International Journal of Electrical and Computer Engineering (IJECE)
Vol. 14, No. 5, October 2024, pp. 5834~5847
ISSN: 2088-8708, DOI: 10.11591/ijece.v14i5.pp5834-5847

Comparing hyperparameter optimized support vector machine, multi-layer perceptron and bagging classifiers for diabetes mellitus prediction

Nuzhat Ahmad Yatoo[1], Ishok Sathik Ali[2], Imran Mirza[3]
[1]Department of Computer Applications, B. S. Abdur Rahman Crescent Institute of Science and Technology, Chennai, India
[2]Department of Information Technology, B. S. Abdur Rahman Crescent Institute of Science and Technology, Chennai, India
[3]Department of Computer Science and Engineering, Poornima University, Jaipur, India

Article Info

Article history:

Received Jan 2, 2024
Revised May 14, 2024
Accepted Jun 4, 2024

Keywords:

Bagging
Classification
Diabetes mellitus
Machine learning
Multi-layer perceptron
Support vector machine
XGBoost

ABSTRACT

Diabetes mellitus (DM) is a chronic metabolic disorder that affects the way the body processes blood glucose levels. Within the medical field, machine learning (ML) has significant potential for accurately forecasting and diagnosing a range of chronic conditions. If an accurate prognosis is achieved early, the risk to health and intensity of DM can be significantly mitigated. In this study, a robust methodology for DM prognosis was proposed, which included anomaly replacement, data normalization, feature extraction, and K-fold cross-validation. Three machine learning methods, support vector machine (SVM), multi-layer perceptron (MLP), and bagging, were employed to predict diabetes mellitus using the National Health and Nutritional Examination Survey (NHANES) 2011-2012 dataset. Accuracy, AUC, and recall were chosen as the evaluation metrics and subsequently optimized during hyperparameter tweaking. From all the comprehensive tests, bagging outperformed the other two models with an accuracy of 0.966, an AUC score of 0.992, and a recall of 0.97. The proposed methodology surpasses other approaches for forecasting DM.

Corresponding Author:

Ishok Sathik Ali
Department of Information Technology, B. S. Abdur Rahman Crescent Institute of Science and Technology
Chennai-48, India
Email: isathikali@crescent.education

1. INTRODUCTION

Diabetes mellitus (DM) is an endocrinological condition that is a group of metabolic or diverse afflictions caused by irregularities in insulin secretions and insulin actions, or both. For our bodies to boast energy, a lot of the food consumed is converted into glucose or sugar [1]. Insulin is a pancreas-formed hormone that acts as the main to allow glucose from the food we eat to shift into the cells in the body from the bloodstream to generate energy. Before 1982, type 1 diabetes was diagnosed only on clinical expertise, with the most common symptoms being severe hyperglycemia and ketoacidosis. Type 2 DM is trivial, so your body is not at all able to make good use of the insulin it produces. DM affects a huge percentage of the world's population. DM is a set of metabolic illnesses that includes hyperglycemia and decreased insulin action and/or secretion. DM accounts for 90% to 95% of all diabetic cases in North America [2]. As the obesity epidemic continues, the prevalence is anticipated to rise [3]. In India, approximately 61.3 million populations aged 20 to 79 years persisted with diabetes in 2011. Diabetes can lead to the dysfunction and worsening of many organs such as neuropathy, nephropathy, and retinopathy disorders [4], [5]. People shall start taking precautionary strides like maintaining an active lifestyle and safety measures like getting their

blood sugar analyzed frequently [6]. It is found that millions can be saved by routine cost analysis for diabetes patients, notably in developing countries like India [7]. Around 1.5 million people died due to diabetes in 2012 all over the world. In 2008, the budgetary hindrance of diabetes was over 2.35 billion dollars. More than 550 million people are expected to be affected by diabetes by 2030 [8]. The number of individuals recognized with diabetes in Canada is pedestrian, according to the Canadian Diabetes Association (CDA), growing from 2.5 million to around 3.7 million. According to the International Diabetes Federation, there were 382 million independents influenced by DM in 2013. Healthcare expenses for diabetes are likely to be $490 billion by 2030 [9]. 90% of diabetes patients are DM all over the world. According to the American Diabetes Association (ADA), the overall cost of around 13,700 USD per year for individuals who are recognized with diabetes is 2.3 times greater than fit individuals in America [10]. Diabetic patients in the world will number 642 million by 2040, implying that one out of every ten persons will have diabetes [11]. All these numbers and facts indicate that there is a need for a predictive method or model that can warn the concerned people about the possibility of getting diabetes.

To enhance the effectiveness of machine learning (ML) techniques in DM diagnosis, it is crucial to identify the key risk factors associated with DM and analyze their correlation. To improve the appropriateness of data, sampling techniques, and analysis are implemented to identify and analyze its relevant attributes. Various ML models were accessed using multiple evaluation metrics to compare their effectiveness and identify the most suitable ML model for diagnosing DM. Grid search optimization technique is employed to optimize the parameters of ML techniques.

2. RELATED WORK

Many researchers developed models using various ML techniques to predict DM. Edelman *et al.* [12] conducted research work spanning three years to assess the utility of hemoglobin A1c (HbA1c) in predicting the risk of getting diabetes. Balkau *et al.* [13] used simple clinical, biological, and genetic approaches to predict the possibility of diabetes in people. The authors determined the best clinical, biological Lang genetic factors for predicting diabetes. Zou *et al.* [14] employed three ML methods decision tree (DT), random forest (RF), and neural network (NN) with 5-fold cross-validation on the Luzhou dataset. The best-performing model was the RF with an efficiency of 0.8084 and for the Pima Indian Diabetes Dataset (PIDD) 0.7721 with all the attributes. Lai *et al.* [15] used two ML approaches gradient boosting machine (GBM) and logistic regression (LR) for the prognosis of DM. Since the dataset had missing values in some of the columns and removed the records with missing values. They used the adjusted-threshold approach and the class weight method to cope with the problem of unbalanced data. The developed GBM model has an area under the receiver operating characteristic (AROC) of 84.7% and a sensitivity of 71.6%, while the proposed LR model has an AROC of 84.0% and a sensitivity of 73.4%. Alam *et al.* [16] applied artificial neural network (ANN), RF, and K-means clustering methods on PIDD. The dataset has some features with missing values which were replaced with median values. They implemented the principal component analysis (PCA) method to extract symbolic features from a dataset. The veracity of the RF approach was 74.7%, the ANN method was 75.7%, and the K-means clustering method was 73.6%. Other approaches are outperformed by ANN. Pradhan *et al.* [17] explored support vector machine (SVM), K-nearest neighbor (KNN), naïve Bayes (NB), LR, AdaBoost (AB), and DT techniques for diabetes prediction using PIDD. In addition to having the highest accuracy and F1-score, the SVM also showed the lowest log loss and increased recall. Tripathi and Kumar [18] considered linear discriminant analysis (LDA), KNN, SVM, and RF techniques for predicting diabetes early using the PIDD, RF demonstrated the highest accuracy (87.66%) among the classifiers evaluated. Hasan *et al.* [19] utilized a range of ML techniques, including KNN, DT, RF, AB, NB, extreme gradient boosting (XGB), and multilayer perceptron (MLP), for diabetes prediction using the PIDD dataset. They adopted a mean imputation to handle the missing values. Their study demonstrated that combining two boosted classifiers, AB and XGB yields the most accurate diabetes forecasts.

Ramesh *et al.* [20] utilized various ML techniques, including Gaussian naive Bayes (GNB), SVM, KNN, and LR for predicting DM on PIDD. By employing three distinct feature selection methodologies along with the synthetic minority oversampling technique (SMOTE) to address dataset imbalance, SVM emerged as the top-performing model. Through ten-fold stratified cross-validation, SVM demonstrated superior accuracy (83.20%), recall (87.20%), and specificity (79%). Azad *et al.* [21] introduced a DM classification framework that leveraged SMOTE, genetic algorithm (GA), and DT for PIDD. They demonstrated that the proposed approach achieved notable performance metrics: an accuracy of 82.1256, precision of 0.8070, sensitivity of 0.8598, F1_measure of 0.8326, and AUC of 0.8511. Khaleel and Bakry [22] employed KNN, NB, and LR techniques to diagnose diabetes using the PIDD. LR exhibited a superior precision of 94%. Joshi and Dhakal [23] utilized LR and DT to predict diabetes, employing the PIDD. Feature selection was conducted using classification trees, resulting in an accuracy of 78.26%. Barik *et al.*

[24] recommended RF and XGB for diabetes prediction using the PIDD. They yield an accuracy of 74% and 71% for XGB and RF, respectively. Nagabushanam *et al.* [25] proposed a convolutional neural network (CNN) model for diabetes prediction, leveraging the PIDD. The model achieved an accuracy of 77.98% by employing convolutional and pooling layers for feature extraction along with fully connected layers for classification.

Khanam and Foo [26] conducted a study employing various ML algorithms, including DT, KNN, RF, NB, AB, LR, SVM, and NN, for predicting diabetes on PIDD. Their analysis revealed that LR and SVM yielded superior performance in diabetes prediction. Additionally, they constructed a NN model with different configurations of hidden layers and epochs, achieving an impressive accuracy of 88.6% when utilizing two hidden layers. Rajendra and Latifi [27] developed a hybrid approach utilizing multiple ML techniques, including DT, NB, KNN, and LR, to predict diabetes on two distinct datasets: PIDD and another dataset sourced from Vanderbilt. They employed two distinct feature selection methods for their analysis. The highest accuracy achieved was approximately 78% for dataset 1 and 93% for dataset 2. Tan *et al.* [28] present a hybrid model for diabetes prediction using desensitization data sourced from Qingdao. Their model integrates GA for attribute selection, a combination of DT, CNN, and SVM as primary learners, and a fully connected layer serving as the meta-learner for prediction. They conducted a comparative analysis of their GA-stacking approach against other methods, including KNN, SVM, LR, NB, and CNN. The results indicate that the GA-stacking model achieves higher prediction accuracy than the alternatives. Ahmed *et al.* [29] integrated two ML techniques, SVM and ANN, to predict DM. The outcomes from the SVM and ANN models were utilized as the input membership functions for the fuzzy technique. Using fuzzy logic, they managed to ascertain whether the DM inference yielded an optimistic or adverse result. The fused ML technique achieved an impressive prognostic efficiency of 94.87%.

Patil *et al.* [30] employed Mayfly optimization and SVM for DM prediction on PIDD and data collected from local hospitals. Mayfly was used to identify significant factors, SMOTE for class balance, and SVM for prediction. The expected model attained an accuracy of 94.5% when tested on a PIDD. Chang *et al.* [31] employed three distinct ML techniques such as NB, RF, and DT to predict DM on PIDD. They addressed missing data by replacing it with medians and explored three feature-selection methods. Among these approaches, RF demonstrated exceptional performance, with accuracy (79.57%), precision (89.40%), specificity (75.00%), F1 (85.17%), and AUC (86.24%). However, among the three techniques, the J48 DT showed the highest recall (88.43 %). Kibria *et al.* [32] proposed a hybrid approach for DM, integrating multiple ML techniques, including ANN, RF, SVM, LR, AB, and XGB on PIDD. They addressed missing values and data imbalance through median imputation and the SMOTETomek technique, achieving an impressive accuracy rate of 90% and an F1-score of 89%. Rastogi and Bansal [33] applied various ML techniques including RF, SVM, LR, and NB to predict Diabetes using PIDD data. Their study revealed that LR surpassed other methods, achieving an accuracy of 82.46% and recall of 68.23%. Kangra and Singh [34] employed a range of ML algorithms, such as SVM, NB, KNN, RF, LR, and DT, to analyze datasets pertaining to PIDD and Germany. Their study revealed that SVM yielded the highest accuracy of 74% for the PIDD, whereas KNN and RF outperformed the others with an accuracy of 98.7% for the German dataset. Tasin *et al.* [35] utilized DT SVM, RF, LR, KNN, and different ensemble methods to analyze the PIDD and Rownak Textile Mills Ltd (RTML) dataset for DM prediction. The XGB method with the Adaptive Synthetic Sampling (ADASYN) demonstrated an accuracy of 81%, an F1-score of 0.81, and an AUC of 0.84. Sai *et al.* [36] devised a hybrid method for predicting diabetes, merging KNN, NB, RF, AB, and light gradient boosting machine (LightGBM). Feature analysis involves evaluating the permutation importance of each feature. Integrating LightGBM, KNN, and AB yielded impressive metrics: accuracy (90.76%), F1-score (86.63), precision (87.45), and recall (85.82).

Previous studies in the field of DM prediction have predominantly focused on the PIDD which leaves behind important factors which are introduced in this study. These factors are determined from NHANES 2011-2012 by implementing an ensemble technique for feature selection. Furthermore, a significant gap exists in the literature regarding the comprehensive pre-processing of this dataset. Although some studies have explored predictive models using ML algorithms, they overlook critical preprocessing steps, such as handling missing values, outliers, data balancing, and feature selection. This study aims to bridge this gap by addressing these preprocessing challenges and evaluating the impact of hyperparameter optimization techniques on model performance, thereby enhancing the accuracy and generalizability of predictive models for DM. This study is concerned with the following research questions:

a) Can ML techniques be further optimized to achieve results that are superior to already existing techniques?
b) Can significant factors that were not determined in earlier studies be identified, and what ensemble techniques are of great applicability for the same?

c) What implementations can be made in various stages of data preprocessing to maximize performance metrics?

3. MATERIAL AND METHOD

3.1. Proposed framework

This study uses NHANES 2011-2012 dataset which is an open-source dataset consisting of 6,561 samples out of which 657 are diabetic and 5,903 are non-diabetic. In this proposed framework, the preprocessing of raw data stands as a fundamental step within the proposed pipeline. The framework involves several steps for preprocessing data, including filling in missing values, outlier replacement, feature selection of the attributes, and data balancing. This is crucial because the classifiers' ability to learn effectively directly relies on the quality of the data. The steps carried out for the progress of ML methods are shown in Figure 1. Steps for carrying out the diabetes mellitus prediction*:*

Step 1: Import the NHANES 2011-2012 dataset into Jupyter Notebook (Python 3.8.5).
Step 2: Preprocess the dataset to handle missing values using the KNN technique.
Step 3: Identification of outliers using local outlier factor and replacement by median imputation.
Step 4: Resample the dataset using SMOTE to balance the diabetic and non-diabetic samples.
Step 5: Scale the features using the Standard Scaler for uniform scaling.
Step 6: Identify significant factors using the XGBoost classifier.
Step 8: Hyperparameter tuning for all models using grid search.
Step 7: Develop the hyperparameter-optimized SVM, MLP, and bagging ML models for DM prediction.
Step 9: The model performance was evaluated using metrics such as accuracy, recall, precision, F1-score, and AUC.
Step 10: The best model is selected based on the highest performance metric scores.

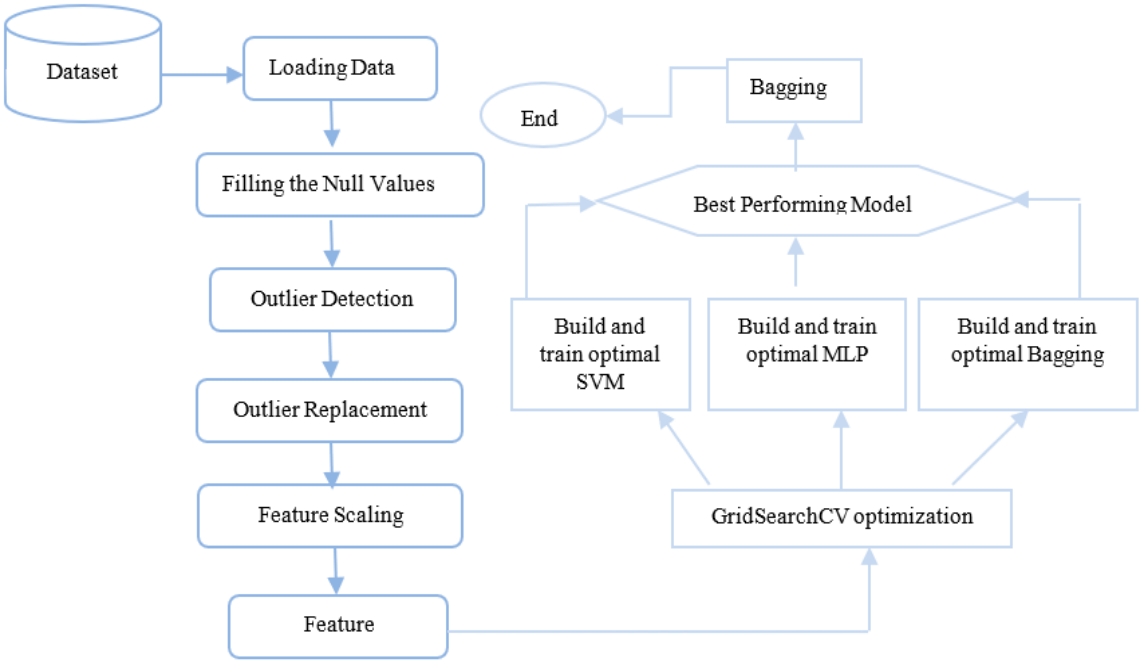


Figure 1. Graphical portrayal of steps executed

3.1.1. K-nearest neighbor

A trivial dilemma in medical datasets is the presence of a significant number of null values which can obscure inclination or even block certain downstream searches. To resolve the issue of missing values, the most popular approach used is KNN [37]. The KNN algorithm is a highly adaptable and non-parametric approach utilized in ML for classification and regression tasks. Unlike other algorithms, KNN does not rely on assumptions about the distribution of the data, which makes it particularly suitable for various data types. The core principle of KNN imputation involves identifying the k-nearest neighbors for missing values and

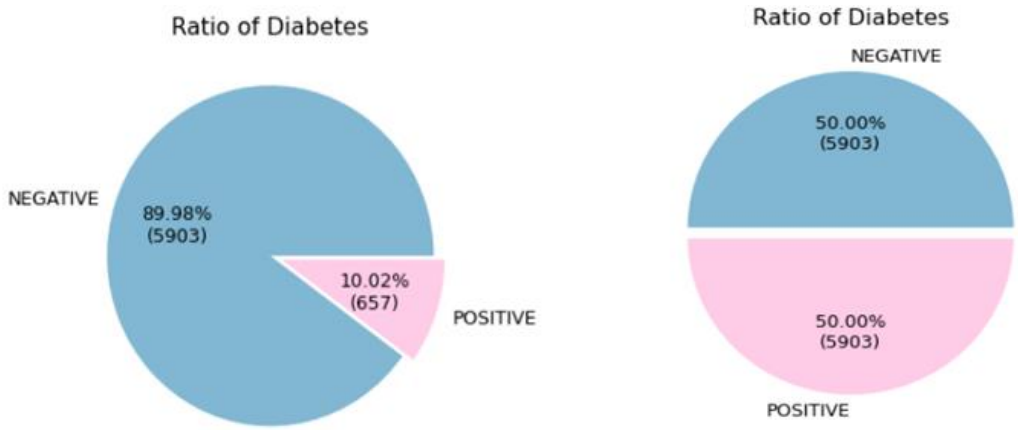
subsequently filling those values by averaging the data of these neighbors. As a result, this imputation method not only enhances the quality of the data but also improves the performance of the model [38].

3.1.2. Local outlier factor

The local outlier factor (LOF) is an outlier detection algorithm known for its robustness and efficacy in pinpointing data points that exhibit noticeable deviations from their immediate surroundings. This algorithm proves particularly advantageous in ML applications, wherein outliers can significantly impact the overall performance of models. LOF's effectiveness lies in its capacity to evaluate the local density of data points and subsequently assign an outlier score to each point based on its relative density compared to neighboring points. By assigning an outlier score to every data point, LOF provides a relative measure quantifying its divergence from the surrounding data. This score enables the ranking of outliers and the selection of those that are most likely to represent anomalies [39]. Next, median imputation is used to operate on the outliers [40].

3.1.3. Data sampling

Dealt with the extant outliers, the balancing of the dataset comes next in the process. Balancing data becomes necessary due to existing instances of the performance of the learning model being affected due to a disparity in training data of one class from another [41]. SMOTE is a popular and effective technique for handling class imbalance, particularly when dealing with high-dimensional datasets. It has been shown to improve the performance of ML models on imbalanced datasets, especially for classification tasks. SMOTE helps reduce bias towards the majority class by balancing the class distribution, making the model less likely to favor the majority class [42]. The distribution of the dataset before and after applying SMOTE is shown in Figure 2 with Figure 2(a) showing data distribution before data balancing and Figure 2(b) showing data distribution after balancing using SMOTE.



(a) (b)

Figure 2. Distribution of dataset (a) data distribution before balancing and (b) data distribution after balancing using SMOTE

3.1.4. Feature scaling

Feature scaling is the procedure for transforming data to a standardized scale, which is crucial to ensuring equitable significance among all features and avoiding bias towards features with higher values by the optimizer. The dataset had distinct characters with various measurement scales, necessitating the scaling of features, for which the StandardScaler was utilized. StandardScaler makes features look like standard normally distributed data with mean zero and unit variance [43].

3.1.5. Feature selection

Extreme gradient boosting (XGB) was used to evaluate the feature's importance. XGB is a potent ML algorithm that excels in many different tasks, including feature selection. XGB feature importance is less sensitive to noisy features compared to traditional methods like correlation analysis. XGBoost captures non-linear relationships between features, making it effective in identifying complex feature interactions [44]. The feature importance produced by XGB is shown in Figure 3.

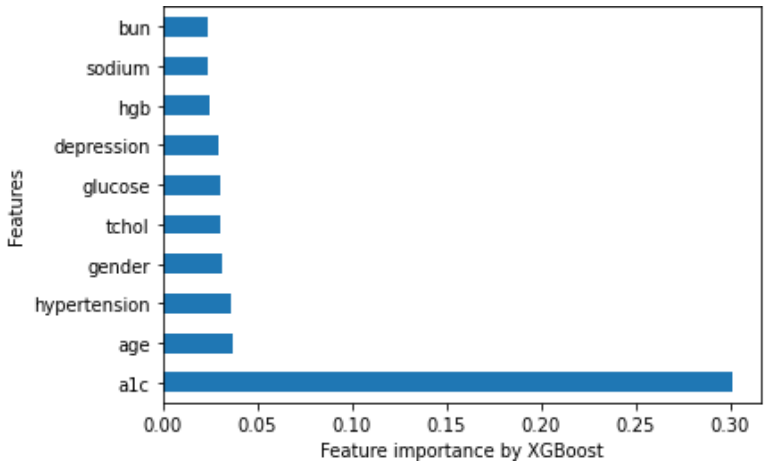


Figure 3. Feature selection with XGBoost

3.2. ML methods

3.2.1. Support vector machine

The SVM is a supervised ML approach that was created from the idea of structure minimization by Vapnik 1998 [45]. It is a fairly new and promising classification technique. To optimize the margin and obtain strong generalization capabilities, this program makes use of a separating hyperplane. The following is how the SVM's decision function is represented concerning the binary classification and regression problem.

$$f(U) = (w, \varphi(u)) + C \tag{1}$$

where $\varphi(U)$ represents a mapping of diabetes sample U from the instruction field to a high-spatial component field; the optimal values of w and b are captured by correcting the following inflation task:

$$Minimize\ h(V, \Psi) = ||V||^2 + d\sum_{k=1}^{M} \Psi_k \tag{2}$$

Subject to the following restraint equation

$$y_k((w, \varphi(U_k)) + +C) \geq 1 - \Psi_{k,}\Psi_k \geq 0 \tag{3}$$

where Ψ_k represents the sloppy variable, $d(> 0)$ represent the regularization variable of the flaws. A kernel function is calculated by (4).

$$k(U_k U_l) = (\varphi(U_k), \varphi(U_l)) \tag{4}$$

$k(.,)$ represents a kernel function and U_k represents the drill pattern vectors. There are four types of kernels in SVM which include linear, polynomial, radial basis function (RBF), and sigmoid. In non-linear classification, RBF-SVM typically produces good performance to other kernel functions [46], [47].

3.2.2. Multi-layer perceptron

One of the primary sub-divisions of feed-forward ANN is MLP. At least three node levels make up an MLP i.e.; an input layer, a hidden layer, and an output layer. MLP employs the supervised learning approach which includes the backpropagation strategy, for its learning [48], [49]. The following calculations are being performed at each neuron in the output and hidden layer.

$$o(U) = F(c(2) + W(2)l(U)) \tag{5}$$

$$l(U) = \varphi(U) = r(c(1) + W(1)U \tag{6}$$

where $c(1), c(2)$ are the bias, $W(1), W(2)$ represent the weights while F and r are the activation functions. The group of variables to be learned is the group $\varphi = \{W(1), c(1), W(2), c(2)\}$.Classic prime for r include tanh function with $\tanh(b) = (e^b - e^{-b})/(e^b - e^{-b})$ or the logistic sigmoid function, with $sigmoid(b) = 1/(1 + e^{-b})$.

This deep learning framework can identify data that cannot be separated linearly. When the data is separable linearly, every neuron uses a linear activation function, which will linearly design the input to the output. In case of data that is not separable linearly, the technique will use activation functions such as a logistic function or a sigmoid function [50].

3.2.3. Bagging

Breiman first suggested bagging in 1996 as a bootstrap aggregating prediction ensemble technique for ML. By mixing the classifications of hastily made training sets, bagging is used to improve generalization ability [51]. The bagging method's benefit is that it can lower the variance of the base methods and closely tune the estimate to the anticipated result, boosting the predictions' reliability [52]–[54]. Consider the following classification problem $U_k, V_k,\ k = 1,2,\dots,m$ where $U_k \in R^n$ denote the n-dimensional predictor variable and the target variable $V_k \in \{0,1,\dots,J-1\}$. The core responsibility for classification is $Q[V = j|U = u]$ $(j = 0,\dots J-1)$. The functional estimator is the outcome of a base process given by (7).

$$\hat{h}(.) = f_n\big((U_1, V_1),\dots,(U_m V_m)\big)(.) :\ R^n \to R \tag{7}$$

where $f_n(.)$ defines the estimated function of the data.

Steps for bagging algorithm:

Step 1: Create a bootstrap sample $(\mathrm{U}_1^*, \mathrm{V}_1^*)$,…,$(\mathrm{U}_\mathrm{n}^*, \mathrm{V}_\mathrm{n}^*)$ by randomly drawing n times with replacements from the data $(\mathrm{U}_1, \mathrm{V}_1),\dots,(\mathrm{U}_\mathrm{n}\mathrm{V}_\mathrm{n})$.

Step 2: Using the plug-in approach, the bootstrap estimator $\mathrm{h}^*(.)$ can be determined by (8).

$$\hat{h}^*(.) = f_n\big((U_1^*, V_1^*),\dots,(U_n^*, V_n^*)\big)(.) \tag{8}$$

Step 3: Rerun steps 1 and 2 N times, where N is usually chosen as 50 or 100 yields $\hat{h}^{*k}(.)$ k $(1,\dots,N)$. Mathematically, the value for the bagged estimation is determined by (9).

$$\hat{h}_{Bag}(.) = N^{-1}\sum_{k=1}^{N}\hat{h}^{*k}(.). \tag{9}$$

Theoretically speaking, the bagging estimator corresponds to the following.

$$\hat{h}_{Bag}(.) = D^*[\hat{h}^*(.)] \tag{10}$$

The theoretical batch in (2) coincides with $= \infty$: the definite number N in the rule carries out the veracity of the Monte Carlo approximation otherwise, it should not be recognized as an adjustment criterion for bagging. For classification, it is proposed to average the estimation possibilities $\hat{h}_j^{*k} = \hat{Q}^*[V^{*k} = j|U^{*k} =.]$, $(j = 0,\dots J-1)$ yields an estimator for $Q[V = j|U =.]$. Because it is sensitive to even small changes in the training data, bagging is an efficient strategy in vulnerability frameworks and may therefore be able to improve prediction accuracy [55].

3.3. Cross validation

Cross-validation is a statistical method utilized in ML and model training for evaluating the effectiveness and adaptability of a predictive model. The fundamental concept of cross-validation involves splitting the dataset into several subsets, commonly known as "folds". The model is trained on a selection of these folds, followed by assessing its performance on the remaining folds. This iterative process is repeated numerous times, and the average performance is considered as an estimation of the model's actual effectiveness. To assess the effectiveness of our suggested algorithm, we leverage the 10-fold cross-validation technique. This method offers an unbiased assessment of generalization error, particularly benefiting reliable classifiers [56].

3.4. Hyperparameter optimization

Efficient optimization of hyperparameters is imperative during the progress of ML models. The choice of hyperparameters can significantly impact the performance of an ML model. Tuning these hyperparameters allows you to find the configuration that leads to the best possible performance on your specific dataset. Properly tuned hyperparameters help in finding the right balance between overfitting, and underfitting and ensuring better generalization to new data. Grid search is used to do an extensive search for the best-adjusted model parameters based on predefined performance criteria, such as accuracy and mean error, or explicitly defined metrics [57].

3.5. Evaluation metrics

In this research work, Python 3.8.5 is used. Python is known for being simple to read and write, allowing developers to work faster and more efficiently integrate systems [58]. The tool used for coding is Jupyter Notebook. Jupyter is free open-source software that facilitates its users in creating documents using text equations and visualization elements [59]. To evaluate the efficacy of our developed methods, multiple metrics are utilized including accuracy, precision, recall, area under the receiver operating characteristic curve (AUC), and F1-score. These metrics decisively measure the performance of an algorithm in a given task, providing an objective assessment.

3.5.1. Accuracy

The classification accuracy serves as the core metric for assessing the categorization and prediction of samples. This measure gauges the proportion of accurate predictions among all the predictions made for the classified samples. To calculate this performance metric, the following formula can be employed:

$$Acc = TP + TN / TP + TN + FP + FN$$

In the realm of prediction accuracy evaluation, true positives (TP) denote the count of accurately predicted positive outcomes, whereas true negatives (TN) represent the quantity of correctly predicted negative outcomes. Conversely, false positives (FP) indicate the number of erroneously predicted positive outcomes, and false negatives (FN) refers to the total of inaccurately predicted negative outcomes.

3.5.2. Recall

This metric calculates the accuracy of positive predictions by dividing the number of true positive results by the number of actual positive instances. Recall measures the model's ability to identify all relevant instances. A high recall indicates that most actual positives are correctly identified.

$$Recall = TP/TP + FN$$

3.5.3. Precision

The true negative rate, also known as the specificity, represents the proportion of correctly identified negative cases to the overall number of actual negative cases. Precision gives insight into the accuracy of the positive predictions made by the model. A high precision indicates a low false positive rate. This calculation can be performed using the subsequent formula:

$$Precision = TN/TN + FP$$

3.5.4. F1-score

The F1-score is a commonly used metric in evaluating classification models, providing a balanced assessment of both recall and precision, and can be calculated.

$$F1 - score = 2 * (recall * precision)/(recall + precision)$$

Precision is how accurate the classifier was in classifying examples as positives, whereas recall measures how accurate the classifier was in classifying all of the examples that needed to be classed as positives. Precision and recall have the advantage of avoiding the problem that the accuracy metric has [60]. For class shifting, F1 varies and the number of correctly categorized negative samples has no bearing on the F1-score [61].

3.5.5. ROC/AUC

In ROC analysis, the performance of a binary classifier is depicted by altering its discrimination threshold. The true positive rate (TPR) and false positive rate (FPR) are plotted at different thresholds. The summary of classifier performance is determined by calculating the area under the ROC curve, commonly referred to as AUC. The classification results are plotted using an ROC curve, which goes from the most positive to the most negative classification. Changing the decision threshold of an algorithm is another way to build the ROC. The AUC defined by a single run is sometimes referred to as balanced accuracy [62].

4. RESULTS AND DISCUSSION

Using the NHANES 2011-2012 and during data analysis, it was found that there were some missing values and outliers in the dataset which were subsequently replaced by KNN, and outliers detected by boxplot and replaced using median values. In addition; diverse characteristics have varying scales of

quantification, which present a challenge for ML methods to achieve exact outcomes. To address this issue, two essential data processing techniques are employed: SMOTE for data balancing and StandardScaler for feature scaling. After balancing and scaling the dataset, the next crucial step was featuring selection, so XGB was used. The top variables that were identified as significant included A1C, age, sex, hypertension, total cholesterol, glucose, depression, hemoglobin, sodium, and blood urea nitrogen. Our analysis revealed a strong correlation between hemoglobin and sodium levels and the onset of DM. These features were used for the model development and the rest of the features were dropped. After completing the data-processing steps, the focus shifted toward developing and reviewing the performance of the SVM, MLP, and bagging models. To determine the best-performing model accuracy, AUC, precision, recall, and F1-score were used as performance metrics, and hyperparameter tuning was used to improve the efficiency of the developed models. After tuning the hyperparameters using a Grid Search, the optimal values were obtained for the SVM, MLP, and bagging models, as shown in Tables 1 to 3. The results indicated robust performance across all models in predicting the risk of DM, with accuracy ranging from 0.9242 to 0.9667, AUC scores ranging from 0.970 to 0.992, and recall from 0.93 to 0.97. Among these models, the bagging exhibited the highest performance, achieving an accuracy of 0.9667, an AUC score of 0.992, and a recall of 0.97. Furthermore, it showcased superior precision and F1-score compared with MLP and SVC as shown in Figures 4 to 6. Results of the three developed models after hyperparameter tuning.

Table 1. Optimal parameters of SVM classifier

Parameter	Value
Kernel	RBF
Gamma	0.1
Max_it	500
Degree	1
C	0.75

Table 2. Optimal parameters of MLP classifier

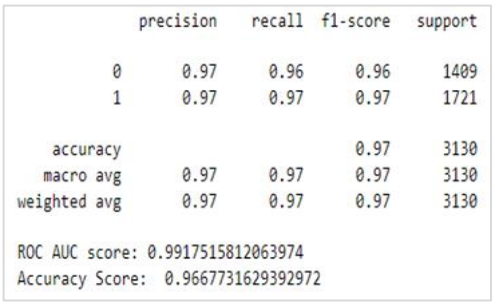
Parameter	Value
Hidden_size	100
Max_it	300
Solver	Adam
Alpha	0.1
Activation	Relu
Learning_rate	Constant

Table 3. Optimal parameters of bagging classifier

Parameter	Value
Max_features	1.0
Max_samples	1.0
N_estimators	100
Bootstrap	True

4.1. Discussion

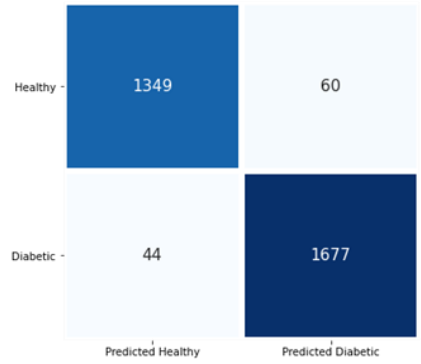
Several publications in the literature focus on identifying high-risk indicators as well as complex diabetic illness classification. Chang *et al.* [63] utilized different ML techniques including NB, LR, DT, KNN, and RF to predict the prognosis of diabetes using the behavioral risk factor surveillance system (BRFSS) dataset. The models were evaluated and compared based on metrics such as accuracy, precision, recall, and F1-score. Among these models, RF stood out as the most successful, achieving an accuracy rate of 82.26%. Ejiyi *et al.* [64] employed RF, AB, XGB, and extra tree (ET) for DM prediction on the PIDD. Subsequently, the Shapley additive explanation (SHAP) was used to identify the most significant features. Based on their findings, XGB and AB demonstrated superior performance compared with the other models, achieving an impressive accuracy of 94.67% and F1-scores of 95.27% and 95.95%. Bhat *et al.* [65] proposed a model for DM prediction using various ML techniques, such as DT, LR, and GBM, on PIDD. The authors demonstrated that DT outperformed other approaches with accuracy (91%), precision (96%), recall (92%), and F1-score (94%). Kumar *et al.* [66] devised a hybrid method that combined multiple ML techniques to investigate the prediction and characterization of DM on PIDD. The researchers demonstrated that the proposed model exhibited superior performance, achieving an accuracy of 95.26%, AUC of 91.15, sensitivity of 96.81, specificity of 97.72, and precision of 90.72.



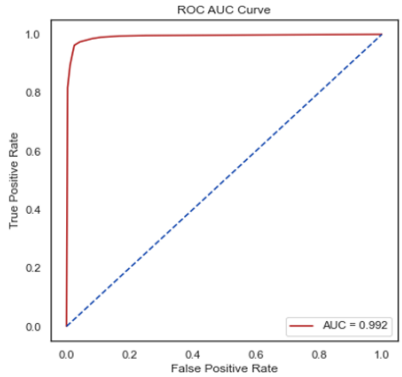
	precision	recall	f1-score	support
0	0.97	0.96	0.96	1409
1	0.97	0.97	0.97	1721
accuracy			0.97	3130
macro avg	0.97	0.97	0.97	3130
weighted avg	0.97	0.97	0.97	3130

ROC AUC score: 0.9917515812063974
Accuracy Score: 0.9667731629392972

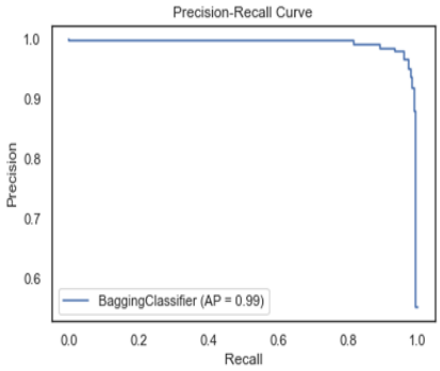
(a)



(b)

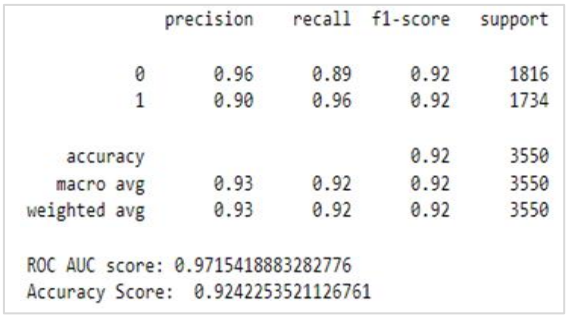


(c)



(d)

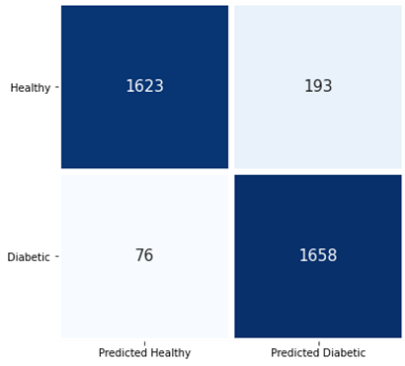
Figure 4. Overall performance of bagging classifier (a) results, (b) confusion matrix, (c) AUC and (d) precision-recall



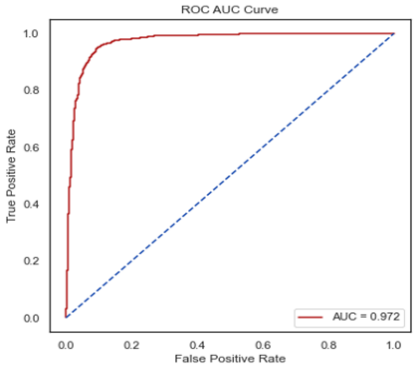
	precision	recall	f1-score	support
0	0.96	0.89	0.92	1816
1	0.90	0.96	0.92	1734
accuracy			0.92	3550
macro avg	0.93	0.92	0.92	3550
weighted avg	0.93	0.92	0.92	3550

ROC AUC score: 0.9715418883282776
Accuracy Score: 0.9242253521126761

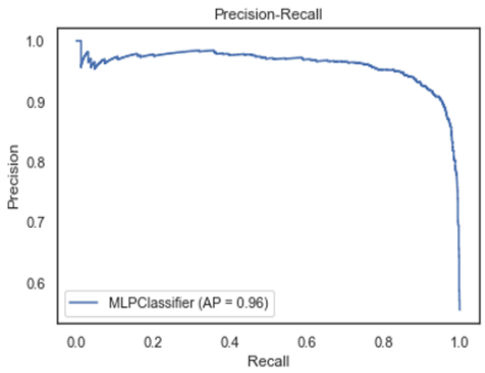
(a)



(b)



(c)



(d)

Figure 5. Overall performance of MLP classifier (a) results, (b) confusion matrix, (c) AUC, (d) precision-recall

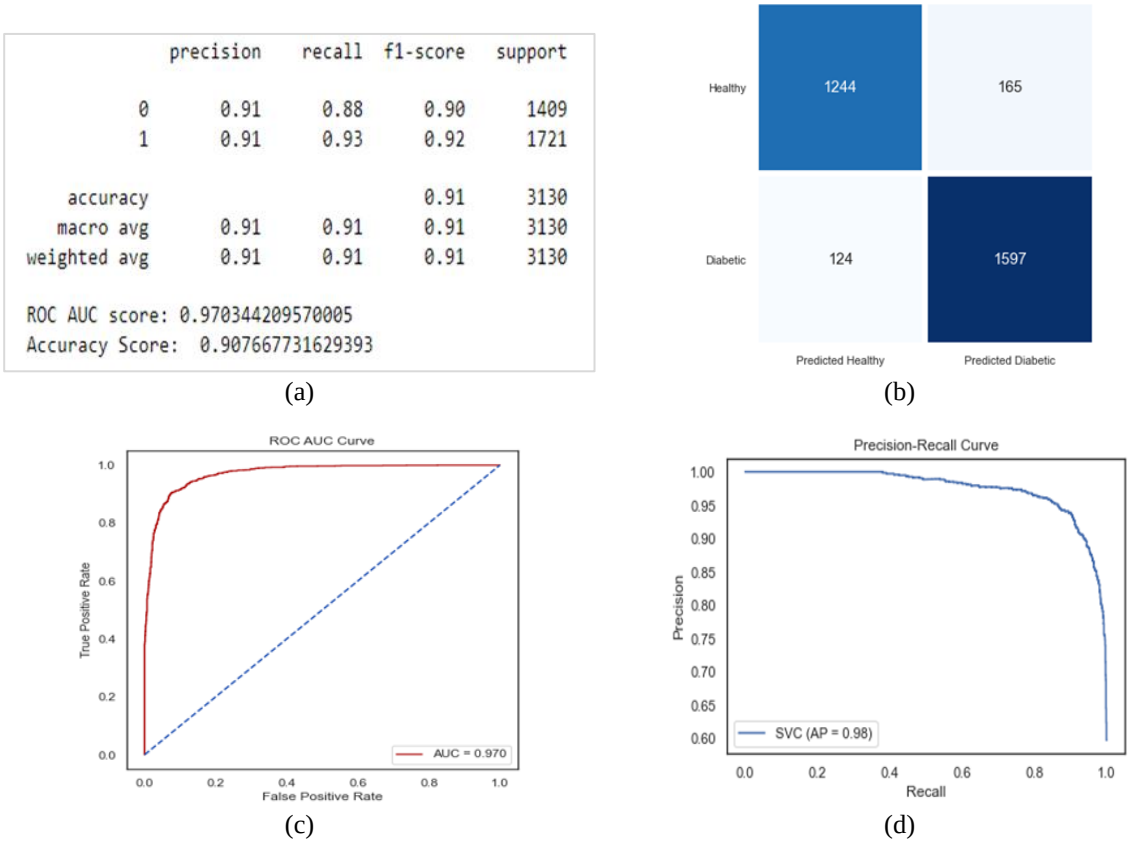


Figure 6. Overall performance of SVM classifier (a) results, (b) confusion matrix, (c) AUC, (d) precision-recall

However, in this study, the KNN technique was employed instead of the commonly used median technique to address missing values and to maintain the integrity of the data. Next, the focus on outliers, an often-overlooked aspect in research, is demonstrated through the use of the LOF to identify them and replace them with median values. This comprehensive approach to data pre-processing enhances the robustness of the analysis. Recognizing the importance of balancing the dataset, SMOTE was used to ensure accuracy and generalizability that addresses this issue, XGB was used to identify the most significant risk factors associated with DM and to tune the hyperparameters of the models by a grid search to mitigate overfitting. The results show that the bagging classifier achieved an accuracy of 0. 9667, a recall of 0.97, and an AUC of 0.992. Upon comparing our findings with those in the existing literature, our study demonstrates significantly improved performance in predicting the occurrence of DM among individuals with unique characteristics. While our study thoroughly investigated the association between newly identified factors and DM prediction, further research is necessary to validate our results, especially concerning the generalizability of our model across diverse populations and the impact of potential confounding variables. Additionally, different ensemble ML methods will be developed using advanced hyperparameter optimization methods to forecast DM.

5. CONCLUSION

This study employs three ML techniques, SVM, MLP, and bagging, for diabetes mellitus forecasting on the NHANES 2011-2012 dataset, where pre-processing is critical for accurate and reliable prognosis. The proposed pre-processing strategy increased the reliability of the dataset, with anomaly replacement, missing value filling, and feature selection being the top priorities. The overall variance of the characteristic distribution can be improved using such preprocessing. A 10-fold stratified cross-validation method was used to verify the stability of the SVM, MLP, and bagging. The proposed approach employs a grid search technique to determine the optimal parameters for all three developed methods. The Accuracy, AUC, and Recall are chosen as metrics to design a generalized algorithm because they give more importance to the classifier with a greater accuracy, AUC and recall. The results suggest that our proposed framework outperforms other existing paradigms in terms of Accuracy, AUC, and Recall, indicating a high tendency for diabetes mellitus forecasting using the NHANES dataset.

REFERENCES

[1] D. B, S. K, L. B, B. RS, and B. PS, "A modern review of diabetes mellitus: an annihilatory metabolic disorder," *Journal of In Silico & In Vitro Pharmacology*, vol. 03, no. 01, 2018, doi: 10.21767/2469-6692.100014.

[2] Y. Lin and Z. Sun, "Current views on type 2 diabetes," *Journal of Endocrinology*, vol. 204, no. 1, pp. 1–11, Jan. 2010, doi: 10.1677/JOE-09-0260.

[3] U. Kampmann, "Gestational diabetes: a clinical update," *World Journal of Diabetes*, vol. 6, no. 8, 2015, doi: 10.4239/wjd.v6.i8.1065.

[4] M. Panwar, A. Acharyya, R. A. Shafik, and D. Biswas, "K-nearest neighbor based methodology for accurate diagnosis of diabetes mellitus," in *2016 Sixth International Symposium on Embedded Computing and System Design (ISED)*, Dec. 2016, pp. 132–136, doi: 10.1109/ISED.2016.7977069.

[5] J. P. Kandhasamy and S. Balamurali, "Performance analysis of classifier models to predict diabetes mellitus," *Procedia Computer Science*, vol. 47, pp. 45–51, 2015, doi: 10.1016/j.procs.2015.03.182.

[6] O. Chandrakar and J. R. Saini, "Development of Indian weighted diabetic risk score (IWDRS) using machine learning techniques for type-2 diabetes," in *Proceedings of the 9th Annual ACM India Conference*, Oct. 2016, pp. 125–128, doi: 10.1145/2998476.2998497.

[7] R. Duggal, S. Shukla, S. Chandra, B. Shukla, and S. K. Khatri, "Predictive risk modelling for early hospital readmission of patients with diabetes in India," *International Journal of Diabetes in Developing Countries*, vol. 36, no. 4, pp. 519–528, Dec. 2016, doi: 10.1007/s13410-016-0511-8.

[8] M. Heydari, M. Teimouri, Z. Heshmati, and S. M. Alavinia, "Comparison of various classification algorithms in the diagnosis of type 2 diabetes in Iran," *International Journal of Diabetes in Developing Countries*, vol. 36, no. 2, pp. 167–173, Jun. 2016, doi: 10.1007/s13410-015-0374-4.

[9] 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, 2016, doi: 10.1016/j.procs.2016.04.016.

[10] M. Maniruzzaman *et al.*, "Comparative approaches for classification of diabetes mellitus data: machine learning paradigm," *Computer Methods and Programs in Biomedicine*, vol. 152, pp. 23–34, Dec. 2017, doi: 10.1016/j.cmpb.2017.09.004.

[11] N. Noori and A. Yassin, "Towards for designing intelligent health care system based on machine learning," *Iraqi Journal for Electrical and Electronic Engineering*, vol. 17, no. 2, pp. 120–128, Dec. 2021, doi: 10.37917/ijeee.17.2.14.

[12] D. Edelman, M. K. Olsen, T. K. Dudley, A. C. Harris, and E. Z. Oddone, "Utility of hemoglobin A1c in predicting diabetes risk," *Journal of General Internal Medicine*, vol. 19, no. 12, pp. 1175–1180, Dec. 2004, doi: 10.1111/j.1525-1497.2004.40178.x.

[13] B. Balkau *et al.*, "Predicting diabetes: clinical, biological, and genetic approaches," *Diabetes Care*, vol. 31, no. 10, pp. 2056–2061, Oct. 2008, doi: 10.2337/dc08-0368.

[14] Q. Zou, K. Qu, Y. Luo, D. Yin, Y. Ju, and H. Tang, "Predicting diabetes mellitus with machine learning techniques," *Frontiers in Genetics*, vol. 9, Nov. 2018, doi: 10.3389/fgene.2018.00515.

[15] H. Lai, H. Huang, K. Keshavjee, A. Guergachi, and X. Gao, "Predictive models for diabetes mellitus using machine learning techniques," *BMC Endocrine Disorders*, vol. 19, no. 1, Dec. 2019, doi: 10.1186/s12902-019-0436-6.

[16] T. M. Alam *et al.*, "A model for early prediction of diabetes," *Informatics in Medicine Unlocked*, vol. 16, 2019, doi: 10.1016/j.imu.2019.100204.

[17] R. Pradhan, M. Aggarwal, D. Maheshwari, A. Chaturvedi, and D. K. Sharma, "Diabetes mellitus prediction and classifier comparitive study," in *2020 International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC)*, Feb. 2020, pp. 133–139, doi: 10.1109/PARC49193.2020.236572.

[18] G. Tripathi and R. Kumar, "Early prediction of diabetes mellitus using machine learning," in *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, Jun. 2020, pp. 1009–1014, doi: 10.1109/ICRITO48877.2020.9197832.

[19] M. K. Hasan, M. A. Alam, D. Das, E. Hossain, and M. Hasan, "Diabetes prediction using ensembling of different machine learning classifiers," *IEEE Access*, vol. 8, pp. 76516–76531, 2020, doi: 10.1109/ACCESS.2020.2989857.

[20] J. Ramesh, R. Aburukba, and A. Sagahyroon, "A remote healthcare monitoring framework for diabetes prediction using machine learning," *Healthcare Technology Letters*, vol. 8, no. 3, pp. 45–57, Jun. 2021, doi: 10.1049/htl2.12010.

[21] C. Azad, B. Bhushan, R. Sharma, A. Shankar, K. K. Singh, and A. Khamparia, "Prediction model using SMOTE, genetic algorithm and decision tree (PMSGD) for classification of diabetes mellitus," *Multimedia Systems*, vol. 28, no. 4, pp. 1289–1307, Aug. 2022, doi: 10.1007/s00530-021-00817-2.

[22] F. Alaa Khaleel and A. M. Al-Bakry, "Diagnosis of diabetes using machine learning algorithms," *Materials Today: Proceedings*, vol. 80, pp. 3200–3203, 2023, doi: 10.1016/j.matpr.2021.07.196.

[23] R. D. Joshi and C. K. Dhakal, "Predicting Type 2 diabetes using logistic regression and machine learning approaches," *International Journal of Environmental Research and Public Health*, vol. 18, no. 14, Jul. 2021, doi: 10.3390/ijerph18147346.

[24] S. Barik, S. Mohanty, S. Mohanty, and D. Singh, "Analysis of prediction accuracy of diabetes using classifier and hybrid machine learning techniques," in *Smart innovation, systems and technologies*, 2021, pp. 399–409.

[25] P. Nagabushanam, N. C. Jayan, C. Antony Joel, and S. Radha, "CNN architecture for diabetes classification," in *2021 3rd International Conference on Signal Processing and Communication (ICPSC)*, May 2021, pp. 166–170, doi: 10.1109/ICSPC51351.2021.9451724.

[26] J. J. Khanam and S. Y. Foo, "A comparison of machine learning algorithms for diabetes prediction," *ICT Express*, vol. 7, no. 4, pp. 432–439, Dec. 2021, doi: 10.1016/j.icte.2021.02.004.

[27] P. Rajendra and S. Latifi, "Prediction of diabetes using logistic regression and ensemble techniques," *Computer Methods and Programs in Biomedicine Update*, vol. 1, 2021, doi: 10.1016/j.cmpbup.2021.100032.

[28] Y. Tan, H. Chen, J. Zhang, R. Tang, and P. Liu, "Early risk prediction of diabetes based on GA-stacking," *Applied Sciences*, vol. 12, no. 2, Jan. 2022, doi: 10.3390/app12020632.

[29] U. Ahmed *et al.*, "Prediction of diabetes empowered with fused machine learning," *IEEE Access*, vol. 10, pp. 8529–8538, 2022, doi: 10.1109/ACCESS.2022.3142097.

[30] R. Patil, S. Tamane, S. A. Rawandale, and K. Patil, "A modified mayfly-SVM approach for early detection of type 2 diabetes mellitus," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 1, pp. 524–533, Feb. 2022, doi: 10.11591/ijece.v12i1.pp524-533.

[31] V. Chang, J. Bailey, Q. A. Xu, and Z. Sun, "Pima Indians diabetes mellitus classification based on machine learning (ML) algorithms," *Neural Computing and Applications*, vol. 35, no. 22, pp. 16157–16173, Aug. 2023, doi: 10.1007/s00521-022-07049-z.

[32] H. B. Kibria, M. Nahiduzzaman, M. O. F. Goni, M. Ahsan, and J. Haider, "An ensemble approach for the prediction of diabetes mellitus using a soft voting classifier with an explainable AI," *Sensors*, vol. 22, no. 19, Sep. 2022, doi: 10.3390/s22197268.

[33] R. Rastogi and M. Bansal, "Diabetes prediction model using data mining techniques," *Measurement: Sensors*, vol. 25, Feb. 2023, doi: 10.1016/j.measen.2022.100605.
[34] K. Kangra and J. Singh, "Comparative analysis of predictive machine learning algorithms for diabetes mellitus," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 3, pp. 1728–1737, Jun. 2023, doi: 10.11591/eei.v12i3.4412.
[35] I. Tasin, T. U. Nabil, S. Islam, and R. Khan, "Diabetes prediction using machine learning and explainable AI techniques," *Healthcare Technology Letters*, vol. 10, no. 1–2, pp. 1–10, Feb. 2023, doi: 10.1049/htl2.12039.
[36] M. J. Sai, P. Chettri, R. Panigrahi, A. Garg, A. K. Bhoi, and P. Barsocchi, "An ensemble of light gradient boosting machine and adaptive boosting for prediction of type-2 diabetes," *International Journal of Computational Intelligence Systems*, vol. 16, no. 1, Feb. 2023, doi: 10.1007/s44196-023-00184-y.
[37] J. Y. Lee and M. P. Styczynski, "NS-kNN: a modified k-nearest neighbors approach for imputing metabolomics data," *Metabolomics*, vol. 14, no. 12, Dec. 2018, doi: 10.1007/s11306-018-1451-8.
[38] A. Jadhav, D. Pramod, and K. Ramanathan, "Comparison of performance of data imputation methods for numeric dataset," *Applied Artificial Intelligence*, vol. 33, no. 10, pp. 913–933, Aug. 2019, doi: 10.1080/08839514.2019.1637138.
[39] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, May 2000, pp. 93–104, doi: 10.1145/342009.335388.
[40] E. Acuña and C. Rodriguez, "The treatment of missing values and its effect on classifier accuracy," in *Classification, Clustering, and Data Mining Applications*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 639–647.
[41] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 20–29, Jun. 2004, doi: 10.1145/1007730.1007735.
[42] M. Ahsan, R. Gomes, and A. Denton, "SMOTE implementation on phishing data to enhance cybersecurity," in *2018 IEEE International Conference on Electro/Information Technology (EIT)*, May 2018, pp. 0531–0536, doi: 10.1109/EIT.2018.8500086.
[43] H. Dhahri, E. Al Maghayreh, A. Mahmood, W. Elkilani, and M. Faisal Nagi, "Automated breast cancer diagnosis based on machine learning algorithms," *Journal of Healthcare Engineering*, vol. 2019, pp. 1–11, Nov. 2019, doi: 10.1155/2019/4253641.
[44] H. Zheng, J. Yuan, and L. Chen, "Short-term load forecasting using EMD-LSTM neural networks with a XGboost algorithm for feature importance evaluation," *Energies*, vol. 10, no. 8, Aug. 2017, doi: 10.3390/en10081168.
[45] V. N. Vapnik, "Adaptive and learning systems for signal processing communications, and control," *Statistical learning theory*, 1998.
[46] H. R. Pourghasemi, A. G. Jirandeh, B. Pradhan, C. Xu, and C. Gokceoglu, "Landslide susceptibility mapping using support vector machine and GIS at the Golestan Province, Iran," *Journal of Earth System Science*, vol. 122, no. 2, pp. 349–369, Apr. 2013, doi: 10.1007/s12040-013-0282-2.
[47] B. T. Pham, B. Pradhan, D. Tien Bui, I. Prakash, and M. B. Dholakia, "A comparative study of different machine learning methods for landslide susceptibility assessment: A case study of Uttarakhand area (India)," *Environmental Modelling & Software*, vol. 84, pp. 240–250, Oct. 2016, doi: 10.1016/j.envsoft.2016.07.005.
[48] S. Abirami and P. Chitra, "Energy-efficient edge based real-time healthcare support system," in *Advances in Computers*, 2020, pp. 339–368.
[49] M. Zubair, J. Kim, and C. Yoon, "An automated ECG beat classification system using convolutional neural networks," in *2016 6th International Conference on IT Convergence and Security (ICITCS)*, Sep. 2016, pp. 1–5, doi: 10.1109/ICITCS.2016.7740310.
[50] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Is object localization for free? - Weakly-supervised learning with convolutional neural networks," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 685–694, doi: 10.1109/CVPR.2015.7298668.
[51] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, Aug. 1996, doi: 10.1007/BF00058655.
[52] P. Bühlmann, "Bagging, boosting and ensemble methods," in *Handbook of Computational Statistics*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 985–1022.
[53] A. Peters, T. Hothorn, and B. Lausen, "ipred: improved predictors," *R news*, vol. 2, no. 2, pp. 33–36, 2002.
[54] D. Tien Bui, T.-C. Ho, B. Pradhan, B.-T. Pham, V.-H. Nhu, and I. Revhaug, "GIS-based modeling of rainfall-induced landslides using data mining-based functional trees classifier with AdaBoost, bagging, and MultiBoost ensemble frameworks," *Environmental Earth Sciences*, vol. 75, no. 14, Jul. 2016, doi: 10.1007/s12665-016-5919-4.
[55] A. Shirzadi *et al.*, "Novel GIS based machine learning algorithms for shallow landslide susceptibility mapping," *Sensors*, vol. 18, no. 11, Nov. 2018, doi: 10.3390/s18113777.
[56] C. Bergmeir and J. M. Benítez, "On the use of cross-validation for time series predictor evaluation," *Information Sciences*, vol. 191, pp. 192–213, 2012, doi: 10.1016/j.ins.2011.12.028.
[57] M. Idhammad, K. Afdel, and M. Belouch, "Semi-supervised machine learning approach for DDoS detection," *Applied Intelligence*, vol. 48, no. 10, pp. 3193–3208, Oct. 2018, doi: 10.1007/s10489-018-1141-2.
[58] B. A. Malloy and J. F. Power, "An empirical analysis of the transition from Python 2 to Python 3," *Empirical Software Engineering*, vol. 24, no. 2, pp. 751–778, Apr. 2019, doi: 10.1007/s10664-018-9637-2.
[59] J. Wang, L. Li, and A. Zeller, "Better code, better sharing," in *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: New Ideas and Emerging Results*, Jun. 2020, pp. 53–56, doi: 10.1145/3377816.3381724.
[60] N. Japkowicz, "Why question machine learning evaluation methods? (an illustrative review of the shortcomings of current methods)," *AAAI Workshop Papers 2006*, 2006.
[61] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, no. 1, Dec. 2020, doi: 10.1186/s12864-019-6413-7.
[62] M. Sokolova, N. Japkowicz, and S. Szpakowicz, "Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation," *In: Sattar, A., Kang, Bh. (eds) AI 2006: Advances in Artificial Intelligence. AI 2006. Lecture Notes in Computer Science, vol 4304*. Springer, Berlin, Heidelberg. 2006, pp. 1015–1021, doi: 10.1007/11941439_114.
[63] V. Chang, M. A. Ganatra, K. Hall, L. Golightly, and Q. A. Xu, "An assessment of machine learning models and algorithms for early prediction and diagnosis of diabetes using health indicators," *Healthcare Analytics*, vol. 2, Nov. 2022, doi: 10.1016/j.health.2022.100118.
[64] C. J. Ejiyi *et al.*, "A robust predictive diagnosis model for diabetes mellitus using Shapley-incorporated machine learning algorithms," *Healthcare Analytics*, vol. 3, Nov. 2023, doi: 10.1016/j.health.2023.100166.
[65] S. S. Bhat, M. Banu, G. A. Ansari, and V. Selvam, "A risk assessment and prediction framework for diabetes mellitus using machine learning algorithms," *Healthcare Analytics*, vol. 4, Dec. 2023, doi: 10.1016/j.health.2023.100273.
[66] A. Kumar, S. Bawa, and N. Kumar, "iDP: ML-driven diabetes prediction framework using deep-ensemble modeling," *Neural Computing and Applications*, vol. 36, no. 5, pp. 2525–2548, Feb. 2024, doi: 10.1007/s00521-023-09184-7.

BIOGRAPHIES OF AUTHORS

Nuzhat Ahmad Yatoo obtained a bachelor's degree in computer applications from the University of Madras. Subsequently, he pursued a master's degree in computer applications from the B.S. Abdur Rahman Crescent Institute of Science and Technology, Tamil Nadu. He is currently pursuing a Ph.D. in the Department of Computer Applications at B.S. Abdur Rahman Crescent Institute of Science and Technology. His research interests include healthcare, machine learning, deep learning, and statistical analysis. He can be contacted at email: nuzhat_ca_2019@crescent.education.

Ishok Sathik Ali is a professor of information technology at BSA Crescent Institute of Science and Technology, Chennai, India. He holds a Ph.D. in information and communication engineering from Anna University and has a strong research background in computer networks, cloud computing, and machine learning. He has published several papers in reputable journals and presented his work at national and international conferences. In addition to his research, he is dedicated to teaching and has taught various courses in computer science and engineering, information technology, and computer applications. He is passionate about mentoring students and fostering their interest in scientific explorations. He can be contacted at email: isathikali@crescent.education.

Imran Mirza is currently working as an assistant professor in the Department of Computer Science and Engineering at Poornima University in Jaipur. He received his Bachelor of Computer Applications from Barkatullah University, Bhopal, Master of Information Technology (MSc. IT) from the University of Madras, Chennai, and Ph.D. degrees from the B S. Abdur Rahman Crescent Institute of Science and Technology, Chennai, Tamil Nadu, India. He has published four research articles indexed in SCI, Web of Science, and Scopus, with reputed publishers, including Springer and Elsevier. He can be contacted at email: imran.mirza@poornima.edu.in.