max Little at the University of Oxford in association with the national center of voice [14]. The dataset consists of the information obtained from voice recordings from a set of 32 persons. Twenty-four of the persons had already PD and 8 of them are healthy. The dataset consists of a total number of 195 voice recordings and 22 features are extracted from each recording. These extracted features are described in Table 1. This is along with a binary PD-score for prediction. Hence, features are extracted from a candidate patient based on the sustained vowel phonation or via running speech test [20]. Regarding the sustained vowel phonation, the persons are requested to enunciate the vowel letter /a/ for a time of fewer than 10 seconds using microphones and recording equipment in the laboratory. A score of 1 refers to a person with PD while a score of 0 means a healthy person. The database that is used in this work is benchmark data available at the machine learning database in Ref. [21].

Table 1. The dataset’s features descriptions

No	Features
1-3	Multi-Dimensional Voice Program (HZ) [min,max,avg]
4	Multi-Dimensional Voice Program (%)
5	Multi-Dimensional Voice Program (abs)
6-8	RAP, PPQ, DDP
9-10	Shimmer/dB
11-12	APQ
13	MDVP
14	DDA
15-16	NHR/HNR
17-18	RPDE
19	DFA
20-21	Spread 1-2
22	PPE

In this work, all of 22 features are considered. The values of these features that belong to the same person are summarized using (min, range, standard deviation, mean, and max) functions, and this process is repeated for each person. Hence, the size of the resulted dataset is 32x110. Then the mRMR feature selection approach is applied to select the best-predefined number of features from the total features space of the dataset. In such a case for the mRMR algorithm, the desired number of features is chosen as (5%, 10%, 25%, 50%, 75%, 100%), respectively.

To prepare the classification data, it has to be divided into two sets or more, one is used for learning the classifier, which is called the training set, while the other set is utilized to check whether the classifier if learns the rule or not, which called the test set.

4.2 RESTful API

RESTful Application Programming Interface (API) is utilized to provide an interfacing mechanism between the mobile application and PD diagnosis model which is located in the cloud. For building and implementing RESTful API on the cloud, the Python framework Flask is used. The mobile application uses the POST method to send data of the patient to the LSTM PD diagnosis model in the cloud and get the prediction result and send it back to the application. The call of RESTful API is seen in Algorithm 2. In order to ensure the operating of the implemented RESTful API that is deployed in the Google cloud for providing the required connection with a mobile application, the API is tested by submitting a data sample of the patient data to the classification model in the cloud computing.

4.3 Mobile application

The Android mobile application for PD diagnosis is written in Java and includes a simple GUI. It contains a record button for recording the sound sample from the patient for diagnosing the PD. The application then displays a result screen with the final diagnosis result to the patient. The decision is received from the ML diagnosis model, which has been deployed in the Google cloud. The pseudocode for the mobile application is described in Algorithm 3 and for the cloud-based PD, the diagnosis Model is described in Algorithm 2. The mobile application is asking the patient to record his/her sound while enunciating the sound /a/ three times in ten seconds period. After recording the sound sample from the patient, the mobile application called Praat software service, which is a Dutch word that means “talk”, that is a sound analysis software to extract the sound features similar to the features used to build the LSTM diagnosis model shown in section (4.1). The obtained features will be converted to JSON format, to be compatible with RESTful API, and then the information has to send to the cloud-based PD the diagnosis model. When the cloud-based model received the data, it produces the diagnosis decision and sends it back to the mobile application.

Algorithm 2: Cloud-based PD Diagnosis Model
Input: Patient sound features
Output: PD predication

```
Model loading and initialization // LSTM - Algorithm1
Wait_Application_request:
If (Data received)
REST_API.GET(Data)
Goto Model_prdiction
Else
Goto Wait_Application_request:
Model_prdiction:
Results=Model.predicate(Data)
Send_model_decision:
RESTful_API.POST(Results) // Return to Algorithm 2
Goto Wait_Application_request:
```

Algorithm 3: Android Application
Input: Sound sample
Output: PD Diagnosis Decision

```
Wait_user_action:
If (Record button pressed)
```

```
Goto Start_recording
Else
Goto Wait_user_action
Start_recording:
Recoder.start()
Wait (10 second)
Recoder.stop()
Recorder.Save(file)
Sound_analysis
Initialize Praat service
Praat.open(file)
Sound_features= Praat.analyze_sound() // Refer to
Table 1for features list
Data_preparation:
Data= format to_JSON (Sound_features)
Submit_data_to_cloud_ml_model
RESTful_API.POST(Data) // Call Algorithm 2
Recive_model_decision_from_colud:
RESTful_API.GET(Results)
Display(Resuts)
If(Results=Negative)
Display("The patient does not have PD")
If(Results=Positive)
Display("The patient have PD")
Goto Wait_user_action.
```

5. RESULTS AND DISCUSSION

This section is divided into two main parts, in the first part, the focus is on the training of the classification models using the data set illustrated earlier in Section 4.1 and a comparison between the results of machine learning (RF) and deep learning (LSTM) is also achieved. In the second part, the best method (LSTM) is compared with the other methods in the related works. The training parameters used in this work for each method are described in Table 2.

Table 2. Models training parameters

Parameter	Value
Loss function (LSTM)	Binary_crossentropy
Activation function (LSTM)	Softmax
Batch Size (LSTM)	64
Learning Rate (LSTM)	0.001
Number of Trees (RF)	20
Loss function (RF)	Entropy

Table 3. Confusion matrix

	Predicted patients with PD	**Predicted healthy persons**
Actual patients with PD	True Positive (TP)	Fales Positive (FP)
Actual healthy persons	True Negative (TN)	False Negative (FN)

In the first part, in order to choose the best classification method for the proposed PD diagnosis system, we carried out a comparison between two classification methods, which are a regular machine learning method called RF and a deep learning method called LSTM. To evaluate and compare the performance of the trained classification methods, two different evaluation metrics are considered including F-score and ROCs [22] to measure the accuracy of the classification models, which can be defined according to the confusion matrix illustrated in Table 3. The classification method with the best performance then will be used in the proposed Parkinson's disease diagnostic system.

- The accuracy of the classification is: (TP + TN) / (TP + TN + FP + FN)
- F-score: is the proportion of actual negative, which are predicted negative: TN / (TN + FP)
- ROC Receiver Operating Characteristics curve is a 2-D curve parameterized with one parameter of the classification algorithm, such as some threshold in the (true positive rate / false positive rate) space, where the Area Under Curve (AUC) is between 0 and 1.

The procedure of the conducted experiment can be summarized as follows: Selecting ***N*** best features from our generated PD dataset using the mRMR feature selection method where N represents several desired features. For each experiment, the percentage number of desired features for the mRMR method is used as (5%, 10%, 25%, 50%, 75%, 100%). The dataset with selected best ***N*** features is used to train each one of the proposed classification methods. After the training process is completed, the results of classification models have tested as well as the evaluation metrics F-score and ROCs are calculated. The evaluation strategy is presented by dividing the dataset into 70% for training samples and 30% for testing samples, which is utilized to evaluate the accuracy of the trained model. Table 4 shows the calculated F-score for each one of the classification methods based on the percentage of desired features selected by the mRMR feature selection method.

Table 4. The F-scores for each classification method based on the percentage of selected features using the mRMR method

Percentage of selected features	**Classification method** Random forest RF	 LSTM deep learning
5%	0.59	0.62
10%	0.69	0.73
25%	**0.84**	**0.95**
50%	0.78	0.90
75%	0.78	0.88
100%	0.80	0.93

For better understanding, the obtained result in Table 3 is visualized as shown in Figure 4 below.

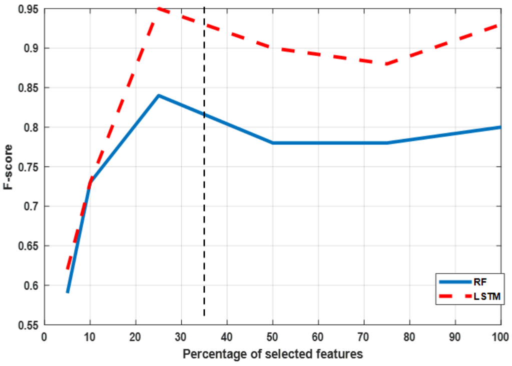


Figure 4. F-scores for each classifier versus the percentage of the selected features

The results in Figure 4 demonstrate that the best performance of the two classification methods is obtained using only 25% of the features selected by the feature selection method. This means that mRMR reduces the number of total features by 75%, which represents selecting only 27 features out of a total of 110 features.

Therefore, the second experiment will involve comparing the two-classification methods based on ROC using the best 25% of the features obtained by the mRMR method as shown in Figure 4. We trained each classifier using the best 25% features selected by the mRMR method and calculate the ROC for each one, as shown in Figure 5.

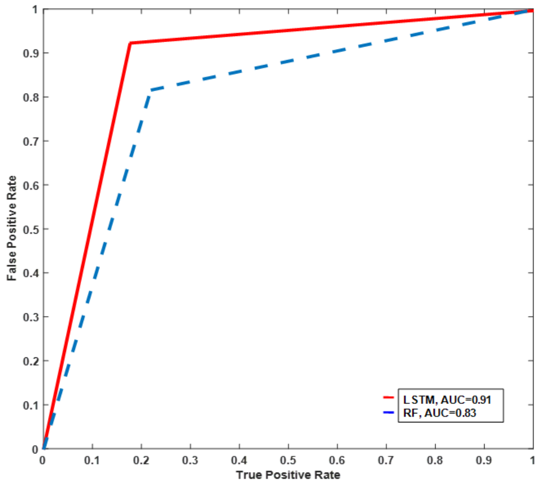


Figure 5. ROC receiver operating characteristic curve results for LTSM and RF classifiers model with the best 27 features selected mRMR

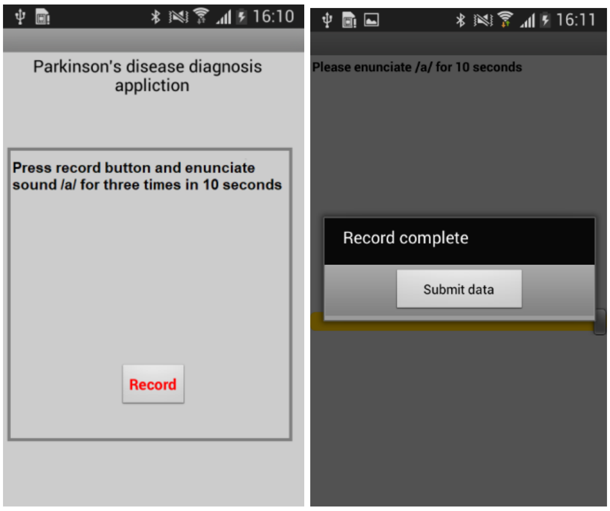
Based on the results of ROC curves shown in Figure 5, it is clearly seen that the LTSM classifier has superior performance compared with the RF classifier since the AUC of LSTM is 0.91, which is greater than the AUC of the RF (0.83). Therefore, the LSTM classifier will be used for building the classification model using the PD dataset with 27 features selected by the mRMR method in designing the cloud-based Parkinson's disease diagnosing system.

Table 5. A comparison between the results of proposed deep learning PD diagnosis model and the other related works

Reference of related work	Method	Accuracy
[5]	mRMR and Deep Neural networks	85
[6]	Convolutional Neural Network (CNN)	91.7
[7]	Convolutional Neural Networks (CNN) and Artificial Neural Network (ANN)	89.1
[8]	CNN-LSTM	88.5
The Proposed System	mRMR and LSTM	95

The LSTM model is built and trained using Python and saved on Google cloud Application Engine to be used in the RESTful API and the mobile application, which represents the implementation of the second stage of this work. Figure 6 shows the results of testing the mobile application for diagnosing PD.

In the second part, the results of the proposed deep learning model (LSTM) are compared with related deep learning methods from the literature that have been applied to PD detection based on voice recordings data. Table 5 shows a brief comparison with the prior research studies in teams of the achieved accuracy. Based on the comparison of Table 5, it can be noticed that the proposed PD diagnosis system outweighs a number of existing studies in terms of detection accuracy.



Parkinson's disease diagnosis appliction

Press record button and enunciate sound /a/ for three times in 10 seconds

Please enunciate /a/ for 10 seconds

Record complete

Figure 6. Mobile application testing screens

6. CONCLUSION

In this paper, machine learning approaches are used in feature selection and classification for designing a cloud-based Parkinson's disease diagnosing system. LSTM and RF classification algorithms are used for generating the diagnosis model while the mRMR feature selection algorithm is applied to minimize the number of features. To gain the advantage of the cloud computing environment, the diagnosis model is deployed in the Google cloud application engine. Hence, an android mobile application is developed with a simple GUI to provide the PD test for a patient anywhere at any time.

The experiment shows that the mRMR feature selection approach yields the best classification performance results by reducing 75% of the overall features of the PD dataset. Based on the obtained classification performance (F-score and ROC) results, the application of LSTM classification shows superior accuracy with 95% of the F-score compared to the RF classifier. Therefore, the LSTM model has been used to implement the proposed cloud-based PD diagnosing system. Further work can be done by using an embedded hardware-based PD diagnosis system as an alternative and redundant to the online cloud-based system, which can diagnose PD offline without Internet availability. Further work can be presented by investigating another type of dataset, deep learning, and classifier methods in order to improve the overall system accuracy.

REFERENCES

[1] Tysnes, O.B., Storstein, A. (2017). Epidemiology of Parkinson's disease. Journal of Neural Transmission, 124(8): 901-905. https://doi.org/10.1007/s00702-017-1686-y

[2] Spencer, K.A., Paul, J., Brown, K.A., Ellerbrock, T.,

Sohlberg, M.M. (2020). Cognitive rehabilitation for individuals with Parkinson's disease: Developing and piloting an external aids treatment program. American Journal of Speech-Language Pathology, 29(1): 1-19. https://doi.org/10.23641/asha.10093493

[3] Miller, N., Nath, U., Noble, E., Burn, D. (2017). Utility and accuracy of perceptual voice and speech distinctions in the diagnosis of Parkinson’s disease, PSP and MSA-P. Neurodegenerative Disease Management, 7(3): 191-203. https://doi.org/10.2217/nmt-2017-0005

[4] De la Prieta, F., Rodríguez-González, S., Chamoso, P., Corchado, J.M., Bajo, J. (2019). Survey of agent-based cloud computing applications. Future Generation Computer Systems, 100: 223-236. https://doi.org/10.1016/j.future.2019.04.037

[5] Wroge, T.J., Özkanca, Y., Demiroglu, C., Si, D., Atkins, D.C., Ghomi, R.H. (2018). Parkinson’s disease diagnosis using machine learning and voice. Proceedings of the 2018 IEEE Signal Processing in Medicine and Biology Symposium (SPMB), Philadelphia, PA, pp. 1-7. https://doi.org/10.1109/SPMB.2018.8615607

[6] Wodzinski, M., Skalski, A., Hemmerling, D., Orozco-Arroyave, J.R., Nöth, E. (2019). Deep learning approach to Parkinson’s disease detection using voice recordings and convolutional neural network dedicated to image classification. Proceedings of the 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Berlin, Germany, pp. 717-720. https://doi.org/10.1109/EMBC.2019.8856972

[7] Johri, A., Tripathi, A. (2019). Parkinson disease detection using deep neural networks. Proceedings of the 2019 Twelfth International Conference on Contemporary Computing (IC3), NOIDA, India, pp. 1-4. https://doi.org/10.1109/IC3.2019.8844941

[8] Mallela, J., Illa, A., Suhas, B.N., Udupa, S., Belur, Y., Atchayaram, N., Ghosh, P.K. (2020). Voice based classification of patients with Amyotrophic Lateral Sclerosis, Parkinson’s disease and healthy controls with CNN-LSTM using transfer learning. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6784-6788. https://doi.org/10.1109/ICASSP40776.2020.9053682

[9] Sarveniazi, A. (2014). An actual survey of dimensionality reduction. American Journal of Computational Mathematics, 4: 55-72. https://doi.org/10.4236/jasmi.2014.42006

[10] Jameel, N., Abdullah, H.S. (2021). Intelligent feature selection methods: A survey. Engineering and Technology Journal, 39(1B): 175-183. https://doi.org/10.30684/etj.v39i1B.1623

[11] Xiu, Y., Zhao, S., Chen, H., Li, C. (2019). I-mRMR: incremental max-relevance, and min-redundancy feature selection. Proceedings of the Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data, Chengdu, China, pp. 103-110. https://doi.org/10.1007/978-3-030-26075-0_8

[12] Peng, H., Long, F., Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(8): 1226-1238. https://doi.org/10.1109/TPAMI.2005.159

[13] Kumar, N., Mitra, S., Bhattacharjee, M., Mandal, L. (2019). Comparison of different classification techniques using different datasets. In Proceedings of International Ethical Hacking Conference 2018, Springer, Singapore, pp. 261-272. https://doi.org/10.1007/978-981-13-1544-2_22

[14] Ellefsen, A.L., Bjørlykhaug, E., Æsøy, V., Ushakov, S., Zhang, H. (2019). Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture. Reliability Engineering & System Safety, 183: 240-251. https://doi.org/10.1016/j.ress.2018.11.027

[15] Albayati, A.Q., Ameen, S.H. (2020). A method of deep learning tackles sentiment analysis problem in Arabic texts. Iraqi Journal of Computers, Communications, Control and Systems Engineering, 20(4): 9-20.

[16] Zhu, M., Xia, J., Jin, X., Yan, M., Cai, G., Yan, J., Ning, G. (2018). Class weights random forest algorithm for processing class imbalanced medical data. IEEE Access, 6: 4641-4652. https://doi.org/10.1109/ACCESS.2018.2789428

[17] Shihab, T. H., Al-Hameedawi, A.N., Hamza, A.M. (2020). Random Forest (RF) and Artificial Neural Network (ANN) Algorithms for LULC Mapping. Engineering and Technology Journal, 38(4A): 510-514. https://doi.org/10.30684/etj.v38i4A.399

[18] Joshi, N., Shah, S. (2019). A comprehensive survey of services provided by prevalent cloud computing environments. In Proceedings of Smart Intelligent Computing and Applications, Springer, Singapore, pp. 413-424. https://doi.org/10.1007/978-981-13-1921-1_41

[19] Belkhir, A., Abdellatif, M., Tighilt, R., Moha, N., Guéhéneuc, Y.G., Beaudry, É. (2019). An observational study on the state of REST API uses in Android mobile applications. In Proceedings of the 2019 IEEE/ACM 6th International Conference on Mobile Software Engineering and Systems (MOBILESoft), IEEE, Montreal, QC, Canada, Canada, pp. 66-75. https://doi.org/10.1109/MOBILESoft.2019.00020

[20] Farzana, W., Hossain, Q.Z.D. (2019). Optimization of features for classification of Parkinson's disease from vocal dysphonia. In Proceedings of the 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE), IEEE, Cox'sBazar, Bangladesh, Bangladesh, pp. 1-6. https://doi.org/10.1109/ECACE.2019.8679126

[21] Shruthi, U., Nagaveni, V., Raghavendra, B. (2019). A review on machine learning classification techniques for plant disease detection. In Proceedings of the 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), IEEE, Coimbatore, India, pp. 281-284. https://doi.org/10.1109/ICACCS.2019.8728415

[22] Kadam, V.J., Jadhav, S.M. (2019). Feature ensemble learning based on sparse autoencoders for diagnosis of Parkinson’s disease. Computing, Communication and Signal Processing, pp. 567-581. https://doi.org/10.1007/978-981-13-1513-8_58