Hindawi
Advances in Human-Computer Interaction
Volume 2022, Article ID 9220560, 14 pages
https://doi.org/10.1155/2022/9220560

Research Article

Diabetes Mellitus Disease Prediction Using Machine Learning Classifiers with Oversampling and Feature Augmentation

B. Shamreen Ahamed,[1] **Meenakshi S. Arya**,[2] **and Auxilia Osvin V. Nancy**[1]

[1]*College of Engineering and Technology, SRM Institute of Science and Technology, Vadapalani Campus, No. 1, Jawaharlal Nehru Road, Vadapalani, Chennai, Tamil Nadu, India*
[2]*MIT World Peace University, Pune, India*

Correspondence should be addressed to B. Shamreen Ahamed; shamu1502@gmail.com

Received 23 April 2022; Revised 27 June 2022; Accepted 24 August 2022; Published 19 September 2022

Academic Editor: Christos Troussas

The technical improvements in healthcare sector today have given rise to many new inventions in the field of artificial intelligence. Patterns for disease identification are carried out, and the onset of prediction of many diseases is detected. Diseases include diabetes mellitus disease, fatal heart diseases, and symptomatic cancer. There are many algorithms that have played a critical role in the prediction of diseases. This paper proposes an ML based approach for diabetes mellitus disease prediction. For diabetes prediction, many ML algorithms are compared and used in the proposed work, and finally the three ML classifiers providing the highest accuracy are determined: RF, GBM, and LGBM. The accuracy of prediction is obtained using two types of datasets. They are Pima Indians dataset and a curated dataset. The ML classifiers LGBM, GB, and RF are used to build a predictive model, and the accuracy of each classifier is noted and compared. In addition to the generalized prediction mechanism, the data augmentation technique is also used, and the final accuracy of prediction is obtained for the classifiers LGBM, GB, and RF. A comparative study and demonstration between augmentation and non-augmentation are also discussed for the two datasets used in order to further improve the performance accuracy for predicting diabetes disease.

1. Introduction

Diabetes mellitus (DM) is a multifactorial progress chronic metabolic disorder. It is a persistent ailment triggered by excessive sugar levels. In the year 2019, 463 million people were affected by diabetes mellitus. It is estimated that in the year 2030, 578 million people are likely to be affected by DM, and in the year 2045, the number of people affected will rise to 700 million as per a study conducted by researchers [1]. DM is also considered as an autoimmune disease; hence, one primary reason cannot be identified as a cause [2]. There could be many reasons, some of them are age, family history, insulin production, body mass index, stress, pregnancy, etc. Some of these attributes can be taken as attributes in the dataset, and a study can be conducted for predicting the disease based on them. These attributes play an important role by affecting a person's health. However, by using individual attributes and combination of these attributes [3], diabetes disease can be predicted using the many advanced technologies in machine learning that have been developed so far. By predicting the disease, a person can avoid developing further complications that the disease can lead to in the future [4].

1.1. Diabetes Is Majorly Classified into 4 Types

1.1.1. Type 1 Diabetes. It is also called juvenile diabetes. It is dependent on insulin. The immune system damages the insulin release cells, removing insulin production in the body. It occurs mostly in adolescence. The treatment aims at blood sugar level, insulin therapy, diet, and exercise. Some of the complications for type 1 include damage in the kidneys,

problems in pregnancy and skin, blood flow becoming weak, problems in skin [5].

1.1.2. Type 2 Diabetes. The human body's production of insulin is insufficient or it resists production of insulin. It is milder when compared to type 1 and can be treated using insulin therapy, medication, diet, and exercise. The complications it can cause include kidney, nerves, and eye problems; heart disease; and stroke. The diagnosis involves tests such as A1C test, FPG (fasting plasma glucose) test, and RPG (random plasma glucose) test. Other tests include oral glucose tolerance test (OGTT), glucose challenge test, fasting blood sugar (FBG) test, and random blood sugar (RBG) test [5].

1.1.3. Gestational Diabetes. It is diagnosed for the initial few days of pregnancy. Some of the complications include fluctuating blood pressure, difficulty in breathing, birth weight complexity, early birth, and diabetes in future. The tests for diabetes are glucose challenge test and glucose tolerance test. The treatment for diabetes includes diet, injections for insulin balance, exercise, and monitoring of blood glucose. The gestational diabetes is one type where the chances for the disease occurrence after birth are reduced or null in most cases. However, if not taken care of more, it may occur in future in the person earlier affected [6].

1.1.4. Prediabetes. It is also called impaired glucose tolerance. It is a case in which blood sugar is high compared to type 2 diabetes. The risk factors are type 2 diabetes and heart diseases like stroke. The glucose level is in the range of 100 to 125 mg/dL. The HbA1c range is 5.7% to 6.4%. Some of the conditions that occur during prediabetes are high BP, low HDL levels, high blood sugar levels, large waist size, and high triglycerides. By combining the three, a term known as metabolic syndrome is created [6].

2. Related Work

2.1. Literature Review. Many researchers have identified and developed work based on DM disease using ML algorithms and classifiers. Some of the work and techniques used are discussed below.

Kopitar et al. [7] have utilized the concept of artificial intelligence for improving the disease prediction accuracy. It consists of three steps: preprocessing, feature selection, and feature classification. Methods such as HSA (harmony search algorithm), GA (genetic algorithm), and PSO (particle swarm optimization) along with k-means clustering for selection of feature are adopted. KNN is used for classification procedures. The accuracy prediction evaluation metrics used include sensitivity, specificity, recall, and precision. An accuracy of 91.65% is obtained.

Deng et al. [8] have constructed a predictive model for diabetes prognosis using DT classifier and SMOTE. The system proposed contains 2 stages: In stage 1, data imbalance is removed using SMOTE. In stage 2, diagnosis of diabetes is made using DT classifier. The dataset is collected from a diagnostic lab in Kashmir Valley containing 734 entries. An accuracy of 94.7% is obtained.

Espino et al. [9] used the concepts of transfer learning and data augmentation to overcome challenges caused by datasets such as imbalanced datasets and small training datasets. Systematic examination of 3 neural network architectures, transfer learning strategies, and augmentation techniques and different loss functions including mixup and generative models are used. The same network architecture for type 1 diabetes using OhioT1DM public dataset is developed. The dataset is collected from Beth Israel Deaconess Medical Center and approved by the International Review Board. An accuracy of 95% is obtained.

Bavkar et al. [10] have designed a pipeline based model using deep learning (DL) techniques to predict diabetes. It includes data augmentation using a variable auto encoder (VAE), feature augmentation using sparse encoder (SAE), and a convolution neural network for classification. The dataset used is Pima dataset taken from UCI Repository. The accuracy obtained is 92.31% by using CNN classifier and training it along with SAE for feature augmentation in comparison with a well-balanced dataset.

Branimir et al. [11] proposed a system to overcome the following two challenges: heterogeneity regarding previous techniques and lack of transparency in features. It used the PRISMA methodology. 18 varieties of model comparison along with algorithms based on trees were performed. The authors concluded that KNN and SVM are majorly used for prediction.

Jyotismita et al. [12] have used the detection and analysis mechanism of diabetes disease using 6 facets, namely, dataset, processing methods, feature extraction, ML identification, and classification and diagnosis of DM, overcoming the flaws of classification. A comparison between various supervised, unsupervised, and clustering techniques is made. Different datasets possess different challenges and significant work that needs to be done to improve the efficiency of detection of various diabetic diseases.

Zhang et al. [13] have used the concept of data augmentation for overcoming the problem of insufficient data. The authors have proposed to use GMM to augment more data when the dataset is numeric. The dataset consists of 1157 samples. 5 regressors are used. They are linear regressor, decision tree regressor, random forest regressor, K-neighbors regressor, ridge regressor.

Safial et al. [14] have proposed a strategy for diabetes diagnosis through DL network by using 5-fold and 10-fold cross validation for training. The dataset used is Pima Indians dataset. The prediction accuracy is obtained as 98.35% when 10-fold cross validation is used.

Nur et al. [15] have focused mainly on data preprocessing which includes the following processes: removing the missing values, balancing the data process, and performing feature importance and data augmentation process. RF and LR are used for algorithm classification. The result obtained is 20% higher for precision and 24% higher for recall when compared to data without preprocessing.

3. Algorithms for Each Classifier

4. Implementation

Machine learning is such a field in which many classification and regression problems can be solved. ML based data-driven approaches have been used along with individual recorded data of each person/patient for implementation. The first step in implementation procedure is to select and identify the dataset. The selected dataset is then input into the predictive model in order to obtain the accuracy percentage. Predictive models are developed using different input parameters of the dataset combinations for diabetes accounting based on the correlation between the attributes used [16]. To obtain the outcome, a correlation matrix can be generated as shown in Figures 1 and2. The attributes producing maximum relevance are chosen and used for further processes. Following the correlation matrix, the data preprocessing is done in order to clean the data, avoid and ignore missing data, correct the incomplete data, etc. The data preprocessing is followed by feature extraction and feature selection [17]. The data are divided into training data (TrD) and testing data (TtD). The model is then built using the machine learning algorithms, and the prediction performance is assessed. The major reason for predicting the disease is to maintain a healthcare regime, create awareness, and prevent a person from further getting affected by other diseases that can be caused by diabetes mellitus.

The various processes are explained below.

4.1. Dataset. The first step in the prediction process is to identify the dataset. The features and attributes in the dataset play an important role in prediction. The datasets used are Pima Indians dataset [18] and a curated dataset [19]. The dataset consists of attributes such as age, blood pressure, insulin, glucose, pregnancy, diabetes pedigree function, skin thickness, outcome, and BMI. These attributes are considered for disease prediction by comparing them with the different classifiers.

In this study, two types of datasets are used as mentioned above. The first dataset is Pima Indians dataset taken from UCI Repository. It consists of 9 attributes: age (A), glucose (G), insulin (I), blood pressure (BP), diabetic pedigree function (DPF), BMI, pregnancy (P), skin thickness (ST), and outcome (O). The second dataset is taken from a survey conducted independently among known sources containing the same 9 attributes mentioned above. This dataset is called the DMS (Diabetes Mellitus Survey) dataset. The accuracy percentage is calculated using the various classifiers suitable for implementation.

The total number of entries in Pima Indians dataset is 768 with 500 negative instances and 268 positive instances. The number of entries in DMS dataset is 1110, the number of positive instances is 372, and the number of negative instances is 738.

The attributes used in the dataset are described in Figure 1, and the correlation of these attributes is taken into account to find out the attributes that play an important role in the disease prediction.

Figure 2 shows the attributes, and the highest relevance to the disease is taken and measured based on correlation matrix as shown in Figure 3 [32].

4.2. Correlation Matrix. Following the dataset selection procedure, a correlation matrix is presented in a diagrammatic form or table to display the relation between various variables or attributes used. The correlation can also be determined using the value of information gain calculated between the attributes. The information gain is used to measure and calculate the information received in bits about the class prediction that needs to be performed [20–22]. It majorly depends on feature distribution and corresponding class distribution. To calculate the information gain, the following formula can be used [23]:

$$IG\left(S_{X1}', x_{1i}\right) = H\left(S_{X1}'\right) - \sum_{v=\text{values}\left(x_{1i}\right)}^{\left|S_{x_{1i}=V}'\right|/S_{X1}'} H\left(S_{x_{1i}=V}'\right), \quad (1)$$

with entropy

$$H(S'1) = -p'1 + (S'1)\log_2\ p'1 + (S'1) - p'1 - (S'1)\log_2\ p'1\,(S\prime 1), \quad (2)$$

where S_X' is the set of training examples, x_{i1} is the vector of ith variable in the set, $|S_{x_{1i}=V}'|/|S_{X1}'|$ is the fraction of examples [19] of the ith variable having value v, and $p1$+(S1) and p1-(S1) are the probability of training sample in the set S1 of the positive and negative class [24].

The value between two attributes in the correlation matrix as shown in Figures 3 and4 denotes the correlation graph that determines how they are dependent on one another.

The information gain calculated for the attributes is shown in Table 1.

Table 1 indicates that glucose has the highest relevance of 47%. The next highest relevance is given by BMI (29%) and pregnancy (22%). Therefore, the attributes which have a threshold value of above 20% are considered the prime attributes. Glucose, BMI, and pregnancy are taken into consideration as the prime attributes as the threshold values of these attributes are more than 20%.

4.3. Data Preprocessing. The data preprocessing is one major step during implementation. It is the process of transforming the data before including it in the algorithm that is going to be used. It is a process of converting raw data into clean and processed data [25]. The purpose of data preprocessing is to obtain a more clarified result in a specific format. It can also be used to modify the data in such a way that more than one type of algorithm can be used during processing and implementation. The steps in data preprocessing include the following [26].

Invalue**:** Data used for n-dimension, X1 $\in$ R1n1 consisting of threshold and samples with variance
Outvalue: k-dimensional data that is reduced, Y1 ϵ R1k1
(1) Given X1 ϵ R1n1 and obtain the mean,
$\overline{X1} = 1/N \sum_{i=1}^{N} X1_{i1}$
where $\overline{X1} \in$ R1n1
(2) Covariance matrix, $n1 \times n1$,
$C1_{n1\times n1} = \sum_{i1=1}^{N} (X1_{i1} - \overline{X1})(X1_{i1} - \overline{X1})^T$
(3) Decomposition of eigenvalue: $C1_{n1\times n1}$ given as P1DP−1, where P1 ϵ R1n1 is the eigenvector matrix and $D1_{n1\times n1}$ denotes the diagonal eigenvalues
(4) The eigenvectors are then sorted in a descending order to select first k1 eigenvectors that is given as
variance $\geq$ *Tvariance*
$W_{n1\times k1}$
(5) The data X1 is given into a k-dimension by Y1 = W^TX1, where Y1 ϵ R1k1.

Algorithm 1: Feature selection using PCA.

Invalue: n-dimensional data (original), X1 ϵ R1n1
Outvalue: k-dimensional data (reduced), Y1 ϵ R1k1
(1) The non-quadratic function is set and considered as nonlinear function and G1 is assumed as negentropy.
(2) Given W1 of $W1 \times H1 = X1$, where W1, H1, and X1, during mixing, are the source ratios
Consisting of multiple components, where the output is mixed separately.
(3) Obtain PCA on X1 by $X1 =$ PCA(X1)
(4) **while** W changes **do**
(5) $W1 =$ mean (X1 $\times$ G1(W1 $\cdot$ X1)) − mean (G0 (W1T1 $\cdot$ X1)),
(6) $W1 =$ orthogonalize (W1)
(7) Execute $Y1 =$ W1 $\cdot$ X1, where Y1 ϵ R1k1.

Algorithm 2: Feature selection based on ICA.

Invalue: n-dimensional data (original), $X1 \epsilon R1n$ and expected outvalue, $Y1T \in R1$
Outvalue: k-dimensional data (reduced), $Y1 \epsilon Rk1$
(1) for $i \leq n$ do
(2) $r1_{iT} = \sum (X1_i - \overline{X1_i})(Y1_T - \overline{Y1}_T)/\sqrt{\sum (X1_i - \overline{X1_i})^2 X \sqrt{(Y1_T - \overline{Y}_{1T})^2}}$
(3) Sort the correlation $r1_{iT}$ in a descending order and choose first $k1$ features for $Y1 \epsilon R1k1$.

Algorithm 3: Feature selection based on correlation.

Invalue: Value that is n-dimensional, $X1 \epsilon R1n1$ and outvalue (target), $Y1 \epsilon R1$
Outvalue: The pp, $P1 \in [0, 1]$ of test data (unseen), x,
$\sum_{i=1}^{C1} P1_i = 1$, $C1 = 2$ (diabetes present ($C1$) or not ($C2$))
(1) The geometric distances are calculated,
$D1_{h1} = \sum_{i1=1}^{k1} |X1_{i1} - x_{i1}|^{q1^{1/q1}}$
$D1h1$ for $k1$ query points, where $X1i1 =$ current instance, $x1i1 =$ query instance, $q1 =$ order
(2) Establish set $S1$ with $k1$ points (closest)
(3) Estimate the pp, $P1$ for each class
$P1(C1 = j1|X1 = x1) = 1/K1 \sum_{i1\varepsilon S1} f1(C1_{i1} = j1)$
$f1(x1)$ is the function to class assignment.
pp means posterior probability.

Algorithm 4: K-nearest neighbor (KNN).

Input: data (n-dimensional), X1 ϵ R1n1 and outvalue (target), Y1 ϵ R1
Output: The pp, P1 ϵ [0, 1] of test data (unseen), x,
$\sum_{i1=1}^{C11} P11_{i1} = 1$, $C1 = 2$. (diabetes in (C11) or not (C12))
(1) Divide θ = (j1, tm1) into Q'$_{\text{left}}$ (θ) and Q'$_{\text{right}}$ (θ) subsets; θ contains feature, j1, threshold, tm1
(2) Calculate the kth node using an impurity(i) function (H1),
$G1 = (Q\prime 1, \theta) = n_{\text{left}}/N_{m1} H1(Q_{\text{left}}'(\theta)) + n_{\text{right}}/N_{m1} H1(Q_{\text{right}}'(\theta))$
$H1 = \sum_{C1} P1_{mC1} \times (1 - P1_{mC1})$ (OR)
$H1 = -P1_{mC1} \times \log(P1_{mC1})$AND
$P1_{mC1} = 1/N_{m1} \sum_{x_{i1\epsilon} R1_{m1}} I1(y_{i1} = c1)$
(3) Reduce the impurity(i) by selecting the right parameters, $\theta* = \text{argmin}\ \theta\ \text{G1}(\text{Q}'1, \theta)$
(4) Repeat the processes for subsets
$Q_{\text{left}}'(\theta*)$ and $Q_{\text{right}}'(\theta*)$ until depth reaches $\text{N}_{\text{m1}} < \text{min samples}$ or $\text{N}_{\text{m1}} = 1$.

Algorithm 5: Decision tree (DT).

Invalue: data (n-dimensional), $X1 \epsilon R1^{n1}$ with N samples and outvalue (target),
$Y1 \epsilon R1$
Outvalue: The pp, $P1 \epsilon [0,1]$ of unseen test data, $x1$, where
$\sum_{i1=1}^{C1} P1_{i1} = 1$, $C1 = 2$ (diabetes in ($C1$) or not ($C2$))
(1) Initiate the sample weight, $D1(i1) = 1/N$, $i1 = 1, 2,...,N$.
(2) for $t1 \leq T1$ ($n\_Classifiers$) do
(3) Weak_learner_training by using distribution $D1_{t1}$.
(4) Select a hypothesis (weak), $h1_{t1}$: $R1^{n1} \longrightarrow R1$ with low weight error,
$\_x000F\_t1 = D1_{t1}\ [h1_{t1}\ (x1_{i1})\ 6 = Y1\ ]$
(5) Choose $\alpha 1_{t1} = 1/2 \ln(1 - \varepsilon_{t1}/\varepsilon_{t1})$ and update $D1_{t1+1}(i1) = D1_{t1}(i1) e^{-\alpha 1_{t1} Y1 h1 t1 (x1_{i1})}/z1_{t1}$ where $i1 = 1,...,N$ and $z1t1$ is the normalization factor.
(6) Output pp: $P1(x1) = \text{sign}(\sum_{t1=1}^{T1} a1_{t1} h1_{t1}(x1))$.

Algorithm 6: AdaBoost (AB).

Invalue: data (n-dimensional), X1 ϵ R1$_{\text{n1,}}$ outvalue (target), Y1 ϵ R1
Outvalue: The pp, P1 ϵ [0, 1] of test data (unseen), x1, where
$\sum_{i1=1}^{C1} P1_{i1} = 1 C1 = 2$ (diabetes in (C1) or not (C2))
(1) for $b1 = 1$ to N ($n_$Bagging) do
(2) Design a sample (bootstrap) (X1$_{\text{b1}}$, Y1$_{\text{b1}}$) from given X1 ϵ R1$_{\text{n1}}$, Y1 ϵ R1
(3) Design an RF tree T1$_{\text{b1}}$ using X1$_{\text{b1}}$ and Y1$_{\text{b1}}$ by recursively repeating.
(4) The pp P1$^{\text{N}}_{\text{RF}}$ (x1) $\widehat{P}_{RF}^{N}(x1) = \text{Voting}\{\widehat{P}_{k1}(x1)\}_1^N$ where $\widehat{P}_{k1}(x1)$ is the prediction of the kth RF.

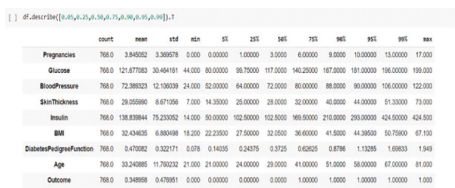
Algorithm 7: Random forest (RF).

Invalue: data (n-dimensional), $X1 \epsilon R1n1$ and outvalue (target), $Y1 \epsilon R1$
Outvalue: The pp, $P1 \epsilon [0, 1]$ of test data (unseen), x, where
$\sum_{i1=1}^{C1} P1_{i1} = 1$, $C1 = 2$ (diabetes present (C1) or not (C2))
(1) Assign the probabilities (prior) for each class,
P1(Y1 = C1) = N_{C1}/N and P1(Y1 = C2) = N_{C2}/N, where N determines the number of samples
(2) The output pp of class for the given predictor (attributes)
$P1(C_{i1}|X1) = P(X1|C_{i1}) \times P1(Y1 = C_{i1})/P1(X1)$
$P1(X1|Ci1)$ is the predictor (likelihood) for a given class and $P(X1)$ is the pp (prior).

Algorithm 8: Naive Bayes (NB).

Invalue: data (*n-dimensional*), $X1 \epsilon R1_{n1}$ and outvalue (target), $Y1 \epsilon R1$
Outvalue: The pp, $P1 \epsilon [0, 1]$ of test data (unseen), $x1$, where
$\sum_{i1=1}^{C1} P1_{i1} = 1$, $C1 = 2$ (diabetes in (C1) or not (C2))
(1) Initiate the model with fixed value:
$F1_0(x1) = \text{argmin}_{\gamma 1} \sum_{i1=1}^{N1} L1(Y1, \gamma 1)$,
L1(Y1,F1(x1)) is the loss functions and N denotes the number of samples
(2) **for** $m = 1$ *to* M (*n_Iterations*) **do**
(3) Calculate pseudo-residuals,
$r1_{im} = \delta L1(Y1, F1(X1_{i1}))/\delta F1(X1_{i1})$
where $i1 = 1, 2,...,N$
(4) Assign a base tree, $h1_{m1}$ using set (training) $(X1_{i1}, r1_{im})$ for $i1 = 1, 2,...,N$
(5) Multiplier $\gamma 1_{m1}$ is calculated by
$\gamma_m = \text{argmin}_{\gamma} \sum_{i=1}^{n} L1(Y1_i 1, F1_{m-1}(X1_{i1}) + \gamma h_m(X1_{i1}))$
(6) Update the model by
$F1_m(x1) = F1_{m-1}(x) + \gamma_m h_m(x1)$
(7) $F1_m(x1)$ is the desired pp, $P1 \epsilon [0, 1]$.

Algorithm 9: XGBoost (XB).



	count	mean	std	min	1%	25%	50%	75%	90%	95%	99%	max
Pregnancies	768.0	3.845052	3.369578	0.000	0.00000	1.00000	3.0000	6.00000	9.0000	10.00000	13.00000	17.000
Glucose	768.0	121.677083	30.464161	44.000	80.00000	99.75000	117.0000	140.25000	167.0000	181.00000	196.00000	199.000
BloodPressure	768.0	72.386323	12.106039	24.000	52.00000	64.00000	72.0000	80.00000	88.0000	90.00000	106.00000	122.000
SkinThickness	768.0	29.055990	8.671056	7.000	14.35000	25.00000	28.0000	32.00000	40.0000	44.00000	51.33000	73.000
Insulin	768.0	138.839844	75.233052	14.000	60.00000	102.50000	102.5000	169.50000	210.0000	293.00000	424.50000	424.500
BMI	768.0	32.434635	6.880498	18.200	22.23500	27.50000	32.0500	36.60000	41.5000	44.39500	50.75900	67.100
DiabetesPedigreeFunction	768.0	0.470082	0.322171	0.078	0.14035	0.24375	0.3725	0.62625	0.8786	1.13285	1.69833	1.949
Age	768.0	33.240885	11.760232	21.000	21.00000	24.00000	29.0000	41.00000	51.0000	58.00000	67.00000	81.000
Outcome	768.0	0.348958	0.476951	0.000	0.00000	0.00000	0.0000	1.00000	1.0000	1.00000	1.00000	1.000

Figure 1: Description of attributes.

4.3.1. Data Cleaning. The main use of data cleaning is to ignore the missing value and compute the value in a filled format. Binning of noisy data along with regression and clustering is done along with removing outlier values that cannot be grouped into clusters [27].

4.3.2. Data Integration. It is used to combine and merge multiple sources into single storage unit known as the data warehouse. It can include detection and data value conflict resolution, object matching, schema integration, and removing redundant attributes [28].

4.3.3. Data Transformation. It involves the process of grouping and consolidating the quality of data into alternate forms by structuring and formatting the data and changing values by processes such as aggregation, attribute selection, normalization, and generalization [29].

4.3.4. Data Reduction. The process is used to reduce the representation of data. Some of the strategies are numerosity reduction, attribute subset selection, discretization, data cube aggregation, data compression, and dimensionality reduction [30].

The outcome of the processes explained above is shown in Figure 5.

Figure 5 demonstrates the difference in the values before and after preprocessing. The values used in the dataset are cleaned, integrated, transformed, and reduced for further procedure.

4.4. Exploratory Data Analysis. Exploratory data analysis is the critical process of performing initial investigations on data to discover patterns and spot anomalies in order to test the hypothesis of summary statistics and graphical representations [31].

It is a method of analyzing the different kinds of data that are being used in the model. Specific statistical functions and techniques you can perform with EDA tools include clustering, univariate visualization, bivariate visualization, and multivariate visualization [32].

Libraries such as NumPy, Seaborn, and Matlab are some of the data visualization methods used in the model.

The histogram shown in Figure 6 is a graphing tool. It is used to consolidate discrete or continuous data values used to measure the interval scale regularly [33].

4.5. Algorithms in Machine Learning. There are various algorithms in machine learning that can be used for prediction concepts, including random forest, support vector machine, light gradient boosting machine, gradient boosting machine, decision tree, XGBoost, and logistic regression. In order to choose the highest accuracy producing algorithm, a model is built for the dataset taken, and each algorithm is applied to it. The classifier producing the highest value is then selected based on the results [34].

The model built during simulation for the proposed system generated the results shown in Figure 7.

From Figure 7, it can be seen that the algorithms LGBM, decision tree, and XGB produced the highest results comparatively.

The results obtained when the process was run again are shown in Figure 8.

From Figure 8, it can be seen that the algorithms LGBM, XGB, and decision tree still produced the highest results comparatively even when the process was executed again.

From the above simulation, the LGBM algorithm can be taken into consideration for actual model processing and execution as it produces the highest accuracy percentage.

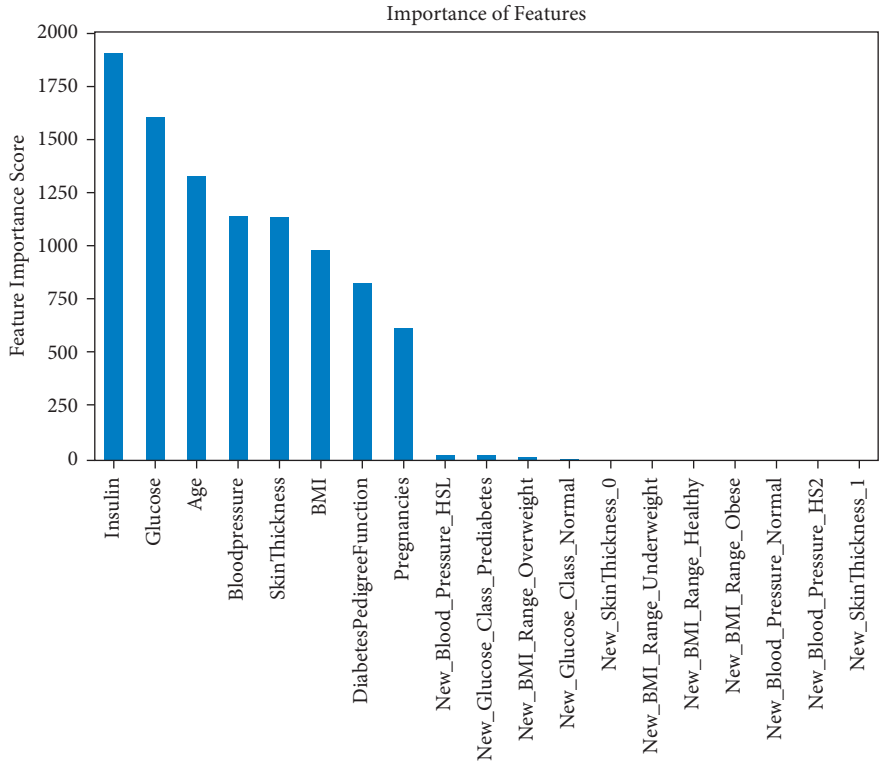


Figure 2: Attributes.

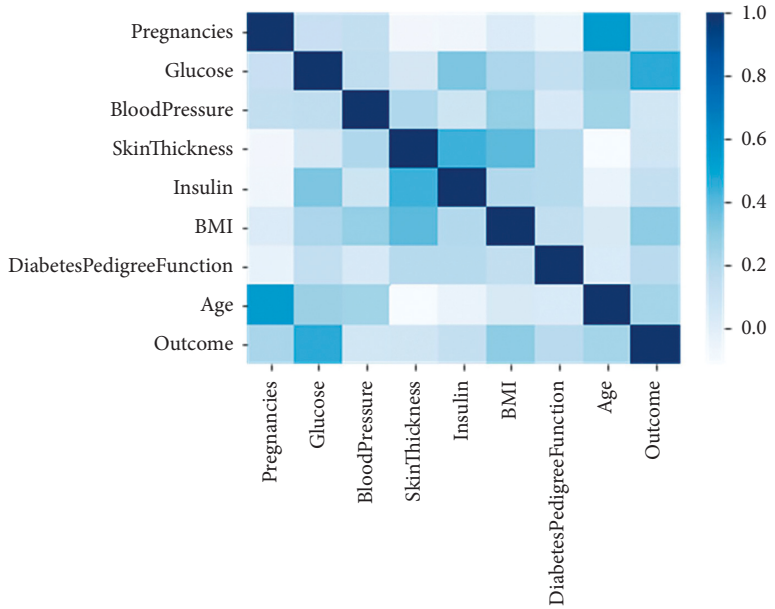


Figure 3: Correlation matrix.

4.6. Theoretical Description of Concepts Used. Many ML methods and techniques can be tested and used along with classifiers for diabetes disease prediction. However, for the datasets used, the best suited classifiers are gradient boosting classifiers (GBM, LGBM, and XGB) and decision tree based on the simulation mechanism used

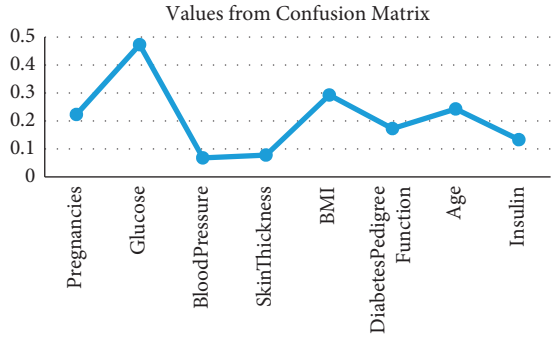


Figure 4: Correlation graph.

Table 1: Correlation matrix values.

Pregnancy	0.22
Glucose	0.47
Blood pressure	0.065
Skin thickness	0.075
BMI	0.29
Diabetes pedigree function	0.17
Age	0.24
Insulin	0.13

earlier. However, other classifiers such as random forest, naive Bayes, and support vector machine are also considered for final accuracy percentage analysis [35].

The theoretical concepts of the machine learning classifiers used are explained below.

4.6.1. Gradient Boosting. A gradient boosting classifier is a combination of many weak learners formed into a predictive model typically in the form of decision trees. The number of trees is based on the number of values in the dataset used. It is mainly used when the bias error in the model needs to be decreased. A gradient-descent technique is chosen to obtain values of the coefficients [35].

In order to obtain the value of the coefficient, the loss function used needs to be calculated. It is calculated using $(\mathrm{y1} - \mathrm{y1}\prime)^2$, where y1 is the actually calculated value and y1′ is the finally predicted value by the model. So y1′ is replaced with $G_n(X)$ which represents the actual target [36]. It is mathematically given as

$$G_{n+1}(X) = G_n(X) + \gamma_n H1(x, e_n), \tag{3}$$

$$L1 = (y1 - y1')^2, \tag{4}$$

$$L1 = (Y - G_n(x))^2. \tag{5}$$

4.6.2. Light Gradient Boosting Machine (LGBM). The LGBM has high performance and is considered as an advanced version of "gradient boosting framework" based on decision tree algorithm. It is majorly used for ranking and classification. It splits the tree leaf-wise with best fit. It can be calculated using many data improvement techniques and can be given by evaluating the variance after diving the values [26]. It can be given by the following equation:

$$Y1 = \text{Base Tree}(X1) - lr1^{*}\text{Tree1}(X1) - lr1^{*}\text{Tree2}(X1). \tag{6}$$

The value determines the way in which the decision tree algorithm can be used to split the data and implement the values. The equation represents the number of trees that can be used in the model depending on the number of instances used in the dataset. When compared with GB, LGBM is comparatively faster and the parameters used are different, which can further increase or decrease the efficiency [37].

4.6.3. XGB. XGBoost is used for supervised regression models. It is used to infer the details about the validity of the objective function and base learners. The concept of ensemble learning is used to combine the results into a single prediction by involving training and combining individual models. XGBoost is a type of the ensemble learning methods. The objective function of XGBoost is given as follows:

$$obj(\theta) = \sum_{1}^{n} l(y_i - \hat{y}_i) + \sum_{j=1}^{j} \delta(f_j), \tag{7}$$

where f_j denotes prediction from the jth tree. The MSE (mean squared error) is given as follows [38]:

$$MSE = \frac{\sum_{i=1}^{n}(y_i - y_i^p)^2}{n}. \tag{8}$$

4.6.4. Decision Tree. The decision tree entropy is generated as follows: A node k is taken, and J class labels are identified. The value of j ranges from 1 to J. It is given mathematically as follows:

$$\text{Entropy}(k) = -\sum_{j=1}^{J} p(j|k)\log_2(j|k). \tag{9}$$

LGBM uses two concepts, namely, GBDT (gradient boosting decision tree) and GOSS (gradient based one-sided sampling). It is used to separate the tree in a leaf-wise manner that provides best fit whereas other boosting algorithms are used to divide it depth-wise. The accuracy results are better when compared with the other existing boosting algorithms [39].

4.6.5. Random Forest (RF). The RF consolidates the outputs or outcomes of a number of decision trees together in order to obtain a single result. The DTs considered are taken as a base row sampling technique as well as column sampling technique. The number of base learners is improved

depending on the inputs, and the variance is reduced to increase the accuracy. It is taken into account as one of the important bagging methodologies [40].

$$\begin{aligned}\text{Random forest} = {} & \text{DT (base learner)} + \text{bagging (row sampling with replacement)} \\ & + \text{faeture bagging (column sampling)} \\ & + \text{aggregation (mean/median, majority vote).}\end{aligned} \tag{10}$$

4.6.6. Naive Bayes (NB). NB is dependent on classification methods that divide the data using the conditional probability values. Naive Bayes is an algorithm that is used for detecting the behavior of the different patients involved. It is a combination of classification logistic regression for classifying the patients into different groups. It is an algorithm that works swiftly for all the classification problems. It is good for predictions involving real time, multiple classes, recommendation system, text based classification, and sentiment analysis. It is easy to implement for large datasets [2].

The Bayesian formula for calculating naive Bayes algorithm is as follows:

$$\mathrm{P(A|B)} = \frac{\mathrm{P(B|A)}P(A)}{P(B)}, \tag{11}$$

where P(A|B) is the posterior probability, P(B|A) is the likelihood probability, P(A) is the class prior probability, and P(B) is the predictor prior probability [41].

4.6.7. Support Vector Machine. Support vector machine belongs to the concept of supervised learning algorithm that can be used for problems of regression and classification. It is used to generate the decision boundary or best line for dividing the n-dimensional space into many classes that are different from one another in order to place the data point in the right category for future purposes. The hyperplane is known as the best decision boundary. To create the hyperplane, SVM selects vectors and extreme points. This introduces the concept of support vectors that further gives rise to the algorithm called support vector machine [42].

SVM uses the Lagrangian formulation mentioned below for classifying the testing samples:

$$d\left(X^T\right) = \sum_{i=1}^{l} y_i a_i X_i X^T + b_0, \tag{12}$$

where the class label of support vector is given as y_i, $X_i X^T$ are test tuple, and α_i and b_0 are numeric parameters [43].

4.7. Data Augmentation. In some cases, there are small datasets. In order to overcome the problem of small datasets or data imbalance, the concept of data augmentation can be applied [44]. Data augmentation in data analysis is a technique that can be utilized for increasing the quantity of data by slightly modifying copies of the already existing data or creating synthetic data from the already existing data [45]. It is useful for enhancing the performance and outcomes of ML models by forming new and different examples to train the datasets. The flow of the process is given in Figure 9.

For text data, techniques such as sampling, tokenization of documents or texts, shuffling sentences, rejoining statements, work replacement, and syntax tree manipulation can be carried out [46]. Some libraries used for data augmentation are Augmentor, Albumentations, Imgaug, and AutoAugment (Deep Augment).

These libraries have to be used along with frameworks for implementation. Some of the libraries already have a predefined or preexisting synergy with specific framework. For example, Albumentations uses PyTorch [47].

4.7.1. Technique. The technique used is oversampling. The concept of oversampling involves randomly duplicating the values in the dataset. The examples are chosen from the minority class by replacing and adding them to the existing training dataset. This process is repeated until the data in the minority class and majority class are equal [48].

4.8. The Algorithm Used for Oversampling. Figures 10 and 11 show the count of the dataset before augmentation and after augmentation. In Figure 10, there is a mismatch in the count of the dataset, whereas in Figure 11, the data count has been balanced.

The metrics used for accuracy prediction include [49] F1-score, precision, recall, sensitivity, and specificity. They can be calculated as follows:

F1-Score: It is a metric used to calculate accuracy. It is used in classification models. It is calculated mathematically as follows:

$$2 * \frac{(\text{precision} * \text{recall})}{(\text{precision} + \text{recall})}. \tag{13}$$

Recall: It is mainly used to identify relevant data among a lot of available data. It is calculated mathematically as follows:

$$\frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}. \tag{14}$$

Precision: It determines the quality of the positive prediction made by the model. It is calculated mathematically as follows:

$$\frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}. \tag{15}$$

Inputs: Class_0: Minority Class, Class_1: Majority Class.
Parameters: used to improve minority class Class_0.
(1) Class_1_over: To get sample of Class_0 and to store in Class_1_over.
(2) Test_over: To concatenate Class_0 and Class_1_over.

Algorithm 10: Consider a class or the whole dataset with n samples and (F) features.

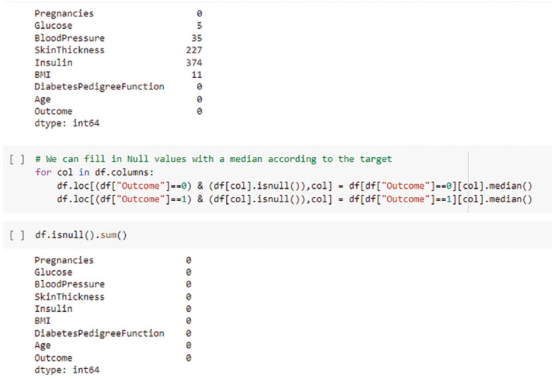


Figure 5: Data preprocessing.

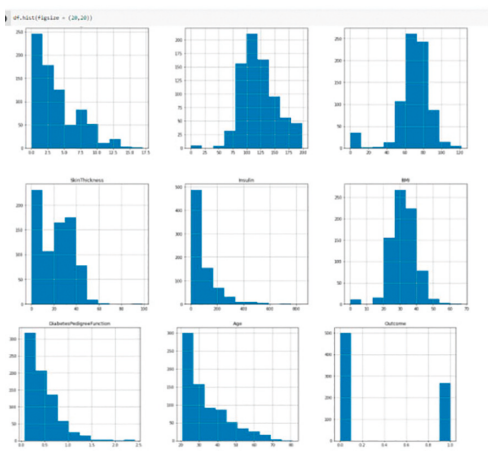


Figure 6: Histogram.

Specificity: It determines the proportion of actual negatives which are true negatives in the model. It is calculated mathematically as follows:

$$\frac{\text{True Negative}}{\text{True Negative} + \text{False Positive}}. \tag{16}$$

Sensitivity: It determines the value of *TruePositive* in proportion to the added values of *TruePositive* and *FalsePositive*. It is calculated mathematically as follows [50]:

$$\frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}. \tag{17}$$

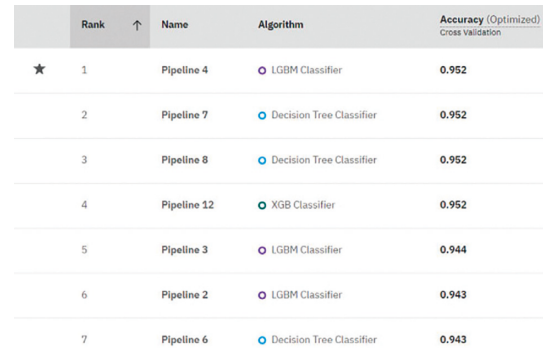


Figure 7: Accuracy percentage during simulation, process 1.

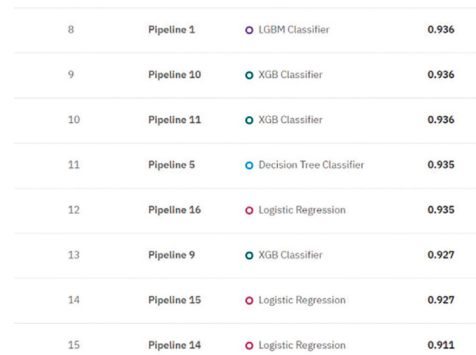


Figure 8: Accuracy percentage during simulation, process 2.

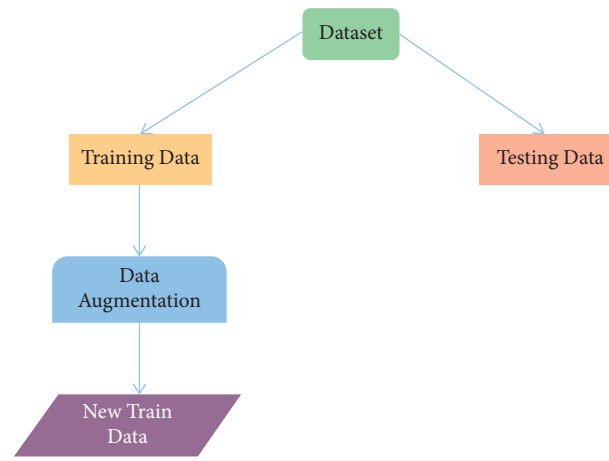


Figure 9: Flow of the process.

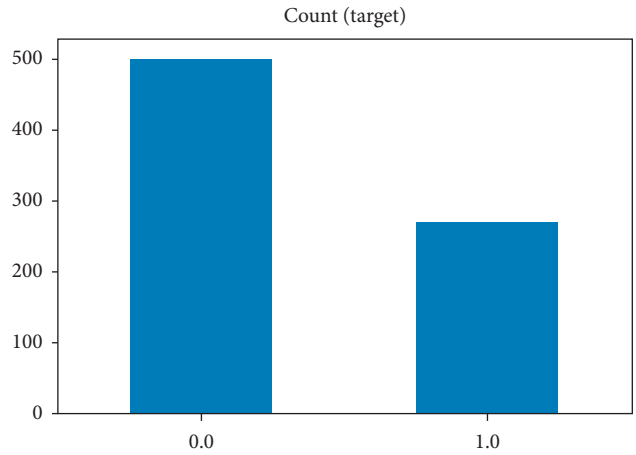


Figure 10: Before augmentation.

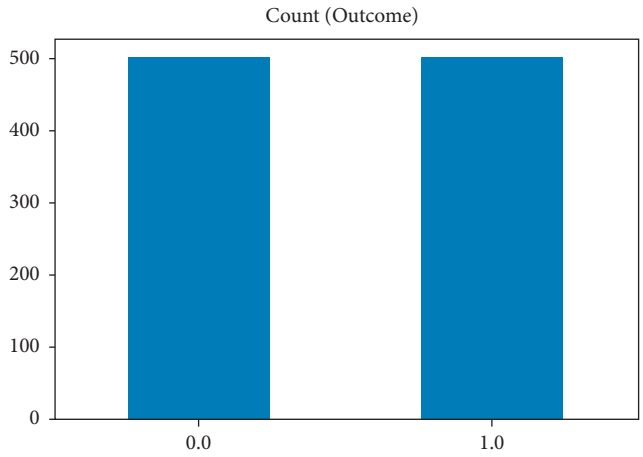


Figure 11: After augmentation.

5. Architecture

The architecture consists of the working flow of the procedure as shown in Figure 12. Initially the data is chosen from the various databases available, and finally the dataset is chosen. The data chosen is preprocessed, and feature selection and extraction are carried out. The data is further preprocessed using EDA until all of the defects are rectified. The dataset is then clean and suitable for training and testing procedures. The dataset is divided into training data, testing data, and validation data. The various classifiers are then compared, and the best suited classifier for the dataset is then chosen and applied. The prediction model is then built based on the classifiers taken, i.e., GB, LGBM, and RF.

After obtaining the prediction accuracy, data augmentation is further applied to the dataset, and this improves the already existing accuracy percentage. This is finally obtained as the best prediction accuracy percentage of the model. In addition to the above methods, the voting classifier is also used to predict the best possible outcome for the disease prediction among the classifiers LGBM, GB, and RF.

Voting Classifier: It is a ML model that can be used to train on numerous models and predict output depending on the highest probability of the class chosen as the output. The voting classifiers are divided into softvoting and hardvoting.

Hard Voting: Based on the higher number of votes Nc(yt), the prediction of class label happens via majority voting of each classifier. Hard voting is mainly used to predict class labels. It is given mathematically as follows:

$$\hat{y} = \arg\max\left(Nc\left(yt^1\right), Nc\left(yt^2\right) \ldots Nc\left(yt^n\right)\right). \tag{18}$$

Soft Voting: The probability vectors for each predicted classifier are summed up, and the average is obtained. The highest value is taken as the winning class. Soft voting is mainly used to predict class membership probabilities. It is given mathematically as follows:

$$\hat{y} = \arg\max \frac{1}{N_{\text{Classifiers}}} \sum_{\text{Classifiers}} \left(P_{1,} P_{2,} \ldots P_n\right). \tag{19}$$

6. Results and Discussion

The prediction for diabetes mellitus is done by the model built using the dataset Pima initially, and then the highest accuracy producing algorithms are chosen and further incorporated in the DMS dataset used.

The accuracy values obtained for the various classifiers are given in Table 2. The abbreviations for the classifiers are as follows: LR (logistic regression), XGB (extreme gradient boosting), GB (gradient boosting), DT (decision tree), ET (extra trees), RF (random forest), and LGBM (light gradient boosting machine).

Figure 13 indicates the bar graph of the accuracy percentage obtained while using classifiers LR, XGB, GBM, DT, ET, RF, and LGBM.

Table 2 shows that LGBM and RF produce the highest accuracy. However, from Figures 6 and 7, XGB also produces high accuracy. Therefore another predictive mechanism for the classifiers RF, LGBM, and GBM is conducted for the datasets Pima and DMS.

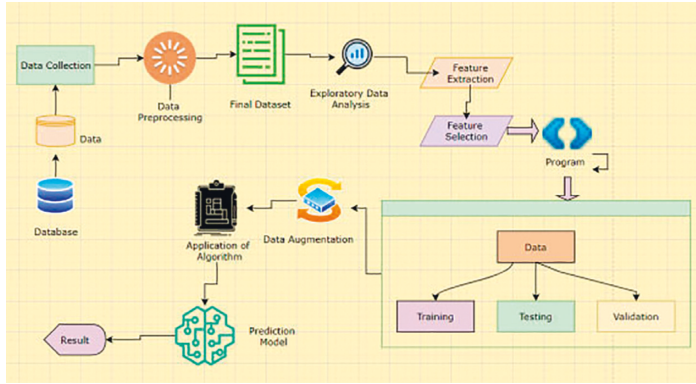


Figure 12: Architecture of proposed model.

Table 2: Comparison of all classifiers.

Dataset	Algorithm	Accuracy (%)
Pima Indians dataset	LR	75.20
	XGB	83.30
	GBM	94.10
	DT	94.40
	ET	94.60
	RF	94.80
	LGBM	95.20

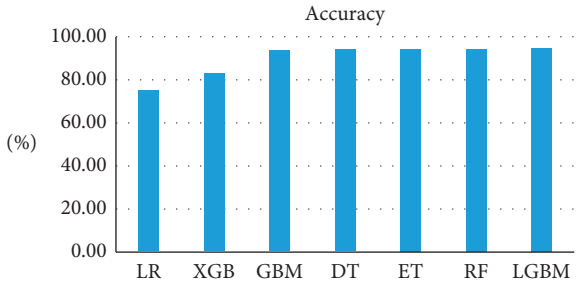


Figure 13: Accuracy percentage.

Table 3: Comparison of 3 classifiers.

Dataset	LGBM	RF	GBM
Pima without preprocessing	89.5	89.5	84.5
DMS without preprocessing	93.92	95.27	87.84
Pima with augmentation and preprocessing	92.5	92	91.5
DMS with augmentation and preprocessing	98.99	96.6	97.64

Diabetes mellitus is predicted using the predictive model built, and the accuracy for the 2 datasets used varies with each machine learning algorithm used. The result obtained after performing data augmentation for both datasets is given in Table 3.

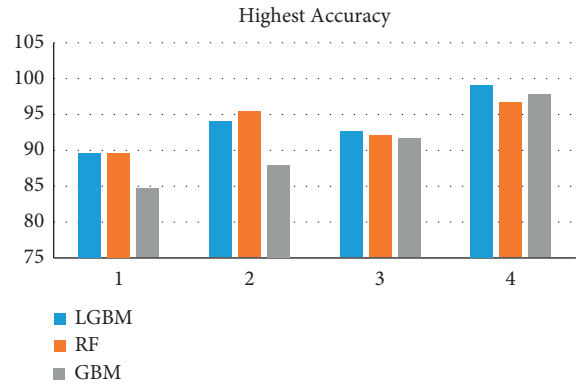


Figure 14: Highest accuracy percentage.

From Table 3, it can be seen that the LGBM algorithm produces the highest accuracy (89.5%) for the Pima dataset and the same accuracy is also obtained when the RF classifier is used without data preprocessing. When the data is preprocessed, the accuracy obtained for Pima dataset is highest (92.5%) when LGBM algorithm is used. For the DMS dataset, the accuracy obtained is highest (95.27%) when RF algorithm is used without preprocessing, and the accuracy obtained after preprocessing is highest (98.99%) for LGBM.

Figure 14 demonstrates the bar graph of the highest accuracy percentage obtained while using classifiers LGBM, RF, and GBM.

7. Conclusion and Future Scope

From the above study, it can be concluded that the LGBM algorithm provides the highest accuracy when compared with RF and GB classifiers. Therefore, the LGBM algorithm is well suited for the Pima dataset and the DMS dataset used in the study.

The LGBM algorithm differs from RF and GB in the following ways: The parameters used in LGBM are different from those in GB and RF. The parameter tuning varies with each algorithm, and the model is built based on the classifier

used. Therefore, in this paper, a predictive model is built using LGBM algorithm, and the accuracy is obtained as shown in Table 3 for the datasets used.

The diabetes mellitus disease prediction can further be improved by enhancing the dataset using other advanced methodologies like transformer based learning. The attributes used can also be employed in different combinations for identification. The classifiers used can be fine-tuned more to predict the disease with higher accuracy, and the probability of occurrence of the disease can be calculated. This will further improve the accuracy percentage and deliver a more profound model to predict diabetes mellitus disease among affected people.

Data Availability

The data used to support the findings of the study are available at https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] A. Mujumdar and V. Vaidehi, "Diabetes prediction using machine learning algorithms," *Procedia Computer Science*, vol. 165, pp. 292–299, 2019.

[2] A. S. Alanazi and M. A. Mezher, "Using machine learning algorithms for prediction of diabetes mellitus," in *Proceedings of the International Conference on Computing and Information Technology (ICCIT-1441)*, pp. 1–3, Tabuk, Saudi Arabia, 2020.

[3] A. Allen, Z. Iqbal, A. Green-Saxena et al., "Prediction of diabetic kidney disease with machine learning algorithms, upon the initial diagnosis of type 2 diabetes mellitus," *BMJ Open Diabetes Research & Care*, vol. 10, no. 1, Article ID e002560, 2022.

[4] K. Anandha Kumar, "A survey on diabetes mellitus prediction using machine learning techniques," *International Journal of Applied Engineering Research*, vol. 11, 2022.

[5] A. Arora, N. Shoeibi, V. Sati, A. González Briones, and P. Chamoso, "Data augmentation using gaussian mixture model on CSV files," *Distributed Computing and Artificial Intelligence*, Springer, Berlin, Germany, 2021.

[6] S. Habibi, M. Ahmadi, and S. Alizadeh, "Type 2 diabetes mellitus screening and risk factors using decision tree: results of data mining," *Global Journal of Health Science*, vol. 7, no. 5, pp. 304–310, 2015.

[7] L. Kopitar, P. Kocbek, L. Cilar, A. Sheikh, and G. Stiglic, "Early detection of type 2 diabetes mellitus using machine learning-based prediction models," *Scientific Reports*, vol. 10, no. 1, Article ID 11981, 2020.

[8] Y. Deng, L. Lu, L. Aponte et al., "Deep transfer learning and data augmentation improve glucose levels prediction in type 2 diabetes patients," *Npj Digital Medicine*, vol. 4, no. 1, p. 109, 2021.

[9] U. M. Butt, S. Letchmunan, M. Ali, F. H. Hassan, A. Baqir, and H. H. R. Sherazi, "Machine learning based diabetes classification and prediction for healthcare applications," *Journal of Healthcare Engineering*, vol. 2021, Article ID 9930985, 17 pages, 2021.

[10] V. C. Bavkar and A. A. Shinde, "Machine learning algorithms for Diabetes prediction and neural network method for blood glucose measurement," *Indian Journal of Science and Technology*, vol. 14, 2021.

[11] B. Ljubic, A. A. Hai, M. Stanojevic et al., "Predicting complications of diabetes mellitus using advanced machine learning algorithms," *Journal of the American Medical Informatics Association*, vol. 27, no. 9, pp. 1343–1351, 2020.

[12] J. Chaki, S. T. Ganesh, S. K. Cidham, and S. Ananda Theertan, "Machine learning and artificial intelligence based diabetes mellitus detection and self-management: a systematic review," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, pp. 1319–1578, 2020.

[13] X. Li, J. Zhang, and F. Safara, "Improving the accuracy of diabetes diagnosis applications through a hybrid feature selection algorithm," *Neural Processing Letters*, pp. 1–17, 2021.

[14] S. Islam Ayon and M. Milon Islam, "Diabetes prediction: a deep learning approach," *International Journal of Information Engineering and Electronic Business*, vol. 11, no. 2, pp. 21–27, 2019.

[15] A. Nur Ghaniaviyanto Ramadhan and R. Ade, "Preprocessing handling to enhance detection of type 2 diabetes mellitus based on random forest," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 12, 2021.

[16] L. Fregoso-Aparicio, J. Noguez, L. Montesinos, and J. A. Garcia-Garcia, "Machine learning and deep learning predictive models for type 2 diabetes: a systematic review," *Diabetology & Metabolic Syndrome*, vol. 13, no. 1, p. 148, 2021.

[17] https://towardsdatascience.com/machine-learning-for-diabetes-562dd7df4d42.

[18] https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database.

[19] https://www.kaggle.com/datasets/mathchi/diabetes-data-set.

[20] https://www.analyticsvidhya.com/blog/2022/01/diabetes-prediction-using-machine-learning/.

[21] J. Martinsson, A. Schliep, B. Eliasson, and O. Mogren, "Blood glucose prediction with variance estimation using recurrent neural networks," *Journal of Healthcare Informatics Research*, vol. 4, no. 1, pp. 1–18, 2020.

[22] 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, p. 101, 2019.

[23] M. Soni and S. Varma, "Diabetes prediction using machine learning techniques," *International Journal of Engineering Research and Technology*, vol. 9, no. 9, 2020.

[24] S. V. K. R. Rajeswari and P. Vijayakumar, "Prediction of diabetes mellitus using machine learning algorithm," *Annals of the Romanian Society for Cell Biology*, vol. 25, pp. 5655–5662, 2021.

[25] M. Shuja, M. Sonu, and Z. Majid, "Decision Support Predictive model for prognosis of diabetes using SMOTE and Decision tree," *International Journal of Applied Engineering Research*, vol. 13, 2018.

[26] N. Sneha and T. Gangil, "Analysis of diabetes mellitus for early prediction using optimal features selection," *J Big Data*, vol. 6, no. 1, p. 13, 2019.

[27] R. Srivastava and R. K. Dwivedi, "A survey on diabetes mellitus prediction using machine learning algorithms," in *ICT Systems and Sustainability*, M. Tuba, S. Akashe, and A. Joshi, Eds., Springer, Berlin, Germany, 2022.

[28] S. A. Narendrakumar, "Diabetes mellitus prediction using ensemble machine learning techniques," 2020, https://ssrn.com/abstract=3642877.

[29] C. Thiyagarajan, K. Kumar, and A. Bharathi, "A survey on diabetes mellitus prediction using machine learning techniques," *International Journal of Applied Engineering Research*, vol. 11, pp. 1810–1814, 2016.

[30] R. J. Valderrábano and M. I. Linares, "Diabetes mellitus and bone health: epidemiology, etiology and implications for fracture risk stratification," *Clinical diabetes and endocrinology*, vol. 4, no. 1, p. 9, 2018.

[31] T. Zhu, K. Li, and J. Chen, "Dilated recurrent neural networks for glucose forecasting in type 1 diabetes," *Journal of Healthcare Informatics Research*, vol. 4, pp. 308–324, 2020.

[32] 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, p. 515, 2018.

[33] 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, p. 515, 2018.

[34] K. Guolin, M. Qi, F. Thomas et al., "LightGBM: a highly effificient gradient boosting decision tree," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 3149–3157, Long Beach, CA, USA, 2017.

[35] B. Omodunbi, "Development of a diabetes melitus detection and prediction model using light gradient boosting machine and k-nearest neighbour," *UNIOSUN Journal of Engineering and Environmental Sciences*, vol. 3, no. 1, 2021.

[36] S. Habibi, M. Ahmadi, and S. Alizadeh, "Type 2 diabetes mellitus screening and risk factors using decision tree: results of data mining," *Global Journal of Health Science*, vol. 7, no. 5, pp. 304–310, 2015.

[37] B. Shamreen Ahamed and Dr. Meenakshi Sumeet Arya, "Prediction of type-2 diabetes using the LGBM classifier methods and techniques," *Turkish Journal of Computer and Mathematics Education*, vol. 12, no. 12, pp. 223–231, 2021.

[38] A. M. Posonia, S. Vigneshwari, and D. J. Rani, *Machine Learning Based Diabetes Prediction Using Decision Tree J48*, in *Proceedings of the 3rd International Conference on Intelligent Sustainable Systems(ICISS)*, pp. 498–502, Thoothukudi, India, 2020.

[39] D. Pei, T. Yang, and C. Zhang, "Estimation of diabetes in a high-risk adult Chinese population using J48 decision tree model," *Diabetes, Metabolic Syndrome and Obesity: Targets and Therapy*, vol. 13, pp. 4621–4630, 2020.

[40] N. Pradhan, G. Rani, V. Singh Dhaka, and R. Chandra Poonia, "14 - diabetes prediction using artificial neural network," *Deep Learning Techniques for Biomedical and Health Informatics*, pp. 327–339, 2020.

[41] https://towardsdatascience.com/xgboost-mathematics-explained-58262530904a.

[42] https://www.saedsayad.com/naive_bayesian.htm.

[43] Y. Li and H. Wu, "A clustering method based on K-means algorithm," *Physics Procedia*, vol. 25, pp. 1104–1109, 2012.

[44] S. Kumari, D. Kumar, and M. Mittal, "An ensemble approach for classification and prediction of diabetes mellitus using soft voting classifier," *International Journal of Cognitive Computing in Engineering*, vol. 2, 2021.

[45] N. S. kumar, T. Eswari, P. Sampath, and S. Lavanya, "Predictive methodology for diabetic data analysis in big data," *Procedia Computer Science*, vol. 50, pp. 203–208, 2015.

[46] K. Saravananathan and T. Velmurugan, "Analyzing diabetic data using classification algorithms in data mining," *Indian Journal of Science and Technology*, vol. 9, no. 43, 2016.

[47] R. Sehly and M. Mezher, "Comparative analysis of classification models for pima dataset," in *Proceedings of the International Conference on Computing and Information Technology (ICCIT-1441)*, pp. 1–5, Tabuk, Saudi Arabia, 2020.

[48] L. Baoli, Y. Shiwen, and L. Qin, "An improved K nearest neighbor algorithm for text categorization," *Expert Systems with Applications*, vol. 39, 2003.

[49] M. Muja and D. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," *VISAPP 2009 - Proceedings of the 4th International Conference on Computer Vision Theory and Applications*, vol. 1, pp. 331–340, 2009.

[50] D. Shetty, K. Rit, S. Shaikh, and N. Patil, "Diabetes disease prediction using data mining," in *Proceedings of the 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, pp. 1–5, Coimbatore, India, 2017.