eDiaPredict: An Ensemble-based Framework for Diabetes Prediction

ASHIMA SINGH, ARWINDER DHILLON, and NEERAJ KUMAR, Thapar University, Patiala, India
M. SHAMIM HOSSAIN, Research Chair of Pervasive and Mobile Computing, and Department of Software Engineering, College of Computer and Information Sciences, King Saud University, Saudi Arabia
GHULAM MUHAMMAD, Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, Saudi Arabia
MANOJ KUMAR, SMVD University, Katra, India

Medical systems incorporate modern computational intelligence in healthcare. Machine learning techniques are applied to predict the onset and reoccurrence of the disease, identify biomarkers for survivability analysis depending upon certain health conditions of the patient. Early prediction of diseases like diabetes is essential as the number of diabetic patients of all age groups is increasing rapidly. To identify underlying reasons for the onset of diabetes in its early stage has become a challenging task for medical practitioners. Continuously increasing diabetic patient data has necessitated for the applications of efficient machine learning algorithms, which learns from the trends of the underlying data and recognizes the critical conditions in patients. In this article, an ensemble-based framework named e*DiaPredict* is proposed. It uses ensemble modeling, which includes an ensemble of different machine learning algorithms comprising XGBoost, Random Forest, Support Vector Machine, Neural Network, and Decision tree to predict diabetes status among patients. The performance of *eDiaPredict* has been evaluated using various performance parameters like accuracy, sensitivity, specificity, Gini Index, precision, area under curve, area under convex hull, minimum error rate, and minimum weighted coefficient. The effectiveness of the proposed approach is shown by its application on the PIMA Indian diabetes dataset wherein an accuracy of 95% is achieved.

CCS Concepts: • **Computing Methodologies** → **Machine Learning**;

Additional Key Words and Phrases: Diabetes prediction, ensembled models, XGBoost, decision tree, random forest

The authors are grateful to the Deanship of Scientific Research at King Saud University, Riyadh, Saudi Arabia for funding this work through the Vice Deanship of Scientific Research Chairs: Chair of Pervasive and Mobile Computing.

Authors' addresses: A. Singh, A. Dhillon, and N. Kumar, Computer Science and Engineering Department, Thapar Institute of Engineering and Technology, Patiala, Pb, India; email: {ashima, arvinder, neeraj.kumar}@thapar.edu; M. S. Hossain (corresponding author), Research Chair of Pervasive and Mobile Computing and Department of Software Engineering, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia; email: mshossain@ksu.edu.sa; G. Muhammad, Department of Computer Engineering, College of Computer, and Information Sciences, King Saud University, Saudi Arabia; email: ghulam@ksu.edu.sa; M. Kumar, SMVD University, Katra, India; email: manoj.kumar@smvdu.ac.in.

1551-6857/2021/06-ART66 $15.00
https://doi.org/10.1145/3415155

ACM Reference format:
Ashima Singh, Arwinder Dhillon, Neeraj Kumar, M. Shamim Hossain, Ghulam Muhammad, and Manoj Kumar. 2021. eDiaPredict: An Ensemble-based Framework for Diabetes Prediction. *ACM Trans. Multimedia Comput. Commun. Appl.* 17, 2s, Article 66 (June 2021), 26 pages.
https://doi.org/10.1145/3415155

1 INTRODUCTION

Recent surveys by the **World Health Organization (WHO)** show an up-growing trend in the amount and fatalities of diabetic patients worldwide. Based on such trends, the WHO predicted diabetes as the seventh driving reason for deaths by 2030 [2]. Diabetes is one of the fastest rising diseases in the world. Diabetes is defined as a collection of metabolic ailments that results in increased glucose levels in human blood. The two underlying reasons for increased glucose levels are as follows: (1) inability of the human body to produce a sufficient amount of insulin and (2) incorrect reaction of the body cells to insulin [1]. Insulin is the pancreatic secreted hormone that helps in regulating blood glucose. The blood sugar should be kept within the standard range (70–120 mg/dl or 3.6–6.9 mmol/l) [39]. Lower concentrations of glucose (<50 mg/dl) are identified as hypoglycemia, resulting in excessive thirst, sweating, seizures, and diabetic coma. Hypoglycemic prediction is a clinically significant task in diabetes management. Since hypoglycemia has hazardous effects such as seizures and coma, it should be predicted well in advance, and preventive actions must be taken. Higher concentrations of glucose (>200 mg/dl) are reported to be hyperglycemia leading to long-term vascular complications comprising diabetic retinopathy, neuropathy, and nephropathy [5]. Therefore, surveillance is necessary to properly regulate the amount of glucose to enhance the quality of life. Diabetes is classified into three types comprising Type-1, Type-2, and gestational diabetes. Type-1 diabetes occurs when the immune system destroys the beta cells that make insulin inside the pancreas. About 10% of patients fall under the Type-1 diabetes category. Though, it is difficult to prevent but, a treatment of supplying insulin externally to the body can be opted [3]. However, if the insulin generated inside the pancreas is not used in the right way, then it is classified as Type-2 diabetes. About 90% of cases of diabetes are of Type-2 and is common in patients aged more than 45 years. The chance of heart disease increases by 2 to 4 times among Type-2 diabetic patients [4]. Diabetes that affects females during pregnancy is known as gestational diabetes [41]. To measure diabetes at discrete times, blood glucometers are used. However, to measure diabetes continuously, Continuous glucose monitoring devices are used, providing a minimally invasive mechanism to record the patient's present glycemic level [42]. Delayed detection of diabetes affects most of the body parts, which include kidneys, eyes, heart, and nerves, and so on. Consequently, an early and accurate prediction of diabetes is important. The diagnosis and interpretation of diabetes, appropriate analysis of data becomes very important while handling them as a classification problem of **machine learning (ML)**. Hence, the application of computational intelligence is much appreciated for the efficient prediction of diabetes.

In today's world, Big data, the Internet of Things, **Artificial Intelligence (AI)**, ML, and **Deep Learning (DL)** are emerging technologies [52]. With the help of ML, patients can easily check their status in the early stage as well as it will also help practitioners for further research [6]. It can be applied to both the classification and regression problems. Diabetes Prediction is a classification problem, which means we can classify the patients in different classes like a patient is diabetic or not [5]. Different ML techniques are useful for examining and synopsizing the data into valuable information from various perspectives. ML involves various steps like preprocessing of the dataset, feature selection and extraction, training and testing, and further evaluation. Data are collected in multiple forms, such as clinical information, text information, and sensor information [46], generated using separate wearable devices [7] that are mostly in raw form. To convert these data in a meaningful form, preprocessing is needed, which includes handling of missing values in the dataset and imputing missing values so that predictions can be made accurately. The dataset contains multiple attributes, and selecting the best attributes from the feature space is the first and foremost injective for attaining the best prediction. There are various ML algorithms [38] including **Support Vector Machine (SVM), Decision Tree (DT), Neural Network (NN)**, and **Random forest (RF)** that can be applied for efficient diabetes prediction [8]. These models are trained, which are further tested using the test dataset to know whether the model is working correctly or not. Ensemble techniques also help the researchers to predict diabetes with the best accuracy, which uses bagging, boosting, and stacking techniques based on majority voting to classify the dataset [40]. It works by blending the best models results by using majority voting in which votes are given, and the one with the highest votes is chosen as the final result. From past years, researchers are working hard to predict diabetes but the achieved performance is not sufficient [19, 20]. So, there is a need to propose some other techniques for efficient and accurate prediction.

Diabetes mellitus is one of the most chronic diseases that has affected humans of all age groups across the globe. It goes undetected for a long period of time, showing no symptoms or very mild symptoms. It is found that undetected diabetes may harm other vital organs of the body. Thus, an early prediction is required to save human lives. Researchers have worked on the early prediction of diabetes using various ML algorithms, including SVM, RF, NN, KNN, and DT, but the performance achieved is not satisfactory. Therefore, advanced techniques are required to predict diabetes with better accuracy. Motivated by Reference [18], an ensemble of the existing ML algorithms is proposed for the detection and prognosis of diabetes, because ensemble algorithms are considered more accurate and flexible as compared to single classifiers. It provides the best solutions with greater accuracy and predictive performance. We have proposed an effective diabetes prediction framework called ***eDiaPredict,*** which deploys the ensemble of the selected ML algorithms for predicting diabetes. Various models that are selected are XGBoost, RF, SVM, NN, and DT. The key contributions are as follows:

- Multiple ML models are applied in the proposed framework to add diversity to the final ensemble model.
- Missing value imputation and normalization are used for pre-processing the PIMA diabetes dataset for diabetes prediction.
- **Recursive Feature Elimination (RFE)** is used for feature space reduction in the dataset. The performance of eDiaPredict is compared with five existing state-of-the-art ML models using traditional performance parameters such as Sensitivity, Specificity, Precision, Accuracy, **area under curve (AUC)**, and four new parameters (**area under convex hull (AUCH), minimum error rate (MER)**, MWL, and **Gini Index (GI)**).

The rest of the article is structured as follows: Section 2 describes the related work and comparison of proposed work with existing state-of-the-art frameworks. Section 3 discusses the background and preliminaries of the algorithms used in eDiaPredict. Section 4 describes the proposed framework in detail. Section 5 and 6 present the experimental analysis that discusses dataset used and performance parameters in detail. Finally, Section 7 concludes the article and presents future research directions.

2 RELATED WORK

Diabetes prediction is taken up as a challenge by medical practitioners as well as data scientists globally [51]. Quan Zou et al. [9] proposed ML algorithms for the detection of Type 2 Diabetes.

Sample of 68,994 healthy patients was taken by collecting data repeatedly for five times. **Principal Component Analysis (PCA)** and Minimum Redundancy Maximum Relevance were used to reduce the dimensionality of the dataset and training was performed. The results proved that the proposed approach worked well with an accuracy of 80%. Anjali et al. [10] use ML algorithms, which include AdaBoost with Decision Stump, DT, SVM, and **Naive Bayes (NB)** for the diagnosis and prognosis of diabetes mellitus. A sample from the PIMA dataset was taken, and the experiment was performed. It became clear from the results that AdaBoost with DT performed best with 80.72% accuracy. Amin et al. [11] make use of **convolutional neural network (CNN)** to predict certain measures. An experiment was performed, and RF uses the DT to combine the meaning of each tree to obtain the last results. Gandhi et al. [12] used data digging innovation to anticipate diabetes and conducted a pre-treatment project to handle the dataset using strategy and standardization to determine features. The technique of ML for SVM is assessed. From the perspective of the new pre-processing schemes and the K-closest neighbor classifier, Panwar et al. [13] suggested the methodology for diagnosing diabetes correctly. Sowjanya et al. [14] used the portable/android application to address the lack of diabetes care. Four ML algorithms [49] are used to organize the collected data, i.e., NB, J48, multilayer perceptron, and SVM. Hashi et al. [15] suggested using a DT and KNN to predict diabetes in a healthcare system. When trained using the PIMA diabetes dataset, the model achieved 90.43% accuracy. Sushant Ramesh et al. [16] proposed a deep NN with **Restricted Boltzmann Machine (RBM)** for the detection of diabetes. The PIMA Diabetes dataset was taken, and RBM with the deep NN was applied. The result shows that the proposed approach performed well with precision and recall values of 77% and 54%. On the other side, Suyash et al. [17] proposed another approach called **Artificial Neural Network (ANN)** to predict diabetes on the PIMA dataset. An experiment was performed, and results show that the proposed approach worked well with 90% accuracy. Sajida et al. [18] used AdaBoost and bagging ensemble techniques to classify diabetic patients with the help of J48DT as a base learner. A sample of 667,907 patients was taken from the Canadian Primary Care Sentinel Surveillance Network dataset, and the experiment was performed. The results show that AdaBoost works well with an AUC value of 98%. Anand [19] suggested ML models comprising classification and regression trees that would consider daily lifestyle operations to predict a patient's blood glucose fluctuations. A PIMA dataset was taken, and the experiment was performed. The result proved that the suggested framework works well with 75% accuracy. Jakhmola [20] suggested a model for predicting diabetes in an individual using supervised binning and multiple regression techniques. The model uses the PIMA dataset to provide 77.85% accuracy. Jarullah [21] used the J48 classifier and pre-processing methods to develop a DT model. The model uses the PIMA diabetes dataset to give 78.17% accuracy. Hamzaatal [22] proposed a hybrid approach called ***k*-means clustering along with SVM (KSVM)** using the feature selection algorithm. The model is tested to obtain the experimental results using the PIMA dataset. Various study fellows use ANN, as mentioned in the prediction of diabetes. Heydari et al. [23] addressed various techniques including ANN, DT, and Bayesian Network to predict diabetes. The experiment was performed, and results were analyzed. It is obtained from the results that Artificial Neural System performed best with an accuracy of 97.44%. Komi et al. [24] investigated early diabetes forecast, and the findings of the experiment have shown that ANN offers the greatest accuracy compared to other methods. Swain et al. [25] used the hybrid **Adaptive Neuro-Fuzzy Inference Scheme (ANFIS)** and ANN to explore Diabetes mellitus prediction and characterization. The ANFIS method is more satisfactory in terms of precision than the ANN method. It is used to model knowledge-based systems, **Fuzzy cognitive maps (FCM)**. Douali et al. [26] described the procedure for predicting gestational diabetes and used the case-based decision support scheme for FCM. Bhatia et al. [27] used particular FCM to discover proximity or non-appearance of diabetes mellitus. The product tool was tested in 50 instances with 96%

accuracy. Han Wu et al. [28] utilizes enhanced *k*-means algorithms and logistic regression for diabetes prediction using the questionnaire method, a dataset was gathered online, and training was carried out. The findings obtained are contrasted with the outcomes of the PIMA dataset, and the experiment has shown that precision increases by 3% compared to other researchers. For the prediction of diabetes, Leila et al. [29] suggested a knowledge-based scheme based on the clustering method, and predictive noise extraction methods were used. Various datasets were used, including PIMA, mesothelioma, WDBC [30], and StatLog, and the experiment was performed. Results demonstrate that CART with noise removal and clustering methods work well by anticipating diabetes efficiently. Adil Hussain et al. [31] proposed an ensemble technique based on a voting method for the prediction of diabetes. NHANES dataset from 2013 to 2014 was taken, which consists of 10,172 patients with 54 features. An experiment was performed, and results show that the proposed approach performed well with an area under the curve value of 75%. A comparison of the proposed framework (eDiaPredict) with existing frameworks based on the performance parameters is taken up in Table 1 below.

3 BACKGROUND AND PRELIMINARIES

In this section, we have provided a brief background and preliminaries of ML algorithms used in the proposed framework “eDiaPredict.” The proposed framework uses five ML models, namely XGBoost, DT, RF, NN, and SVM. The function and features of each model are discussed below.

3.1 XGBoost

XGBoost is an extreme gradient boosting algorithm. It is an advanced implementation of gradient boosting algorithm in which a new model is added that predicts the errors or residual from the prior models and then combines the new model with the previous model. This new model is retrained to remove the error from earlier models, followed by another set of retrains until all the errors are removed. The stochastic gradient boosting sub-samples each column and row and reduces the error. The regularized gradient boosting reduces overfitting using L1 (Lasso regression) and L2 (ridge regression) [54]. By doing this, performance is enhanced, and the model works well to give an accurate result. It is a highly efficient, reliable, and portable ML model that works by reducing the error recursively until we get accurate and efficient results. XGBoost works on the following criteria:

Consider a scenario with model *f(x)* having y as the actual value, gamma (γ) as the predicted value, and L as the loss function. First model $f_0(x)$ can be given by:

$$f_0\left(x\right) = argminL\left(y, \gamma\right) \tag{1}$$

Now, find the difference between the target value and gamma, which is called residual value, and retrain the data to build new model $h_1(x)$. This $h_1(x)$ is added to $f_0(x)$ to get $f_1(x)$, i.e.,

$$f_1\left(x\right) < -\, f_0\left(x\right) + h_1\left(x\right). \tag{2}$$

The residual is calculated from $f_1(x)$ and retrains it to build another model $h_2(x)$. In this way, the process is repeated recursively until all the errors are removed to get the efficient results, i.e.,

$$f_m\left(x\right) < -\, f_{m-1}\left(x\right) + \alpha h_m\left(x\right). \tag{3}$$

Here,

$$\alpha = argmin_{\alpha} L\left(y_i,\ f_{m-1}\left(x_1 +\ \alpha h_m\left(x\right)\right)\right), \tag{4}$$

$$h_m\left(x\right) = \sum_{j=1}^{T} b_{jm} 1_{Rjm}\left(x\right), \tag{5}$$

Table 1. Comparison of Proposed Framework with Existing Frameworks

Authors	Work Done	Result	Pros	Cons
Quan Zou et al. [9]	Proposed DT, RF, and NN to predict diabetes. Used PCA to reduce the dimensionality of features	RF effectively predict diabetes with accuracy, sensitivity, and specificity value of 80%, 89%, and 85% respectively	—	Only glucose index performed well. More indexes are required for effective results.
Anjali et al. [10]	Used SVM, DT, and NB algorithms to predict and diagnose diabetes.	SVM performed well by effectively predicting diabetes with 80% accuracy.	The proposed framework can also be used for the prediction of different diseases.	Basic algorithms are used for prediction. Other powerful algorithms like KNN and ANN can be used for better results.
Amin et al. [11]	Used SVM and CNN, to train the samples and get the final result.	CNN outperforms with accuracy, and specificity value of 85.4% and 94.1% respectively.	—	—
Gandhi et al. [12]	In this F-score and *k*-means clustering is used to reduce the features space, and then SVM is proposed to predict diabetes.	SVM works well by effectively predicting diabetes with 90% accuracy.	F-score achieves better performance as compared to other filter methods.	Only a single model is used. Better performance can be achieved by using advanced algorithms.
Panwar et al. [13]	KNN classifier is proposed to predict diabetes. Both raw and preprocessed dataset is used to obtain the result.	KNN works well with 85% accuracy.	This proposed model helps the physicians to predict diabetes in a better way.	This algorithm is useful in the prediction of diabetes only.
Sowjanya et al. [14]	DT, SVM, Multilayer perceptron, and NB algorithm are presented to predict diabetes effectively.	DT effectively predicts diabetes with sensitivity, specificity, and AUC values of 89%, 90%, and 91% respectively.	—	—
Hashi et al. [15]	DT and KNN are proposed to diagnose diabetes on the PIMA diabetes dataset using an app to be installed on mobile phones.	DT outperforms with 90.43% accuracy.	Provides application which can be used by practitioners to diagnose diabetes via the internet.	The algorithms used for prediction are fundamental. Other powerful algorithms that are SVM, NN, RF, or ensemble of these can be used for better results.
Sushant et al. [16]	Used DL framework, i.e., ANN to identify the risk present in diabetic patients.	Effectively identify the risk with MSE and RMSE of 0.30 and 0.39, respectively.	It shows an effective performance in terms of precision.	The dataset used was of small size. The efficiency of ANN in terms of speed and accuracy can be improved by increasing the number of patients using AI approaches.
Suyash et al. [17]	Used NN to predict diabetes.	NN attained an accuracy of 90% and an AUC value of 89%.	This approach is valuable for health decision leaders, who will take preventive action before diabetes happens in significant numbers.	The size of the dataset is small. Better performance can be achieved by using the dataset of large size.
Sajida et al. [18]	Adaboost and Bagging method with DT as a Base learner is used to predict diabetes.	Adaboost outperforms with 90% accuracy.	It can be applied to predict other diseases like hypertension, heart diseases, and dementia.	Only DT is used as a base learner. Other algorithms comprising SVM, Naïve Bayes, and NN can be used for better results.
Anand et al. [19]	Classification and Regression trees are used to predict diabetes. Along with late sleeping, roadside eating, and family history, Blood Pressure is identified as a significant factor that causes diabetes.	The proposed approach attains an accuracy of 75% by effectively predicting diabetes.	—	The size of the dataset was small, because it is collected manually. Effective performance can be achieved by using the dataset of large size.

(Continued)

Table 1. Continued

Authors	Work Done	Result	Pros	Cons
Jarullah [21]	DT was proposed to identify patients with developing diabetes. Two phases are considered comprising the preprocessing of the dataset, followed by the prediction phase using DT.	DT works well by identifying patients correctly with an accuracy of 78%.	—	The size of the dataset used is small. Effective performance can be achieved by using the dataset of large size.
Hamza et al. [22]	Integrated SVM and KSVM are used to predict diabetes.	KSVM accurately predicts diabetes with 89% accuracy.	This research sought to address the issue of diabetes diagnosis being identified wrongly	Optimization techniques can be used for more accurate results.
Heydari et al. [23]	ML algorithms comprising DT, SVM, ANN, and Bayesian networks are used to predict diabetes.	ANN outperforms with an accuracy of 89%.	—	These can be used on different datasets.
Komi et al. [24]	SVM, ANN, and Extreme Learning Machine (ELM) were presented to predict diabetes.	ANN works well by effectively predicting diabetes with 89% accuracy.	—	This study's primary drawback is its limited sample size, which rendered it very challenging to obtain statistical significance for all of the endpoints
Swain et al. [25]	ANFIS and ANN were proposed to predict diabetes.	ANFIS gives good results with accuracy and MSE of 91% and 0.042.	Less error is obtained, which shows that the proposed work effectively predicts diabetes.	Computational complexity can be calculated to prove the effectiveness of the work.
Douali et al. [26]	A Case-Based FCM decision support system was anticipated to predict gestational diabetes.	The presented approach effectively predict gestational diabetes with 90% accuracy.	CBFCM offers control rules describing health symptoms and enabling patient classification dependent on knowledge from patients	—
Bhatia et al. [27]	FCM had been proposed based on the symptoms recorded by a fuzzy system to identify the presence of diabetes among patients.	FCM predicts the presence of diabetes with 70% accuracy.	The proposed solution was built as an alternate knowledge-based method that inherits the benefits of fuzzy relationships with consistency, versatility, clarity, and ease of use	—
Han Wu et al. [28]	K-means clustering and Logistic Regression were used on the PIMA dataset to diagnose diabetes.	The proposed approach attains an accuracy of 92%.	The proposed approach can be applied on other datasets also.	Consumes a lot of time during preprocessing of the dataset.
Adil et al. [31]	Used Ensemble method with the majority voting to predict diabetes.	Effectively predict diabetes with 90% accuracy.	Provides a user-friendly environment to predict diabetes effectively.	Goes under high training time overhead, under-fitting, and over-fitting problem.
eDiaPredict	The ensemble of the existing ML algorithms (XGBooost and RF) is proposed to predict diabetes.	Effectively predict diabetes with 95% accuracy.	It is very effective in the prediction of diabetes and can be applied to other datasets also.	PIMA dataset has data for female patients only. Male diabetic patients are not considered.

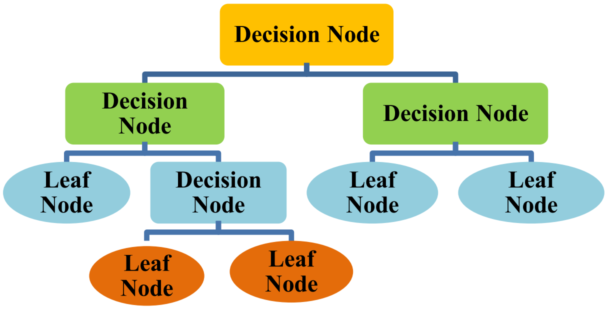

Fig. 1. Structure of DT [43].

where b_{jm} is predicted mean or voted values. For XGBoost, nodes are made by using the gain function, which is calculated by minimizing the loss function, i.e.,

$$Loss = \sum_{i=1}^{n} L(y_i,\ f_{m-1}(x_i) +\ h_m(x_i)) + \sum_{j=1}^{T} \Omega\ (h_m(x_j)). \tag{6}$$

On solving the above using Taylor theorem, we get the gain value, which helps in building XGBoost tree. One of the main advantages of XGBoost is that it supports parallel processing, which makes it faster compared to gradient boosting, stochastic gradient boosting, and regularized gradient boosting. The time taken by XGBoost is 0.04 s for 100 nodes which increases to 0.16 s for 1,000 nodes. For 10,000 nodes, it takes 0.24 s, which shows a linear increase in time. Therefore, its time complexity is $O\,(\log\, n)$.

3.2 Decision Tree

Decision Tree is a supervised learning method used for regression and classification problems. There are nodes and edges in a DT where nodes represent features and edges represent the outcome. An algorithmic approach is followed to split the data into child nodes by calculating each feature's entropy and information gain value. The feature with the highest information gain value is selected as the root node. This process is done for each feature until the last node stops splitting further [32]. Entropy is calculated using the following equation:

$$H\,(S) = \sum_{c\varepsilon C} -p\,(c)\ log_2 p\,(c). \tag{7}$$

Here *H(S)* is the entropy, *C* is the set of classes, *S* is the set of features in the data, and *p(c)* is the probability of *C* with respect to *S*. Entropy is used to calculate information gain. The equation for information gain is given below:

$$IG\,(A, S) = H\,(S) - \sum_{t\varepsilon T} p\,(t)\,H\,(t). \tag{8}$$

Here, *H(S)* is the entropy, *T* is the subset on which decision to be made, *p*(*t*) is the probability of *T* with respect to *S*, and *H*(*t*) is entropy on subset *T*. The structure and pseudocode of the DT are shown in Figure 1 and Algorithm 1, respectively.

In the present work, we get glucose as the first decision node with a gain value of 127, followed by BMI with a gain value of 26.4. We checked the condition, and if the value of gain in BMI is less than 26.4, it can be classified as the patient having no diabetes. Moving ahead, we get to age as the next splitting node, followed by insulin and diabetes pedigree function. The complexity of the DT is O (*n*log *n*) where *n* defines the number of leaves. In the present research, the time taken to build and test a model for 100 nodes is 0.14 s. For 1,000 nodes, it takes 0.2 s followed by 10,000 nodes that take 0.41 s to build a model that increases $n\log n$ times. One of the DT's main advantages is that it takes very little time and effort in the preprocessing of the dataset and does not require normalization and scaling of the dataset. Sometimes overfitting occurs when the model fits the training data so well that it falsely predict the value of testing data [33]. Overfitting is removed with the help of pruning, which works without affecting the accuracy of the model.

ALGORITHM 1: Algorithm for DT

```
Input: Set of Features (S), S ≠ Φ, n_attr > 0
Output: Final DT
Begin:
  Repeat
  Maximum_Gain <- 0, Split <- NULL, E <- Entropy of all Attributes // Initially set values to 0
  for all Attributes i in S
  Gain <- InformationGain (i, E)          // calculate information gain using entropy
       if (Gain >Maximum_Gain)             // Check if gain is greater than maximum gain
           Maximum_Gain <- Gain            // if yes, assign gain value to maximum gain
               Split <- i                  // set ith attribute as decision node
           Else
               Stop
             End if else
  End for
  Partition (S, Split)                     // Call the function on remaining attributes
  End
End
```

3.3 Random Forest

Random Forest is a popular ML algorithm that contains multiple decision trees built on various subsets of the dataset, wherein all trees' outcome is used to make predictions [8]. It is a classifier that can solve both regression and classification problems [50]. In regression, the average of all the decision trees outcome is calculated, and in classification, the votes from the different decision trees are aggregated to decide the final output. RF is considered as an ensemble of various simple decision trees. For the present research, the complete dataset is taken, partitioned in k bootstrap samples called bags. On each bootstrap sample, the DT algorithm runs. The outcome of each decision tree is collected, and voting is performed. Based on the result of the voting, one with the highest vote is chosen as a classification result. The RF structure is shown in Figure 2 followed by the pseudo-code is given in Algorithm 2, which briefly explains the working of RF.

In the present research, we make 100 trees on different subsets of the dataset. We select the highest voted tree from all the trees, which is 217 with attributes BMI, glucose level, age, and insulin. The time complexity of RF is $O(Mn\log n)$ where M is the total number of decision trees formed and $n\log n$ is the time complexity for n DTs. The runtime for 100 iterations is 0.16 s followed by 1.54 s and 14 s for 1,000 and 10,000 iterations, respectively. In RF, performance is improved due to an ensemble of multiple decision trees [8]. One of the biggest problems in the DT is over-fitting. This problem is resolved in RF by optimizing the tuning parameters. It can also handle large datasets efficiently without any variable deletion.

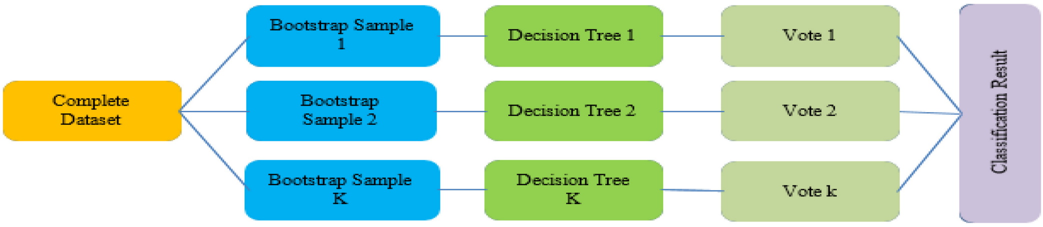

Fig. 2. Structure of RF.

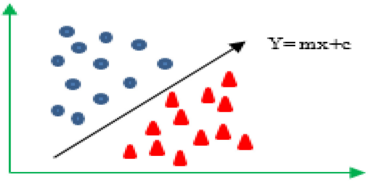

Fig. 3. Linearly Separable classes.

ALGORITHM 2: Algorithm for RF

Input: A training set with n individuals, k, t attributes
Output: Predicted Result based on voting method
Begin:

```
For i = 1 to n
    Select k features randomly from n such that k << n // k implies bootstrap sample
    For each node of the tree
        Select p′ attributes
        Calculate best split from p attributes
    End
End for
DT <- { T1, T2, T3, ...} // All the decision trees are assigned to one variable DT
Vote <- { DT} // voting is performed
Select the high voted predicted value as the final result
```

End

3.4 Support Vector Machine

SVM is a supervised learning method in which the decision plane separates two classes using a hyperplane [34]. The hyperplane can be two-dimensional (2D) or 3D. It represents a line in the case of 2D and a plane in the case of 3D. Sometimes, the data are linearly separable, as shown in Figure 3. They can be separated using a line whose equation is given by:

$$y = mx + c. \tag{9}$$

Here, m is the slope, and c is constant.

But sometimes the data are not linearly separable as shown in Figure 4. In that case, it is not possible to separate the classes on the x-y axis. Therefore, a new axis called the z-axis is required to classify the classes on a 3D plane. The equation for z-axis is given as:

$$z = x^2 + \; y^2. \tag{10}$$

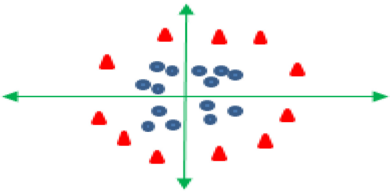

Fig. 4. Non-linearly separable classes.

The following non-linearly separable classes can be separated by making a circle along with the origin.

SVM works on the Maximum Margin Classifier principle, which maximizes the geometric margin and minimizes the error value. The pseudocode for SVM is given in Algorithm 3.

ALGORITHM 3: Pseudocode for SVM

```
Input: N_in (input vectors), N_sv (Support Vectors), N_ft (features in support vectors), SV[N_sv]
(Support Vector Array), IN[N_in] (input vector array), b (bias)
Output: Decision function output F
Begin:
For i from 1 to N_in
    Output = 0
    For j from 1 to sv
        assign Margin distance<-0            // initially assign margin to 0
        For k from 1 to Nft
            Set margin = margin + (sv[j].feature [k] – IN.feature [k]^2)
        End for
        Set k= exp (−β X Margin)
        Output = Output + SV[j].α X k
    End for
    Output = Output + bias.
End for
END
```

The kernel used for diabetes prediction is RBF kernel, which correctly classifies 156 instances of 230. The time taken by the training model in diabetes prediction for 100 nodes is 0.22 s. For 1,000 nodes it takes 0.28 s, followed by 0.42 s for 10,000 nodes, respectively, which shows that it is taking $O(n^2)$ time. SVM can be used in various applications like medical diagnosis. One of the main advantages of SVM is that it can work efficiently with all types of data, i.e., structured and unstructured data comprising text and images data.

3.5 Neural Network

The biological nervous system inspired the working of the neural networks. The main element is the novel structure of the data processing framework. A group of highly interconnected processing elements worked together to solve particular problems. It works as follows:

For a given training sample, $\{x_i, y_i | \; x_i \in R^P, \; y_i \in R^m\}_{i=n}^{n}$, if n defines total observations, p gives the dimension of covariates, y_i defines the target for each observation, then NN with n hidden layers can be written as:

$$f_L(x) = \sum_{i=1}^{L} g(x, w_{i,} b_i)\, \beta_i = h(x)\, \beta. \tag{11}$$

Here, g defines activation function, w_i defines input weights, b_i defines the bias variable, $h(x)$ defines hidden layers, and β defines the output target variable. The hidden layer for NN can be expressed as:

$$\begin{bmatrix} h\,(x1) \\ h\,(x2) \\ . \\ . \\ h\,(xn) \end{bmatrix} = \begin{bmatrix} g\,(w_1, b_1, x_1) & \dots & g\,(w_L, b_L, x_1) \\ g\,(w_1, b_1, x_2) & \dots & g\,(w_L, b_L, x_2) \\ & . & \\ & . & \\ g\,(w_1, b_1, x_n) & \dots & g\,(w_L, b_L, x_n) \end{bmatrix}_{(n*l)} . \tag{12}$$

And the target matrix is given by:

$$Y = \begin{bmatrix} y_1^T \\ y_2^T \\ . \\ . \\ y_n^T \end{bmatrix} = \begin{bmatrix} y_{11} & \dots & y_{1m} \\ y_{21} & \dots & y_{2m} \\ & . & \\ & . & \\ y_{n1} & \dots & y_{nm} \end{bmatrix} . \tag{13}$$

The output weights can be solved with the given Equation:

$$\beta = H^Y \left(\frac{I}{C} H H^Y \right)^{-1} Y, \tag{14}$$

where I is an $n^* m$ matrix.

One of NN's main advantages is that they can work even when complete information about the data is unknown. It can also work for multiple computations simultaneously, which makes it faster compared to other models.

3.6 Ensemble Modeling

The ensemble method combines two or more classification algorithms to improve or boost overall performance. Bagging and Boosting are the two most prominent ensemble-based strategies. Boosting follows a sequential process in which the subsequent model corrects the previous model's errors while Bagging works by combining the results of multiple models to get the final result. Various ensemble methods are described below:

3.6.1 Bagging. In Bagging, the results of different models are combined to obtain a generalized result. These models are trained on different subsets of the original dataset that are created with the help of bootstrapping sampling technique. These subsets are known as bags and are of the same size. The models are trained on each subset in parallel. The results obtained from these models are combined to determine the final prediction [40].

3.6.2 Boosting. Boosting is a sequential process that converts weak base learners to strong learners [40]. In this approach, a base model is created to make predictions on the dataset. If a base learner causes any prediction error, then the full attention is paid to the prediction error, and a new model is created to remove the error. This process is executed repeatedly until all the prediction errors are removed, and higher accuracy is achieved.

3.6.3 Stacking. Stacking is an ensembling technique in which different classification algorithms are combined with the help of meta-classifier or meta-regressor. A base model is created and trained on a complete dataset and predictions are made. The meta-model is trained on the output obtained from the base model predictions [40]. In our proposed framework, a boosting-based voting method is used to ensemble the current research models.

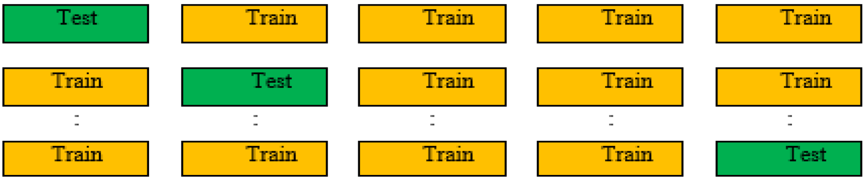
Test Train Train Train Train
Train Test Train Train Train
: : : : :
Train Train Train Train Test

Fig. 5. *k*-fold cross-validation.

3.7 Cross Validation

Cross-validation is the process of analyzing the performance of algorithms by dividing the dataset into two parts. One part is used for training and the other part is used for validating purposes [35]. Cross-validation aims to ensure that every part from the original dataset has the same chance of appearing in the training and testing set. The final output got from the cross-validation is used to check robustness. One of the techniques for validation is *k*-fold cross-validation. In *k*-fold cross-validation, the validate dataset is divided into *k* parts in which *k*-1 parts are used for training purposes, and one part is used for evaluating the performance of the trained model. This is repeated again and again until all the parts go through the train and test phase. The result is obtained in the form of performance parameters at the end of iterations, which are averaged to get the final results. If the achieved results are close or equal to training results, then it means the models are working correctly [36]. It is shown in Figure 5. The main advantage of this approach is that each data point comes precisely once in the validation process, leading to low bias and variance in the model's overall performance [36].

4 PROPOSED FRAMEWORK

In this research, Ensemble modelling based on ML models comprising XGBoost, RF, SVM, NN, and DT is used to predict diabetes based on the PIMA dataset [55]. The dataset is preprocessed for missing values, and further feature extraction is performed with the help of RFE [47]. The individual models are trained and performance is evaluated. Based on the majority voting method, the best models are selected that are ensembled together to predict the desired outcome. Tenfold cross-validation is used to check the robustness of the models. The flowchart of the whole process is shown in Figure 6.

4.1 Data Preprocessing

A dataset is an accumulation of information portraying distinctive appreciations (factors) of various real products (units). A PIMA diabetes dataset of 768 patients is taken, including data from 21- to 65-year-old female patients. The dataset consists of eight features and one label attribute comprising the number of pregnancies, glucose level, blood pressure of the patient, skin thickness, insulin level, body mass index, diabetes pedigree function, age, and class, which describes whether a patient has diabetes or not. The preprocessing of the dataset is performed in three steps described in the following sections.

4.1.1 Checking of Missing Values. Missing value and **not allowed (NA)** [48] values are checked. Among the eight features, only the number of pregnancies for a patient can be zero. Rest all values should be other than zero. If the value found to be zero other than pregnancies, then that value is treated as a missing value. Dataset is passed for missing value algorithm, which is given in Algorithm 4. It is found that except age and pedigree function, rest all features have some missing values. The number of missing values for each attribute is shown in Table 2.

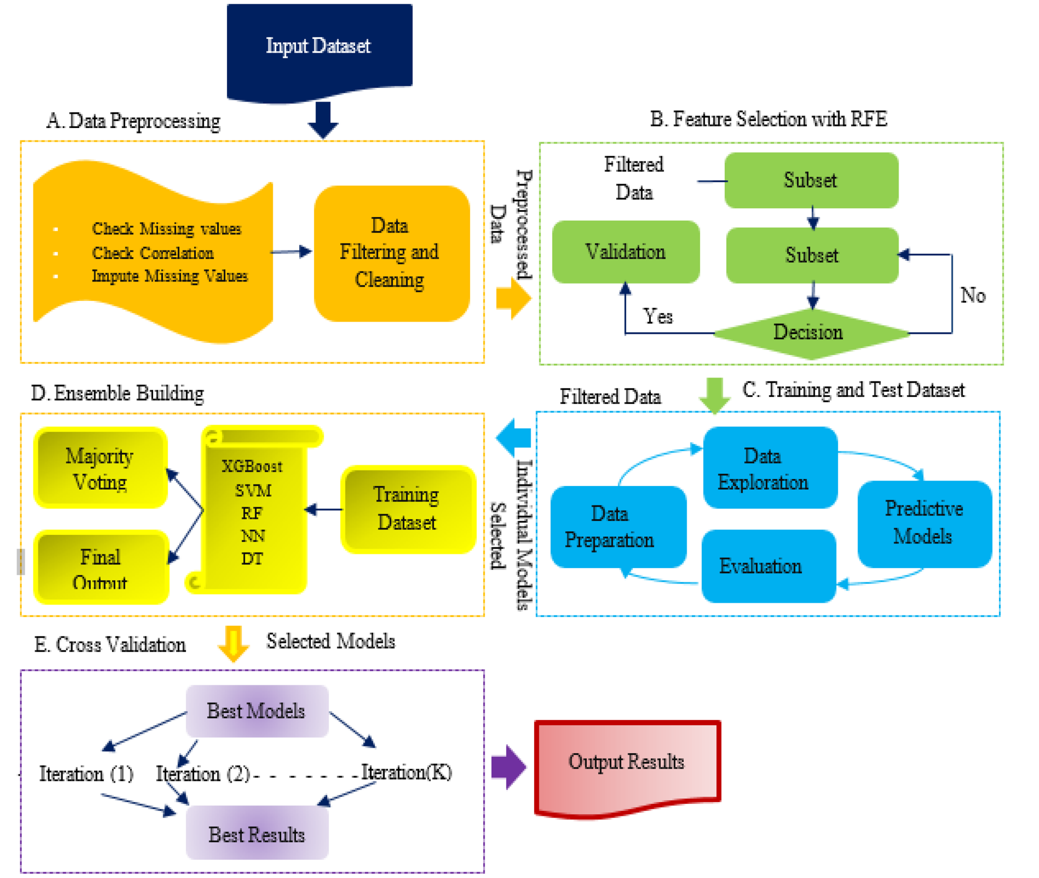

Fig. 6. Workflow of eDiaPredict.

Table 2. No of Missing Entries for Each Attribute

Attribute Name	No. of Missing Values	Attribute Name	No. of Missing Values
Insulin	374	Glucose level	5
Skin Thickness	227	Pedigree Function	0
Blood Pressure	35	Age	0
Body Mass Index	11		

4.1.2 Correlation among the Attributes. Correlation is the measure of the relationship between two quantitative attributes. It can be positive or negative. When the value of both attributes increases, it is known as a positive correlation. When one of the values increases and others decrease, then it is known as a negative correlation [37]. Correlation tells how the attributes are dependent on each other to predict diabetes. A correlation matrix is made in our proposed framework, which shows that there is no high correlation among the independent variables. It also tells that there is no correlation between age and blood pressure with diabetes. The scatter plot for our dataset is shown in Figure 7.

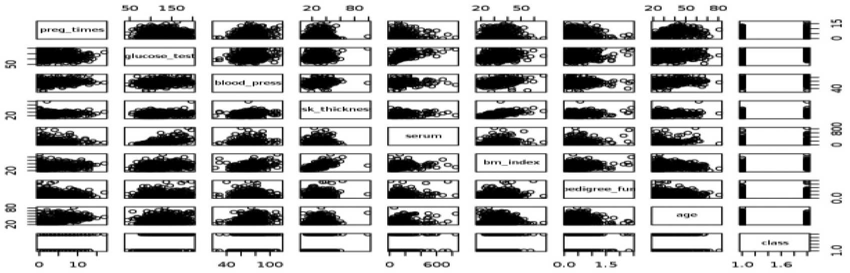

Fig. 7. Correlation plot.

4.1.3 Missing Value Imputation. Missing values create problems while analyzing the dataset. It is necessary to impute some value in place of it. So, missing value imputation is the process of replacing the missing value with some other value. Here, the median of the attribute values is calculated and imputed in place of the missing values. Algorithm 4 gives the pseudocode for finding and imputing missing values.

ALGORITHM 4: Pseudocode for Missing Values Imputation

```
Input: Input Dataset
Output: No of missing values for each attribute
Begin:
    Pass the dataset to is.NA function in R.
    NA <- 0// Set all NA values to 0.
    Calculate the sum of all 0 values in each attribute.
    Print Missing Values          // Print all missing value in each attribute as shown in
                                   Table 2
    if (attribute value = 0)
        Dataset [attribute] <- median (Dataset$Attribute) //calculate median of that column &
                                                            replace 0 with median
    else
        Dataset [attribute] <- Dataset [attribute]          // keep the value as it is
    End if else
    Normalize Dataset
End
```

4.2 Feature Selection

Screening, diagnosis, and editing of data are performed in three steps. First, all the missing values and *not allowed* values are removed from the dataset. Once the data are preprocessed feature selection is performed. The RFE [47] package available in R is used for feature selection. In the previously existing work, PCA [44] was used to extract the features. But it returned the combination of principal features. It was unable to give significant features [44]. RFE eliminates such issues by returning optimal features that are relevant to the target variable. Previously, RFE was used in other applications [45] to extract the important features. This technique starts with building a model on the complete set of features and calculates a rank for each feature. The features with the least rank are removed, the model is re-built, and the rank is computed for the remaining features. This is repeated until the desired sets of features are achieved. Therefore, it will return the important features by recursively eliminating the least important features. The pseudocode for RFE is described in Algorithm 5.

Table 3. Rank for Each Attribute

Attribute Name	Rank	Attribute Name	Rank
Glucose level	8	Insulin	4
Body Mass Index	7	Skin Thickness	3
Age	6	Pregnancies	2
Pedigree Function	5	Blood Pressure	1

ALGORITHM 5: Recursive Feature Elimination

```
Inputs: A training set T, a set of n features, i.e., F = {f1, f2, f3,..fn}, Ranking Method M {T, F}
Output: Best Features
Begin:
  for i in [1: n]                            // For each feature in the dataset
      Rank set F using M {T, F}              // rank each feature using ranking method
      f* <- last ranked feature in F         // assign last ranked features to f*
      R (n- i+1) <-f*                        //Set final rank to the features
      F<- F-f*                               // remove last rank features
  End for
End
```

A loop is started, which traverses all the features and ranks them at the end of each iteration according to their importance. The last ranked features are removed, and then the loop is iterated for the remaining set of features. By doing this, all the dependencies and collinearity exits in the model are removed. In our proposed framework, all features are chosen in predicting diabetes with body mass index and glucose level having the highest rank and number of pregnancies and blood pressure having the lowest. Table 3 shows the rank of features according to their importance.

4.3 Training and Test Dataset

In this phase, a dataset is divided into training, validation, and testing phase with 70:10:20 ratios. The selected models NN, RF, SVM, DT, and XGBoost are trained individually on the dataset. In SVM, a radial basis kernel is used to predict diabetes. In RF, 500 trees are built on which DT runs individually and the results are obtained. From the results, voting is done, and the highest voted result is chosen. Similarly, the hidden layers are used for training the NN model, which uses sigmoid activation for prediction. In XGBoost, 10 repeats are used, which works by removing errors in each iteration and no error is achieved at the end of 10th iteration. XGBoost and RF achieve better results when compared to NN, SVM, and DT. The pseudocode of training the models is described in Algorithm 6.

4.4 Ensembling of Models

In this phase, the best models obtained from the previous training phase are ensembled using the voting method. Majority voting and weighted voting are two types of voting methods [56]. For the present research, majority voting is used. The results for each model are calculated and the ones with higher votes are chosen for ensembling. The prediction with higher votes is considered as the final prediction. The algorithm for the ensemble is shown in Algorithm 7. In the proposed work, XGBoost and RF being the best performing models are ensembled to achieve the final results. It is discussed in Section 5.

ALGORITHM 6: Pseudocode for training models

```
Input: Dataset
Output: Diabetes prediction
  Begin:
    x<-missing values // assign all the missing values to x
    Count x                  // count all missing values
    x[missing values]<- impute median                  // impute median in missing values
    RFE [x]                 // feature selection using RFE
    Divide x into a:b:c with 70:10:30 ratio
    Train a                  // Train the models
    Calculate performance parameters
  End
```

ALGORITHM 7: Pseudocode for Majority Voting

```
Input: ML Models
Output: Higher Voted Models
Begin:
  Set Votes = v
  Set Table= {}// Initially table is empty
  For all v
  if v in table                  // if there is a vote entry in table
  Table [v] = Table [v] + 1                  // increment the vote value
  Else Table [v] = 1                  // keep it as it is
  End if else
  End for
  Return max
End
```

4.5 Cross-validation

In the proposed framework, 10% of the dataset is used for validation to check the performance of the models. 10-fold cross-validation is used in which datasets are divided randomly into 10 samples. Every time, 9 samples are used for training the model and the 10th part is used to test the performance of the trained model and calculate the error. There are a total of 10 iterations for 10 samples. Each sample goes through the training and testing phase at least once followed by performance evaluation. An average result of each sample is obtained and compared with the training results. There is no need to retrain the models as the training model results are equal to cross-validation results in the present research. The pseudocode for cross-validation is described in Algorithm 8.

The time complexity for cross-validation is $O(n)$, where n defines the sample size. The data are processed K times for each sample. Therefore, the time complexity becomes $O(Kn)$, where k is a constant. $K-1$ folds are used for training in each iteration and then the remaining folds are used to evaluate the performance of the model. There are total n samples, and each sample should be traversed once for the prediction of diabetes. Therefore, the complexity of traversing the n samples is $O(n)$.

Table 4. H/W and S/W Requirement (Minimum)

Sr. No.	H/W & S/W	Memory	Sr. No.	H/W & S/W	Memory
1.	Processor	32 bit	4.	Operating System	Windows 7
2.	RAM	2 GB	5.	Programming Language	R (Rattle)
3.	Hard Disk	80 GB	6.	Platform	R Studio

ALGORITHM 8: Pseudocode for cross-validation

Input: PIMA Diabetes Dataset

Output: Results

Begin:

```
Divide dataset D randomly in k1, k2,…k10 samples    // We divide the diabetes dataset into 10 samples
for i = 1 to k                                       // Run the loop for all the 10 samples
    Train (k-1) [i] samples using ensemble models    // Train the 9 samples using ensemble modelling
    Test kth sample                                  // Test the remaining one sample
    D[i] <- Calculate performance                    // Check the performance for each sample
End For
Results = Average D[i].                              // Calculate the average of each sample and assign it to results
If (Results = = Training data results)               // if training and cross-validation results are equal
    Models are trained Properly                      // We can say that the models are properly trained
Else Retrain them
End if else
End
```

4.6 Experimental Setup

Table 4 represents the minimum **Hardware (H/W)** and **Software (S/W)** requirements required to implement the proposed framework eDiaPredict. The different packages used to implement the present work are given in Table 5.

4.7 Experimental Dataset

In this document, we used "PIMA Indian Diabetes Data." There are 768 instances of women patients aged 21 or older in the dataset. Nine characteristics are available in the information set [54]. Table 2 provides a short description of the dataset. This dataset is divided into 70:10: 20 ratios, i.e., 70% is used for training data, 10% for cross-validation, and 20% testing data.

4.8 Performance Parameters

Table 6 shows performance parameters that are used to test the efficiency of the proposed framework.

5 RESULTS AND ANALYSIS

First, we applied the selected models comprising RF, SVM, DT, NN, XGBoost individually and obtained the results in terms of accuracy, sensitivity, specificity, precision, GI, AUC, AUCH, MER, and **Minimum Weighted Coefficient (MWC)**. The results demonstrate that XGBoost performed better among all the models with 92.21% accuracy individually. Boosting the performance of the base learners and reducing the bias recursively is the reason behind the outstanding performance

Table 5. Packages used to Implement the Present Work

Package	Description
XGBoost	It is an Extreme Gradient Boosting package used in the *R* [“*R*” is italic and roman; please make consistent throughout]interface, which includes efficient decision trees and linear models. It is 10 times faster than gradient boosting, because it performs multiple parallel computations. It is also used for regression, classification, and ranking.
randomForest	Random Forest is a forest of multiple trees that are used for classification and regression purposes. The results of each tree are achieved, and the final outcome is selected using the majority voting method.
Caret	This package is used to implement decision trees that stand for short Classification and Regression Training. It is used for complex regression and classification problems. It includes 30 packages in itself that are loaded based on the model used.
Kernlab	It stands for Kernel Based Machine Lab, which includes various methods to solve classification, regression, clustering, and dimensionality reduction problem. In our work, it is used for SVM to predict diabetes.
nnet	It is a package for a feed-forward NN with a single hidden layer and is used to train the model for diabetes prediction in our research.
hmeasure	This package is used to calculate the performance metrics comprising sensitivity, specificity, accuracy, AUC, true positive, false positive, and so on, to predict the results.

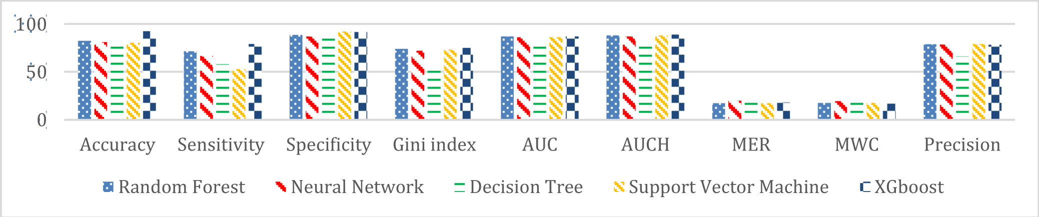

Fig. 8. Bar plots for each performance parameter.

of XGBoost. However, RF produces the second-best result with 80.95% accuracy and is shown in Table 7.

XGboost and RF work by reducing the bias recursively and finding the best solution. XGBoost is an implementation of gradient boosting, in which a tree is drawn, and bias is calculated. Again, the model is retrained and repeated until all the bias is eliminated. XGBoost is also implemented very fast because of the single tree trained repeatedly. Similarly, RF works by randomly drawing multiple trees and predicting the results based on voting to attain the final tree. Therefore, XGBoost and RF based on their performance, are selected for the ensemble. Table 7 and Figure 8 showcase the performance of various models taken individually. It is cleared from the plot that XGBoost performed best for each parameter with accuracy, sensitivity, specificity, Gini-index, AUC, AUCH, MER, MWL, and precision value of 92.21%, 79%, 91.5%, 0.746%, 0.873, 0.888, 0.182, 0.168, and 0.783, respectively.

To improve the performance, ensembling of the selected models is performed to obtain the results. Table 8 represents the results of each performance parameter before parameter optimization.

Table 6. Important Parameters Used in the Present Work

Parameter	Description
Sensitivity= $\frac{\text{TP}}{\text{TP+FN}}$	Sensitivity is defined as the degree of total true positive (TP) or positive cases that are predicted as true. It is also known as recall. It is given as TP's ratio with the sum of TP and False Negative (FN).
Specificity = $\frac{\text{TN}}{\text{TN+FP}}$	Specificity is defined as the degree of actual negative value and is predictive as negative. It is the true negative (TN) ratio with the sum of TN and False Positive (FP).
Precision = $\frac{\text{TP}}{\text{TP+FP}}$	Precision is defined as the percentage of actual results that are true or relevant. It is given by the ratio of TP with the sum of TP and FP.
Accuracy = $\frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}$	Accuracy is defined as the percentage of predicted results that are expected correctly concerning actual values.
AUC	The AUC shows the relation between the true-positive rate (TPR) and the false-positive rate (FPR) at different threshold values. A high AUC means both TPR and FPR values are high. Higher the value of AUC, the better the model is.
Area under Convex Hull (AUCH)	The AUCH finds the ROC curve's optimal points under some class and cost distribution and calculates the area. A point is said to be optimal if it lies under the convex hull. A convex hull contains the data points that are connected with each other to form line segments.
Gini Index (GI)	Gini index is used to find the unevenness in the variable data. Its value lies between ranges 0 to 1. 1 means the distribution is uneven and 0 means even distribution.
Minimum Error Rate	It is used to find the error value. Its value lies between ranges 0 to 1. 1 means the error rate is more so need to train the model again, and 0 means less error rate.
MWC = L(c, T)	Minimum Weighted Coefficient works by selecting a threshold value that will minimize the loss for each value of the cost.

TP means patient having cancer is suffering from cancer and FN means a person having no cancer are predicted as having cancer. TN means a person not suffering from cancer has no cancer in actual. FP means the person predicted as having cancer has actual no cancer.

Table 7. Results for Individual Models

Model	Accuracy	Sensitivity	Specificity	GI	AUC	AUCH	MER	MWC	Precision
RF	82.25	71.4	88.4	0.739	0.87	0.884	0.173	0.176	0.787
NN	80.95	66.7	87	0.72	0.86	0.87	0.199	0.196	0.786
DT	78.79	57.8	90.5	0.57	0.78	0.81	0.22	0.21	0.66
SVM	80.52	52.8	92.1	0.73	0.86	0.88	0.173	0.177	0.79
XGBoost	92.21	79	91.5	0.746	0.873	0.888	0.182	0.168	0.783

Table 8. Results Obtained by Ensemble Models before Parameter Optimization

Model	Accuracy	Sensitivity	Specificity	GI	AUC	AUCH	MER	MWL	Precision
DT + NN	91.5	75.8	80.12	0.50	0.70	0.76	0.34	0.23	0.80
DT+RF	70.6	71.4	80	0.53	0.80	0.78	0.21	0.24	0.75
DT+SVM	72.5	91.4	80.25	0.50	0.73	0.8	0.19	0.21	0.80
SVM+ RF	50	82	82.1	0.46	0.60	0.61	0.23	0.21	0.71
RF+NN	70.1	76.4	81.5	0.41	0.75	0.72	0.22	0.14	0.82
SVM+ NN	89	90.5	91.04	0.34	0.81	0.81	0.29	0.25	0.71
XGBoost+ RF	92	92.32	9.8	0.41	0.924	0.89	0.12	0.15	0.81
XGBoost+ NN	80	68	90.0	0.44	0.61	0.78	0.32	0.22	0.73
XGboost+ DT	89	59	90	0.42	0.83	0.86	0.20	0.20	0.72
XGboost+ SVM	84	65	87	0.37	0.612	0.650	0.28	0.21	0.8

Table 9. Results Obtained by Ensemble Models after Parameter Optimization

Model	Accuracy	Sensitivity	Specificity	GI	AUC	AUCH	MER	MWL	Precision
DT + NN	94.5	79.5	83.12	0.52	0.78	0.79	0.16	0.18	0.87
DT+RF	76.6	71.4	83.7	0.47	0.81	0.81	0.19	0.18	0.75
DT+SVM	78.5	94.4	82.25	0.45	0.75	0.76	0.21	0.25	0.81
SVM+ RF	35.8	80	84.1	0.54	0.68	0.68	0.27	0.26	0.76
RF+NN	76.1	76.4	83.5	0.48	0.73	0.74	0.17	0.13	0.86
SVM+ NN	87.9	96.5	90.04	0.46	0.85	0.86	0.19	0.24	0.73
XGBoost+ RF	95	90.32	96.8	0.44	0.90	0.901	0.142	0.145	0.88
XGBoost+ NN	89	60	90	0.41	0.70	0.708	0.303	0.276	0.78
XGboost+ DT	87	53	90	0.40	0.86	0.87	0.18	0.17	0.79
XGboost+ SVM	88	71	89	0.39	0.69	0.69	0.30	0.28	0.84

Here each model comprising DT, RF, NN, SVM, and XGBoost are ensemble with other models using the voting method to obtain the final results.

Table 9 presents the results after parameter optimization. It is visible that the performance of proposed work after parameter optimization is better than the performance before optimization with an improvement of 5% in accuracy. The results show that the ensemble XGBoost with the RF model produces the best result by accurately predicting the diabetic patient with 95% accuracy. eDiaPredict gives more efficient results with 95% accuracy, which shows an improvement of 15% and 5% when compared to previously proposed approaches Anjali et al. [10] and Hashi et al. [15], respectively. Individually, XGBoost is a boosting method, and RF is a bagging approach. When both XGBoost with RF model are ensembled together, they work more efficiently as compared individually. For ensembling, majority voting is used, which provided more efficient results as this voting approach reduces error in individual models dramatically.

Table 10 shows the value of FP and FN at 90%, 95%, and 99% CI. There is a slight difference between the values of FP and FN at different confidence intervals. The value of FP for ensemble XGBoost and RF is 34 at 90% CI and 35 and 39 for 95% and 99% CI respectively. In the same way, FN value for ensemble XGBoost and RF is 39, 40, and 43 at 90%, 95%, and 99% CI, respectively.

The bar plots for ensembled models based on accuracy, sensitivity, specificity, GI, AUC, AUCH, MER, MWC, precision are shown in Figures 9, 10, and 11 below. Results of Figure 9 shows that

Table 10. Value of FP and FN at Different Confidence Intervals (CI)

Model	90% CI		95% CI		99% CI	
	FP	FN	FP	FN	FP	FN
DT + NN	94.5	79.5	83.12	0.52	0.78	0.79
DT+RF	41	36	37	42	41	46
DT+SVM	50	55	52	57	55	59
SVM+ RF	48	50	49	51	51	52
RF+NN	38	42	40	44	42	46
SVM+ NN	40	44	41	46	42	48
XGBoost+ RF	34	39	35	40	39	43
XGBoost+ NN	40	36	37	42	41	46
XGboost+ DT	40	41	42	43	44	44
XGboost+ SVM	34	40	38	44	42	48

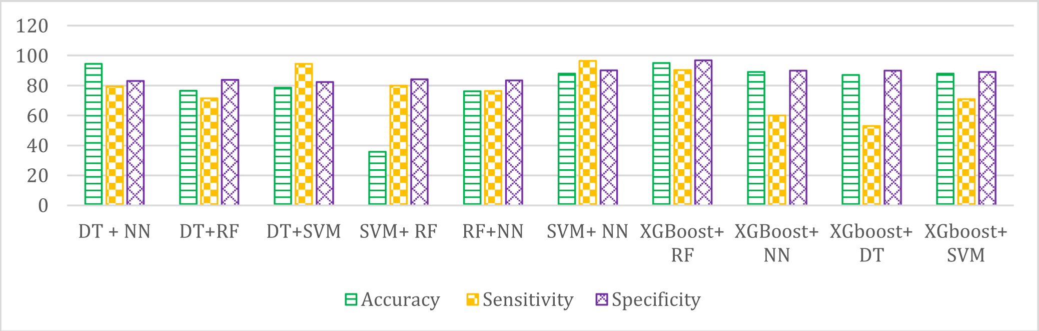

Fig. 9. Bar plots for ensemble models for parameters Accuracy, Sensitivity, and Specificity.

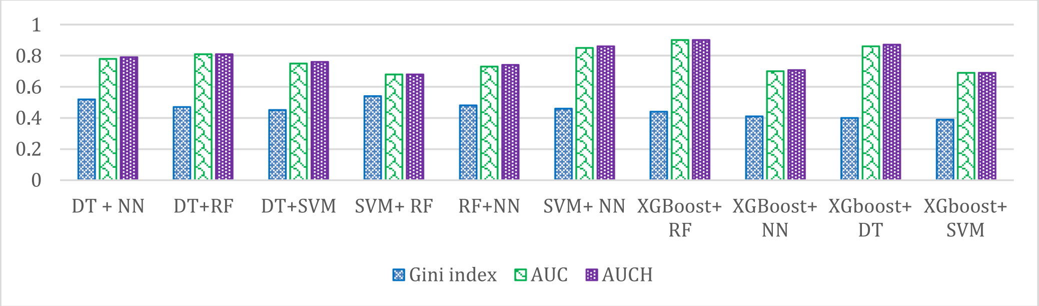

Fig. 10. Bar plots for ensemble models for parameters Gini Index, AUC, and AUCH.

XGboost ensembled with RF have higher bar graphs with accuracy, sensitivity, and specificity value of 95%, 90.32%, and 96.8% respectively. They achieve a 5% improvement when compared to models trained alone. The GI, AUC, and AUCH results are shown in Figure 10 and are cleared from the results that ensemble RF with XGBoost performed best with 0.44, 0.90, and 0.91, respectively. GI value must be low for good results, which is evident from the achieved results. The MER, MWL, and precision values are also good for ensemble XGBoost and RF and are 0.142, 0.145, and 0.88, respectively. The bar plots for MER, MWL, and precision are shown in Figure 11.

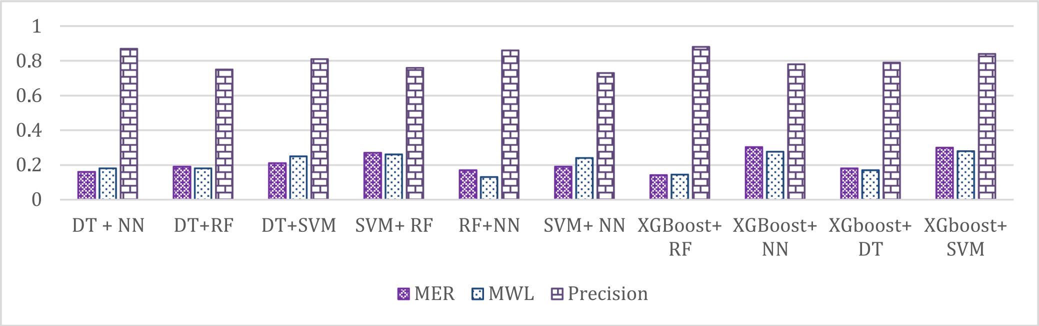

Fig. 11. Bar plots for ensemble models for parameters MER, MWL, and Precision.

The proposed eDiaPredict shows the suitability of the models based on the performance parameters comprising accuracy, sensitivity, specificity, GI, precision, AUC, AUCH, MER, and, MWC. The ensemble XGBoost with the RF model produces the best result by accurately predicting the diabetic patient with 95% accuracy. Also, the value of other performance parameters is higher for these models. Individually, XGBoost is a boosting method and RF is a bagging approach. XGBoost is an implementation of gradient boosting, in which a tree is drawn, and bias is calculated. Bias is eliminated by repeatedly training the model. XGBoost is also implemented very fast because of the single tree trained repeatedly. Similarly, RF works by randomly drawing multiple trees and predicting the results based on voting. Eventually, the final tree is drawn based upon the voting results.

The time complexity of the ensemble of XGBoost and RF is $O(Mn\log\ n + \log\ n)$. Therefore, the theoretical time complexity of the model comes out to be the $O(Mn\log\ n)$. Additionally, the application of XGBoost provides eDiaPredict a faster scheme due to parallelized computation and effective storage in in-memory units called blocks [53]. It is scalable in distributed as well as limited memory environments. This proves the suitability of the proposed framework.

6 CONCLUSIONS

The proposed framework “eDiaPredict” is used to predict diabetic patients based on glucose concentrations. PIMA Indian diabetes dataset is used, which is preprocessed based on missing value calculations and imputations. Further, feature selection based on Recursive Feature Elimination is performed to select the best features. Multiple ML models are applied in the proposed framework to add diversity to the final ensemble model. The findings indicate that the “eDiaPredict” can provide patients with an effective and precise prediction of diabetes based on glucose concentrations. In the proposed approach. XGBoost individually achieves an accuracy of 92%. XGBoost ensembled with RF gives the prediction accuracy of 95%. The obtained results using “eDiaPredict” are compared with the results of previously proposed approaches having an accuracy of 80% and 90%, which shows an improvement of 15% and 5%, respectively. The reason behind the improvement in results is that both XGboost and RF algorithms work on the principle of reducing the bias recursively and finding the best solution. Doctors, clinical businesses, medical researchers, and scientists working in healthcare will get benefitted from the proposed framework. In the future, the proposed framework can also be validated using real-life clinical diabetic data. It can also be extended for large diabetic datasets.

REFERENCES

[1] Chitra Jegan, V. Anuja Kumari, and R. Chitra. 2018. Classification of diabetes disease using support vectormachine. *Int. J. Eng. Res. Appl.* 3, 2 (2018), 1797–1801. Retrieved from https://www.researchgate.net/publication/320395340.

[2] Parampreet Kaur, Neha Sharma, Ashima Singh, and Bob Gill. 2019. CI-DPF: A cloud IoT based framework for diabetes prediction. In *Proceedings of the 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON'18)*, 654–660. DOI: https://doi.org/10.1109/IEMCON.2018.8614775

[3] Kevin Plis, Razvan Bunescu, Cindy Marling, Jay Shubrook, and Frank Schwartz. 2014. A Machine Learning Approach to Predicting Blood Glucose Levels for Diabetes. AAAI Workshop Technical Report WS-14-08 (2014), 35–39.

[4] Tao Zheng, Wei Xie, Liling Xu, Xiaoying He, Ya Zhang, Mingrong You, Gong Yang, and You Chen. 2017. A machine learning-based framework to identify type 2 diabetes through electronic health records. *Int. J. Med. Inform* 97, (2017) 120–127. DOI: https://doi.org/10.1016/j.ijmedinf.2016.09.014

[5] Ambika Choudhury and Deepak Gupta. 2019. *Recent Developments in Machine Learning and Data Analytics*. Springer Singapore. DOI: https://doi.org/10.1007/978-981-13-1280-9

[6] Radia Belkeziz and Zahi Jarir. 2017. A survey on internet of things coordination. In *Proceedings of the 2016 3rd International Conference on Systems of Collaboration (SysCo'16)*, 619–635. DOI: https://doi.org/10.1109/SYSCO.2016.7831328

[7] M. S. Hossain. 2017. Cloud-supported cyber–physical localization framework for patients monitoring. *IEEE Syst J.* 11, 1 (2017), 118–127. DOI: 10.1109/JSYST.2015.2470644

[8] Usha Devi Gandhi, Priyan Malarvizhi Kumar, R. Varatharajan, Gunasekaran Manogaran, Revathi Sundarasekar, and Shreyas Kadu. 2018. HIoTPOT: Surveillance on IoT devices against recent threats. *Wireless Pers. Commun.* 103, 2 (2018), 1179–1194. DOI: https://doi.org/10.1007/s11277-018-5307-3

[9] Quan Zou, Kaiyang Qu, Yamei Luo, Dehui Yin, Ying Ju, and Hua Tang. 2018. Predicting diabetes mellitus with machine learning techniques. *Front. Genet.* 9, (2018) 1–10. DOI: https://doi.org/10.3389/fgene.2018.00515

[10] V. Veena Vijayan and C. Anjali. 2016. Prediction and diagnosis of diabetes mellitus—A machine learning approach. In *Proceedings of the 2015 IEEE Recent Advances in Intelligent Computational Systems (RAICS'15)*, 122–127. DOI: https://doi.org/10.1109/RAICS.2015.7488400

[11] S. U. Amin et al. 2019. Cognitive smart healthcare for pathology detection and monitoring. *IEEE Access.* 7 (2019), 10745–10753. DOI: 10.1109/ACCESS.2019.2891390

[12] Khyati K. Gandhi and Nilesh B. Prajapati. 2014. Diabetes prediction using feature selection and classification. *Int. J. Adv. Eng. Res. Dev* 1, 05 (2014), 1–7. DOI: https://doi.org/10.21090/ijaerd.0105110

[13] Madhuri Panwar, Amit Acharyya, Rishad A. Shafik, and Dwaipayan Biswas. 2017. K-nearest neighbor based methodology for accurate diagnosis of diabetes mellitus. In *Proceedings of the 2016 6th International Symposium on Embedded Computing and System Design (ISED'16)*, 132–136. DOI: https://doi.org/10.1109/ISED.2016.7977069

[14] K. Sowjanya, Ayush Singhal, and Chaitali Choudhary. 2015. MobDBTest: A machine learning based system for predicting diabetes risk using mobile devices. In *Proceedings of the Souvenir 2015 IEEE International Advanced Computing Conference (IACC'15)*, 397–402. DOI: https://doi.org/10.1109/IADCC.2015.7154738

[15] Emrana Kabir Hashi, Md Shahid Uz Zaman, and Md Rokibul Hasan. 2017. An expert clinical decision support system to predict disease using classification techniques. In *Proceedings of the International Conference Electrical Computer and Communications Engineering ECCE 2017.*(2017), 396–400. DOI: https://doi.org/10.1109/ECACE.2017.7912937

[16] H. Balaji, N. Ch. S. N. Iyengar, and Ronnie D. Caytiles. 2017. Optimal predictive analytics of pima diabetics using deep learning. *Int. J. Database Theory Appl.*10, 9 (2017), 47–62. DOI: https://doi.org/10.14257/ijdta.2017.10.9.05

[17] S. Srivastava, L. Sharma, V. Sharma, A. Kumar, A. and H. Darbari. 2019. Prediction of diabetes using artificial neural network approach. In *Engineering Vibration, Communication and Information Processing*. Springer, Singapore, 679–687.

[18] Sajida Perveen, Muhammad Shahbaz, Aziz Guergachi, and Karim Keshavjee. 2016. Performance analysis of data mining classification techniques to predict diabetes. *Proc. Comput. Sci.* 82, (2016) 115–121.

[19] Ayush Anand and Divya Shakti. 2016. Prediction of diabetes based on personal lifestyle indicators. In *Proceedings of the 2015 1st International Conference on Next Generation Computing Technologies (NGCT'15)*, 673–676. DOI: https://doi.org/10.1109/NGCT.2015.7375206

[20] Shivani Jakhmola and Tribikram Pradhan. 2015. A computational approach of data smoothening and prediction of diabetes dataset. *ACM Intnational Conference Proceeding Series*, 744–748. DOI: https://doi.org/10.1145/2791405.2791572

[21] A. A. A. Jarullah. 2011. Decision tree discovery for the diagnosis of type II diabetes. In *Proceedings of the 2011 International Conference on Innovations in Information Technology*. IEEE.

[22] Ahmed Hamza and Hani Moetque. 2017. Diabetes disease diagnosis method based on feature extraction using K-SVM. *Int. J. Adv. Comput. Sci. Appl* 8, 1 (2017), 236–244. DOI: https://doi.org/10.14569/ijacsa.2017.080130

[23] Mahmoud Heydari, Mehdi Teimouri, Zainabolhoda Heshmati, and Seyed Mohammad Alavinia. 2016. Comparison of various classification algorithms in the diagnosis of type 2 diabetes in Iran. *Int. J. Diabetes Dev. Ctries.* 36, 2 (2016), 167–173. DOI: https://doi.org/10.1007/s13410-015-0374-4

[24] Messan Komi, Jun Li, Yongxin Zhai, and Zhang Xianguo. 2017. Application of data mining methods in diabetes prediction. In *Proceedings of the 2nd International Conference on Image, Vision and Computing (ICIVC'17)*, 1006–1010.

[25] A. Swain, S. N. Mohanty, and A. C. Das. 2016. Comparative risk analysis on prediction of diabetes mellitus using machine learning approach. In *Proceedings of the 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT'16).*

[26] N. Douali, J. Dollon, and M. Jaulent. 2015. Personalized prediction of gestational Diabetes using a clinical decision support system. In *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'15).* 1–5. DOI: 10.1109/FUZZ-IEEE.2015.7337813

[27] Nitin Bhatia and Sangeet Kumar. 2015. Prediction of severity of diabetes mellitus using fuzzy cognitive maps. *Life Sci. Adv. Tech.* 29 (2015), 71–79.

[28] Han Wu, Shengqi Yang, Zhangqin Huang, Jian He, and Xiaoyi Wang. 2018. Type 2 diabetes mellitus prediction model based on data mining. *Informat. Med. Unlocked* 10, (2018), 100–107.

[29] Mehrbakhsh Nilashi, Othman bin Ibrahim, Hossein Ahmadi, and Leila Shahmoradi. 2017. An analytical method for diseases prediction using machine learning techniques. *Comput. Chem. Eng.* 106, (2017), 212–223.

[30] WDBC. Retrieved 2019 from https://datahub.io/machine-learning/wdbc.

[31] AdilHusain and Muneeb Khan. 2018. Early diabetes prediction using voting based ensemble learning. In *Proceedings of the International Conference on Advances in Computing and Data Sciences,* Springer, Singapore. 2018, 95–103.

[32] S. Rasoul Safavian and David Landgrebe. 1991. A survey of decision tree classifier methodology. *IEEE Trans. Syst. Man Cybernet.* 21, 3 (1991), 660–674. DOI: https://doi.org/10.1109/21.97458

[33] Mohamed Ahmed Ahmed, Ahmet Rizaner, and Hakan Ulusoy Ali. 2018. A novel decision tree classification based on post-pruning with Bayes minimum risk. *PLoS One* 13, 4 (2018), 1–12. DOI: https://doi.org/10.1371/journal.pone.0194168

[34] C. Cortes and V. Vapnik. 1995. Support-vector networks. *Mach. Learn.* 20, 3 (1995), 273–297.

[35] M. S. Hossain, S. U. Amin, M. Alsulaiman, and G. Muhammad. 2019. Applying deep learning for epilepsy seizure detection and brain mapping visualization. *ACM Trans. Multimed. Comput. Commun. Appl* 15, 1 (2019), 1–17. DOI: 10.1145/3241056

[36] S. U. Amin et al. 2019. Deep Learning for EEG motor imagery classification based on multi-layer CNNs feature fusion. *Future Gener Comput Syst.* 101 (2019), 542–554. DOI: 10.1016/j.future.2019.06.027

[37] What Is Correlation. Retrieved 2019 from https://www.displayr.com/what-is-correlation/.

[38] Arwinder Dhillon, Ashima Singh 2019. Mach. Learn. *Healthcare.* 8, (July 2019), 92–109.

[39] Diseases Conditions. Retrieved 2019 from https://www.mayoclinic.org/diseases-conditions/diabetes/diagnosis-treatment/drc-20371451.

[40] Ensemble Learning to Improve Machine Learning Results. Retreived 2019 from https://blog.statsbot.co/ensemble-learning-d1dcd548e936.

[41] Gestational Diabetes and Pregnancy. Retrieved 2019 from https://www.cdc.gov/pregnancy/diabetes-gestational.html.

[42] How Does a Continuous Glucose Monitor Work? Retrieved 2019 from https://www.webmd.com/diabetes/guide/continuous-glucose-monitoring#1.

[43] Decision Tree Classification in Python. Retrieved 2020 from https://www.datacamp.com/community/tutorials/decision-tree-classification-python.

[44] Feature Selection Is Python—Recursive Feature Elimination. Retreived 2020 from https://towardsdatascience.com/feature-selection-in-python-recursive-feature-elimination-19f1c39b8d15.

[45] M. Chen, J. Yang, L. Hu, M. S. Hossain, and G. Muhammad. 2018. Urban Healthcare Big Data System Based on Crowd-sourced and Cloud-Based Air Quality Indicators. *IEEE Commun. Mag.* 56, 11 (2018), 14–20. DOI: 10.1109/MCOM.2018.1700571

[46] Gagangeet Singh Aujla, Anish Jindal, Rajat Chaudhary, Neeraj Kumar, Sahil Vashist, Neeraj Sharma, and Mohammad S. Obaidat. 2019. DLRS: Deep learning-based recommender system for smart healthcare ecosystem. In *Proceedings of the IEEE International Conference on Communications.* DOI: https://doi.org/10.1109/ICC.2019.8761416

[47] Pratt. 2018. Anti-drug antibodies: emerging approaches to predict, reduce or reverse biotherapeutic immunogenicity. *Antibodies* 7, 2 (2018), 19. DOI: https://doi.org/10.1142/S0219720018500178

[48] Arwinder Dhillon and Ashima Singh. 2020. eBreCaP: Extreme learning based model for breast cancer survival prediction. *IET Sys. Biol.* (2020), 12. DOI: https://doi.org/10.1049/iet-syb.2019.0087

[49] Parampreet Kaur, Ashima Singh, and Inderveer Chana, 2021. Computational techniques and tools for omics data analysis: State-of-the-art, challenges, and future directions. *Arch. Computat. Methods Eng.* (2021). DOI: https://doi.org/10.1007/s11831-021-09547-0

[50] G. Muhammad, M. S. Hossain, and N. Kumar. 2021. EEG-based pathology detection for home health monitoring. *IEEE J. Sel. Areas Commun.* 39, 2 (2021), 603–610. DOI: 10.1109/JSAC.2020.3020654

[51] Neha Sharma and Ashima Singh. 2018. Diabetes detection and prediction using machine learning/IoT: A survey. In *Proceedings of the IEEE International Conference on Advanced Informatics for Computing Research*, Springer, Singapore, (2018), 471–479. DOI: https://doi.org/10.1007/978-981-13-3140-4_42

[52] Thinking Before Building: XGBoost Parallelization. Retreived 2020 from https://medium.com/blablacar-tech/thinking-before-building-xgboost-parallelization-f1a3f37b6e68.

[53] Arwinder Dhillon, Ashima Singh, Harpreet Vohra, Caroline Ellis, Blesson Varghese, and Sukhpal Singh Gill. 2020. IoTPulse: Machine learning-based enterprise health information system to predict alcohol addiction in Punjab (India) using IoT and fog computing. *Enter. Inform. Sys.* (2020), 1–33. DOI: https://doi.org/10.1080/17517575.2020.1820583

[54] How XGBoost Works. Retreived 2020 from https://docs.aws.amazon.com/sagemaker/latest/dg/xgboost-HowItWorks.html.

[55] PIMA INDIAN DIABETES. Retreived 2019 from https://www.kaggle.com/rnmehta5/pima-indian-diabetes-binary-classification.

[56] Emsemble Methods. Retreived 2020 from https://www.toptal.com/machine-learning/ensemble-methods-machine-learning.

Received January 2020; revised July 2020; accepted August 2020