IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Diabetes Prediction Using Ensembling of Different Machine Learning Classifiers

MD. KAMRUL HASAN[1], MD. ASHRAFUL ALAM[1], DOLA DAS[2], EKLAS HOSSAIN[3] (Senior Member, IEEE), MAHMUDUL HASAN[2]

[1]Department of Electrical and Electronic Engineering, Khulna University of Engineering & Technology, Khulna-9203, Bangladesh (e-mail: m.k.hasan@eee.kuet.ac.bd, alam1603001@stud.kuet.ac.bd)
[2]Department of Computer Science and Engineering, Khulna University of Engineering & Technology, Khulna-9203, Bangladesh (e-mail: {dola.das, mahmudul}@cse.kuet.ac.bd)
[3]Department of Electrical Engineering & Renewable Energy, Oregon Renewable Energy Center (OREC), Oregon Institute of Technology, OR 97601, USA (e-mail: Eklas.hossain@oit.edu)

Corresponding author: Md. Kamrul Hasan (m.k.hasan@eee.kuet.ac.bd)

ABSTRACT Diabetes, also known as chronic illness, is a group of metabolic diseases due to a high level of sugar in the blood over a long period. The risk factor and severity of diabetes can be reduced significantly if the precise early prediction is possible. The robust and accurate prediction of diabetes is highly challenging due to the limited number of labeled data and also the presence of outliers (or missing values) in the diabetes datasets. In this literature, we are proposing a robust framework for diabetes prediction where the outlier rejection, filling the missing values, data standardization, feature selection, K-fold cross-validation, and different Machine Learning (ML) classifiers (k-nearest Neighbour, Decision Trees, Random Forest, AdaBoost, Naive Bayes, and XGBoost) and Multilayer Perceptron (MLP) were employed. The weighted ensembling of different ML models is also proposed, in this literature, to improve the prediction of diabetes where the weights are estimated from the corresponding Area Under ROC Curve (AUC) of the ML model. AUC is chosen as the performance metric, which is then maximized during hyperparameter tuning using the grid search technique. All the experiments, in this literature, were conducted under the same experimental conditions using the Pima Indian Diabetes Dataset[a]. From all the extensive experiments, our proposed ensembling classifier is the best performing classifier with the sensitivity, specificity, false omission rate, diagnostic odds ratio, and AUC as 0.789, 0.934, 0.092, 66.234, and 0.950 respectively which outperforms the state-of-the-art results by $2.00\,\%$ in AUC. Our proposed framework for the diabetes prediction outperforms the other methods discussed in the article. It can also provide better results on the same dataset which can lead to better performance in diabetes prediction. Our source code for diabetes prediction is made publicly available[b].

[a]https://www.kaggle.com/uciml/pima-indians-diabetes-database
[b]https://github.com/kamruleee51/Diabetes-Prediction-Using-ML-Classifiers

INDEX TERMS Diabetes Prediction, Ensembling Classifier, Machine Learning, Multilayer Perceptron, Missing Values and Outliers, Pima Indian Diabetic Dataset.

I. INTRODUCTION

DIABETES is a very familiar word in the present world and crucial challenges in both developed and developing countries [1]. The insulin hormone in the body produced by the pancreas allows glucose to pass from the food into the bloodstream. The lack of that hormone due to malfunctioning of the pancreas forms diabetes which can result in coma, renal and retinal failure, pathological destruction of pancreatic beta cells, cardiovascular dysfunction, cerebral vascular dysfunction, peripheral vascular diseases, sexual dysfunction, joint failure, weight loss, ulcer, and pathogenic effects on immunity [2]. Research on diabetes patients demonstrates that diabetes among adults (over 18 years old) has risen from $4.7\,\%$ to $8.5\,\%$ in 1980 to 2014 respectively and rapidly growing up in second and third world countries [3]. Statistical results in 2017 show that 451 million people were living with diabetes worldwide, which will increase to 693 million by 2045 [4]. Another statistical study in [5] shows the severity of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2989857, IEEE Access

diabetes, where they reported that half a billion people have diabetes worldwide, and the number will increase to $25\,\%$ and $51\,\%$ respectively in 2030 and 2045. However, there is no long term cure for diabetes, but it can be controlled and prevented if an early prediction is accurately possible. The prediction of diabetes is a challenging task, as the distribution of classes for all attributes is not linearly separable as depicted in Fig. 1.

In recent years, plenty of methods have been proposed and published for diabetes prediction. A ML based framework was proposed in [7] where authors implemented the Linear Discriminant Analysis (LDA) [8], Quadratic Discriminant Analysis (QDA) [9], Naive Bayes (NB) [10], Gaussian Process Classification (GPC) [11], Support Vector Machine (SVM) [12], Artificial Neural Network (ANN) [13], AdaBoost (AB) [14], Logistic Regression (LR) [15], Decision Tree (DT) [16], and Random Forest (RF) [17] with different dimensionality reduction and cross-validation techniques. They also performed extensive experiments on the outlier rejection and filling missing values for boosting the performance of the ML model, where they were able to obtain the highest possible AUC of 0.930. In [18], authors employed three different ML classifiers such as DT, SVM, and NB to prognosticate the likelihood of diabetes with maximum accuracy. They demonstrated that NB is the best performing model with the AUC of 0.819. The AB and bagging ensemble techniques using $J48$ ($c4.5$)-DT, as a base learner and standalone data mining technique ($J48$), have been studied and implemented in [19] for the classification of diabetes mellitus. The experimental results of them prove that the AB ensemble method is better than bagging and standalone $J48$-DT. Genetic programming for the prediction of diabetes had proposed in [20] where the framework outperformed as compared to other implemented techniques by them. Authors, in [21], employed four ML methods such as DT, ANN, LR, and NB to classify the risk of diabetes mellitus, where they boosted the robustness by bagging and boosting techniques. The experimental results show that the RF algorithm gives optimum results among all the employed algorithms. Gaussian Process (GP)-based classification technique was proposed, in [22], using three different kernels (linear, polynomial, and radial basis function) and compared against the traditional LDA, QDA, and NB. The authors also performed extensive experiments to search for the best cross-validation protocol. Their experiments demonstrate that the GP-based classifier with the $K10$ cross-validation protocol is the best performing classifier for the diabetes prediction. Although there are numerous frameworks already been published, in recent years, still, the improvement requires in the preciseness and robustness for diabetes prediction.

In this literature, We propose a new pipeline for diabetes prediction from the PIMA Indians Diabetes dataset. Preprocessing, in the proposed pipeline, is the heart of achieving the state-of-the-art result, which consists of outlier rejection, filling missing values, data standardization, feature selection, and K-fold cross-validation. We consider the mean value in the missing position of attribute rather than median value, as it has a more central tendency toward the mean of that attribute distribution. The folding of the dataset for cross-fold validation is performed carefully to preserve the percentage of class proportion, as same as in the original dataset. Different ML classifiers (k-nearest Neighbour (k-NN), RF, DT, NB, AB, and XGBoost (XB)) and MLP were implemented in our proposed pipeline. We apply the grid search technique for selecting the number of hidden layers, number of neurons in each hidden layer, activation function, neuron initializer, batch size, learning rate, epoch, percentage of dropped neurons, loss function, an optimizer of MLP and hyperparameters of ML models. Extensive experiments are performed on different combinations of preprocessing and ML classifiers for maximizing the AUC of diabetes prediction under the same experimental conditions and dataset. The best ML classifier is then set as a baseline model to evaluate our proposed classifier quantitatively for the prediction of diabetes precisely. Moreover, we propose an ensembling classifier by the combination of the ML models for boosting the diabetes prediction. To ensemble the ML models, soft weighted voting is employed, where the weight for the individual model was estimated from the respective AUC. The AUC of the ML model is chosen as the weight of that model for voting ensembling rather than accuracy since AUC is unbiased to the class distribution. Extensive experiments on different combinations of the ML models are accomplished for searching the best ensemble classifier where the best performing preprocessing from the previous experiments is employed.

The organization of the remaining paper is as follows: Section II presents the dataset, proposed methodology, and evaluation metrics. In section III, the different experimental results are reported with the interpretation. Finally, the paper is concluded with future works in section IV.

II. MATERIALS AND METHODS

This section focuses on materials and methods used for this study, in the literature, where the subsections II-A, II-B, and II-C respectively explain the dataset, proposed framework, and hardware & metrics used to evaluate the framework.

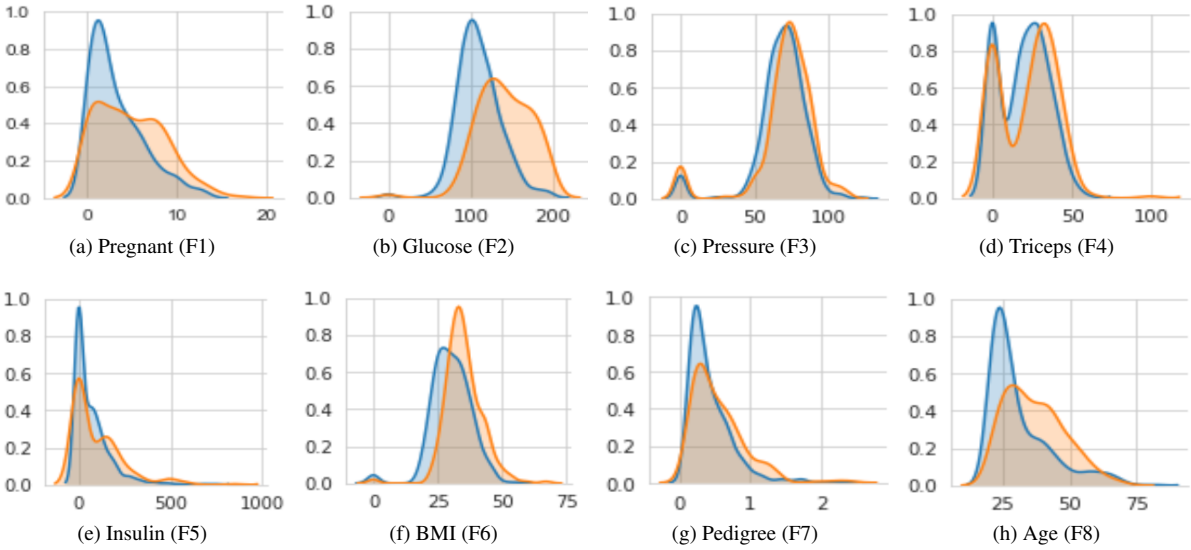
A. DATASET

The ML models were trained and tested on publicly available PIMA Indians Diabetes (PID) dataset of 768 female diabetic patients from the Pima Indian population near Phoenix, Arizona [6]. This dataset consists of 268 diabetic patients (positive) and 500 non-diabetic patients (negative) with eight different attributes. The descriptions of the attributes and brief statistical summary are shown in Table 1. The Pedigree (Diabetes Pedigree Function) was calculated [6] as in (1).

$$Pedigree = \frac{\sum_i K_i(88 - ADM_i) + 20}{\sum_j K_j(ALC_j - 14) + 50} \tag{1}$$

where i and j respectively denote the relatives who had developed and NOT developed diabetes. K is the percentage

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2989857, IEEE Access



(a) Pregnant (F1) (b) Glucose (F2) (c) Pressure (F3) (d) Triceps (F4)

(e) Insulin (F5) (f) BMI (F6) (g) Pedigree (F7) (h) Age (F8)

FIGURE 1: The population distribution of all attributes in the PIMA Indian Diabetes Dataset [6] where blue and orange color distribution respectively denotes non-diabetes and diabetes class.

TABLE 1: The overview of the diabetic patient cohort.

SN	Attributes	Description	Mean ± Std
1	Pregnant (F1)	Number of times pregnant	3.85 ± 3.37
2	Glucose (F2)	Plasma Glucose Concentration at 2 Hours in an Oral Glucose Tolerance Test	120.90 ± 31.97
3	Pressure (F3)	Diastolic Blood Pressure (mm Hg)	69.11 ± 19.36
4	Triceps (F4)	Triceps Skin Fold Thickness (mm)	20.54 ± 15.95
5	Insulin (F5)	2-Hour Serum Insulin (μU/ml)	79.81 ± 115.24
6	BMI (F6)	Body Mass Index (Weight in kg / (Height in inches)^2)	32.00 ± 7.88
7	Pedigree (F7)	Diabetes Pedigree Function	0.47 ± 0.33
8	Age (F8)	Age in years	33.24 ± 11.76

of shared genes by the relatives ($K = 0.500$ for the parent or full sibling, $K = 0.250$ for a half-sibling, grandparent, aunt or uncle and $K = 0.125$ for a half aunt, half-uncle or first cousin). ADM_i and ACL_j is the age of relatives, in years, at the time of diagnosing and at the last non-diabetic test respectively.

B. PROPOSED FRAMEWORK

The proposed framework, in this literature, has been illustrated in Fig. 2 where the preprocessing of raw data is the integral step in the proposed pipeline, as the quality of data can drive the classifiers to learn directly.

1) Preprocessing

In the proposed framework, the preprocessing step includes outlier rejection (P), filling missing values (Q), standardization (R), and feature selection of the attribute which are briefly described as follows:

The outlier [23] is a markedly deviated observation from other observations. It requires to be rejected from data distribution as the classifiers are very much sensitive to the data range and distribution of the attributes. The mathematical formulation for the outlier rejection in this literature can be written as in (2).

$$P(x) = \begin{cases} x, & \text{if } Q_1 - 1.5 \times IQR \leq x \leq Q_3 + 1.5 \times IQR \\ reject, & \text{otherwise} \end{cases} \tag{2}$$

where x is the instances of the feature vector that lies in n-dimensional space, $x \in \mathcal{R}^n$. Q_1, Q_3, and IQR is the first quartile, third quartile, and interquartile range of the attributes respectively, where Q_1, Q_3, $IQR \in \mathcal{R}^n$.

The attributes, after outlier rejection, were processed to fill the missing or null values [24] as they could lead to the wrong prediction for any classifiers. In the proposed framework, the missing or null values were imputed by the mean values of the attributes rather than dropping, which can be formulated as in (3). The imputation with the mean is beneficial as it imputes the continuous data without introducing outliers.

$$Q(x) = \begin{cases} mean(x), & \text{if } x = null/missed \\ x, & \text{otherwise} \end{cases} \tag{3}$$

where x is the instances of the feature vector that lies in n-dimensional space, $x \in \mathcal{R}^n$.

The standardization or Z-score normalization is the technique to rescale the attributes for achieving standard normal

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2989857, IEEE Access

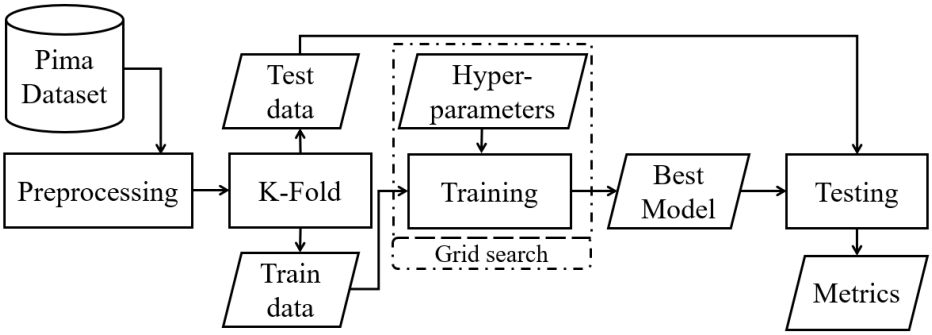


FIGURE 2: The proposed block diagram of a robust and automatic diabetes prediction.

distribution with zero mean and unit variance. The standardization (R), as shown in (4), also reduces the skewness of the data distribution.

$$R(x) = \frac{x - \bar{x}}{\sigma} \tag{4}$$

where x is the n-dimensional instances of the feature vector, $x \in \mathcal{R}^n$. $\bar{x} \in \mathcal{R}^n$ and $\sigma \in \mathcal{R}^n$ are the mean and standard deviation of the attributes. However, in many ML models such as tree-based models are probably the models, where feature standardization can't provide a guarantee for significant improvement.

The accuracy of the classifiers increases with the increment of the attribute's dimension. However, the performance of the classifiers will tend to reduce when the attribute's dimension increases without increasing the samples. Such a scenario, in machine learning, is referred to as a curse of dimensionality. Due to a curse of dimensionality, the space of the feature becomes sparser and sparser which forces the classifiers to be overfitted by loosing generalizing capability. In this literature, three most commonly used methods for the feature selection namely Principle Component Analysis (PCA) [25], Independent Component Analysis (ICA) [26], and Correlation-based [27] technique were used to compare their performance for the PID dataset. The details algorithm of PCA, ICA and Correlation-based technique are given in Appendix A, Appendix B, and Appendix C respectively.

2) Cross-fold Validation

The K-fold Cross-validation (KCV) technique is one of the most widely used approaches by practitioners for model selection and error estimation of classifiers [28]. The pictorial presentation of the data splitting (5-fold cross-validation), used in this literature, is shown in Fig. 3. The PID dataset has partitioned into K folds. The $K-1$ folds are used to train and fine-tune the hyperparameters in the inner loop where the grid search algorithm [29] was employed. In the outer loop (K times), the best hyperparameters and the test data were used to evaluate the model. Since the PID dataset contains an imbalanced positive and negative samples, the stratified KCV [30] has been used to preserve the percentage of samples for each class as same as in the original percentage. The final performance metric was estimated using the equation as in (5).

$$M = \frac{1}{K} \times \sum_{n=1}^{K} P_n \pm \sqrt{\frac{\sum_{n=1}^{K}(P_n - \bar{P})^2}{K-1}} \tag{5}$$

where M is the final performance metric for the classifiers and $P_n \in \mathcal{R}$, $n = 1, 2, ..., K$ is the performance metric for each fold.

3) ML Model and Ensembling

Different ML models such as k-NN [31], DT, AB, RF, NB, and XB [32] have been trained (see Appendix D, Appendix E, Appendix F, Appendix G, Appendix H, and Appendix I respectively) and tested in the proposed framework. The hyperparameters which will tune, in the inner loop, are shown in Table 2. The ensembling of the ML model is the well-known technique to boost the performance using a group of classifiers [33], [34]. In ensembling, the aggregation of the output from different models can improve the precision of the prediction. The output from each model, $Y_j\,(j = 1, 2, 3, ..., m = 6) \in \mathcal{R}^C$ assigns $C = 2$ (either having diabetes, C_1 or not, C_2) confidence values $P_i \in \mathcal{R}\,(i = 1, 2)$ to the unseen test data where $P_i \in [0, 1]$ and $\sum_{i=1}^{C} P_i = 1$. The weighted aggregation of different ML models in this literature was performed using the equation as in (6).

$$P_i^{en} = \frac{\sum_{j=1}^{m=6}(W_j \times P_{ij})}{\sum_{C=2}^{i=1}\sum_{m=6}^{j=1}(W_j \times P_{ij})} \tag{6}$$

where the weight, W_j is the corresponding AUC of that j^{th} classifier. Since we are proposing a weighted soft voting ensemble, we need an imbalanced, as in the PID dataset, unbiased metric as a weight. That is why we choose AUC as a weight for the proposed ensembling classifier. The output of the ensembled model, $Y \in \mathcal{R}^C$ has the confidence values $P_i^{en} \in [0, 1]$. The final class label of the unseen data,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2989857, IEEE Access

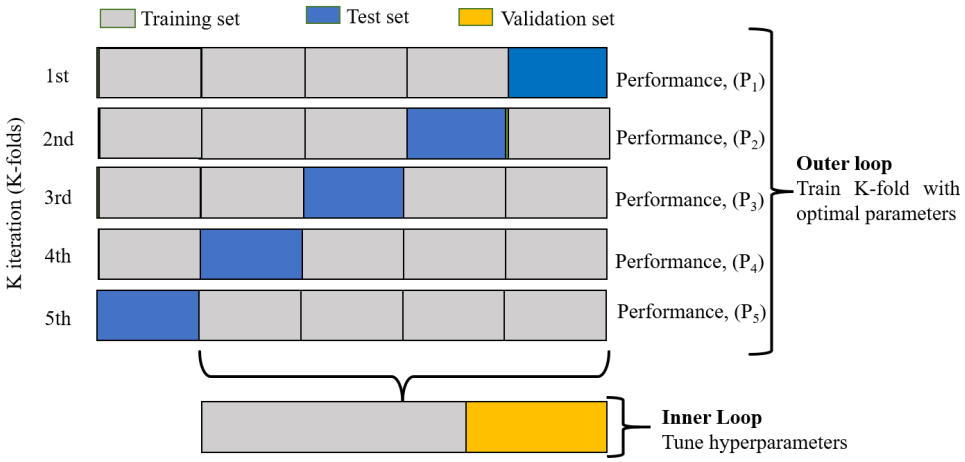


FIGURE 3: The partitioning of the PID dataset for KCV for both the hyperparameters tuning and evaluation.

TABLE 2: Different ML models with hyperparameters to be tuned by the grid search technique in the inner loop.

ML Models	hyperparameters
k-NN	1) Number of neighbors for queries 2) Computing algorithm for nearest neighbors • **Ball Tree (BT):** Node defines a D-dimensional hypersphere or ball • **KD Tree (KDT):** Leaf node is a D-dimensional point • **Brute:** Based on the brute-force search 3) Leaf size for BT or KDT which depends on the nature of problem 4) Metric (Manhattan distance (L_1-norm) or Euclidean distance (L_2-norm))
DT	1) Measuring function: **Gini impurity** or **Entropy** 2) The strategy used to choose the split at each node 3) The minimum samples for an internal node 4) The minimum samples for a leaf node.
RF	1) The trees in the forest. 2) Measuring function: **Gini impurity** or **Entropy**
AB	1) The boosting algorithm (Real boosting or Discrete boosting) 2) Learning rate to shrink the contribution of each classifier 3) The maximum number of estimators to terminate the boosting
NB	1) Portion of the largest variance of the attributes
XB	1) Minimum sum of instance weight (Hessian) 2) Minimum loss reduction for further partitioning on the leaf node 3) Subsample ratio of the training instance 4) Subsample ratio for constructing each tree 5) Maximum tree depth

$X \in \mathcal{R}^n$ from ensembled model will be C_i if $P_i^{en} = max(Y(X))$.

4) Multilayer Perceptron (MLP)

A neural network consists of processing units, called neurons, where each neuron is connected to other neurons by unidirectional connections of different weights [35]. A feed-forward neural network or MLP used, in this paper, is shown in Fig. 4 which consists of an input-output layer and several hidden layers. The D-dimensional input vector of any layer of MLP produces N-dimensional output vector, $f(x) : \mathbb{R}^D \rightarrow \mathbb{R}^N$. The output of each processing unit can be expressed as in (7).

$$f(x) = \Phi\left(\sum_j w_j x_j + b\right) \tag{7}$$

where the x_j, w_j, b and Φ are the inputs, weights, bias to the neuron and the nonlinear activation function respectively. The parameters of the neuron are updated as in (8) during the training using back-propagation [36] to minimize the error, $\gamma = y_{true} - y_{output}$.

$$w_{new} = w_{old} + \eta \times \gamma \tag{8}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2989857, IEEE Access

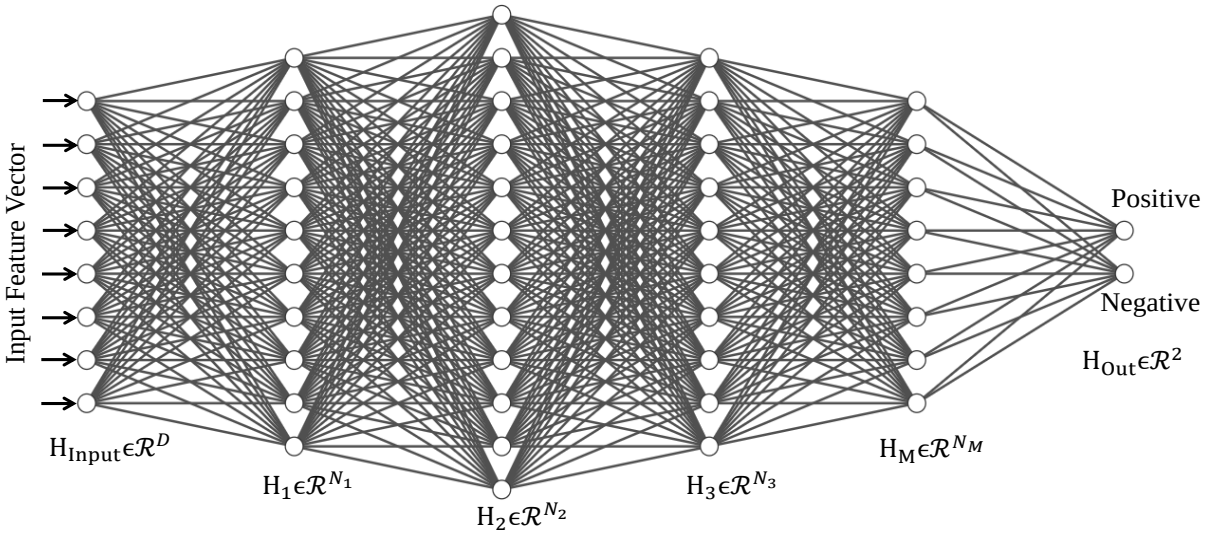


FIGURE 4: The MLP architecture, with M hidden layers (H) and N_M neurons in H_M layer, for diabetes prediction in the proposed framework.

where η is the learning rate, which is the amount at which the weights are updated during the training. However, it is very uncertain to guesstimate the number of the hidden layers (H_M) and neurons (N_M) at each hidden layer as they highly depend on the dataset. The more number of layers and neurons will have more parameters that can not provide any guarantee to have better performance. The more the parameters, the more the samples require in the training dataset. However, in this paper, we are learning those hyperparameters from the PID dataset. The hyperparameters such as the number of hidden layers, number of neurons in each hidden layer, activation function, neuron initializer, batch size, learning rate, epoch, percentage of dropped neurons, loss function, the optimizer will be used in the grid search for optimizing to maximize the AUC.

C. EVALUATION METRICS

The models were implemented using the Python programming language with different Python and Keras APIs and the experiments were carried out on a machine running *Windows-10* operating system with the following hardware configuration: Intel® Core™ i7-7700 HQ CPU @ $2.80\,GHz$ processor with Install memory (RAM): $16.0\,GB$ and GeForce GTX 1060 GPU with $6\,GB$ GDDR5 memory.

All the extensive experiments were evaluated using several metrics where each metric has a different meaning of evaluation. The confusion matrix of True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN) along with different metrics e.g. Sensitivity (Sn), Specificity (Sp), Precision (Pr), False Omission Rate (FOR), and Diagnostic Odds Ratio (DOR) [37] has been reported. The Sn and Sp are respectively used to quantify the type-II error (the patient having positive symptoms, but erroneously fails to be rejected) and type-I error (the patient having negative symptoms, but detected as positive). Pr, FOR, and DOR have been used to evaluate the percentage of correctly classified diabetes patients having positive conditions, the proportion of the individuals with a negative test result, for which the true condition is positive, and the effectiveness of a diagnostic test respectively. Additionally, the Receiver Operating Characteristics (ROC) with Area Under the ROC Curve (AUC) is also reported to measure how well predictions are ranked, rather than their absolute values.

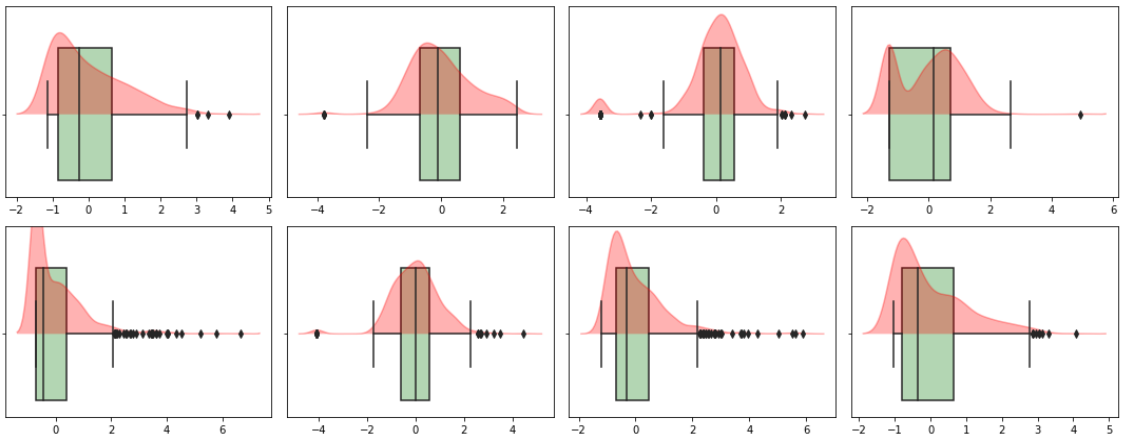
III. RESULTS AND DISCUSSION

This section presents the different extensive experiments with the corresponding results in several subsections. The results for preprocessing and ML model are described in subsections III-A and III-B respectively. The subsections III-C and III-D are dedicated to represent the results for MLP and ensembling classifiers respectively, and the subsection III-E compares the results.

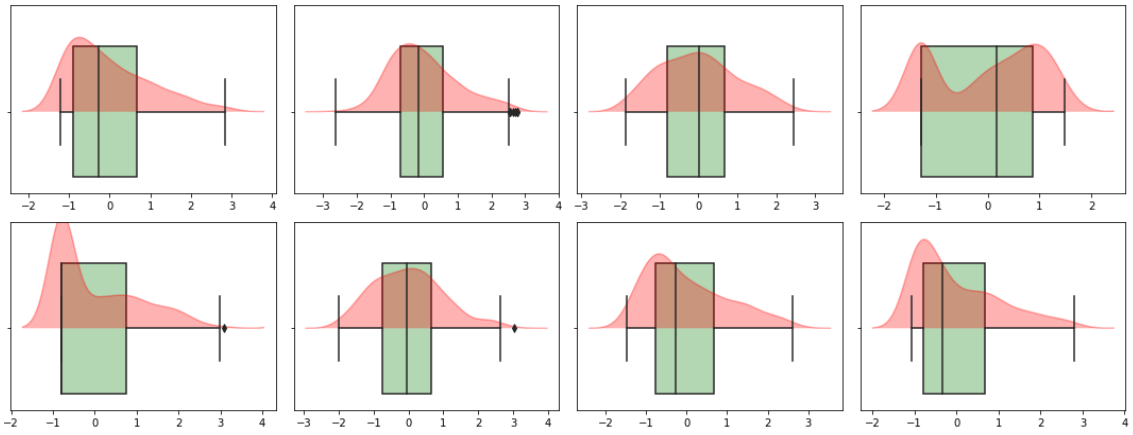
A. RESULTS FOR PREPROCESSING

The class-wise distribution of the attributes (see Fig. 1) demonstrates the complexity of distinguishing positive and negative diabetes in the PID dataset. Most of the attributes also have the skewness (positive and negative) and leptokurtic distribution. However, the presence of the outlier introduces the skewness and kurtosis (see Fig. 5 (a)) in the attribute's distribution where the high kurtosis is an indicator of heavy tails or outliers in the PID dataset. The presence of the skewness and kurtosis will tend to underestimate and overestimate the expected value respectively. The result for the outlier rejection (see Fig. 5) demonstrates that the skewness of the distribution moves to the zero means, which indicates the mean and median of the attribute have coincided approximately (see Fig. 5 (b)). The leptokurtic ($kurtosis > 3$) distribution of the PID dataset also moves to a mesokurtic distribution ($kurtosis = 3$). The confusion

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2989857, IEEE Access

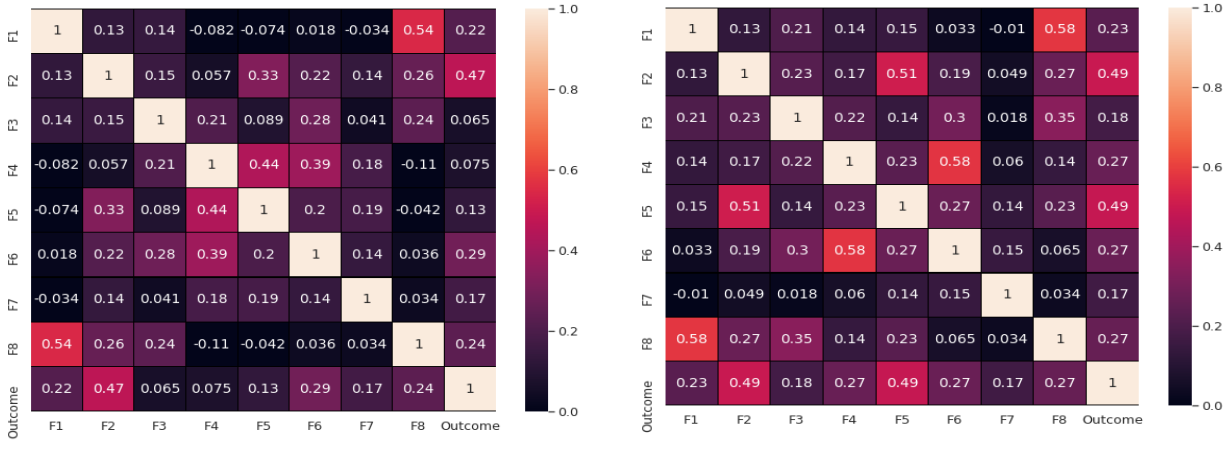


(a) Presence of outlier in data.



(b) After rejection of outlier.

FIGURE 5: The distribution of attributes with box plot (a) with and (b) without outliers, where the first row is for F1, F2, F3, and F4 attributes (left to right) and the second row is for F5, F6, F7, and F8 attributes (left to right) for both (a) & (b).



(a) (b)

FIGURE 6: The confusion matrix of the attribute's correlation with the outcome for (a) raw and (b) preprocessed PID dataset.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2989857, IEEE Access

matrix of the correlation (see Fig. 6) presents the result for the outlier rejection and filling missing values together. The qualitative and quantitative analysis on the Fig. 6 (a) and Fig. 6 (b) demonstrate that the correlation of the attribute with the target outcome has improved after applying outlier rejection and filling the missing values where the correlation coefficient, especially for the $F3$, $F4$, and $F5$, have improved significantly. The improved correlation is the beneficiary for the correlation-based feature selection (see Appendix C).

B. RESULTS FOR ML MODEL

Table 3 shows the quantitative results for the selection of the best performing preprocessing and ML model where the AUC with standard deviation is reported for the comparison among them. The summary of each model's capability of achieving the best AUC from the proposed pipeline, with corresponding best preprocessing and attribute selection algorithm as well as the number of selected attributes, has reported in Table 4. The best-tuned hyperparameters using the grid search are also shown in Table 4. The investigation on Table 3 provides evidence of getting better results from different models when we employ suitable preprocessing for them.

All the classifiers demonstrate their respective best results for outlier rejection and filling missing values when the correlation-based feature selection is employed (see Table 3 and Table 4). The first two experiments, as shown in Table 3, show that the boosting classifiers (AB & XB) beat all the classifiers in AUC. The AB performs better for the raw data ($x \in \mathcal{R}^8$), and XB performs better when only the outliers are rejected ($x \in \mathcal{R}^8$) from the PID dataset. The performance of the XB has improved by a $0.6\,\%$ margin when only the outliers are rejected (P). These two experiments show that XB is affected by the outlier, in the PID dataset, more than AB, although XB has extreme gradient boosting capabilities. There is a possibility of overfitting in XB as it assigns equal weight to all the weak base-learners, whereas AB assigns more weight to the weak base-learners having better performance. The building of a new tree depends on the residuals of the previous tree, where the outliers will have much larger residuals than non-outliers. XB does not penalize those residuals as in AB. Moreover, after applying PCA and ICA on outlier rejected data, the NB classifier yields better performance to AUC by improving the AUC of all other classifiers (k-NN, DT, and RF), even the boosting classifiers (AB & XB). The reason can claim that the PCA and ICA return the feature vector with mutually exclusive and uncorrelated features. For which, NB performs better than others. However, for correlation-based feature selection, the XB outperforms other classifiers, even the NB classifiers for the preprocessing of P. Since features from the correlation-based selection are correlated with the outcome and are no more uncorrelated with each other as in PCA and ICA-based feature selection. For which, NB fails to be a winner in this experiment.

When the missing values are filled (Q) with the mean rather than rejection along with outlier rejection (P), the classification performance has boosted significantly. The XB has won for all the cases of feature selection when both the P and Q are employed. For P+Q and PCA or ICA, the XB outperforms the NB, where the NB was the best classifier for the process, P and PCA or ICA. The preprocess ($P + Q$) has more samples comparing the preprocess, P alone, as the samples were rejected when it was an outlier or missed in P alone. For the preprocess ($P + Q$) and correlation-based feature selection, all the classifiers show their tremendous success, as there are no missed values and outliers, where the RF and XB outperform the state-of-the-art by a $0.9\,\%$ and $1.6\,\%$ margin in AUC respectively.

Further addition of standardization as a preprocessing could not increase the performance of the classifiers as it is not always guaranteed to improve the performance. Tree-based models are not distance-based models, and hence standardization could not improve the performance of most of the ML models in this literature (see Table 3). Moreover, the standardization of the smaller dataset with fewer instances used in this literature can increase the possibility of losing information regarding the mean and standard deviation since the variability is less.

Remarkably, the employing of correlation-based feature selection rather than employing PCA and ICA-based techniques improves the AUC of all ML models when we apply the processing P and Q. The PCA transformed the higher dimensional space into a lower-dimensional space based on the orthogonal projections that contain the highest variance. The higher variance between the features will have lower covariance, whereas the uncorrelated data is only partially independent according to the ICA theory. The performance of the PCA algorithm depends on the number of PCs are being used, where the separation of the classes is more pronounced in the direction of smaller variance. Since the ICA finds the new predetermined mutually independent components, there is a possibility of losing correlation with the target outcome. Both the PCA and ICA find the new components in an unsupervised technique. For which, there is no guarantee of getting better performance in the PID dataset using PCA or ICA. On the other hand, the correlation-based feature selection uses the correlation between the feature and target outcome to select the features.

From the Table 4, it is also noticed that most of the classifiers performed better with 6 attributes comparing 4 or 8 attributes which are F1, F2, F4, F5, F6, and F8. This experiment also shows that the features such as diastolic blood pressure and diabetes pedigree function can be discarded from the PID dataset for diabetes prediction, as they carry less information of diabetes comparing other features, as in the PID dataset. Comparing all the ML models in Table 3 and Table 4, the XB provides the best performance with AUC ($\pm$ std.) of 0.946 ± 0.020, as it has extreme gradient boosting capability to minimize the loss when adding new models in parallel. The best performance of the diabetes prediction from the proposed pipeline using the XB model is achieved

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2989857, IEEE Access

TABLE 3: The summary of all extensive experiments for the selection of the best performing preprocessing, feature selection methods with selected attribute numbers, and classifier. The last column represents the best performing classifier for any preprocessing, whereas the underlined blue color denotes the best preprocessing for each classifier.

Preprocessing	Algorithm	N	k-NN	DT	RF	AB	NB	XB	Winner
Raw Data	N/A	8	0.813 ± 0.034	0.790 ± 0.052	0.826 ± 0.035	0.831 ± 0.026	0.816 ± 0.027	0.828 ± 0.030	AB
P	N/A	8	0.811 ± 0.046	0.772 ± 0.056	0.827 ± 0.039	0.827 ± 0.043	0.815 ± 0.030	0.834 ± 0.039	XB
	PCA	4	0.802 ± 0.027	0.760 ± 0.041	0.796 ± 0.056	0.794 ± 0.050	0.803 ± 0.040	0.795 ± 0.061	NB
		6	0.816 ± 0.004	0.784 ± 0.059	0.803 ± 0.037	0.810 ± 0.040	0.818 ± 0.036	0.812 ± 0.048	NB
	ICA	4	0.801 ± 0.055	0.751 ± 0.043	0.776 ± 0.035	0.780 ± 0.040	0.802 ± 0.039	0.790 ± 0.044	NB
		6	0.808 ± 0.037	0.754 ± 0.038	0.801 ± 0.037	0.809 ± 0.053	0.815 ± 0.038	0.811 ± 0.046	NB
	Corr	4	0.785 ± 0.044	0.765 ± 0.060	0.763 ± 0.045	0.803 ± 0.034	0.801 ± 0.029	0.807 ± 0.038	XB
		6	0.816 ± 0.039	0.805 ± 0.046	0.810 ± 0.041	0.837 ± 0.041	0.824 ± 0.037	0.838 ± 0.044	XB
P+Q	N/A	8	0.926 ± 0.022	0.899 ± 0.030	0.934 ± 0.014	0.938 ± 0.016	0.869 ± 0.022	0.943 ± 0.022	XB
	PCA	4	0.912 ± 0.024	0.880 ± 0.021	0.915 ± 0.023	0.905 ± 0.022	0.867 ± 0.024	0.915 ± 0.019	XB
		6	0.913 ± 0.023	0.871 ± 0.019	0.918 ± 0.016	0.912 ± 0.017	0.869 ± 0.030	0.919 ± 0.015	XB
	ICA	4	0.923 ± 0.015	0.912 ± 0.019	0.927 ± 0.009	0.941 ± 0.014	0.874 ± 0.02	0.943 ± 0.013	XB
		6	0.886 ± 0.023	0.883 ± 0.030	0.897 ± 0.025	0.891 ± 0.021	0.871 ± 0.038	0.904 ± 0.020	XB
	Corr	4	0.923 ± 0.015	0.912 ± 0.019	0.927 ± 0.009	0.941 ± 0.014	0.874 ± 0.020	0.943 ± 0.013	XB
		6	0.926 ± 0.022	0.911 ± 0.007	0.939 ± 0.019	0.940 ± 0.018	0.879 ± 0.025	0.946 ± 0.020	XB
P+Q+R	N/A	8	0.912 ± 0.018	0.899 ± 0.030	0.935 ± 0.015	0.938 ± 0.016	0.876 ± 0.024	0.943 ± 0.022	XB
	PCA	4	0.889 ± 0.039	0.880 ± 0.021	0.915 ± 0.023	0.905 ± 0.022	0.861 ± 0.032	0.915 ± 0.019	XB
		6	0.904 ± 0.020	0.871 ± 0.019	0.918 ± 0.016	0.912 ± 0.017	0.872 ± 0.028	0.919 ± 0.017	XB
	ICA	4	0.891 ± 0.040	0.852 ± 0.058	0.905 ± 0.020	0.885 ± 0.031	0.857 ± 0.034	0.904 ± 0.028	RF
		6	0.886 ± 0.023	0.883 ± 0.030	0.897 ± 0.025	0.891 ± 0.021	0.871 ± 0.038	0.904 ± 0.020	XB
	Corr	4	0.918 ± 0.013	0.912 ± 0.019	0.927 ± 0.008	0.941 ± 0.014	0.875 ± 0.018	0.943 ± 0.013	XB
		6	0.922 ± 0.021	0.911 ± 0.007	0.938 ± 0.017	0.940 ± 0.018	0.877 ± 0.025	0.946 ± 0.020	XB

Note: P: Outlier Rejection, Q: Filling Missing Value, R: Standardization, N: Number of Attributes, and Corr: Correlation-based Feature Selection.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2989857, IEEE Access

TABLE 4: The best performing ML model and preprocessing along with tuned hyperparameters with highest possible AUC.

ML Models	Best preprocessing	Best hyperparameters	Performance
k-NN	P+Q Correlation (n_Attributes = 6)	n_neighbors = 27 leaf_size = 30 algorithm = brute L_1-norm (manhattan_distance)	0.926 ± 0.022
DT	P+Q Correlation (n_Attributes = 4)	criterion = gini min_samples_split = 0.1 min_samples_leaf = 1 splitter = best	0.912 ± 0.019
RF	P+Q Correlation (n_Attributes = 6)	criterion = gini n_estimator =100	0.939 ± 0.019
AB	P+Q Correlation (n_Attributes = 4)	algorithm = SAMME.R n_estimator =200 learning_rate= 0.1	0.941 ± 0.014
NB	P+Q Correlation (n_Attributes = 6)	var_smoothing = 0.01	0.879 ± 0.025
XB	P+Q Correlation (n_Attributes = 6)	min_child_weight = 5 gamma = 1.5 subsample= 1.0 colssample_bytree = 0.6 max_depth = 5	$\mathbf{0.946 \pm 0.020}$

when the sum of instance weight in a leaf node less than 5 with the tree depth 5. The minimum loss reduction to make a further partition on a leaf node of the tree and the subsample ratio of to construct the tree were 1.5 and 0.6 respectively to obtain the highest possible results using the XB model from the proposed pipeline.

C. RESULTS FOR MLP

The extensive experiments were conducted on the PID dataset for diabetes prediction to obtain the best MLP architecture. Eight different models of MLP, with $1 \sim 8$ hidden layers, were implemented and tested, where the number of neurons was the hyperparameter to select optimum numbers. The experimental results are shown in Fig. 7, where it shows that the MLP architecture of $M = 3$ hidden layers (H_1, H_2, and H_3) with $N_1 = 16$, $N_2 = 64$, and $N_3 = 64$ neurons was chosen as the best architecture. The addition of

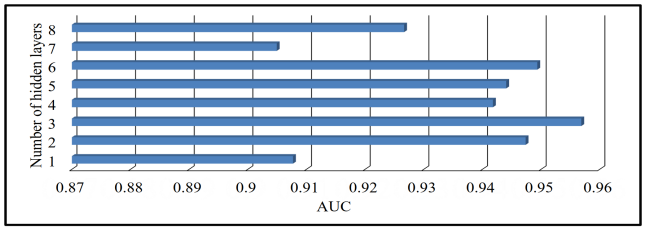


FIGURE 7: The performance of different MLP architectures to select the best one with the highest AUC, where the best corresponding models are shown in Table 5.

more hidden layers with fewer samples as in the PID dataset will tend to limit the generalizing capability of the MLP model, as depicted in Fig. 7. The extensive depth in the MLP model may also lead the model to be overfitted and often has gradient fading problems due to the limited numbers of data, as in the PID dataset.

The results on the best MLP architecture for different preprocessing are shown in Table 6, where all the neurons were initialized and activated by a normal distribution and ReLU function [38] respectively. We use the dropout layer [39] by randomly dropping $60\,\%$ neurons to tackle the overfitting. We trained our MLP model on 200 epochs with respective learning rate and batch sizes as 0.001 and 8. The results in Table 6 demonstrate that the outliers rejection and filling missing values drive the performance of the MLP model by a $7.1\,\%$ margin in AUC from raw data. Only the preprocess (P) can not improve the performance due to fewer samples, as both outliers and missing values are rejected in the process, P. The highest AUC from the MLP model is 0.902 with a standard deviation of 0.020 when we perform both the outliers rejection and filling missing values (P+Q). It is also demonstrated that the correlation-based feature selection is better in the PID dataset for diabetes prediction as similar to previous experiments on ML models (see subsection III-B). The ICA also performed as same as the correlation-based feature selection, the standard deviation for later one is much less than the former. For which later one has less inter-fold variation. Further addition of standardization with outliers rejection and filling missing values can not improve the results, as there is a possibility of losing information regarding the mean and standard deviation due to the less variability in the PID dataset.

D. RESULTS FOR ENSEMBLING MODEL

Since the ML models are ensembled for boosting the performance of the diabetes prediction, the best preprocessing

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2989857, IEEE Access

TABLE 5: The different MLP architectures with the corresponding number of hidden layers and the number of neurons.

Different Architectures	Number of hidden layers with corresponding neurons
Architecture-1	$H_1 \in \mathcal{R}^{32}$
Architecture-2	$H_1 \in \mathcal{R}^{64}$, $H_2 \in \mathcal{R}^{16}$
Architecture-3	$H_1 \in \mathcal{R}^{16}$, $H_2 \in \mathcal{R}^{64}$, $H_3 \in \mathcal{R}^{64}$
Architecture-4	$H_1 \in \mathcal{R}^{32}$, $H_2 \in \mathcal{R}^{32}$, $H_3 \in \mathcal{R}^{16}$, $H_4 \in \mathcal{R}^{16}$
Architecture-5	$H_1 \in \mathcal{R}^{16}$, $H_2 \in \mathcal{R}^{16}$, $H_3 \in \mathcal{R}^{16}$, $H_4 \in \mathcal{R}^{16}$, $H_5 \in \mathcal{R}^{32}$
Architecture-6	$H_1 \in \mathcal{R}^{32}$, $H_2 \in \mathcal{R}^{32}$, $H_3 \in \mathcal{R}^{64}$, $H_4 \in \mathcal{R}^{32}$, $H_5 \in \mathcal{R}^{16}$, $H_6 \in \mathcal{R}^{64}$
Architecture-7	$H_1 \in \mathcal{R}^{16}$, $H_2 \in \mathcal{R}^{16}$, $H_3 \in \mathcal{R}^{16}$, $H_4 \in \mathcal{R}^{16}$, $H_5 \in \mathcal{R}^{64}$, $H_6 \in \mathcal{R}^{32}$, $H_7 \in \mathcal{R}^{32}$
Architecture-8	$H_1 \in \mathcal{R}^{64}$, $H_2 \in \mathcal{R}^{32}$, $H_3 \in \mathcal{R}^{64}$, $H_4 \in \mathcal{R}^{64}$, $H_5 \in \mathcal{R}^{32}$, $H_6 \in \mathcal{R}^{16}$, $H_7 \in \mathcal{R}^{32}$, $H_8 \in \mathcal{R}^{16}$

TABLE 6: The summary of all extensive experiments on the MLP model, where all the hyperparameters from the grid search were kept constant throughout the experiment.

Raw Data	P							P+Q							P+Q+R						
	N/A	PCA		ICA		Corr		N/A	PCA		ICA		Corr		N/A	PCA		ICA		Corr	
8	8	4	6	4	6	4	6	8	4	6	4	6	4	6	8	4	6	4	6	4	6
0.821 ± 0.040	0.796 ± 0.032	0.738 ± 0.029	0.770 ± 0.044	0.787 ± 0.045	0.829 ± 0.045	0.793 ± 0.051	0.818 ± 0.045	0.892 ± 0.019	0.846 ± 0.029	0.874 ± 0.025	0.901 ± 0.037	0.887 ± 0.039	0.902 ± 0.020	0.890 ± 0.013	0.884 ± 0.024	0.890 ± 0.032	0.881 ± 0.019	0.889 ± 0.019	0.885 ± 0.031	0.867 ± 0.016	0.884 ± 0.015

Note: P: Outlier Rejection, Q: Filling Missing Value, R: Standardization, and Corr: Correlation-based Feature Selection.

from the subsection III-B and Table 3 & Table 4 are used in this experiment. The combination of the above ML models ($N = 6$) provides $\sum_{i=1}^{N} {}^{N}C_i = 63$ ensemble models. Among them only the best performing ensemble model with 2, 3, 4, 5, and 6 baseline models are reported in Table 7 with their corresponding results. The combination of AB and XB provides the best results for diabetes prediction for the three metrics out of the five, as shown in Table 7, by beating the other combinations by the $1.20\,\%$, $14.81\,\%$, and $0.90\,\%$ margin in Sp, DOR, and AUC respectively. The prevalence independent measure (DOR) of the AB+XB (see Table 7) has a greater value than the other combinations, which is considered to be a very good test [40] for the diabetes prediction. The confusion matrix and ROC curve of the best ensemble model (AB+XB) are shown in Fig. 8 (a) and Fig. 8 (b) respectively. The fraction of correctly classified patients among all the positive predictions is $84.2\,\%$ using the combination of AB and XB. From the ROC curve (see Fig. 8 (b)), it is seen that for false-positive rate of 0.066, the probability of getting true-positive rate is 0.788 at the model's accuracy (see the red star point in Fig. 8 (b)). From the ROC curve, it is also observed that the inter-fold variation of the AUC is also less which proves the robustness of the best ensembling classifier (AB+XB). The performance of AB+XB for diabetes prediction on the PID dataset is the superior, as both the AB and XB are the boosting type classifiers, where AB is the sequential boosting and XB is the parallel boosting. The combination of other ML models with the boosting type models (AB & XB) can not predict diabetes as good as the boosting types alone, as shown in Table 7 ($2 \sim 5^{th}$ rows). Although the combination of all the 6 models (see Table 7 (5^{th} row)) beats the best combination (AB+XB) in two metrics out of five, it has defeated in unbiased measurement (AUC) by a margin of $1.0\,\%$. As a consequence, we can claim that for the diabetes prediction from the PID dataset, the soft weighted voting of serial and parallel boosting classifiers performs better than serial or parallel boosting classifier alone.

E. RESULTS COMPARISON

In this subsection, all the three experiments (see subsection III-B, III-C, and III-D) are compared and summarized. Finally, the best experiment is compared with the state-of-the-art to validate our contributions in this literature.

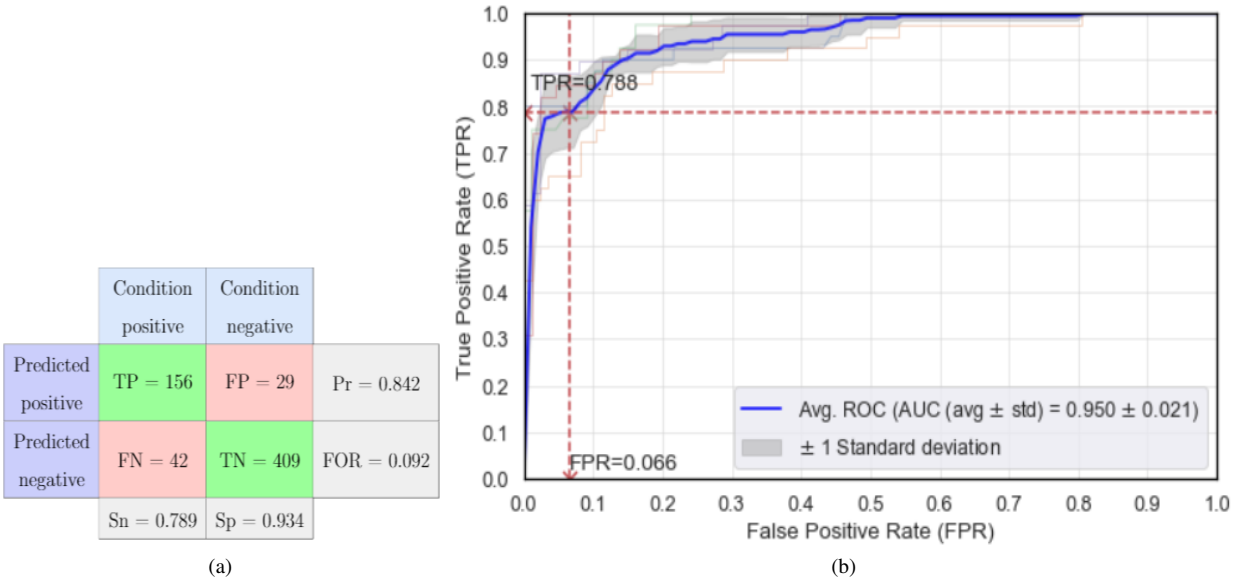
Table 8 demonstrates that the proposed weighted-ensemble of AB and XB produces the best prediction for the three metrics out of the five metrics, whereas performs as a second highest with respect to Sp and prevalence independent measurement (DOR). The proposed ensemble model (AB+XB) yields the best performance concerning Sn, FOR, and AUC by improving the XB by the margin of $2.1\,\%$, $0.8\,\%$, and $0.6\,\%$ respectively. It also beats MLP model in Sn, Sp, FOR, and AUC respectively by the margin of $3.2\,\%$, $3.4\,\%$, $1.5\,\%$, and $4.8\,\%$. The ensembling model (AB+XB) improves the true-positive rate compare to the XB model alone, as there is less possibility of miss-classification in the ensembling model. The less FOR values in the ensembling model (see Table 8) demonstrates that negative predictive value is high with less Type II error in the diabetes prediction. Furthermore, it is also observed that the proposed ensembling model (AB+XB) yields the best performances for balanced accuracy (average of Sn and Sp) by improving the XB and MLP results by $0.6\,\%$ and $3.3\,\%$ respectively, when the proposed preprocessing (P+Q and correlation-based feature selection) is employed. As a consequence from the above discussions in subsections III-B, III-C, III-D, and III-E, it can be concluded as follows:

The proposed ensembling classifier (AB+XB) appears better suited for diabetes prediction from the PID dataset. For

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2989857, IEEE Access

TABLE 7: Comparing different ensembling models for selecting the best classifier.

Ensemble Models	Sn	Sp	FOR	DOR	AUC
AB+XB	0.789 ± 0.077	0.934 ± 0.012	0.092 ± 0.032	66.234 ± 33.323	0.950 ± 0.021
k-NN+DT+XB	0.793 ± 0.064	0.920 ± 0.019	0.092 ± 0.026	53.614 ± 26.766	0.941 ± 0.015
DT+AB+RF+XB	0.793 ± 0.057	0.922 ± 0.015	0.091 ± 0.024	50.367 ± 13.421	0.943 ± 0.013
k-NN+DT+RF+XB+NB	0.808 ± 0.047	0.920 ± 0.013	0.086 ± 0.020	54.135 ± 20.053	0.939 ± 0.016
k-NN+DT+RF+AB+NB+XB	0.813 ± 0.052	0.920 ± 0.013	0.084 ± 0.022	57.688 ± 24.538	0.940 ± 0.016



(a) (b)

FIGURE 8: (a) Confusion matrix of the highest possible diabetes classification (b) ROC curve of our proposed ensemble model.

TABLE 8: Comparing all the implemented models for diabetes prediction.

Ensemble Models	Sn	Sp	FOR	DOR	AUC
XB	0.768 ± 0.072	0.943 ± 0.016	0.100 ± 0.030	71.369 ± 41.245	0.946 ± 0.020
MLP	0.757 ± 0.059	0.900 ± 0.045	0.107 ± 0.022	32.748 ± 7.570	0.902 ± 0.020
AB+XB	0.789 ± 0.077	0.934 ± 0.012	0.092 ± 0.032	66.234 ± 33.323	0.950 ± 0.021

ensembling, the base classifiers should have a minimum correlation between them to achieve higher precision in diabetes prediction (see Table 7). The ensembling of two boosting (adaptive (AB) and gradient (XB)) type classifier is the best combination for diabetes prediction. The best combination (AB+XB), along with our proposed preprocessing (P+Q and correlation-based feature selection), can achieve tremendous success for diabetes prediction in the PID dataset.

From Table 9, it is observed that all the models perform better either in positive or negative diabetes prediction, whereas the proposed model beats them with improved balanced accuracy or AUC or both. The framework proposed in [43], [44] used the k-NN technique to impute the missing values, where the algorithm searches the k^{th} neighbor as a missing value. In such a technique, the new imputed value could be far from the central tendency of the population distribution. The performance in the pipeline (see Table 9) employed in [18], [20], [41], [42], [46] is less as comparing the proposed framework and others in [7], [44], [45]. Those fewer performances clearly indicate the role of outlier rejection and filling missing values in the PID dataset. The manual feature selection [42] without considering the correlation and covariance with the features and target label is the possible reason for getting less true-positive rates. The above discussion and Table 9 confirm that our proposed ensembling classifier (AB+XB) for predicting diabetes is a better diagnosis, with an AUC of 0.950, when the AUC-weighted soft voting and proposed preprocessing pipeline were employed compared to others.

IV. CONCLUSION AND FUTURE WORK

In this literature, diabetes prediction has been accomplished using the proposed ensemble model from the PID dataset, where the preprocessing plays a crucial role in robust and precise prediction. The quality of the dataset was improved by the proposed preprocessing scheme, where outlier rejection and filling missing values was a core concern. Such a preprocessing can improve the kurtosis and skewness of the attribute distribution in the PID dataset. The correlation-based attribute selection can improve the correlation between

TABLE 9: Comparative performance of our proposed method against the state-of-the-art works on the same dataset as shown in Table 1.

SL #	Authors and Year	MVIT	ORT	FRT	NSF	Classifier	Performance
01	Li (2014) [41]	NA	NA	NA	8	Ensembling of SVM, ANN, & NB	AUC : — Sn : 0.583 Sp : 0.868
02	M. Pradhan et al. (2015) [20]	NA	NA	NA	8	GPA	AUC : — Sn : 0.880 Sp : 0.900
03	A. K. Dewangan et al. (2015) [42]	NA	NA	Manual	6	Ensembling of MLP & NB	AUC : — Sn : 0.641 Sp : 0.909
04	S. Bashir et al. (2016) [43]	k-NN impute	ESD	NA	8	HM-BagMoov	AUC : — Sn : 0.787 Sp : 0.926
05	M. Maniruzzaman et al. (2017) [22]	Median	NA	NA	8	GPC	AUC : — Sn : 0.918 Sp : 0.633
06	H. Kaur et al. (2018) [44]	k-NN impute	NA	BWA	4	k-NN	AUC : 0.920 Sn : — Sp : —
07	D. Sisodia et al. (2018) [18]	NA	NA	NA	8	Naive Bayes	AUC : 0.819 Sn : 0.763 Sp : —
08	M. Maniruzzaman et al. (2018) [7]	Group median	Median	RF	4	RF	AUC : 0.930 Sn : 0.960 Sp : 0.797
09	Q. Wang et al. (2019) [45]	NB method	-	-	8	RF	AUC : 0.928 Sn : 0.854 Sp : —
10	S. P. Chatrati et al. 2020 [46]	NA		NA	8	DA	AUC : 0.700 Sn : 0.720 Sp : 760
11	**Our Proposed (2020)**	Attribute's Mean	IQR	Correlation	6	Ensembling of AB & XB	AUC : 0.950 Sn : 0.789 Sp : 0.934

Note: MVIT: Missing Value Imputing Technique, ORT: Outlier Rejection Technique, FRT: Feature Reduction Technique, NSF: Number of Selected Feature, GPA: Genetic Programming Algorithm, IQR: Interquartile range, BWA: Boruta Wrapper Algorithm, ESD: Extreme Studentized Deviate, DA: Discriminant Analysis, and GPC: Gaussian Process Classification.

attribute and target outcome, whereas PCA and ICA care only the inter-attribute redundancy. In case of tree-based classifier, data standardization can not provide any guarantee to improve the performance. The robustness validation of the XB, MLP, and proposed ensemble classifier was verified by using the 5-fold cross-validation. Hyperparameters of different classifiers can drive the learning capability of those classifiers, which were optimized using a grid search technique in our proposed framework. The AUC as a weight to build a generic ensembling classifier is better, as it considers more priority to the model having more AUC. Random tree-based classifiers are well suited for the data to be classified when inter-class redundancy is much higher (not linearly separable), as in the PID dataset. The comparative results demonstrate that our proposed framework has outperformed other frameworks on AUC, which has shown great potentiality for diabetes prediction from the PID dataset. The ensembling of two boosting type classifiers (AB and XB) is the best combination for diabetes prediction, as the base classifiers should have a minimum correlation between them. The higher precision in diabetes prediction from the PID dataset using the best combination (AB+XB) can be achieved when our proposed preprocessing (P+Q and correlation-based feature selection) is applied. In the future, the proposed trained model will be used to build a web app with a user-friendly interface. Additionally, the proposed framework will be applied to other medical contexts to verify their generality and versatility to predict the disease classes.

CONFLICTS OF INTEREST

Authors haven't any conflicts to disclose this research.

REFERENCES

[1] A. Misra, H. Gopalan, R. Jayawardena, A. P. Hills, M. Soares, A. A. Reza-Albarrán, and K. L. Ramaiya, "Diabetes in developing countries," Journal of Diabetes, vol. 11, no. 7, pp. 522-539, Mar. 2019.

[2] R. Vaishali, R. Sasikala, S. Ramasubbareddy, S. Remya, and S. Nalluri, "Genetic algorithm based feature selection and MOE Fuzzy classification algorithm on Pima Indians Diabetes dataset," in Proc. International Conference on Computing Networking and Informatics, Oct. 2017, pp. 1-5.

[3] Emerging Risk Factors Collaboration and other, "Diabetes mellitus, fasting blood glucose concentration, and risk of vascular disease: a collaborative meta-analysis of 102 prospective studies," The Lancet, vol. 375, no. 9733, pp. 2215-2222, Jul. 2010.

[4] N. H. Choac, J. E. Shaw, S. Karuranga, Y. Huang, J. D. R. Fernandes, A. W. Ohlrogge, and B. Malandaa, "IDF Diabetes Atlas: Global estimates of diabetes prevalence for 2017 and projections for 2045," Diabetes Research and Clinical Practice, vol. 138, pp. 271-281, Apr. 2018.

[5] P. Saeedi, I. Petersohn, P. Salpea, B. Malanda, S. Karurangaa, N. Unwin, S. Colagiuri, L. Guariguata, A. A. Motala, K. Ogurtsova, J. E. Shaw, D. Bright, R. Williams, and IDF Diabetes Atlas Committee, "Global and regional diabetes prevalence estimates for 2019 and projections for 2030 and 2045: Results from the International Diabetes Federation," Diabetes Research and Clinical Practice, vol. 157, pp. 107843, Nov. 2019.

[6] J. W. Smith, J. E. Everhart, W. C. Dickson, W. C. Knowler, and R. S. Johannes, "Using the ADAP learning algorithm to forecast the onset of diabetes mellitus," in Proc. Annual Symposium on Computer Application in Medical Care, Nov. 1988, pp. 261-265.

[7] M. Maniruzzaman, M. J. Rahman, M. A. M. Hasan, H. S. Suri, M. M. Abedin, A. El-Baz, and J. S. Suri, "Accurate diabetes risk stratification using machine learning: role of missing value and outliers," Journal of Medical Systems, vol. 42, no. 5, pp. 92, May 2018.

[8] G. J. McLachlan, "Discriminant analysis and statistical pattern recognition," Journal of the Royal Statistical Society: Series A (Statistics in Society), vol. 168, no. 3, pp. 635-636, Jun. 2005.

[9] T. M. Cover, "Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition," IEEE Transactions on Electronic Computers, vol. 14, no. 3, pp. 326-334, Jun. 1965.

[10] G. I. Webb, J. R. Boughton, and Zhihai Wang, "Not So Naive Bayes: Aggregating one-dependence estimators," Machine learning, vol. 58, no. 1, pp. 5-24, Jan. 2005.

[11] S. B. Belhouari and A. Bermak, "Gaussian process for nonstationary time series prediction," Computational Statistics & Data Analysis, vol. 47, no. 4, pp. 705-712, Feb. 2004.

[12] C. Cortes and V. Vapnik , "Support-vector networks," Machine Learning, vol. 20, pp. 237-297, Sep. 1995.

[13] A. Reinhardt and T. Hubbard, "Using neural networks for prediction of the subcellular location of proteins," Nucleic Acids Research, vol. 26, no. 9, pp. 2230-2236, Mar. 1998.

[14] B. Kégl, "The return of AdaBoost. MH: Multi-class Hamming trees," arXiv:1312.6086, Dec. 2013.

[15] T. BP and H. WH, "A multivariate logistic regression equation to screen for diabetes: development and validation," Diabetes Care, vol. 25, no. 11, pp. 1999-2003, Nov. 2002.

[16] I. Jenhani, N. B. Amor, and Z. Elouedi, "Decision trees as possibilistic classifiers," International Journal of Approximate Reasoning, vol. 48, no. 3, pp. 784-807, Aug. 2008.

[17] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5-32, Oct. 2001.

[18] D. Sisodia and D. S. Sisodia, "Prediction of diabetes using classification algorithms," Procedia Computer Science, vol. 132, pp. 1578-1585, Jan. 2018.

[19] S. Perveen, M. Shahbaz, A. Guergachi, and K. Keshavjee, "Performance analysis of data mining classification techniques to predict diabetes," Procedia Computer Science, vol. 82, pp. 115-121, Mar. 2016.

[20] M. Pradhan and G. R. Bamnote, "Design of Classifier for Detection of Diabetes Mellitus Using Genetic Programming," in Proc. third International Conference on Frontiers of Intelligent Computing: Theory and Applications, Nov. 2015, pp. 763-770.

[21] N. Nai-arun and R. Moungmai, "Comparison of classifiers for the risk of diabetes prediction," Procedia Computer Science, vol. 69, pp. 132-142, Dec. 2015.

[22] M. Maniruzzaman, N. Kumar, M. M. Abedin, M. S. Islam, H. S. Suri, A. S. El-Baz, and J. S. Suri, "Comparative approaches for classification of diabetes mellitus data: Machine learning paradigm," Computer Methods and Programs in Biomedicine, vol. 152, pp. 23-34, Dec. 2017.

[23] R. Bansal, N. Gaur, and S. N. Singh, "Outlier Detection: Applications and techniques in data mining," in Proc. sixth International Conference-Cloud System and Big Data Engineering, Jan. 2016, pp. 373-377.

[24] D. Cousineau and S. Chartier, "Outliers detection and treatment: A review," International Journal of Psychological Research, vol. 3, no. 1, pp. 58-67, Mar. 2010.

[25] C. R. Rao, "The use and interpretation of principal component analysis in applied research," Sankhyā: The Indian Journal of Statistics, Series A, pp. 329-358, Dec. 1964.

[26] A. Hyvärinen and E. Oja, "Independent Component Analysis: Algorithms and applications," Neural Networks, vol. 13, no. 4-5, pp. 411-430, Jun. 2000.

[27] F. Han and H. Liu, "Statistical analysis of latent generalized correlation matrix estimation in transelliptical distribution," Bernoulli: official journal of the Bernoulli Society for Mathematical Statistics and Probability, vol. 23, no. 1, pp. 23-57, Feb. 2017.

[28] S. Arlot and A. Celisse, "A survey of cross-validation procedures for model selection," Statistics surveys, vol. 4, pp. 40-79, Jul. 2010.

[29] D. Krstajic, L. Buturovic, D. E. Leahy, and S. Thomas, "Cross-validation pitfalls when selecting and assessing regression and classification models," Journal of Cheminformatics, vol. 6, no. 1, pp. 10, Mar. 2014.

[30] X. Zeng and T. R. Martinez, "Distribution-balanced stratified cross-validation for accuracy estimation," Journal of Experimental & Theoretical Artificial Intelligence, vol. 12, no. 1, pp. 1-12, Nov. 2000.

[31] P. Cunningham and S. J. Delany, "k-Nearest neighbour classifiers," Multiple Classifier Systems, vol. 34, no. 8, pp. 1-17, Mar. 2007.

[32] T. Chen and C. Guestrin, "XGboost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Aug. 2016, pp. 785-794.

[33] S. Hsieh, S. Hsieh, P. Cheng, C. Chen, K. Hsu, I. Wang, and F. Lai, "Design ensemble machine learning model for breast cancer diagnosis," Journal of Medical Systems, vol. 36, no. 2012, pp. 2841-2847, Jul. 2011.

[34] B. Harangi, "Skin lesion classification with ensembles of deep convolutional neural networks," Journal of Biomedical Informatics, vol. 86, pp. 25-32, Oct. 2018.

[35] A. S. Miller, B. H. Blott, and others, "Review of neural network applications in medical imaging and signal processing," Medical and Biological Engineering and Computing, vol. 30, no. 5, pp. 449-464, Jan. 1992.

[36] D. E. Rumelhart, G. E. Hinton, and J. Ronald, "Review of neural network applications in medical imaging and signal processing," Nature, vol. 323, no. 6088, pp. 533-536, Oct. 1986.

[37] A. S. Glas, J. G. Lijmer, M. H. Prins, G. J. Bonsel, and P. M. M. Bossuyt, "The Diagnostic Odds Ratio: A single indicator of test performance," Journal of Clinical Epidemiology, vol. 56, no. 11, pp. 1129-1135, Nov. 2003.

[38] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," arXiv:1710.05941, Oct. 2017.

[39] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," The journal of machine learning research, vol. 15, no. 1, pp. 1929-1958, Jan. 2014.

[40] J. J. Deeks, "Systematic reviews of evaluations of diagnostic and screening tests," BMJ, vol. 323, no. 7305, pp. 157-162, Jul. 2001.

[41] L. Li, "Diagnosis of Diabetes Using a Weight-Adjusted Voting Approach," in Proc. IEEE International Conference on Bioinformatics and Bioengineering, Nov. 2014, pp. 320-324.

[42] A. K. Dewangan and P. Agrawal, "Classification of diabetes mellitus using machine learning techniques," International Journal of Engineering and Applied Sciences, vol. 2, no. 5, pp. 145-148, May 2015.

[43] S. Bashir, U. Qamar, and F. H. Khan, "IntelliHealth: A medical decision support application using a novel weighted multi-layer classifier ensemble framework," Journal of Biomedical Informatics, vol. 59, pp. 185-200, Feb. 2016.

[44] H. Kaur and V. Kumari, "Predictive modelling and analytics for diabetes using a machine learning approach," Applied Computing and Informatics, Dec. 2018.

[45] Q. Wang, W. Cao, W. Guo, J. Ren, Y. Cheng, and D. N. Davis, "DMP_MI: An effective diabetes mellitus classification algorithm on imbalanced data with missing values," IEEE Access, vol. 7, pp. 102232-102238, Jul. 2019.

[46] S. P. Chatrati, G. Hossain, A. Goyal, A. Bhan, S. Bhattacharya, D. Gaurav, and S. M. Tiwari, "Smart Home Health Monitoring System for Predicting Type 2 Diabetes and Hypertension," Journal of King Saud University-computer and Information Sciences, Jan. 2020.

[47] L. Jiang, Z. Cai, D. Wang, and S. Jiang, "Survey of Improving K-Nearest-Neighbor for Classification," in Proc. Fourth International Conference on Fuzzy Systems and Knowledge Discovery, Aug. 2007, pp. 679-683.
[48] R. E. Schapire, "Explaining AdaBoost," in Empirical Inference, pp. 37-52, Oct. 2013
[49] S. Taheri and M. Mammadov, "Learning the naive Bayes classifier with optimization models," International Journal of Applied Mathematics and Computer Science, vol. 23, no. 4, pp. 787-795, Dec. 2013.

MD. KAMRUL HASAN was born in Tangail, Bangladesh in 1992. He received B. Sc. and M. Sc. engineering degree in Electrical and Electronic Engineering (EEE) from Khulna University of Engineering & Technology (KUET) in 2014 and 2017 respectively. He received another M. Sc. in Medical Imaging and Application (MAIA) from France (University of Burgundy), Italy (University of Cassino and Southern Lazio), and Spain (University of Girona) as an Erasmus scholar in 2019. He is a University gold medalist due to securing 1^{st} position in his class at KUET. Currently, he is working as an Assistant Professor at KUET in the EEE department.

During the studding of MAIA, he focused on different modalities of medical image analysis and machine learning to build a generic computer-aided diagnosis system. He has published several international journal and conference papers on medical image and signal processing. His research interest includes medical image and data analysis, machine learning, deep convolutional neural network, medical image reconstruction, and surgical robotics. Currently, he is working with several undergraduate students as a supervisor on different modalities of medical image classification, segmentation, and registration.

MD. ASHRAFUL ALAM is studying in Electrical and Electronic Engineering (EEE) at Khulna University of Engineering & Technology (KUET). Currently, he is working on medical data analysis, skin cancer classification, and multi-label whole heart segmentation from CT and MRI as a B. Sc. thesis. His interests include medical image and data processing, computer vision, and deep learning.

DOLA DAS was born in Khulna, Bangladesh in 1997. She received B. Sc. Engineering degree in Computer Science and Engineering (CSE) from Khulna University of Engineering & Technology (KUET) in 2019. Currently, she is working as a Lecturer in the CSE department of KUET as well as pursuing M. Sc. Engineering in CSE at KUET.

She has research interest in machine learning, deep neural networks, data mining, and biomedical engineering. She has published some conference papers in these domains. Currently, she is working on some papers about these topics with her students and colleagues.

EKLAS HOSSAIN (M'09, SM'17) received his Ph. D. from the College of Engineering and Applied Science at University of Wisconsin Milwaukee (UWM). He received his MS in Mechatronics and Robotics Engineering from International Islamic University of Malaysia, Malaysia in 2010 and BS in Electrical and Electronic Engineering from Khulna University of Engineering & Technology, Bangladesh in 2006. Dr. Hossain has been working in the area of distributed power systems and renewable energy integration for the last ten years and he has published several research papers and posters in this field. He is now involved with several research projects on renewable energy and grid-tied microgrid system at Oregon Tech, as an Assistant Professor in the Department of Electrical Engineering and Renewable Energy since 2015.

He is the senior member of the Association of Energy Engineers (AEE). He is currently serving as an Associate Editor of IEEE Access. He is working as an Associate Researcher at the Oregon Renewable Energy Center (OREC). He is a registered Professional Engineer (PE) in the state of Oregon, USA. He is also a Certified Energy Manager (CEM) and Renewable Energy Professional (REP). His research interests include modeling, analysis, design, and control of power electronic devices; energy storage systems; renewable energy sources; integration of distributed generation systems; microgrid and smart grid applications; robotics, and advanced control system. He is the winner of the Rising Faculty Scholar Award in 2019 from Oregon Institute of Technology for his outstanding contribution to teaching. Dr. Hossain, with his dedicated research team, is looking forward to exploring methods to make the electric power systems more sustainable, cost-effective and secure through extensive research and analysis on energy storage, microgrid system, and renewable energy sources.

MAHMUDUL HASAN born in February 1994 in Bangladesh. Mr. Hasan completed his B. Sc. degree in Computer Science and Engineering at Khulna University of Engineering & Technology (KUET), Khulna-9203, Bangladesh in 2018. Currently, Mr. Hasan is admitted for Ph. D. degree in computer science at Stony Brook University, Stony Brook, New York – 11790, USA.

He was a Lecturer at KUET from 2018-2020. Currently, he is a graduate teaching assistant at Stony Brook University, Stony Brook, New York – 11790, USA. His previous research interest was Applicable to Machine Learning and Deep Learning. His current research interest lies in Computer Vision. Mr. Hasan was an active IEEE student member from year 2017 to 2018. Mr. Hasan also worked as IEEE student branch president at the same time.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2989857, IEEE Access

APPENDIX. ALGORITHMS FOR THE FEATURE SELECTION AND ML CLASSIFIERS FOR DIABETES PREDICTION

A. PCA-BASED FEATURE SELECTION

Algorithm 1: The steps of implementing the PCA-based feature selection

Input: The original n-dimensional data, $X \in \mathcal{R}^n$ with N number of sample and variance threshold, $T_{variance}$
Output: The reduced k-dimensional data, $Y \in \mathcal{R}^k$

Load $X \in \mathcal{R}^n$ and compute it's mean, $\bar{X} = \frac{1}{N}\sum_{i=1}^{N} X_i$, where $\bar{X} \in \mathcal{R}^n$
Compute the $n \times n$ covariance matrix, $C_{n\times n} = \sum_{i=1}^{N}(X_i - \bar{X})(X_i - \bar{X})^T$
Compute eigen decomposition of $C_{n\times n}$ as PDP^{-1}, where $P \in \mathcal{R}^n$ is the matrix of eigen vectors and $D_{n\times n}$ is the diagonal matrix with eigenvalues on the diagonal
Sort the eigen vectors by descending order to choose first k eigen vectors that will have variance $\geq T_{variance}$ and form a new projection matrix, $W_{n\times k}$
Project data X into a new k-dimensional space by $Y = W^T X$, where $Y \in \mathcal{R}^k$

B. ICA-BASED FEATURE SELECTION

Algorithm 2: The steps of implementing the ICA-based feature selection

Input: The original n-dimensional data, $X \in \mathcal{R}^n$
Output: The reduced k-dimensional data, $Y \in \mathcal{R}^k$

Set non-quadratic nonlinear function, G for the approximation of neg-entropy
Initialize W of $W \times H = X$, where W, H, and X are the ratios of the sources during mixing, the matrix containing the different components, and the mixed output respectively.
Perform PCA on X by $X = PCA(X)$ as in A
while W *changes* **do**
    $W = mean(X * G(W \cdot X)) - mean(G'(W^T \cdot X))$, where G' is the first derivative of non-quadratic nonlinear function, G
    $W = orthogonalize(W)$
Compute, $Y = W \cdot X$, where $Y \in \mathcal{R}^k$

C. CORRELATION-BASED FEATURE SELECTION

Algorithm 3: The steps of implementing the correlation-based feature selection

Input: The original n-dimensional data, $X \in \mathcal{R}^n$ and expected outcome, $Y_T \in \mathcal{R}$
Output: The reduced k-dimensional data, $Y \in \mathcal{R}^k$

for $i \leq n$ **do**
    $r_{iT} = \frac{\sum(X_i - \bar{X_i})(Y_T - \bar{Y_T})}{\sqrt{\sum(X_i - \bar{X_i})^2} \times \sqrt{\sum(Y_T - \bar{Y_T})^2}}$
Sort the correlation, r_{iT} by descending order to choose first k features for $Y \in \mathcal{R}^k$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2989857, IEEE Access

D. ALGORITHM FOR IMPLEMENTING K-NEAREST NEIGHBOUR

Algorithm 4: The steps of implementing k-nearest Neighbour (k-NN)

Input: The n-dimensional data, $X \in \mathcal{R}^n$ and target outcome, $Y \in \mathcal{R}$

Output: The posterior probability, $P \in [0,\ 1]$ of unseen test data, x, where $\sum_{i=1}^{C} P_i = 1$ and $C = 2$ (diabetes present (C_1) or not (C_2))

1 Calculate geometric distances, D_h for k query points, $D_h = \sum_{i=1}^{k} |X_i - x_i|^{q}$ $^{\frac{1}{q}}$, where X_i = current instance, x_i = query instance, q = order [47].

2 Form a set, S with closest k points

3 Estimates the posterior probability, P for each class $P(C = j|X = x) = \frac{1}{K} \sum_{i \in S} f(C_i = j)$, where $f(x)$ is the indicator function to assign the class (1 when patient having diabetes and 0 otherwise)

E. ALGORITHMS FOR IMPLEMENTING DECISION TREE

Algorithm 5: The steps of implementing Decision Tree (DT)

Input: The n-dimensional data, $X \in \mathcal{R}^n$ and target outcome, $Y \in \mathcal{R}$

Output: The posterior probability, $P \in [0,\ 1]$ of unseen test data, x, where $\sum_{i=1}^{C} P_i = 1$ and $C = 2$ (diabetes present (C_1) or not (C_2))

1 Split $\theta = (j, t_m)$ into $Q_{left}(\theta)$ and $Q_{right}(\theta)$ subsets, where θ consisting of a feature, j and threshold, t_m

2 Compute the impurity at k^{th} node using an impurity function (H),
$G(Q,\ \theta) = \frac{n_{left}}{N_m} H(Q_{left}(\theta)) + \frac{n_{right}}{N_m} H(Q_{right}(\theta))$, where $H = \sum_{C} P_{mC} \times (1 - P_{mC})$ or
$H = -\sum_{C} P_{mC} \times log(p_{mC})$ and $P_{mC} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = C)$

3 Minimise the impurity by selecting the parameters, $\theta^* = argmin_\theta\, G(Q,\ \theta)$

4 Repeat the above processes for subsets $Q_{left}(\theta^*)$ and $Q_{right}(\theta^*)$ until depth reach to $N_m < min_{samples}$ or $N_m = 1$

F. ALGORITHMS FOR IMPLEMENTING ADABOOST

Algorithm 6: The steps of implementing AdaBoost (AB)

Input: The n-dimensional data, $X \in \mathcal{R}^n$ with N number of sample and target outcome, $Y \in \mathcal{R}$

Output: The posterior probability, $P \in [0,\ 1]$ of unseen test data, x, where $\sum_{i=1}^{C} P_i = 1$ and $C = 2$ (diabetes present (C_1) or not (C_2))

1 Initialize weight sample, $D(i) = \frac{1}{N}$, where $i = 1, 2, ..., N$

2 **for** $t \leq T(n\_Classifiers)$ **do**

3 Train a weak learner using distribution D_t [48].

4 Select a weak hypothesis, $h_t : \mathcal{R}^n \rightarrow \mathcal{R}$ with low weight error, $\epsilon_t = Pr_{i \sim D_t}[h_t(x_i) \neq Y]$

5 Choose $\alpha_t = \frac{1}{2} ln(\frac{1-\epsilon_t}{\epsilon_t})$ and update, $D_{t+1}(i) = \frac{D_t(i) e^{-\alpha_t Y h_t(x_i)}}{z_t}$, where $i = 1, ..., N$ and z_t is the normalization factor.

6 Output posterior probability: $P(x) = sign(\sum_{t=1}^{T} a_t h_t(x))$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2989857, IEEE Access

G. ALGORITHMS FOR IMPLEMENTING RANDOM FOREST

Algorithm 7: The steps of implementing Random Forest (RF)

Input: The n-dimensional data, $X \in \mathcal{R}^n$ and target outcome, $Y \in \mathcal{R}$

Output: The posterior probability, $P \in [0,\ 1]$ of unseen test data, x, where $\sum_{i=1}^{C} P_i = 1$ and $C = 2$ (diabetes present (C_1) or not (C_2))

for $b = 1\, to\, N\ (n\_Bagging)$ **do**
- Draw a bootstrap sample, (X_b, Y_b) from given $(X \in \mathcal{R}^n, Y \in \mathcal{R})$
- Grow a random-forest tree T_b using X_b and Y_b by repeating recursively using the following steps until the minimum node size is n_{min}.
  1) Randomly select m variables from the given n variables
  2) Pick the best variable or split-point among the m variables
  3) Split the node into two daughter nodes
- Output the ensemble of trees will be $\{T_b\}_1^N$

The posterior probability, $\hat{P}_{RF}^N(x) = Voting\{\hat{P}_k(x)\}_1^N$, where $\hat{P}_k(x)$ is the class prediction of the k_{th} random-forest.

H. ALGORITHMS FOR IMPLEMENTING NAIVE BAYES

Algorithm 8: The steps of implementing Naive Bayes (NB)

Input: The n-dimensional data, $X \in \mathcal{R}^n$ and target outcome, $Y \in \mathcal{R}$

Output: The posterior probability, $P \in [0,\ 1]$ of unseen test data, x, where $\sum_{i=1}^{C} P_i = 1$ and $C = 2$ (diabetes present (C_1) or not (C_2))

Compute the prior probabilities for each of the class [49], $P(Y = C_1) = \frac{N_{C1}}{N}$ and $P(Y = C_2) = \frac{N_{C2}}{N}$, where N is the number of sample

The output posterior probability of class for the given predictor (attributes), $P(C_i|X) = \frac{P(X|C_i) \times P(Y=C_i)}{P(X)}$, where $P(X|C_i)$ is the likelihood of the predictor for a given class and $P(X)$ is the prior probability of predictor.

I. ALGORITHMS FOR IMPLEMENTING XGBOOST

Algorithm 9: The steps of implementing XGboost (XB)

Input: The n-dimensional data, $X \in \mathcal{R}^n$ and target outcome, $Y \in \mathcal{R}$

Output: The posterior probability, $P \in [0,\ 1]$ of unseen test data, x, where $\sum_{i=1}^{C} P_i = 1$ and $C = 2$ (diabetes present (C_1) or not (C_2))

Initialize the model with constant value: $F_o(x) = \text{argmin}_\gamma \sum_{i=1}^{N} L(Y, \gamma)$ [32], where $L(Y,\ F(x))$ is the differentiable loss function and N is the number of sample

for $m = 1\, to\, M\ (n\_Iterations)$ **do**
- Compute pseudo-residuals, $r_{im} = -[\frac{\delta L(Y, F(X_i))}{\delta F(X_i)}]$, where $i = 1, 2, ..., N$
- Fit a base tree, h_m using training set (X_i, r_{im}) for $i = 1, 2, ..., N$
- Compute multiplier γ_m by $\gamma_m = \text{argmin}_\gamma \sum_{i=1}^{n} L(Y_i, F_{m-1}(X_i) + \gamma h_m(X_i))$
- Update the model by $F_m(x) = F_{m-1}(x) + \gamma_m h_m(x)$

$F_m(x)$ is the desired posterior probability, $P \in [0,\ 1]$