Utilizing Various Machine Learning Techniques for Diabetes Mellitus Feature Selection and Classification

Alaa Sheta[1], Walaa H. Elashmawi[2], Ahmad Al-Qerem[3], Emad S. Othman[4]
Department of Computer Science, Southern Connecticut State University, New Haven, CT, USA[1]
Department of Computer Science-Faculty of Computers & Informatics, Suez Canal University, Ismailia, Egypt[2]
Computer Science Department-Faculty of Information Technology, Zarqa University, Zarqa, Jordan[3]
Management Information System Department, AL-Shorouk Academy, Cairo, Egypt[4]

***Abstract*—Diabetes mellitus is a chronic disease affecting over 38.4 million adults worldwide. Unfortunately, 8.7 million were undiagnosed. Early detection and diagnosis of diabetes can save millions of people's lives. Significant benefits can be achieved if we have the means and tools for the early diagnosis and treatment of diabetes since it can reduce the ratio of cardiovascular disease and mortality rate. It is urgently necessary to explore computational methods and machine learning for possible assistance in the diagnosis of diabetes to support physician decisions. This research utilizes machine learning to diagnose diabetes based on several selected features collected from patients. This research provides a complete process for data handling and pre-processing, feature selection, model development, and evaluation. Among the models tested, our results reveal that Random Forest performs best in accuracy (i.e., 0.945%). This emphasizes Random Forest's efficiency in precisely helping diagnose and reduce the risk of diabetes.**

***Keywords*—Diabetes; machine learning; random forest; SMOTE technique**

I. Introduction

The Centers for Disease Control (CDC) reported several statistics on the number of people diagnosed with diabetes. It was found that 38.4 million people were diagnosed with diabetes. This ratio presents 11.6% of the adult of 18 years or older. The healthcare system afforded in 2022 more than $413 billion [1]. For example, by 2030, it is anticipated that more than 20% of the population of West Virginians will be diagnosed. The public's health is being devastated by this. Following Alabama, the following two states with the highest disease rates are Mississippi and Florida. South Americans have a significantly high chance of being diagnosed with diabetes by 2030.

The Economic Report published by the American Diabetes Association in 2022 shows that the total yearly cost of diabetes exceeds $412 billion, including a direct and indirect medical cost of $306.6 billion and $106.3, respectively. It is worth mentioning that diabetes is the eighth reason of death in the United States. There were more than 399,401 deaths linked to diabetes in the USA.

Diabetes is a metabolic condition that supports the development of high blood sugar levels. When left undiagnosed, the high sugar in the blood could lead to severe damage to organs such as the kidneys, heart, and eyes [2]. Diabetes can emerge in two different ways: type 1 complications and type 2 complications. Those who have type 2 diabetes have insulin production that is either insufficient or nonexistent. Sometimes, the patient's body is not reacting to the effects of insulin appropriately. Although type 2 is more dangerous than type 1, it is widespread for people aged 19 and over [3]. The authors of [4] investigated the possibility of utilizing Artificial Neural Networks (ANN), Support Vector Machines (SVM), and Decision Trees (DT) to classify the Pima Indians Diabetes and Breast Cancer Coimbra datasets that are available in the UCI Machine Learning Repository.

Diagnosing diabetes is currently very challenging for several reasons, including the following:

1) The availability of an adequate dataset to build an ML model with high confidence [5]. It is normally a lengthy process to get permission to access the patient's medical records, given that the patient has a medical history and is always checking his medical condition citenoisy-data-diabetes. The Obama Administration invested over $27 billion to support hospitals and medical service providers to implement electronic health records (EHR). Currently, clinics adopt a software platform to store medical data. The problem arises when trying to integrate these HER systems. Thus, medical data is commonly unstructured since each software platform has a different design, and integrating this system is always challenging.

2) A multidisciplinary method is essential to develop reliable diagnosis (i.e., prediction) models. Experts from diverse fields such as medicine, statistics, and data scientists need to collaborate to verify the correct diagnosis of the disease [6], [7].

3) There is always a need to develop diagnosis models that are explainable and easy for physicians to interpret. Physicians are always interested in understanding the cause and being able to generate a resonating of the findings.

4) Finally, in many cases, it is important to integrate these diagnosis models to perform on a computer platform or mobile devices [8], [9]. These models should be integrated into the EMR systems.

For these reasons, this research aims to demonstrate the effectiveness of machine learning, particularly Random Forest, in efficiently diagnosing diabetes. By selecting the most compelling features collected from patients and providing a comprehensive process of data handling, pre-processing, model

development, and evaluation, we have achieved a high accuracy diagnosis rate of 94.5%. This emphasizes the potential of machine learning algorithms like Random Forest to help physicians diagnose diabetes early and effectively moderate its risks.

The subsequent sections delineate the structure of this paper. Machine learning models for classification are covered in Section II. Section III provides a comprehensive explanation of the machine learning approaches employed. The steps of classifying diabetes, from dataset preparation to the evaluation of machine learning models, are illustrated in Section IV. Sections V and VI outline the results of three distinct machine-learning algorithms for classifying diabetes. Additionally, various evaluation criteria are used to evaluate the compared algorithms. Section VII presents this research's main findings, and some future directions are mentioned.

II. Machine Learning

Traditional diagnosis models adopted correlation methods between symptoms and cause(s) [10]. Additional approaches were also utilized, including examining environmental and genetic factors that influence the development and risk of type 1 and type 2 diabetes [11], [12]. AI has helped accelerate the diagnosis of medical diseases and the advancement of drugs and medicines. Healthcare systems with AI and ML have become more modernized. ML techniques significantly support advancing diagnosis methods such that they enhance the precision in medical diagnosis [4], [13], [14]. Diagnosis using ML involves the development of models that utilize input data to build a relationship between various medical features (i.e., attributes) to produce a corresponding diagnosis (i.e., label). This process involves training a model to recognize if there is a disease or not. As seen in Fig. 1, there are several stages to the ML diagnostic process, including pre-processing of the dataset, selection of the most promising features, utilizing the most appropriate model, and finally assessing the model. The medical industry has successfully used this technique for diagnosis and prediction, leading to improved patient outcomes [15], [16]. Various research has validated using artificial intelligence in conjunction with machine learning [15]–[20] in solving real-world problems.

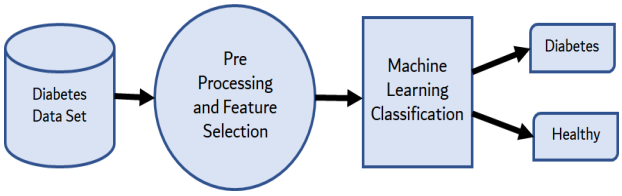

Fig. 1. Machine learning classification process.

III. Methods

This section outlines the basic concepts of diverse machine-learning techniques for developing the proposed diabetes classification model.

A. Artificial Neural Networks

Prominent machine-learning models include artificial neural networks (ANNs). It draws its inspiration from the biological neurons of the human brain. Multiple layers make up an ANN. These layers include input, hidden, and output. These layers are organized sequentially so that the output from the first layer feeds into the next one. The input layer contains neurons corresponding to the model's input features. Depending on the specific application, the number of neurons in the hidden layer might vary from a few to many. Ultimately, the number of neurons in the output layer equals the number of labels, or classes, in the data set. We use the sigmoid function to produce model nonlinearity, which gives the model more flexibility. The literature has well-known functions, such as the tanh and ReLU functions. ANN was used in many medical diagnosis applications [13], [21]. The process of using ANNs for classification involves the following steps:

- *Pre-processing of Dataset:* It is an essential process for preparing the data for modeling to clean it by various means, such as dealing with noise, outliers, missing values, normalization/scaling, data imbalance, and many others.
- *Network Architecture:* The adoption of a specific architecture of the ANN is domain-independent. An adequate number of layers and neurons in the hidden layer is essential for the ANN to model the input and output data successfully.
- *Training the Network:* Many models have been utilized in the literature for training ANN, which mainly depends on the adopted structure. A famous example is the employment of a backpropagation learning method for training the Feedforward ANN model, which is based on gradient descent [22].
- *Testing and Validation:* To verify the ANN model's ability to diagnose a disease, we utilize a new dataset to test the ANN-developed model and calculate its performance, such as accuracy and precision.
- *Deployment:* The ANN model can now be deployed in real-world applications.

Fig. 2 shows a Feedforward ANN model with five inputs: $x_i, i = 1, \ldots, 5$. given that, the network has four hidden nodes $h_j, j = 1, \ldots, 4$ and one output node. The output y can be presented in Eq. 1. w_i and b_i correspond to the weights and biases of the ANN.

$$y = f\left(\sum_{i=1}^{n} w_i x_i + b\right) \tag{1}$$

B. Decision Tree

Given the features of the data, a powerful machine-learning technique called a Decision Tree (DT) may be constructed according to a set of rules. DT can be used for various machine learning classification and regression applications [23]. DT learning algorithm depends on picking up the best-split point on each node. The process of splitting utilizes the concept of Entropy and Information Gain [23], [24] and provides

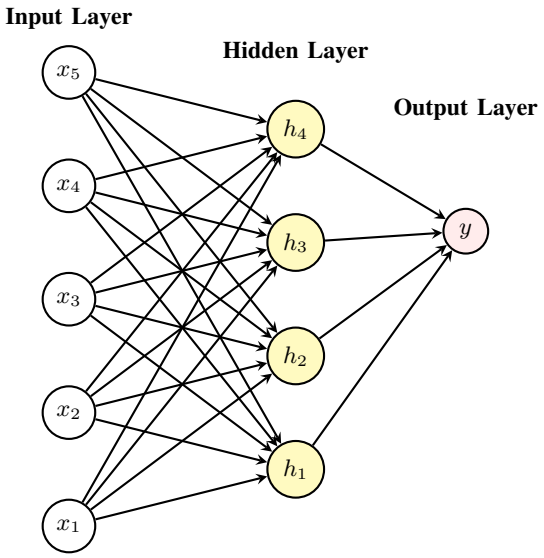

Fig. 2. Feedforward ANN model.

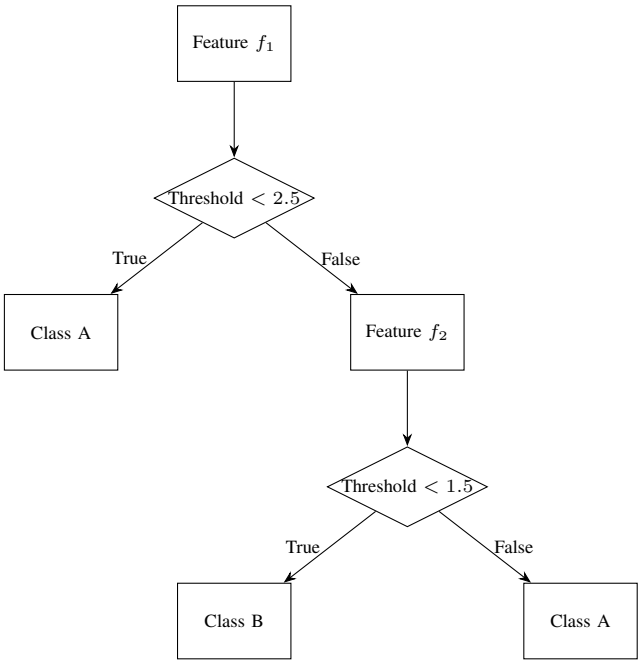

Fig. 3. Example of a simple decision tree.

the best data splitting. Information theory inspires entropy, determining the sample values' impurity. The entropy (i.e., $S(Z)$) is calculated using Eq. 2.

$$S(Z) = -\sum_{i} P(Z = z_i) \cdot \log_2(P(Z = z_i)) \tag{2}$$

Given that $S(Z)$ represents the entropy of the random variable Z and $P(Z = z_i)$ denotes the probability of the occurrence $Z = z_i$, the table summarizes key symbols and their descriptions.

The process for creating a decision tree for diagnosis (i.e., classification) consists of the following phases:

1) Utilize the training data to explore the best feature to be considered as a root node for data splitting using entropy.
2) Based on step 1, several child nodes will be created. The process adopted in phase one is repeated to build the new tree level and create new sets of children nodes.
3) Repeat phases 1 and 2 pending a stopping criterion is satisfied. For example, approaching the maximum tree depth or having a minimum number of samples per leaf. Fig. 3 illustrates a simplified example of the development of a decision tree, showcasing the creation of child nodes at each step.

To minimize the complexity of the DT and avoid overfitting, we adopt the concept of pruning. Pruning allows the DT to overcome the problem of overfitting and supports the reduction of the tree's complexity.

C. Random Forest

One of the ensemble learning algorithms used for regression analysis and classification is the random forest (RF) [25]. The RF model's central concept is to generate many decision trees, each constructed using a random subset of the training data and features.

The basic idea of bagging may be depicted as follows: Assume we have a dataset $U = \{(f_1, c_1), (f_2, c_2), \ldots, (f_m, c_m)\}$. Assuming that f_i represents the feature vector of the i-th sample and c_i denotes the class or label. The RF algorithm bagging starts by generating multiple bootstrap samples U_i^* from the original dataset U. Each bootstrap sample produces DT models, as Fig. 4 shows. A rule of thumb for RF is to utilize $\sqrt{m}$ features for each split. To predict the class or label of a new dataset b, we adopt Eq. 3.

$$\hat{P}(b) = \frac{1}{L} \sum_{i=1}^{L} Q_i(x) \tag{3}$$

Given that the random forest has L decision trees. The trees' prediction outputs are denoted by $Q_i(x)$.

It can be seen that the bagging process in RF encompasses training multiple DTs using bootstrap training data and merging the output predictions of trees to produce the overall output of the model. This collaborative approach is very beneficial since it avoids overfitting and reduces the model's sensitivity to noise. It was reported that RF was positively utilized in many application domains, such as healthcare and medicine [12], stock market prediction [27], [28], and many others [29].

D. K-Nearest Neighbor

K-Nearest Neighbors (KNN) is a nonparametric and essential technique used in supervised machine learning. The process of KNN involves classifying objects within the input space based on the distance to the nearest samples. The KNN classification method addresses the challenges of classification

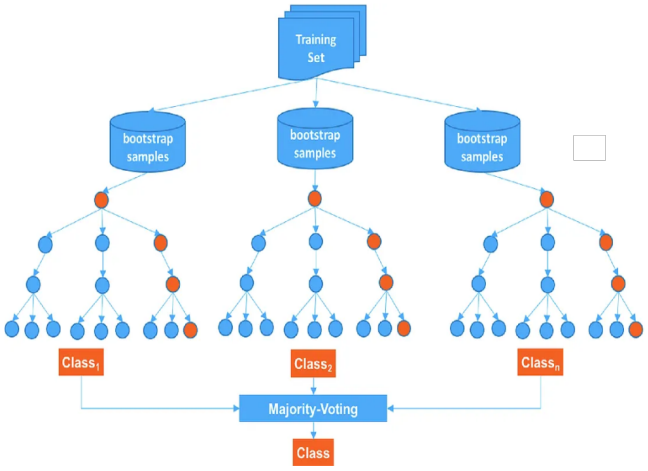

Fig. 4. Illustrating of RF-based bagging method [26].

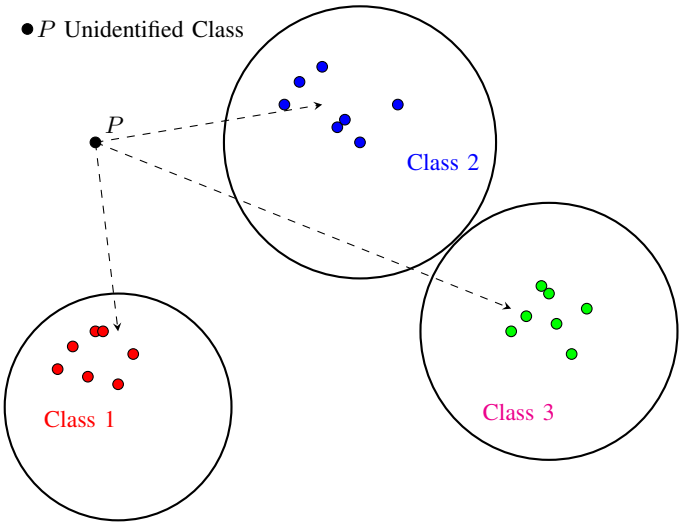

Fig. 5. Illustration of a K-nearest neighbors model.

and regression. Here is a basic overview of how to use KNN for data classification:

- *Data preparation:* Commence by collecting and organizing the dataset. Every data point must possess distinct characteristics (attributes) that provide a description and a matching label with the appropriate format for classification.
- *Choosing K:* For prediction purposes, the parameter 'K' indicates how many nearest neighbors should be considered. A reasonable value for K must be selected. Unreliable predictions result from a small K value, while a large one could introduce bias. Obtaining an optimal K value requires utilizing techniques like cross-validation. Our model utilized a k value equal to 5 for better results.
- *Calculating Distance:* To find a new data point's K-nearest neighbors, the distance between it and all of the points in the dataset is calculated. Assuming two data points X and Y with n features for each such as $X = (x_1, x_2, \ldots, x_n)$ and $Y = (y_1, y_2, \ldots, y_n)$, the Euclidean distance (ED) can be computed according to Equation 4.

$$ED(X,Y) = \sqrt{\frac{\sum_{i=1}^{n}(x_i - y_i)^2}{n}} \tag{4}$$

- *Sorting & Selecting k-neighbors:* Sort the data points based on their distance from the new data point in ascending order. Consequently, the K-nearest neighbors are selected from the sorted list and corresponding data points.
- *Voting for the majority class:* Set the predicted class label or target value for the new data point based on the majority class.
- *Model evaluation:* Analysis of the KNN classifier using several metrics, including F-measure, recall, and accuracy, demonstrates the classification algorithm's performance.

Generally, the kNN algorithm uses a voting system-like approach for assigning a new data point's class, considering the majority class label among its nearest 'k' neighbors in the feature space, as illustrated in Fig. 5.

E. Support Vector Machine

According to [30], a Support Vector Machine (SVM) is one of the classification techniques for supervised machine learning. SVM selects the optimal hyperplane for class separation by aligning the most significant number of points from the same class on one side. The SVM classifier stretches the interval of each class to a hyperplane, which isolates the spots. The hyperplane's nearest points provide the basis of the support vectors. The shortest distance between any two points in a given class and any given hyperplane is from the class to the hyperplane. For a simple linear separable problem, the hyperplane and SVM classifier can be defined according to Eq. 5 and 6.

$$w^T x + b = 0 \tag{5}$$

$$\hat{y} = \begin{cases} 1 : w^T x + b \geq 0 \\ 0 : w^T x + b < 0 \end{cases} \tag{6}$$

The variables in the equation are as follows: w represents a weight vector, x represents a vector, b represents a bias, and $\hat{y}$ represents the projected output class. Minimizing the Euclidean norm of the weight vector w ($\|w\|$) is necessary to optimize the margin. Therefore, this can be formulated as an objective function (i.e., $\min f : 1/2\|w\|^2$).

Here is a basic overview of how to use SVM for data classification:

- *Data Preparation:* The data must be prepared for classification before anything further can be done. Achieving this requires gathering, cleaning, and arranging data so the SVM can readily process it.
- *Train & Test Split:* Splitting the entire dataset into training and testing sets enables us to assess the model's accuracy.

- *Trains SVM with Kernal:* The SVM searches for the optimal hyperplane that divides the classes with the most significant margin using kernel functions. Support vector machines (SVMs) may make use of a wide variety of kernel functions, such as linear, polynomial, sigmoid, and radial basis functions (RBFs) [31].
- *SVM model prediction:* During the training phase, the objective is to determine the hyperplanes that best discriminate between the two classes. During the testing phase, the classification is determined by evaluating the position of the test input relative to the hyperplane.
- *SVM model evaluation:* Several measures, including the confusion matrix and accuracy, may be used to assess the SVM model's performance on the tested dataset.

Fig. 6 shows an illustrative example of finding the best hyperplane for classifying data points. The hyperplane H1 fails to classify the data points, whereas H2 classifies the data points but has the narrowest margin. The hyperplane H3 is considered the ideal classifier due to its ability to classify data points effectively and its greatest marginal width.

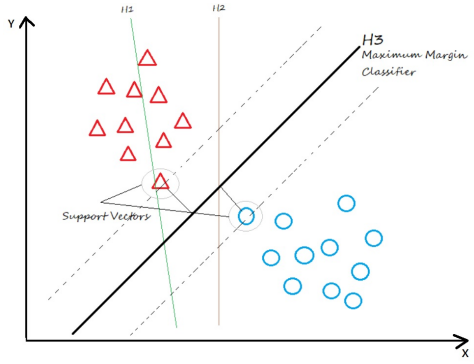

Fig. 6. SVM model.

F. Gradient Boosting

In machine learning, Gradient Boosting (GB) is a very effective method that may be utilized for classification [32] as well as regression tasks. Boosting is based on transforming weak learners into strong ones. To train weak learners, the gradient boosting (GB) approach sequentially adds estimators by modifying their weights one by one [33]. Using an iterative approach for continuous improvement, the GB seeks to estimate residual errors from prior estimators and minimize the difference between predicted and actual values. The overall process can be illustrated below and shown in Fig. 7.

1) Prepare the dataset in a way that the GB algorithm can easily handle through various processes, including cleaning the data, defining the feature variables, and defining the target variable.
2) Select a base model for gradient boosting to fit the data. It is a straightforward model with low variance and high bias. Decision trees are employed as a base learner.
3) Initialize the model by starting predictions based on simple rules or some default values.
4) Calculate the residual error by subtracting the model's predictions from the actual values of the training data.
5) Construct a decision tree and predict the residuals of the prior model. Adjusting the model's parameters in a gradient descent fashion minimizes the loss function as in Eq. 7 during the training of the weak model.

$$F_0(x) = \underset{\gamma}{\operatorname{argmin}} \sum_{i=1}^{n} L\left(y_i, \gamma\right) \tag{7}$$

According to the equation, the predicted and actual values are γ and y_i, respectively. The loss function, denoted as $L = \frac{1}{n}\sum_{i=0}^{n}\left(y_i - \gamma_i\right)^2$, applies to a set of data points n.

6) Update and adjust the model so that the weak model's predictions combine with the prior model's predictions, resulting in an updated set of predictions using Equations 8 and 9.

$$\gamma_m = \arg\min_{\gamma} \sum_{i=1}^{n} L\left(y_i, F_{m-1}\left(x_i\right) + \gamma h_m\left(x_i\right)\right) \tag{8}$$

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \alpha * h_m(\mathbf{x}) \tag{9}$$

In the given context, m denotes the total number of weak learners (e.g., a decision tree), $h_m left(x_i right)$ represents the residual-based constructed model, and α signifies the learning rate.

7) The steps from 4 to 6 are repeated iteratively until the model achieves its highest accuracy (i.e., a negligible residual error has been reached) or until no more enhancements can be achieved.
8) A robust predictive model is produced by adding all of the weak models' predictions to arrive at the final prediction.

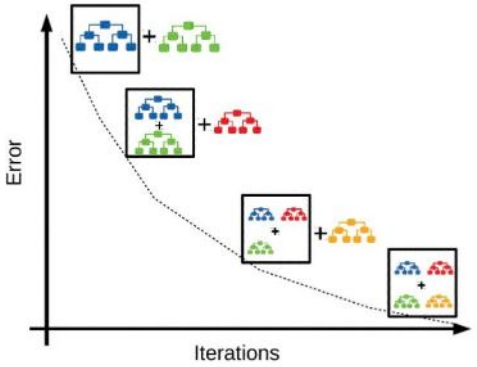

Fig. 7. GB classifier model.

IV. Classification Process

Machine learning faces a significant challenge in classifying people with diabetes, which requires a multi-step data preparation process. The process includes data collection, cleaning, scaling, feature selection, data partitioning (into

training and testing sets), and algorithm utilization. Fig. 8 illustrates the complete classification process for handling the Pima data [21] classification problem.

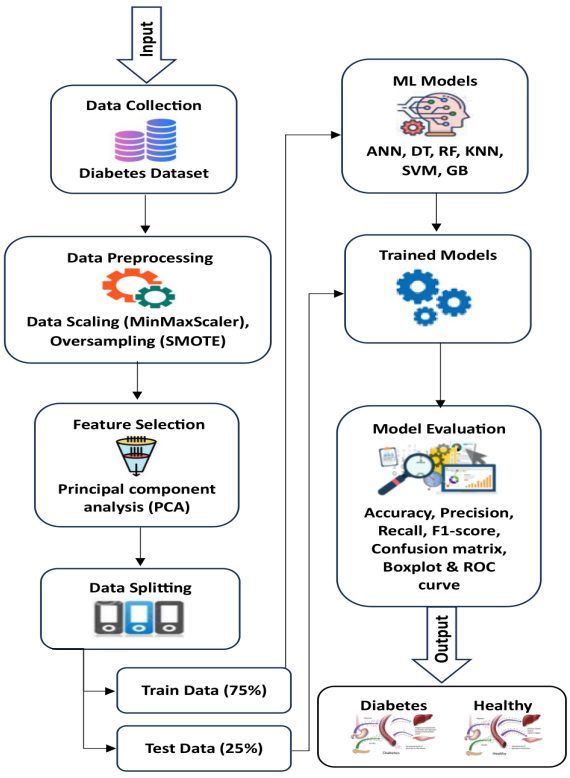

Fig. 8. The workflow of the classification process for diabetes.

TABLE I. SAMPLES OF THE PIMA INDIANS DIABETES

Preg.	Gluc.	BP	Skin Th.	Insulin	BMI	Pedig.	Age	Label
6	148	72	35	0	33.6	0.627	50	1
1	85	66	29	0	26.6	0.351	31	0
8	183	64	0	0	23.3	0.672	32	1
1	89	66	23	94	28.1	0.167	21	0
0	137	40	35	168	43.1	2.288	33	1
5	116	74	0	0	25.6	0.201	30	0
3	78	50	32	88	31	0.248	26	1
10	115	0	0	0	35.3	0.134	29	0
2	197	70	45	543	30.5	0.158	53	1
8	125	96	0	0	0	0.232	54	1
4	110	92	0	0	37.6	0.191	30	0
10	168	74	0	0	38	0.537	34	1
10	139	80	0	0	27.1	1.441	57	0
1	189	60	23	846	30.1	0.398	59	1
5	166	72	19	175	25.8	0.587	51	1

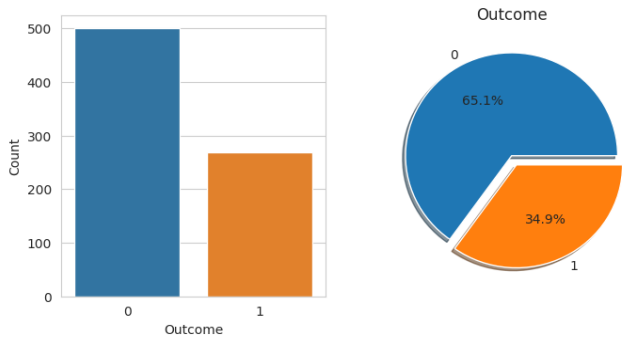

Fig. 9. Distribution of the dataset (0: non-diabetic, 1: diabetic).

A. Pima Indian Diabetes Dataset

The Pima Indian Diabetes dataset is a popular public resource frequently employed for diabetes-related classification issues [34]. The dataset comprises information from 768 female Pima Indians aged 21 and older, initially gathered by the National Institute of Diabetes and Digestive and Kidney Diseases.

Among the numerous features of the diabetes data collection are the following: age, pedigree function, pregnancy, blood pressure, skin thickness, insulin, body mass index, and the output class or label. The dataset is extensively utilized in machine learning applications for evolving predictive models for the diagnosis of diabetes [35], [36]. Table I shows a sample of the data set. The diabetic dataset has 768 records, with 500 and 268 records of non-diabetic and diabetic cases, respectively. As seen in Fig. 9, the dataset exhibits an imbalance.

In Fig. 10, we present a heat map demonstrating the correlation between the sample data label and the various variables in the adopted dataset. Fig. 11 shows the box plot for various dataset features. The Distribution of a dataset and any hidden outliers can be better understood using boxplots.

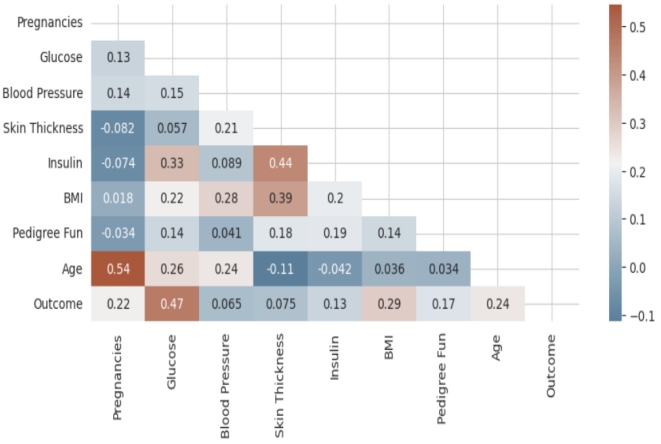

Fig. 10. A heatmap showing the correlation between various features in the dataset.

B. Oversampling

Creating an accurate machine learning model when the data is imbalanced is challenging. One issue arises from the possibility that the model can learn the class with more data records than the other. It is essential to strike a balance between classes as much as possible. Imbalanced data can lead to biased models and poor performance in the minority class. To address this issue, oversampling techniques can be used

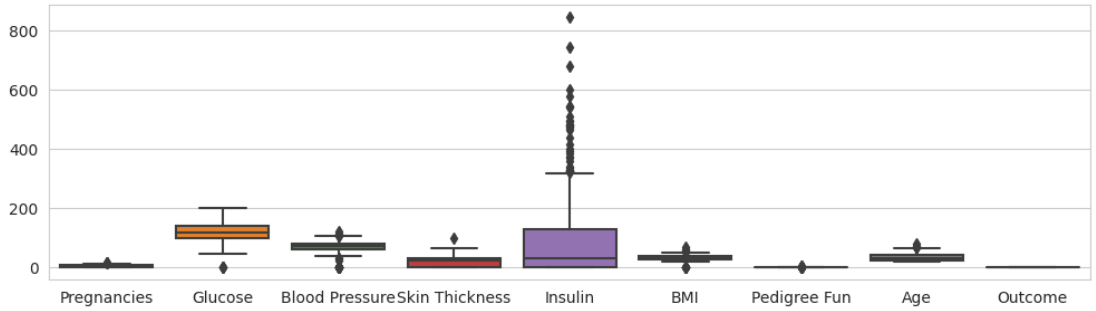

Fig. 11. Box Plot for various attributes of the pima indian diabetes dataset.

to balance the dataset and improve model performance [37], [38]. However, oversampling can also lead to overfitting if not done carefully. Our study addressed the imbalance using the Synthetic Minority Oversampling Technique (SMOTE). The basic concept of SMOTE is to generate synthetic data points between each sample from the minority class and its "k" nearest neighbors according to Eq. 10.

$$x_{syn} = x_i + \gamma\left(x_{knn} - x_i\right) \tag{10}$$

Where x_{syn}, and x_{knn} are the synthetic data point and the closest neighbor to the point x_i, respectively. γ is a randomly generated number between 0 and 1. Subsequently, following the oversampling process, the number of instances in both classes becomes equal.

C. Feature Selection

An essential method for machine learning is feature selection. This strategy can improve model performance, reduce the time required for training, boost interpretability, and reduce overfitting. Selecting the most pertinent features enhances the machine learning models' accuracy. This is because the model can focus on the most critical predictors rather than being distracted by noisy or irrelevant features. Therefore, Principal component analysis (PCA) can be utilized for feature selection in this study.

To extract the most variation from the data, the PCA approach converts the initial features into a new collection of independent features known as principal components (see Algorithm 1). In this research, the top five features are selected for further processing, which are "Pregnancies," "Glucose," "BMI," "Pedigree Function," and "Age."

D. Data Scaling

Data scaling is an essential preprocessing step in machine learning that can improve machine learning models' performance, convergence, and efficiency. Scaling methods depend on the nature of the data and the machine-learning model's requirements.

Many data scaling methods are reported in the literature [39]. MinMaxScaler method is among these methods. This method scales data features to a domain between 0 and 1. Eq. 11 shows how the MinMaxScaler method works.

Algorithm 1 Principal Component Analysis (PCA)

Input: Training data X, number of desired principal components k.
Output: Transformed data X'
Step 1: Calculate the mean vector $\overline{X}$ for each feature in X using $\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$, where n is the number of samples in X and X_i is the i-th sample in X.
Step 2: Compute the covariance matrix C for X as $C = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X})(X_i - \overline{X})^T$.
Step 3: Obtain the eigenvectors V and eigenvalues λ of C using $\lambda, V = \text{eig}(C)$, where $\text{eig}(C)$ returns the eigenvalues and eigenvectors of C.
Step 4: Build the transformation matrix W by picking the top k eigenvectors and sorting them in descending order by eigenvalue.
Step 5: Transform the data using the transformation matrix W as $X' = XW$.
Step 6: Return the transformed data X'.

$$f_{scaled} = \frac{f - f_{min}}{f_{max} - f_{min}} \tag{11}$$

where the feature's minimal value, its maximum value, and its scaled value are denoted by f_{min}, f_{max}, and f_{scaled}, respectively.

E. Evaluation Metrics

Various evaluation metrics can be used to assess the utilized diagnostic (i.e., classification) models [40] based on the actual and predicted results. As an illustration, consider the case when the classifier's output and the actual value are positive; use the notation TP. Meanwhile, the notation TN indicates that the real value and the classifier's output are negative. If the classifier's result is opposed to the actual value, this indicates either a FP or FN. Various metrics for evaluation were calculated based on these values.

- Accuracy (Acc): It indicates the percentage of correct predictions compared to the entire number of predictions, denoted by T ($T = TP + FP + TN + FN$).

$$Acc = \frac{TP + TN}{T} \tag{12}$$

- Precision (P): It denotes the proportion of positive predictions that were accurate to the overall count of positive predictions.

$$P = \frac{TP}{TP + FP} \tag{13}$$

- Recall (R): It quantifies the proportion of correctly predicted positive cases relative to the total number of positive cases.

$$R = \frac{TP}{TP + FN} \tag{14}$$

- F-measure (F): It is a single-value representation of the well-balanced combination of recall and precision.

$$F - measure = \frac{2 \times P \times R}{P + R} \tag{15}$$

- At various classification thresholds, the Area under the Receiver Operating Characteristic (ROC) Curve shows how the true positive and false positive rates relate to one another. To find the Area under the curve (AUC-ROC), we integrate the TP rate from 0 to 1 (where FPR is the independent variable).

V. Experimental Results

Over the past several years, diabetes has become the leading cause of mortality among humans. The prevalence of this disease is on the rise due to several factors, including unhealthy dietary habits and the availability of unhealthy food options. Early detection of diabetes can aid in clinical management decision-making. In our research, we have utilized various evaluation measures to determine and quantify the performance of our ensemble of algorithms, which include ANN, DT, RF, KNN, SVM, and GB classifiers. These techniques were tested and evaluated on the Pima Indian Diabetes Dataset. However, picking the most effective one was a top priority, so we measured each algorithm accurately, even after five iterations, to find which one was superior. The results of each algorithm are illustrated below.

A. ANN Results

In our research, we investigated different designs of Artificial Neural Networks (ANN) with varying complexities to achieve the best classification results. Many benefits may be achieved by increasing the number of neurons in an ANN's hidden layers, as listed below:

- It enhances the model's capacity to learn complex patterns and relationships in the data.
- It can lead to better fitting the model to the data, resulting in improved accuracy and lower error rates.
- A more extensive network can better generalize to unseen data as it has learned a more comprehensive representation of the underlying patterns in the data.

Table II shows three different ANN models were considered, each with varying numbers of neurons in its hidden layer. Furthermore, Fig. 12 shows the convergence curve of the three developed ANN models. From Fig. 12, the ANN model with several neurons equal to 20 and 10 at hidden layers 1 and 2, respectively, has achieved superior convergence.

TABLE II. Various ANN Model Structures

ANN Models	Input	Hidden (1)	Hidden (2)	Output
Model #1	5	5	2	1
Model #2	5	10	5	1
Model #3	5	20	10	1

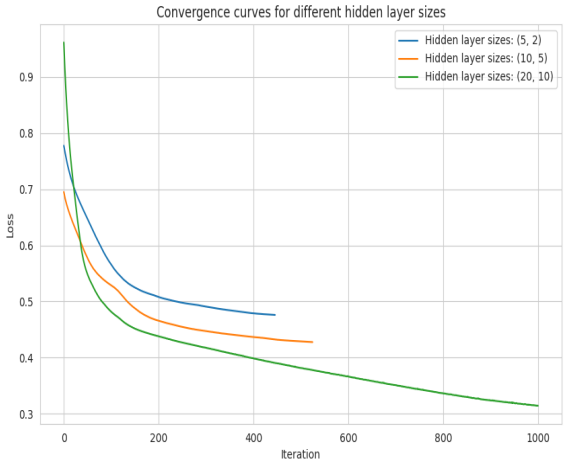

Fig. 12. Convergence curves of the three ANN models.

The confusion matrix summarizes predicted against actual classification results, making it easy to assess a classification model's performance and identify its weak spots. The corresponding confusion matrix for the superior ANN model (Model #3) is shown in Fig. 13. Table III lists the results of the developed ANN models concerning evaluation metrics for both the training and testing datasets to assess the ANN models' efficiency.

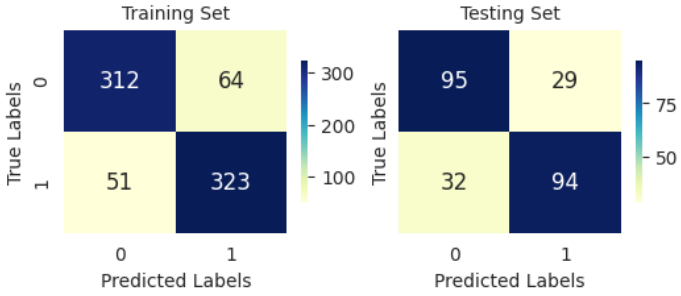

Fig. 13. Confusion matrix for ANN.

Regarding the classification results, the model trained and tested had 323 and 94 diabetic patients predicted, respectively, as TP. However, the model was incorrectly classified as diabetic, with 51 positive data points belonging to a negative class, and the predicted values, denoted as FN, were falsely predicted.

Based on the results listed in Table III, the first model

TABLE III. THE PERFORMANCE OF DIFFERENT ADOPTED ANN MODEL ARCHITECTURES

Model No.	Train				Test			
	Acc	P	R	F	Acc	P	R	F
Model #1	0.748	0.736573	0.770053	0.752941	0.744	0.738462	0.761905	0.75
Model #2	0.797333	0.776119	0.834225	0.804124	0.752	0.766667	0.730159	0.747967
Model #3	0.846667	0.834625	0.863636	0.848883	0.756	0.764228	0.746032	0.75502

may have been overfitted because its accuracy score was lower on the testing dataset than on the training dataset. Although the second model performed better on both train and test datasets, it had difficulty generalizing to the testing dataset due to lower accuracy, recall, and F-measure scores. The third model had the highest accuracy score on the training dataset but a significantly lower accuracy score on the testing dataset, indicating possible overfitting. However, the precision, recall, and F-measure are better than other models in testing.

B. DT Results

The decision tree is an effective tool for interpretation, as it can be presented visually and comprehended quickly, even by those without expertise in the field. It follows a similar process to a physician's diagnostic criteria for identifying diseases. The decision tree algorithm employs a greedy approach for recursive binary splitting, selecting the optimal split at each step rather than anticipating future steps and choosing a split that may lead to a more optimal tree. This allows patients to undergo laboratory tests in the sequence of the nodes and potentially stop the testing process earlier if they meet certain conditions [41].

Fig. 14 illustrates the decision tree used for diabetes classification. The tree is composed of nodes, which are further divided into sub-nodes. The parent node has one or more child nodes. In this case, the tree has 13 nodes, with Glucose being the root node. Then, we split the tree into another branch whose root node is 'Age,' with BMI as the child node. The tree's root node can be interpreted as "Is the glucose level less than 43 (mg/dl)?". If the patient's glucose level is less than 43 (mg/dl), the sub-tree is followed to check the patient's age.

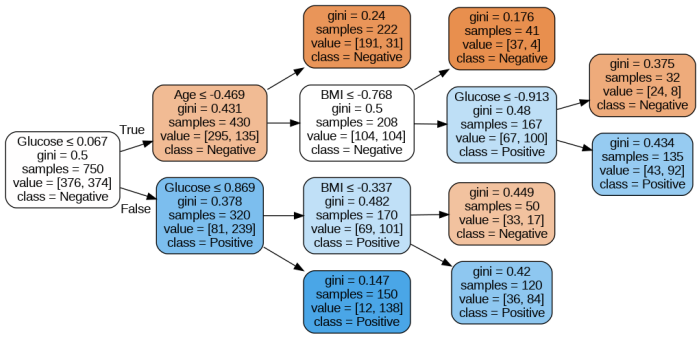

Fig. 14. Diabetic model using pruned DT.

We utilized the Minimal Cost Complexity Pruning (CCP) approach to avoid overfitting and control the decision tree's complexity. This method adds a regularization parameter to the criterion used to divide nodes in the tree. The parameter, α_{ccp}, governs the balance between the tree's complexity (i.e., its depth and breadth) and its capability to fit the training data. By increasing the α_{ccp}, the algorithm can reduce the tree's depth and breadth, effectively curbing overfitting. Selecting an appropriate evaluation metric is crucial in building effective classification models. The accuracy of our model's predictions is evaluated by examining the confusion matrix shown in Fig. 15.

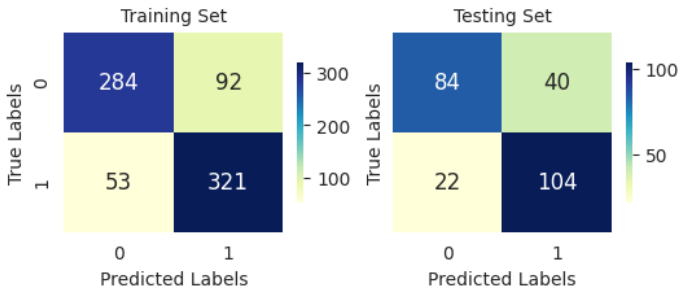

Fig. 15. Confusion matrix for DT.

The model correctly classified 604 out of 750 samples in training and 188 out of 250 in testing. The number of samples was classified as FP equals 92 in training and 40 in testing (i.e., incorrect predictions).

C. RF Results

As an ensemble approach, a random forest uses many decision trees to arrive at one prediction. Since each decision tree is constructed separately, the random forest may be enhanced by pruning each tree before combining them.

"Bagging" represents the ensemble learning process known as "bootstrap aggregating." This method uses bootstrapping to divide the training data into B separate sets and then builds a new decision tree for each iteration. The output is then aggregated to give the class with the most votes from the B trees. Bagging reduces variance and helps to avoid overfitting since it aggregates multiple trees. Random forests are a modified version of bagging that builds B number of de-correlated sample trees. Like bagging, random forest builds B decision trees on bootstrapped training samples. The difference is that random forest builds de-correlated trees.

There is no specific algorithm to prune a random forest tree. Nonetheless, one may indirectly affect the amount of overfitting by controlling the tree complexity by RF algorithm hyper-parameter adjustment. Furthermore, cost complexity pruning can be used to post-prune the individual decision trees. Fig. 16 shows the pruned RF tree.

The confusion matrix that was generated using the RF approach is also shown in Fig. 17. The matrix demonstrates superior performance evaluation in training and testing the Pima diabetic dataset. RF achieved a level of accuracy in

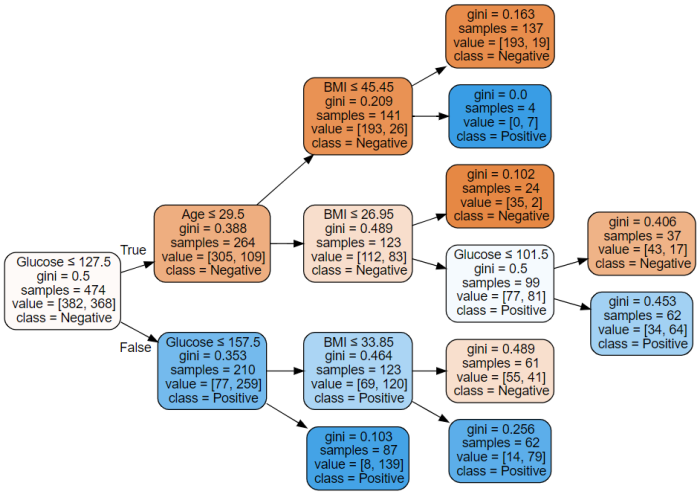

Fig. 16. Diabetic model using pruned RF tree.

patient classification of 364 during the training phase and 101 during the testing phase (TP). The number of correctly classified negative class data points (TN) during testing is 91, while during training, it is 345.

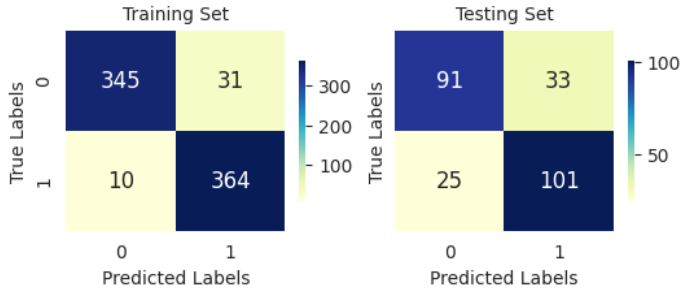

Fig. 17. Confusion matrix for RFC.

D. KNN Results

One of the well-known machine learning algorithms is KNN. It uses a variety of distance metrics. The fact that KNN does not instantaneously start learning from the training set has prompted some to refer to it as a lazy learner algorithm. However, it retains the dataset and performs a calculation while doing classification. The data points are classified accordingly based on the value of k, which determines the number of data points chosen from the nearest neighbors. Overall, the KNN algorithm operated into two primary phases (training phase and classification phase). In the training phase, the algorithm keeps track of the features of the training samples and matches class labels. In the classification phase, the test samples are classified based on the value of k and by calculating the feature similarity. A voting procedure takes place to conclude the classification process ultimately. The value of k determines how well the KNN algorithm works. Based on our model, $k = 5$ for better performance.

Fig. 18 shows a visualization of three (e.g., 'Pregnancies,' 'BMI,' 'Age') of best-selected features with each other at $k = 5$ according to the target class. The generated confusion matrix from the KNN classifier is shown in Fig. 19. For example, "the number of patients that are healthy (i.e., negative) and are predicted as a diabetic disease (i.e., positive) equal to 78 in training and 36 in testing."

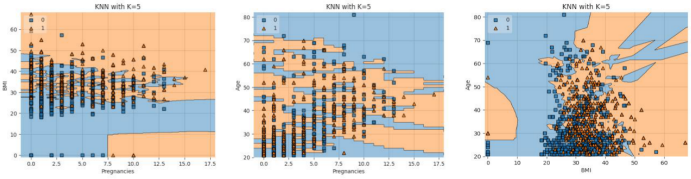

Fig. 18. Feature visualization (Pregnancies, BMI, Age at $k = 5$.

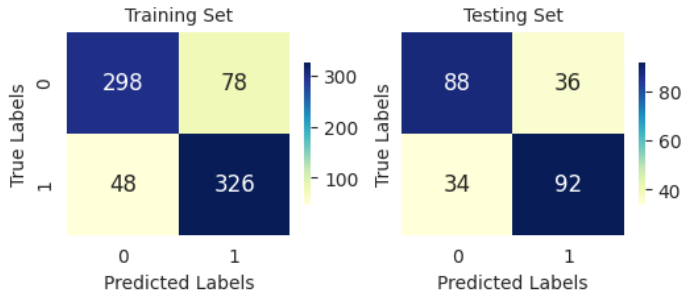

Fig. 19. Confusion matrix for KNN.

E. SVM Results

Support vector machines (SVMs) are standard supervised ML algorithms. The SVM classifier aims to locate the hyperplane with the most significant margin separating the classes. The optimal hyperplane is located by finding the maximum point of the hyperplane's margin. Dealing with high-dimensional data requires kernel functions to transform the input space into the feature space. The Radial Basis Function (RBF) is a popular kernel function that employs the similarity between the two points as presented in Eq. 16.

$$K\left(X_1, X_2\right) = \exp\left(-\frac{\|X_1 - X_2\|^2}{2\sigma^2}\right) \tag{16}$$

where σ is a hyperparameter and $\|X_1 - X_2\|$ is the L_2 norm distance between two data points X_1 and X_2.

The SVM's performance is impacted by two hyperparameters: C, a punishment parameter, and $gamma$, a control parameter. A small number of C leads to a decision boundary with a large margin higher chosen at the expense of more misclassification. On the contrary, a more significant value of C minimizes the misclassified samples with a smaller margin due to the high penalty. The $gamma$ parameter specifies how much a single training sample may be influenced; low values indicate 'far' and large values 'close.' In our case, the values of C and $gamma$ are set to the default values to produce the best results according to our dataset. Fig. 20 shows the decision boundary of the target class in both training and test of diabetic data.

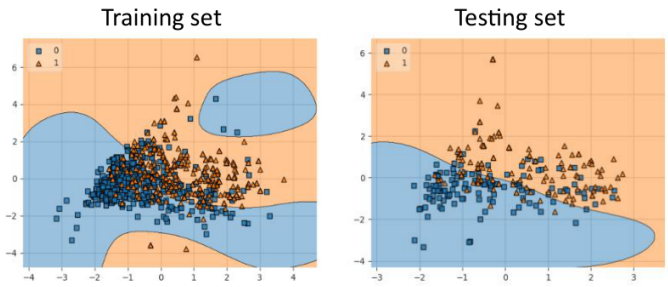

Fig. 20. Decision boundary at training and testing of SVM.

The confusion matrix resulting from the evaluation of the SVM on the diabetes dataset is depicted in Fig. 21. It has proven its efficiency in correctly classifying 602 instances (positive and negative) out of 750 in the training phase, while in the testing phase, it correctly classified 191 instances out of 250.

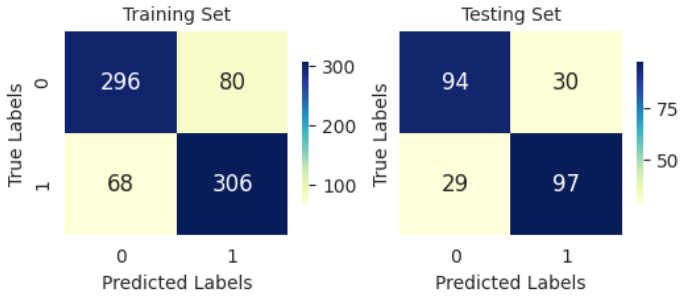

Fig. 21. Confusion matrix for SVM.

F. GB Results

As a subset of ensemble learning, boosting algorithms repeatedly train a series of weak models to improve the accuracy of predictions. Each model addresses the weaknesses of its predecessors until a final robust model has been reached. Boosting should specify a weak model (e.g., decision tree, random forest) as a learner to improve it.

Gradient boosting is a technique that combines many weak prediction models, often decision trees, in a sequential manner to create a robust predictive model. GB iteratively improves the algorithm based on the loss function [42] (i.e., minimizing the residual errors) by fitting each new weak learner to the residuals of the previous model. To simplify the gradient-boosting classifier approach, one has to tweak parameters like the learning rate and the number of estimators. The learning rate determines the relative significance of each new tree in the ensemble, while the number of estimators determines the overall number of trees incorporated into the model. Maintaining a balance between these two parameters is necessary to prevent overfitting.

Moreover, pruning the tree can influence the optimization of gradient boosting by improving the generalization and reducing the overfitting. Fig. 22 shows the initial estimator (i.e., DT) with a depth equal to 3 for the trained GB classifier. Due to the ensemble's overall classifier nature, each tree in the ensemble calculates values in the floating point value format. Consequently, the resulting confusion matrix for training and testing is shown in Fig. 23. The GB classifier has achieved reasonable classification results in TP, which" reached up to 346 and 100 instances in training and testing, respectively. At the same time, it misclassifies 128 instances over the train and test.

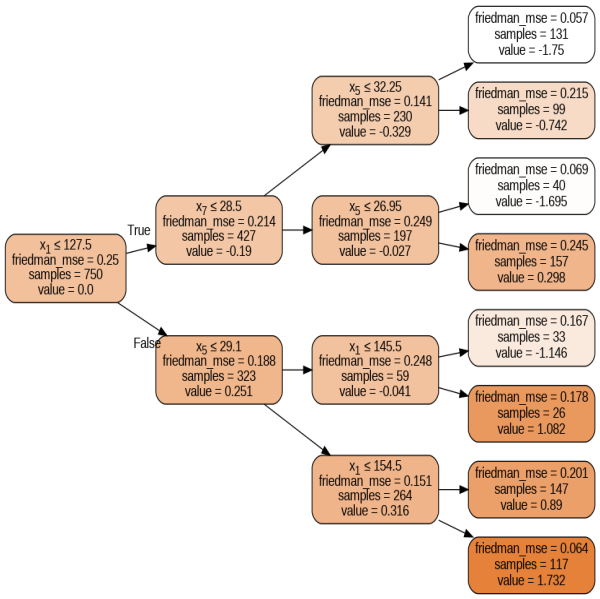

Fig. 22. Diabetic model using pruned GB classifier.

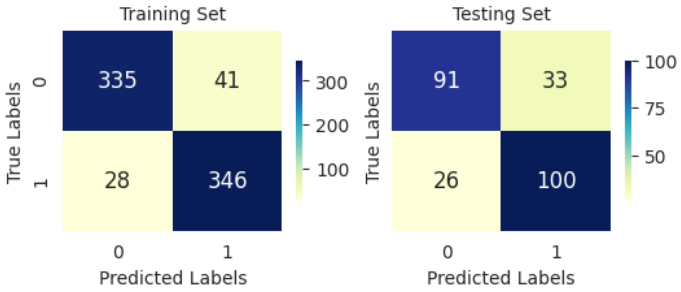

Fig. 23. Confusion matrix for GBC.

VI. Performance Analysis

Table IV displays the results of all the machine learning algorithms used in various assessment measures, with the top-performing algorithms shown in bold.

The RF model has achieved a superior result in terms of accuracy when compared with other algorithms in training and testing, reaching up to 95% and 77%, respectively. Although the ANN and DT performed impressively on the testing set, showcasing high values for precision and recall, they still achieved lower F-measure values than RF. According to other compared algorithms, the GB got higher accuracy (91%) than others in training. However, the SVM has achieved the lowest accuracy values in training but a reasonable value in testing.

Analyzing Table IV, it is evident that the Random Forest classifier (RF) achieved the highest training and testing

TABLE IV. COMPARATIVE PERFORMANCE OF ML ALGORITHMS ON VARIOUS MEASURES

ML Algorithm	Train				Test			
	Acc	*P*	*R*	*F*	*Acc*	*P*	*R*	*F*
ANN	0.868	0.857143	0.882353	0.869565	0.768	**0.788136**	0.738095	0.762295
DT	0.806667	0.77724	0.858289	0.815756	0.752	0.722222	**0.825397**	0.77037
RF	**0.945333**	**0.921519**	**0.973262**	**0.946684**	**0.768**	0.753731	0.801587	**0.77692**
KNN	0.832	0.806931	0.871658	0.838046	0.72	0.71875	0.730159	0.724409
SVM	0.802667	0.792746	0.818182	0.805263	0.764	0.76378	0.769841	0.766798
GB	0.908	0.894057	0.925134	0.90933	0.764	0.75188	0.793651	0.772201

accuracy among the evaluated algorithms. Additionally, RF displayed notable precision, recall, and F-measure on the training and testing sets. These results suggest that the RF model performs effectively on the given dataset and exhibits solid predictive capabilities.

Furthermore, Table V listed the total number of correctly (CC) and mis-correctly (MC) classified instances in each comparative algorithm's training and testing phases. The Random Forest algorithm counted the most prominent correctly classified instances against other algorithms, with 709 out of 750 in training and 192 out of 250 in testing. It achieved the lowest value of mis-correctly instances in training and testing, with 41 out of 750 in training and 58 out of 250 in testing.

TABLE V. COMPARATIVE PERFORMANCE OF ML ALGORITHMS OVER CLASSIFICATION INSTANCES

ML Algorithm	Train		Test	
	# CC	# MC	# CC	# MC
ANN	635	115	189	61
DT	605	145	188	62
RF	**709**	**41**	**192**	**58**
KNN	624	126	180	70
SVM	599	148	191	59
GB	681	69	191	59

Furthermore, Fig. 24 shows the Boxplot of the six compared ML algorithms. The ANN and SVM classifier's box plot reveals a positively skewed, which indicates a more significant frequency of highly rated scores in the data (i.e., a slight deviation from the data's central tendency). However, the GBC and DT boxplots show the median closer to the upper quartile, indicating a negative skew with low-valued scores occurring more frequently in the data classified by the ANN. Concerning overall data distribution, the RF classifier is superior to the normal Distribution. With more scattered data points and a smaller range, the RF classifier indicates less variability. RF appears more robust and stable among the ML models examined, as evidenced by its box plot characteristics.

Fig. 25 represents all classification algorithms' ROC curve (AUC) area. It reveals the random probability that a positive instance would receive a higher score than a negative one. A classification method's ability to discriminate between classes is directly proportional to the AUC value, meaning that a higher AUC indicates better performance. The random forest classifiers had the most excellent ROC value of 0.95 compared to ANN, DT, RF, SVM, and GB algorithms.

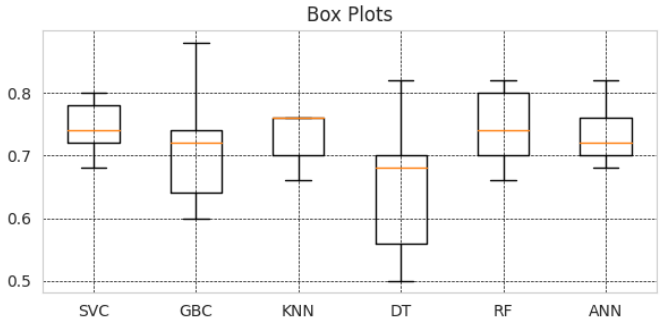

Fig. 24. Comparision of utilized ML models (BoxPlot Curves).

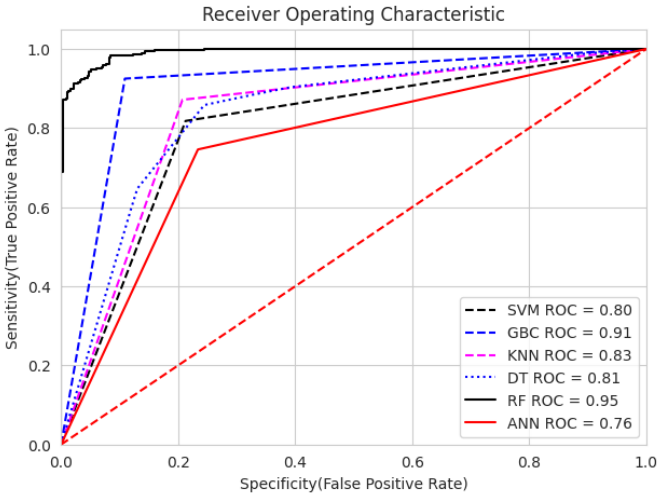

Fig. 25. Comparision of utilized ML models (ROC curves).

VII. CONCLUSION

The study evaluated six employed ML algorithms, ANN, KNN, DT, RF, GB, and SVM, to assess their performance in classifying diabetes. By utilizing an oversampled dataset, we applied various machine learning models and identified five crucial features - "Pregnancies," "Glucose," "BMI," "Pedigree Function," and "Age" - for diabetes classification. Our results indicated that the RF model had the best level of accuracy in diagnosing diabetes. The developed system ensures consistent predictions, enabling more practical application to other diseases. For future research, it would be beneficial to investigate the potential advantages of utilizing algorithm combinations instead of only depending on the top-performing algorithm within the ensemble.

REFERENCES

[1] A. Steele, "Projected diabetes rates in america." [Online]. Available: https://psydprograms.org/projected-diabetes-rates-in-america/

[2] J. Smith, M. Johnson, and D. Williams, "Diabetes mellitus: a comprehensive review," *Journal of Diabetes Research*, vol. 2021, pp. 1–15, 2021.

[3] M. A. Rogers, B. S. Rogers, and T. Basu, "Prevalence of type 1 diabetes among people aged 19 and younger in the united states," *Preventing Chronic Disease*, vol. 15, p. 180323, 2018.

[4] K. Bond and A. Sheta, "Medical data classification using machine learning techniques," *International Journal of Computer Applications*, vol. 183, pp. 1–8, 06 2021.

[5] K. Patel, K. Kalia, and N. M. Patel, "Challenges and opportunities in diabetes research: a machine learning perspective," *Current diabetes reviews*, vol. 14, no. 1, pp. 15–22, 2018.

[6] K. Al-Rubeaan, A. Al-Manaa, H. K. Al-Qumaidi, A. H. El-Malki, M. A. Nasir, A. M. Al-Dhukair, and E. S. Ibrahim, "Diabetes mellitus, hypertension and obesity—common multi-factorial disorders in saudis," *Journal of family & community medicine*, vol. 22, no. 1, p. 1, 2015.

[7] K. J. Gaulton, T. C. Nammo, T. Pasquali, N. M. Matqevalli, H. Benazzo, P. A. Ostrowski, M. L. Johnson, J. Dannenberg, M. L. Kameswaran, M. E. Brandt *et al.*, "A map of open chromatin in human pancreatic islets," *Nature genetics*, vol. 42, no. 3, pp. 255–259, 2010.

[8] I. Kavakiotis, O. Tsave, A. Salifoglou, N. Maglaveras, I. Vlahavas, and I. Chouvarda, "Machine learning and data mining methods in diabetes research," *Computational and structural biotechnology journal*, vol. 15, pp. 104–116, 2017.

[9] S. K. Roy, A. Ali, M. Radeef, A. Alzahrani, and N. Khan, "Machine learning-based diabetes prediction models: a review," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 9, pp. 8951–8974, 2021.

[10] S.-J. Xia, B.-Z. Gao, S.-H. Wang, D. S. Guttery, C.-D. Li, and Y.-D. Zhang, "Modeling of diagnosis for metabolic syndrome by integrating symptoms into physiochemical indexes," *Biomedicine & Pharmacotherapy*, vol. 137, p. 111367, 2021.

[11] A. D. Association, "Classification and Diagnosis of Diabetes," *Diabetes Care*, vol. 40, pp. S11–S24, 12 2016.

[12] M. Z. Alam, M. S. Rahman, and M. S. Rahman, "A random forest based predictor for medical data classification using feature ranking," *Informatics in Medicine Unlocked*, vol. 15, p. 100180, 2019.

[13] A. Sheta, H. Turabieh, M. Braik, and S. R. Surani, "Diagnosis of obstructive sleep apnea using logistic regression and artificial neural networks models," in *Proceedings of the Future Technologies Conference*. Springer, 2019, pp. 766–784.

[14] A. Sheta, H. Turabieh, T. Thaher, J. Too, M. Mafarja, M. S. Hossain, and S. R. Surani, "Diagnosis of obstructive sleep apnea from ECG signals using machine learning and deep learning classifiers," *Applied Sciences*, vol. 11, no. 14, 2021.

[15] C. Haberfeld, A. Sheta, M. S. Hossain, H. Turabieh, and S. Surani, "SAS mobile application for diagnosis of obstructive sleep apnea utilizing machine learning models," in *2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, 2020, pp. 0522–0529.

[16] I. Aiyer, L. Shaik, A. Sheta, and S. Surani, "Review of application of machine learning as a screening tool for diagnosis of obstructive sleep apnea," *Medicina*, vol. 58, no. 11, 2022.

[17] S. Afzali and O. Yildiz, "An effective sample preparation method for diabetes prediction," *The International Arab Journal of Information Technology*, vol. 15, no. 6, November 2018.

[18] M. K. Hossain, S. M. Ehsan, K. Abdullah-Al-Mamun, and S. Baharun, "Machine learning techniques for diabetes decision support: A review," *Journal of medical systems*, vol. 43, no. 9, p. 268, 2019.

[19] A. F. Sheta, S. E. M. Ahmed, and H. Faris, "A comparison between regression, artificial neural networks and support vector machines for predicting stock market index," *International Journal of Advanced Research in Artificial Intelligence*, vol. 4, no. 7, 2015. [Online]. Available: http://dx.doi.org/10.14569/IJARAI.2015.040710

[20] B. Byers and A. Sheta, "Design of convolutional neural networks for fish recognition and tracking," *Artificial Intelligence and Machine Learning AIML*, vol. 22, no. 1, pp. 1–9, 5 2022.

[21] V. Chang, J. Bailey, Q. Xu, and Z. Sun, "Pima indians diabetes mellitus classification based on machine learning (ml) algorithms," *Neural Computing and Applications*, 03 2022.

[22] S. Ruder, "An overview of gradient descent optimization algorithms," *CoRR*, vol. abs/1609.04747, 2016.

[23] J. Fürnkranz, *Decision Tree*. Boston, MA: Springer US, 2010, pp. 263–267.

[24] A. Saud, S. Shakya, and B. Neupane, "Analysis of depth of entropy and gini index based decision trees for predicting diabetes," *Indian Journal of Computer Science*, vol. 6, pp. 19–28, 01 2022.

[25] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[26] U. Bollikonda, "Random forest machine learning algorithm," 2021, accessed: December 8, 2021. [Online]. Available: https://medium.com/@uma.bollikonda/random-forest-machine-learning-algorithm-401bdcd7a0b8

[27] A. B. Omar, S. Huang, A. A. Salameh, H. Khurram, and M. Fareed, "Stock market forecasting using the random forest and deep neural network models before and during the covid-19 period," *Frontiers in Environmental Science*, vol. 10, 2022. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fenvs.2022.917047

[28] S. Du, D. Hao, and X. Li, "Research on stock forecasting based on random forest," in *2022 IEEE 2nd International Conference on Data Science and Computer Application (ICDSCA)*, 2022, pp. 301–305.

[29] P. Josso, A. Hall, C. Williams, T. Le Bas, P. Lusty, and B. Murton, "Application of random-forest machine learning algorithm for mineral predictive mapping of fe-mn crusts in the world ocean," *Ore Geology Reviews*, vol. 162, p. 105671, 2023.

[30] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends," *Neurocomputing*, vol. 408, pp. 189–215, 2020.

[31] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," *The Annals of Statistics*, vol. 36, no. 3, pp. 1171 – 1220, 2008. [Online]. Available: https://doi.org/10.1214/009053607000000677

[32] E. G. Dada, J. S. Bassi, H. Chiroma, S. M. Abdulhamid, A. O. Adetunmbi, and O. E. Ajibuwa, "Machine learning for email spam filtering: review, approaches and open research problems," *Heliyon*, vol. 5, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:189930761

[33] N. Aziz, E. Akhir, A. P. D. I. Aziz, J. Jaafar, M. H. Hasan, and A. Abas, "A study on gradient boosting algorithms for development of ai monitoring and prediction systems," 10 2020, pp. 11–16.

[34] R. Saxena, S. Sharma, and M. Gupta, "Analysis of machine learning algorithms in diabetes mellitus prediction," *Journal of Physics: Conference Series*, vol. 1921, p. 012073, 05 2021.

[35] 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, 2021.

[36] J. Chaki, S. Thillai Ganesh, S. 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, no. 6, Part B, pp. 3204–3225, 2022.

[37] A. Moreo, A. Esuli, and F. Sebastiani, "Distributional random oversampling for imbalanced text classification," in *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 805–808.

[38] T. Wongvorachan, S. He, and O. Bulut, "A comparison of undersampling, oversampling, and smote methods for dealing with imbalanced classification in educational data mining," *Information*, vol. 14, no. 1, 2023.

[39] M. M. Ahsan, M. A. P. Mahmud, P. K. Saha, K. D. Gupta, and Z. Siddique, "Effect of data scaling methods on machine learning algorithms and model performance," *Technologies*, vol. 9, no. 3, 2021.

[40] M. Ucar, "Classification performance-based feature selection algorithm for machine learning: P-score," *IRBM*, vol. 41, 02 2020.

[41] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning*. Springer, 2013, vol. 103.

[42] J. Friedman, "Stochastic gradient boosting," *Computational Statistics & Data Analysis*, vol. 38, pp. 367–378, 02 2002.