

## Research Article

# An Efficient Prediction System for Diabetes Disease Based on Deep Neural Network

Tawfik Beghriche <sup>1</sup>, Mohamed Djerioui <sup>2</sup>, Youcef Brik <sup>2</sup>, Bilal Attallah <sup>2</sup>,  
and Samir Brahim Belhaouari <sup>3</sup>

<sup>1</sup>Department of Electronics, Faculty of Technology, University of M'sila, Algeria

<sup>2</sup>LASS Laboratory, Faculty of Technology, University of M'sila, Algeria

<sup>3</sup>Division of Information and Computing Technology, College of Science and Engineering, Hamad Bin Khalifa University, Education City, Qatar Foundation, Doha, Qatar

Correspondence should be addressed to Samir Brahim Belhaouari; [sbelhaouari@hbku.edu.qa](mailto:sbelhaouari@hbku.edu.qa)

Received 19 August 2021; Accepted 10 November 2021; Published 17 December 2021

Academic Editor: Hocine Cherifi

Copyright © 2021 Tawfik Beghriche et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

One of the main reasons for disability and premature mortality in the world is diabetes disease, which can cause different sorts of damage to organs such as kidneys, eyes, and heart arteries. The deaths by diabetes are increasing each year, so the need to develop a system that can effectively diagnose diabetes patients becomes inevitable. In this work, an efficient medical decision system for diabetes prediction based on Deep Neural Network (DNN) is presented. Such algorithms are state-of-the-art in computer vision, language processing, and image analysis, and when applied in healthcare for prediction and diagnosis purposes, these algorithms can produce highly accurate results. Moreover, they can be combined with medical knowledge to improve decision-making effectiveness, adaptability, and transparency. A performance comparison between the DNN algorithm and some well-known machine learning techniques as well as the state-of-the-art methods is presented. The obtained results showed that our proposed method based on the DNN technique provides promising performances with an accuracy of 99.75% and an F1-score of 99.66%. This improvement can reduce time, efforts, and labor in healthcare services as well as increasing the final decision accuracy.

## 1. Introduction

Diabetes is a noncommunicable chronic disease that disrupts the body's natural blood glucose concentration management with disorders of carbohydrate, fats, and protein metabolism due to imperfections in insulin secretion, insulin action, or both of them [1–5]. The chronic hyperglycemia of diabetes is associated with long-term damage, dysfunction, and failure of different organs, especially the eyes, kidneys, nerves, heart, and blood vessels [1, 6, 7]. According to the World Health Organization, an estimation of about 422 million people worldwide have diabetes, and this number is expected to grow up to 693 million by 2045, and 1.6 million deaths are directly attributed to diabetes each year [8]. On the other hand, the worldwide economic expenditures for

diabetes were estimated to be approximately USD 760 billion, and it is expected to reach over USD 802 billion in 2040 [9]. Day by day, both the number of cases and the prevalence of diabetes have been steadily increasing over the past few decades especially in the second- and third-world countries [2].

Medical diabetes diagnosis is one of the most challenging and important tasks in medicine [1]. To get the prediction of the disease, several parameters must be collected such as plasma glucose concentration, diastolic blood pressure, triceps skinfold thickness, serum insulin, body mass, and age [2, 4], which may take a long time to analyze and make the final decision [1]. Therefore, advanced computer and information technologies such as machine learning algorithms are used rather than traditional approaches [6]. This latter can help the physicians make critical medical decisions in a

short time with small effort and little money with more accurate decisions [1].

Actually, machine learning techniques have been widely used in healthcare systems to make decisions based on clinical data [6, 10–19]. In this context, many researchers have used them for the diagnosis of diabetes. Yuvraj and his colleagues [20] have proposed an implementation of machine learning algorithms like Random Forest (RF), Decision Tree (DT), and Naïve Bayes (NB) in Hadoop based clusters environment for diabetes prediction. The RF algorithm produces the highest accuracy compared to other algorithms. In [21], the authors developed a prediction model using DT approach to identify low-risk individuals for incidence of type 2 diabetes for the Tehran Lipid and Glucose Study (TLGS) database. Moreover, different classification algorithms, such as Support Vector Machine (SVM), Multilayer Perceptron (MLP), Logistic Regression (LR), RF, and DT, have been compared in [22]. The K-fold cross-validation technique has been used to accurately classify diabetes. The MLP classifier achieved the highest accuracy. According to Jakka and Vakula [23], the performance of the diabetes prediction has been evaluated using several classification algorithms such as K-Nearest Neighbor (KNN), DT, NB, SVM, LR, and RF. The best accuracy achieved was with LR algorithm compared to other algorithms. Similarly, the authors in [24] have used many machine learning classification techniques such as DT, SVM, NB, RF, KNN, and LR to predict the disease, where LR and SVM algorithms work well on diabetes prediction compared to other techniques. In [25], the authors have proposed a comparative study on the disease diagnosis by using Levenberg-Marquardt (LM) and probabilistic MLP techniques, where the first one gave the highest classification accuracy. In [26], T. Roopesh et al. have employed a system to assess the performance of diabetes prediction using different machine learning algorithms by classification, regression, and clustering. Both the SVM and linear regression have obtained the highest accuracy in comparison with other techniques. Besides, Zou et al. [27] have made a comparative study between three classifiers (Neural Network, RF, and DT), where the latter was the better. In [28], a comprehensive comparative study was applied on various machine learning algorithms such as SVM, KNN, DT, NB, and LR for the disease classification, where LR gave the most accurate results. Likewise, Mujumdar and Vaidehi [29] have implemented many machine learning algorithms for diabetes prediction such as SVM, RF, DT, Extra Tree Classifier, AdaBoost algorithm, Perceptron, Linear Discriminant Analysis (LDA), LR, KNN, Gaussian NB, Bagging, and Gradient Boost. The LR gave the highest accuracy with 96%. Eventually, the authors in [30] used several machine learning algorithms including SVM, KNN, LR, DT, RF, and NB to predict diabetes disease. Both SVM and KNN algorithms provided the highest accuracy rate compared to the other algorithms.

However, machine learning techniques present some limitations in terms of precision and feature selection [1]. This drawback has been lifted by the Deep Learning (DL) algorithms, which are used widely in many forms in medical fields [31–37]. Numerous studies show that DL techniques

give better results by minimizing the error rate, increasing the precision, and better resisting the noise, compared to other techniques [1, 3]. DL techniques can perfectly handle a massive number of datasets and have the ability to deal with complex problems at ease [1], which makes them very adequate for our diabetes disease prediction system [6].

In this paper, we propose a diabetes prediction system for better diagnosis. Our work focuses on the following points:

- (1) Set up a system architecture for diabetes prediction based on DNN algorithm in order to make an efficient decision to the diabetes diagnosing; • An evaluation of four different DNN architectures to get the best model.
- (2) A comparison of best DNN model's results against those of many well-known ML classifiers such as LR, SVM, XGBoost, DT, and RF.
- (3) Furthermore, we compare our proposed method with the state-of-the-art methods that used the same datasets, the same experimental protocol, and the same performance measurements.

The rest of the paper is planned as follows: The second section provides an overview of the proposed system. The section that follows presents results and analyses. Then, we show the comparison of the state-of-the-art techniques. At last, Section 5 concludes the paper.

## 2. Proposed System

The proposed diabetes disease prediction system consists of many steps which are perfectly linked to each other to get the desired results. The first step consists of splitting the used dataset into two subsets, training and testing data. Then, we applied two different categories (ML and DL methods) in order to carry out the training phase using the training samples with the best parameters. Eventually, the trained models will be able to predict the testing samples. The overall flowchart of the proposed system is shown in Figure 1.

*2.1. Dataset Description.* To evaluate the performance of this work, we used the famous diabetes dataset taken from Frankfurt Hospital, Germany [38]. This latter contains 2000 records with 9 attributes for each one. A brief overview of the attributes can be found in Table 1, while the 9th one is considered as the target that shows the absence or presence of the disease (value of 0 or 1, respectively). In this dataset, 32.4% of the records had a value of 1 and the rest had a value of 0 (67.6%), taking into consideration the fact that all the patients are females and their ages are between 21 and 81. The first attribute “Pregnancies” shows the pregnancy frequency and it is described from 0 to 17. The Glucose attribute is the result of Glucose Tolerance Test, which examines how the body moves sugar from the blood into tissues such as muscle and fat; it has values ranging from 0 to 199. BloodPressure is the pressure in the arteries when the heart stops between beats; it has been recorded with a range of values from 0 to 122. Insulin is a hormone that aids in the

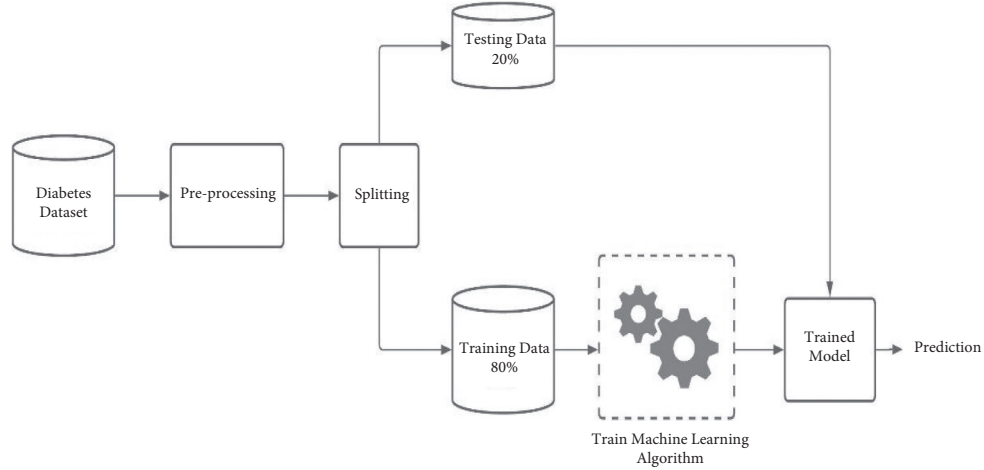


FIGURE 1: The proposed system flowchart.

TABLE 1: Frankfurt Hospital, Germany, dataset description.

No.	Attributes	Description	Range values
1	Pregnancies	Number of times pregnant.	0 to 17
2	Glucose	Oral Glucose Tolerance Test result in terms of milligrams.	0 to 199
3	BloodPressure	Diastolic blood pressure (mm hg).	0 to 122
4	SkinThickness	Triceps skinfold thickness in (mm).	0 to 99
5	Insulin	Patient insulin in the blood.	0 to 846
6	BMI	Body mass index (weight in kg/(height in m) <sup>2</sup> ).	0 to 67.1
7	DiabetesPedigreeFunction	Define the diabetes according to heredity factor.	0.078 to 2.42
8	Age	Patient age in years.	21 to 81
9	Outcome	Presence or absence of diabetes disease.	1 or 0

movement of glucose (blood sugar) from the bloodstream into the cells, and its values are from 0 to 864. The Skin-Thickness attribute provides information about the fat reserves of the body, it has values from 0 to 99. The BMI attribute offers a quick and accurate way to determine whether a patient is overweight or underweight. It has been recorded with a range of values from 0 to 67.1. Finally, DiabetesPedigreeFunction provides a synthesis of the diabetes mellitus history in relatives and the genetic relationship of those relatives to the subject, which can take float values from 0.078 to 2.42.

**2.2. Dataset Preprocessing.** Data preprocessing is a crucial stage that transforms the data into a usable and efficient format, so that it can fit as an input to the machine learning algorithm. In our system, only one technique has been used for data preprocessing, which is data normalization. This latter is generally considered as the process of data structuring. It is also called StandardScaler normalization, where all the values of the attributes are within  $[-1, 1]$ . The StandardScaler formula is shown below in equation (1), where  $X$  represents the input columns of the dataset to transform and  $X\_STS$  represents the transformed ones [39].

$$X\_STS = \frac{(X - \text{mean}(X))}{\text{std\_dev}(X)}. \quad (1)$$

**2.3. Prediction Methods.** In this subsection, we briefly describe the different machine learning methods as well as the Deep Neural Networks that we used for evaluating the proposed system.

**2.3.1. Logistic Regression.** Logistic Regression (LR) is a subset of generalized linear models which deals with the analysis of binary data, which seeks out the best-fitting model for describing the connection between dependent and independent predictors [40, 41]. When it comes to predicting sickness or health status, the LR model is most commonly used [42, 43]. Based on the risk factors given, the LR model can calculate the likelihood of an individual acquiring diabetes disease [43].

If a person suffers from diabetes disease, the value of target is 1; otherwise, target is 0. We determined that the probability of an individual developing diabetes disease is  $P(X)$ . The LR model's formula is defined as follows:

$$\text{logistic}(p) = \ln \left[ \frac{p(X)}{1 - p(X)} \right] = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_k X_k. \quad (2)$$

After exponentiating both sides, we obtain

$$\frac{p(X)}{1 - p(X)} = e^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_k X_k}. \quad (3)$$

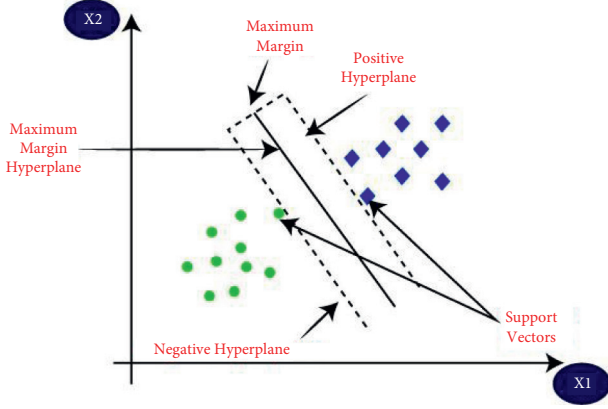


FIGURE 2: Support Vector Machines (SVM) [6].

The probability of an individual developing diabetes disease can be written as

$$p(X) = \frac{e^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_k X_k}}{1 + e^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_k X_k}}, \quad (4)$$

where  $X_1, X_2, X_3, \dots, X_k$  represent the risk factors and  $\beta_1, \beta_2, \beta_3, \dots, \beta_k$  are regression coefficients.

**2.3.2. Support Vector Machine.** SVM is a nonprobabilistic classifier with a separating hyperplane as its formal definition. The technique creates an ideal hyperplane with the greatest distance from the support vectors based on the available training data (supervised learning). This hyperplane is a line that divides a plane into two classes in two-dimensional space. The epsilon  $\epsilon$ , regularization, and kernel parameters are the SVM classifier's tuning parameters [6, 44]. The principle of SVM is shown below in Figure 2.

**2.3.3. Extreme Gradient Boosting (XGBoost).** The Extreme Gradient Boosting is an improved supervised algorithm proposed by Chen and Guestrin [45] based on the Gradient Boosting Decision Tree algorithm [46]. XGBoost can be used to solve problems for regression and classification, which has been chosen to be used by data scientists because of its high execution speed and the high accuracy that it supplies [47]. The XGBoost objective function includes its loss function and regularization term, which can help to prevent overfitting by smoothing the final learned weights to obtain an optimal solution [48]. The loss function  $l(\hat{y}_i, y_i)$  controls the ability of the prediction, which determines the deviation between predicted label  $\hat{y}_i$  and the actual label  $y_i$ . The regularization term  $\Omega(f_k)$  controls the complexity of the model and it can also handle the overfitting issue [48, 49]. XGBoost can also optimize the loss function using first-order and second-order gradient statistics. The objective function for XGBoost is defined as follows [49]:

$$\text{Obj} = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k). \quad (5)$$

The predicted label  $\hat{y}_i$  of the tree boosting model can be expressed as the total sum of all the trees prediction scores

$f_k(x_i)$ , where  $k$  refers to how many trees are in XGBoost model and  $x_i$  refers to the instances samples for a given dataset. Finally  $F$  is the space of classification and regression trees (also referred as CART) [46–48]:

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i), \quad f_k \in F. \quad (6)$$

The regularization term for penalizing the complexity of each tree is shown in equation (7), where  $T$  denotes the number of leaves in the tree,  $\lambda$  is a regularization hyperparameter for controlling the L2-norm of the weights of leaf  $W$ , and  $\gamma$  is a regularization hyperparameter for the simplicity cost by introducing additional leaf depending on each dataset [49, 50].

$$\Omega(f_k) = \gamma T + \frac{1}{2} \lambda \|W\|^2. \quad (7)$$

The main concept behind boosting is to create a more accurate model by combining a lot of simple trees with low accuracy, which will create a new tree for each iteration. There are many different methods for creating a new tree [50]. The common one is called Gradient Tree Boosting which is an improved version of tree boosting by training tree model using the gradient descent to generate the new tree based on all previous trees. Therefore,  $\hat{y}_i$  can be represented by  $\hat{y}_i^{(t-1)} + f_t(x_i)$ , and the objective function in the step  $t$   $\text{Obj}^{(t)}$  is as follows [48]:

$$\text{Obj}^{(t)} = \sum_{i=1}^n l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \sum_t \Omega(f_t). \quad (8)$$

The first-order and second-order gradient statistics of the loss function are shown below in the two following equations, respectively:

$$g_i = \partial_{\hat{y}_i} l(y_i, \hat{y}_i^{(t-1)}), \quad (9)$$

$$h_i = \partial_{\hat{y}_i}^2 l(y_i, \hat{y}_i^{(t-1)}). \quad (10)$$

It is worth noticing that  $g_i$  and  $h_i$  can help to find the optimal weights  $W$ . Hence, the objective function becomes [47, 49]

$$\text{Obj}^{(t)} = -\frac{1}{2} \sum_{j=1}^T \frac{\left( \sum_{i \in I_j} g_i \right)^2}{\left( \sum_{i \in I_j} h_i \right) + \lambda} + \gamma T. \quad (11)$$

**2.3.4. Decision Tree.** DT is a nonparametric supervised learning algorithm for regression and classification tasks. DT (Figure 3) can be seen as a construction model that includes root node, division, and leaf node. Each internal node represents a test on an attribute, each division represents the outcome of test, and each leaf node grips the class label. The opening node in the tree is the root node. First, an attribute is selected and sited at the root node. Then, a division is made for each possible value. This splits dataset into subgroups, one for every value of the attribute. The tree process is

recursively repeated for each division using only those cases that reach the branch. When all cases on a node have the same classification, the tree progress can be stopped. Usually, entropy or classification error is used to define the best tree division [51, 52].

**2.3.5. Random Forest.** RF is one of the most common uses of classifier integration. As shown in Figure 4, RF is made up of numerous separate Decision Tree classifiers that vote on test samples according to a set of criteria [53, 54]. The steps are as follows:

- (i) Extracting some samples from the training set as a training subset using the bootstrap method, which is a self-help sampling approach.
- (ii) A number of features are randomly picked from the feature set for the training subset as the basis for splitting each node of the Decision Tree.
- (iii) Repeat steps (i)-(ii) to generate a large number of training subsets and Decision Trees, which are then combined to build a Random Forest.
- (iv) The test set's samples are fed into the Random Forest, where each Decision Tree makes a choice based on the data. After receiving the findings, the results are voted on using a voting technique to determine the sample categorization results.
- (v) Repeat step (iv) until all of the test sets have been classified [55].

**2.3.6. Deep Neural Networks.** The Deep Neural Networks (DNNs) are one of the architectures of Deep Learning [56]. DNNs have the same basic architecture as ANNs, with the exception that DNNs may have several hidden layers; that is why we use the term “deep.” A Deep Neural Network can hold almost 150 hidden layers [1], and each layer can have several neurons as shown in Figure 5 and, in each layer of neurons, the input of a layer depends on the previous layer's output and so on until we get the prediction of our model in the output layer [57].

The final output value of the first neuron for hidden layer (1) is  $Z_1$ , which is the sum of the products of the various weights and inputs with the bias as shown in equation (12). The value that  $Z_1$  can take is any number from  $-\infty$  to  $+\infty$  so the neuron cannot decide whether to fire or not. Activation functions  $F$  are responsible for deciding whether the neuron will fire or not and calculating  $A_1$  which would be the input for the next layer and so on [57]. The two activation functions used in the proposed model are the ReLU for the hidden layers and the Sigmoid for the output layer (binary classification).

$$\begin{aligned} Z_1 &= \sum_n (X_n * W_{n1}) + b_1, \\ A_1 &= F(Z_1). \end{aligned} \quad (12)$$

### 3. Experimental Results

In this section, we evaluate the performance of DNN algorithm by using the testing data to assess the effectiveness

of our system based on several evaluation metrics. Besides, comparison between our proposed model and the machine learning algorithms described in section (2.3) has been conducted in order to demonstrate the superiority of our model. The used dataset was split into two subsets, the first one for training which contains 80% of the whole data (547 diabetics/1053 nondiabetics) and the other for testing which contains 20% of the whole data (137 diabetics/263 nondiabetics).

**3.1. Evaluation Metrics.** The confusion matrix (Figure 6) is considered as a great tool to show the results summary of a model with the classification issues [1, 56]. In the classification, the prediction can be one of four special cases as follows.

If the actual value of the target in the dataset is True and the classifier predicts it as such, then the prediction is a True Positive (TP). On the contrary, if the classifier predicts it as False, then the prediction is a False Negative (FN). Similarly, if the actual value of the target in the dataset is False and the classifier predicts it as such, then the prediction is True Negative (TN). On the contrary, if the classifier predicts it as True, then the prediction is False Positive (FP) [58].

Finding out how the developed predictive model performs becomes easy with the help of the confusion matrix, which is clearly shown above in Figure 6. The following metrics are used to evaluate the proposed model [49, 56–59].

**Accuracy (Acc)** is the percentage of the correct predictions that a classifier has made compared with the actual values of the target in the testing phase.

$$\text{Acc} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{TN} + \text{FP} + \text{FN})} * 100\%. \quad (13)$$

**Sensitivity (Sens)** gives information about the percentage of True Positives that are correctly classified during the test.

$$\text{Sens} = \frac{\text{TP}}{(\text{TP} + \text{FN})} * 100\%. \quad (14)$$

**Specificity (Spec)** gives information about of True Negatives that are correctly classified during the test.

$$\text{Spec} = \frac{\text{TN}}{(\text{TN} + \text{FP})} * 100\%. \quad (15)$$

**Precision (Pre)** is the percentage of instances that a classifier has labelled as positive with respect to the total predictive positives (the exactness of a classifier).

$$\text{Pre} = \frac{\text{TP}}{(\text{TP} + \text{FP})} * 100\%. \quad (16)$$

**F1-score** shows the harmonic mean of precision and recall.

$$\text{F1-score} = \frac{2 * \text{TP}}{(2 * \text{TP} + \text{FN} + \text{FP})} * 100\%. \quad (17)$$

**3.2. Prediction with ML Methods.** A comparative analysis of all the conventional machine learning algorithms has been

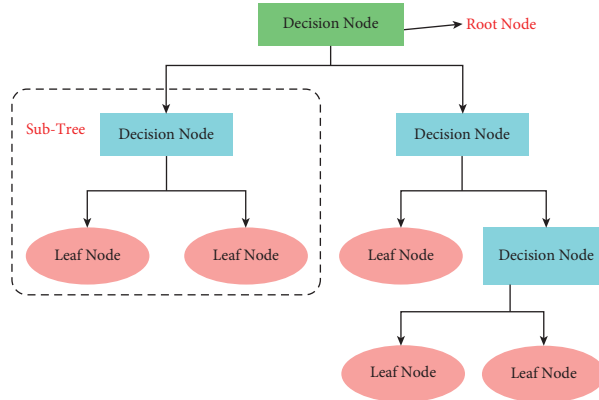


FIGURE 3: Decision Tree principle.

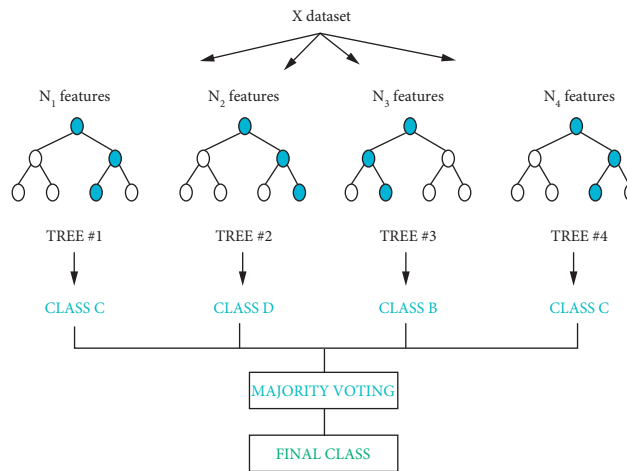


FIGURE 4: Random Forest first stage.

done in this section for diabetes prediction. It has been done for comparing and analyzing accuracies of all the conventional algorithms.

**3.2.1. Hyperparameter Optimization.** Hyperparameter optimization (i.e., tuning) is important because it directly controls the behavior of the training process of the algorithm and has a significant impact on the performance of the model. There are four common methods of hyperparameter optimization: Manual search, Random search, Bayesian optimization, and Grid search [56, 58]. In this work, we applied the Grid search method for each algorithm which systematically builds and evaluates a model for each combination of parameters in a specific grid.

We implemented five machine learning classifiers for binary classification by determining whether or not the patient has diabetes, where each classifier has many different hyperparameters that are not necessary to change, but the main of them needs to be altered to get a good model. Thus, to achieve better results, these parameters and their default values for each algorithm are shown in Table 2.

Now in order to show the impact of hyperparameters optimization on the overall system results, we compare the

performances of the selected ML algorithms with and without the use of this process. Table 3 presents the average score obtained from each classifier using five metrics. We clearly see that all prediction methods give better results than without optimization, while RF gives the highest performance among the others.

**3.3. Evaluation of the DNN Method.** There are different types of layers in DNN. In this work, three types of layers were implemented: a dense layer, which consists of a matrix of weights and the bias; a dropout layer, which can prevent an overfitting issue by dropping out certain fractions of layer's inputs units at each stage of training [1, 60]; and a batch normalization layer, which performs synchronized rescaling for the layer's inputs. We used the Early Stopping technique, which controls the improvement of our model [61]. We have made many experiments by changing the number of layers, the number of neurons in each layer, and different types of layers as shown in Table 4.

As shown in Table 5, the DNN model number 4 is the best one with the following parameters: Epochs=500, Batch\_size=200, and Random\_state=0. Therefore, this model is considered for the rest of this study. The confusion

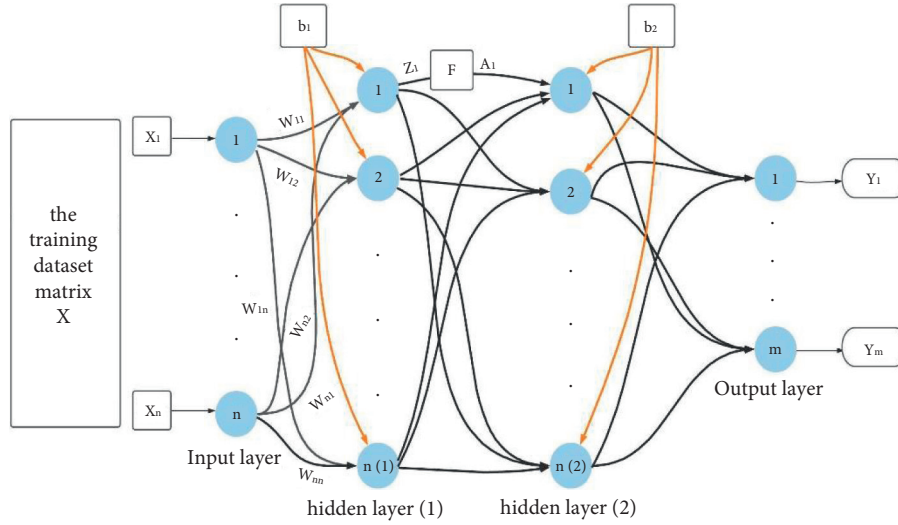


FIGURE 5: A simple DNN architecture with two hidden layers.

		predicted values	
		0	1
Actual values	0	True Negative (TN)	False Positive (FP)
	1	False Negative (FN)	True Positive (TP)

FIGURE 6: Confusion matrix for binary classification.

matrix of DNN prediction results is shown in Figure 7. The performance of the model can be easily got using this confusion matrix by determining the metrics summarized in Table 5.

The behavior of the accuracy is shown in Figure 8, where the blue line represents the training phase, and the orange one represents the testing phase resulting in the best values of the accuracy, 99.0% and 99.75%, respectively.

**3.4. Performance Comparison.** To give an idea of how the proposed DNN has superior performance, we compared it with other prediction methods evaluated above  $x$ . In the following, we discuss the obtained performance for each classifier using Boxplot diagrams.

**3.4.1. Accuracy.** The accuracy performance of the proposed DNN in comparison with five ML methods is shown in Figure 9. Obviously, DNN achieved the highest ACC with 99.75%, where all the implemented ML methods also perform excellently. Only LR performs relatively poorly with an ACC less than 80%.

**3.4.2. Specificity.** Figure 10 shows the specificity performance of the proposed DNN in comparison with other ML methods that performed excellently with more than 96%,

except LR that shows the lowest specificity. The highest value of specificity is 99.60%, and it was achieved by the DNN method.

**3.4.3. Sensitivity.** The sensitivity performance of the proposed DNN and ML methods is shown in Figure 11. The proposed DNN has achieved the highest sensitivity with 100.0%. The other ML methods performed excellently with more than 95%, except LR method that presented a very bad performance.

**3.4.4. Precision.** The precision performance of the proposed DNN and ML methods is presented in Figure 12. The highest precision achieved (99.32%) was that obtained with DNN method. In addition, the ML methods have achieved a good range of precision with more than 93%, except the LR method that gave the worst precision.

**3.4.5. F1-Score.** The F1-score performance of the proposed DNN and other ML methods is shown in Figure 13. Except LR technique, all used methods performed excellently with F1-score greater than 94%. The highest value of F1-score (99.66%) was achieved by using DNN.

Based on these statistics, it was observed that the proposed DNN is the better prediction model among the other implemented ML methods.

## 4. Comparison with the State-of-the-Art Methods

To present how well our diabetes prediction system performs, we compared it with other works that used the same dataset and the same performance measures. It is worth noting that this comparison was based only on the accuracy metric because the other evaluation metrics are not available. As observed from Table 6, the proposed DNN prediction outperforms works reported in literature.

TABLE 2: Hyperparameters optimization for ML classifiers.

Models	Parameters	Description	Default value	Hyperparameters optimization process	
				Considered combinations	Optimal value
LR	C	Inverse of regularization strength.	1.0	np.logspace (-5, 5, 10)	46.42
	Solver	Algorithm to use in the optimization problem.	“lbfgs”	[“lbfgs”, “liblinear”]	“lbfgs”
SVM	Gamma	Kernel coefficient for “rbf,” “poly,” and “sigmoid.”	“scale”	[0.1, 1, 5, 10]	5
	C	Regularization parameter.	1.0	[0.1, 1, 10, 100, 1000]	1000
	Degree	Degree of the polynomial kernel function “poly” (ignored by all other kernels).	3	[1–6]	6
	Kernel	Specifies the kernel type to be used in the algorithm.	“rbf”	[“linear,” “poly,” “rbf,” “sigmoid”]	“poly”
XGBoost	learning_rate	Step size shrinkage used to prevent overfitting.	0.3	[0.1, 0.01, 0.05]	0.1
	n_estimators	The number of trees.	100	[10, 100, 1000]	1000
	max_depth	Maximum depth of a tree.	6	[3, 4, 5]	5
	min_child_weight	Minimum sum of weights needed in child.	1	[0, 5, 10]	0
	colsample_bytree	The features’ subsample ratio when constructing each tree.	1	[0.1, 0.8, 1.0]	1
	Subsample	The training instances’ subsample ratio.	1	[0.2, 0.7, 1.0]	0.7
	Gamma	The minimum loss reduction needed for.	0	[0, 0.25, 1.0]	0
	reg_lambda	L2 regularization term on weights.	1	[0, 1.0, 10.0]	0
	reg_lambda	L1 regularization term on weights.	1	[0, 7.0, 10.0]	0
	max_depth	Maximum depth of the tree.	None	[9, 11, 13, 15]	15
max_features	The number of features to consider when looking for the best split.	None	[“auto,” “sqrt”]	“auto”	
DT	min_samples_leaf	The minimum number of samples required to be at a leaf node.	1	[1, 3, 5, 7, 9]	1
	min_samples_split	The minimum number of samples required to split an internal node.	2	[2, 4, 6]	2
	Criterion	The function to measure the quality of a split.	“gini”	[“gini,” “entropy”]	“entropy”
	Splitter	The strategy used to choose the split at each node.	“best”	[“best,” “random”]	“best”
RF	n_estimators	The number of trees in the forest.	100	[10, 30, 50, 70, 90, 100]	50
	max_features	The number of features to consider when looking for the best split.	“auto”	[“auto,” “sqrt”]	“auto”
	Criterion	The function to measure the quality of a split.	“Gini”	[“gini,” “entropy”]	“entropy”
	max_depth	Maximum depth of the tree.	None	[5, 10, 15]	15
	min_samples_split	The minimum number of samples required to split an internal node.	2	[2, 4, 6]	6
	min_samples_leaf	The minimum number of samples required to be at a leaf node.	1	[1, 2, 3]	1
	Bootstrap	Whether bootstrap samples are used when building trees.	True	[True, False]	False

TABLE 3: The obtained performance with and without hyperparameters optimization.

Methods	Evaluation metrics									
	With default values					With optimal values				
	Acc	Spec	Sens	Pre	F1-score	Acc	Spec	Sens	Pre	F1-score
LR	78.25	92.0	54.42	80.0	64.77	79.0	91.63	57.14	80.0	66.66
SVM	81.25	93.02	61.22	83.33	70.58	96.75	96.38	97.22	93.95	95.56
XGBoost	86.5	92.30	74.62	83.33	78.74	97.25	98.11	95.58	96.29	95.94
DT	98.00	96.77	100.0	94.93	97.40	98.75	98.81	98.59	97.90	98.24
RF	98.5	99.20	97.22	98.59	97.90	<b>99.00</b>	<b>99.20</b>	<b>98.59</b>	<b>98.59</b>	<b>98.59</b>



TABLE 4: Different model's architectures for DNN technique.

	# of hidden layers	# of batch layers	# of dropout layers	Model's architecture
Model 01	2	2	2	256 > B > 0.2 > 64 > B > 0.3
Model 02	3	3	1	64 > B > 0.1 > 32 > B > 16 > B
Model 03	3	4	3	B > 512 > B > 0.3 > 128 > B > 0.3 > 64 > B > 0.3
Model 04	4	4	4	B > 512 > B > 0.4 > 256 > B > 0.3 > 128 > B > 0.3 > 16 > B > 0.1

It is worth noticing that some important parameters can enhance the prediction results by choosing their best values, which we have experimented as well.

TABLE 5: Performance evaluation of different models for DNN method.

Models	Parameters			Evaluation metrics				
	Epochs	Batch_size	Random_state	Acc	Spec	Sens	Pre	F1-score
Model 01	100	10	0	90.24	95.22	79.68	88.69	83.95
	200	70	1	97.25	96.65	98.47	93.47	95.91
	500	200	42	98.25	98.02	98.63	96.66	97.64
	500	200	0	97.00	97.79	95.31	95.31	95.31
Model 02	100	10	0	88.74	92.27	81.25	83.20	82.21
	200	70	1	96.74	97.02	96.18	94.02	95.09
	500	200	42	97.00	96.04	98.63	93.54	96.02
	500	200	0	99.50	100.0	98.63	100.0	99.21
Model 03	100	10	0	84.24	91.17	69.53	78.76	73.85
	200	70	1	96.49	96.65	96.18	93.33	94.73
	500	200	42	95.99	96.83	94.55	94.55	94.55
	500	200	0	98.25	98.16	98.43	96.18	97.29
Model 04	100	10	0	84.75	91.17	71.09	79.13	74.89
	200	70	1	94.24	94.05	94.65	88.57	91.51
	500	200	42	97.00	97.23	96.65	95.30	95.94
	<b>500</b>	<b>200</b>	<b>0</b>	<b>99.75</b>	<b>99.60</b>	<b>100.0</b>	<b>99.32</b>	<b>99.66</b>

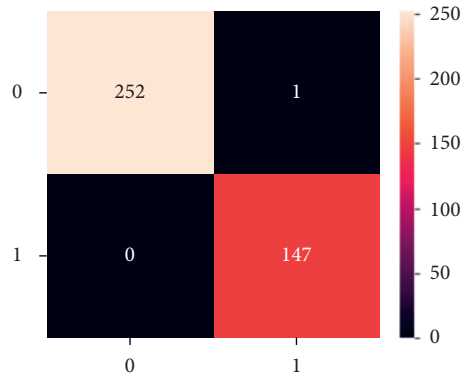


FIGURE 7: Confusion matrix.

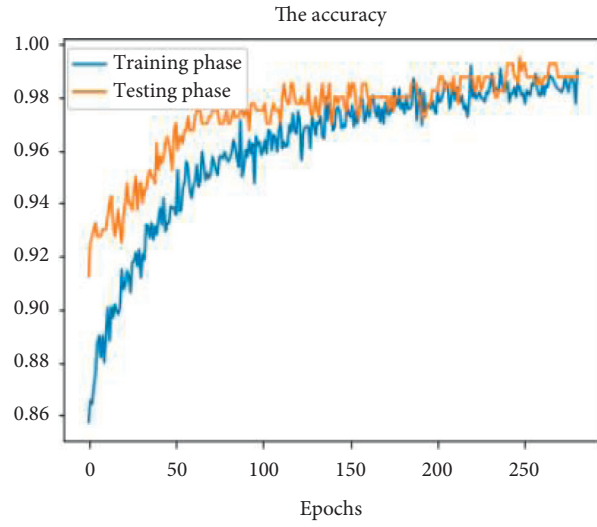


FIGURE 8: The accuracy behavior during the training and the testing phases.

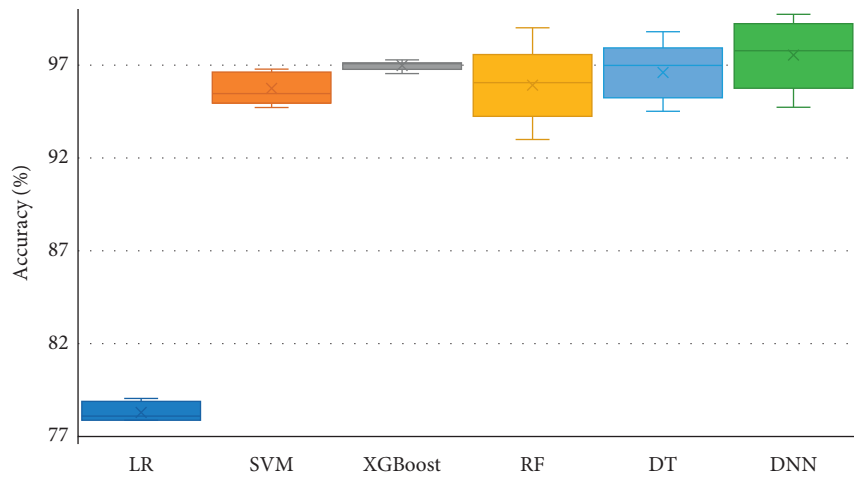


FIGURE 9: Performance comparison of classifiers based on accuracy.

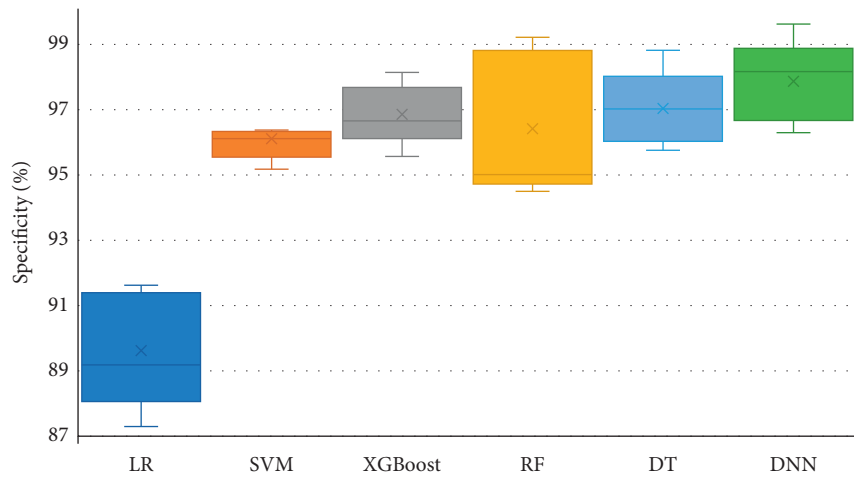


FIGURE 10: Performance comparison of classifiers based on specificity.

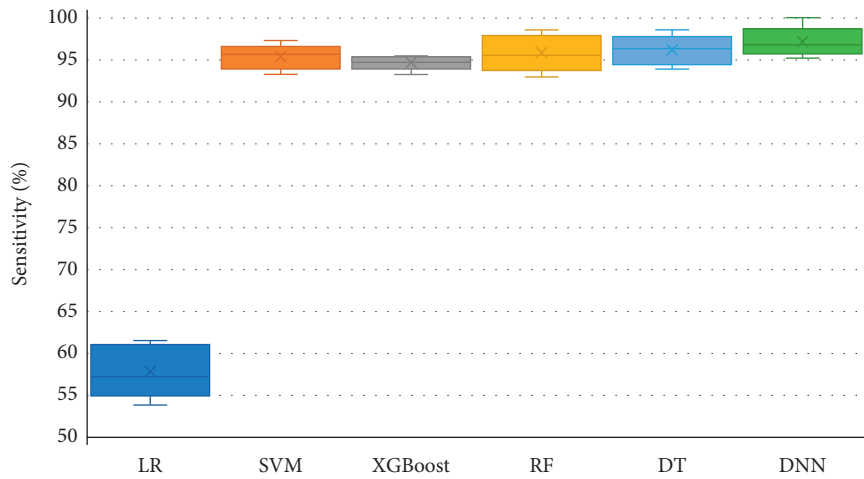


FIGURE 11: Performance comparison of classifiers based on sensitivity.

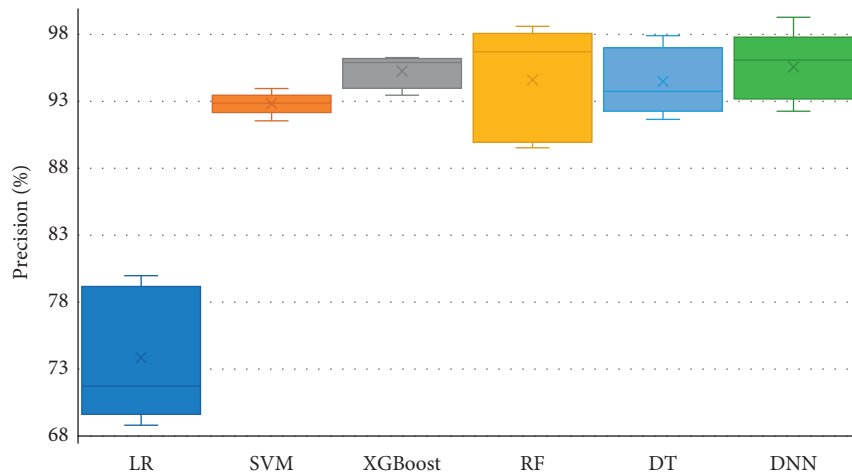


FIGURE 12: Performance comparison of classifiers based on precision.

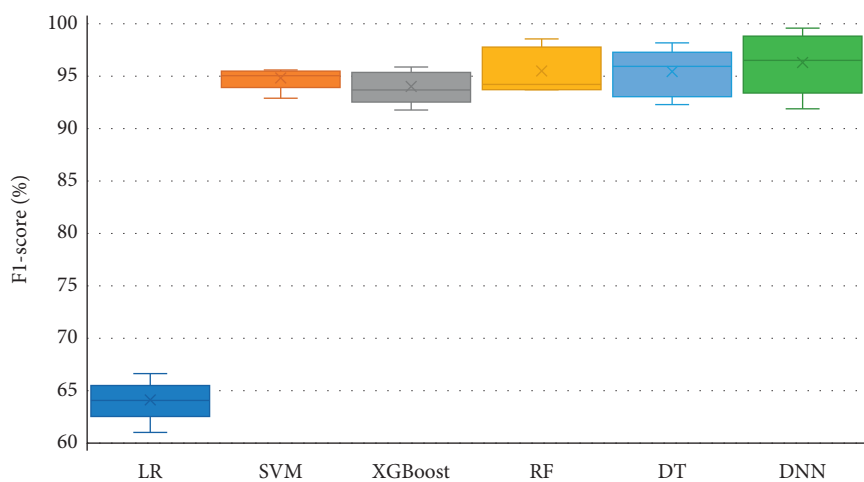


FIGURE 13: Performance comparison of classifiers based on F1-score.

TABLE 6: Comparison of the proposed system with state-of-the-art methods in terms of accuracy.

Authors	Method	Used database	Accuracy (%)
Jakka and Vakula [23]	Logistic Regression	Pima Indian diabetes (PID)	77.60
Ramraj et al. (2016) [62]	XGBoost	PID	77.95
Dwivedi (2017) [59]	Logistic Regression	PID	78.00
Ahmed et al. (2013) [63]	Improved GA	PID	80.40
Khanam and Foo [24]	Neural Network	PID	88.60
Malik et al. [64]	Random Forest	Frankfurt Hospital, Germany	98.80
<b>Our work</b>	<b>Deep Neural Network</b>	<b>Frankfurt Hospital, Germany</b>	<b>99.75</b>

## 5. Conclusion

In this study, we proposed an efficient diabetes prediction system based on Deep Neural Network (DNN) algorithm to identify whether or not a person has diabetes. We presented a comparative study between the Deep Neural Network (DNN) and several machine learning techniques. The performance evaluation of these models that have been studied and evaluated on various performance metrics such as accuracy, specificity, sensitivity, precision, and F1-score proved the superiority of the proposed DNN method. Furthermore, we performed a comparison between our system and the state-of-the-art methods. This comparison showed that a diabetes prediction system based on DNN algorithm could significantly provide promising, better performances compared to the state-of-the-art techniques. Applying this method can have a direct impact and economic saving on the design and development of diabetes disease prediction system in healthcare.

## Data Availability

The data used to support the findings of this study are freely available.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

The authors of this study would like to express their gratitude to the members of LASS Laboratory (University of M'sila, Algeria) for their support and assistance in publishing this work. The publication of this article was funded by the Qatar National Library. The authors would like to thank Qatar National Library (QNL) for supporting the publication charges of this article.

## References

- [1] S. Islam Ayon, Md. M. Islam, 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.
- [2] F. A. Khan, K. Zeb, M. AL-Rakhami, A. Derhab, and S. A. C. Bukhari, "Detection and prediction of diabetes using data mining: a comprehensive review," *IEEE Access*, vol. 9, pp. 43711–43735, 2021.
- [3] M. K. Hasan, M. A. Alam, D. Das, E. Hossain, and M. Hasan, "Diabetes prediction using ensembling of different machine learning classifiers," *IEEE Access*, vol. 8, pp. 76516–76531, 2020.
- [4] American Diabetes Association, "Diagnosis and classification of diabetes mellitus," *Diabetes Care*, vol. 37, no. 1, pp. 81–90, 2014.
- [5] K. S. Jacob, "The challenge of medical diagnosis: a primer on principles, probability, process and pitfalls," *The National medical journal of India*, vol. 28, no. 1, pp. 24–28, 2015.
- [6] M. Djerioui, Y. Brik, M. Ladjal, and B. Attallah, "Neighborhood component analysis and support vector machines for heart disease prediction," *Ingénierie des Systèmes d'Information*, vol. 24, no. 6, pp. 591–595, 2019.
- [7] S. G. Siva and K. Manikandan, "Diagnosis of diabetes diseases using optimized fuzzy rule set by grey wolf optimization," *Pattern Recognition Letters*, vol. 125, pp. 432–438, 2019.
- [8] World health organization report on diabetes diseases: [https://www.who.int/health-topics/diabetes#tab=tab\\_1](https://www.who.int/health-topics/diabetes#tab=tab_1).
- [9] Id Federation, "Idf diabetes atlas," 2019, <https://diabetesatlas.org/en/>, 9th Edition.
- [10] F. Mohanty, S. Rup, and B. Dash, "Automated diagnosis of breast cancer using parameter optimized kernel extreme learning machine," *Biomedical Signal Processing and Control*, vol. 62, pp. 102–108, 2020.
- [11] S. S. Yadav and S. M. Jadhav, "Detection of common risk factors for diagnosis of cardiac arrhythmia using machine learning algorithm," *Expert Systems with Applications*, vol. 163, Article ID 113807, 2021.
- [12] E. Martinez-Rios, L. Montesinos, M. Alfaro-Ponce, and L. Pecchia, "A review of machine learning in hypertension detection and blood pressure estimation based on clinical and physiological data," *Biomedical Signal Processing and Control*, vol. 68, Article ID 102813, 2021.
- [13] H. Tjahjadi, K. Ramli, and H. Murfi, "Noninvasive classification of blood pressure based on photoplethysmography signals using bidirectional long short-term memory and time-frequency analysis," *IEEE Access*, vol. 8, pp. 20735–20748, 2020.
- [14] N. Hameed, A. M. Shabut, M. K. Ghosh, and M. A. Hossain, "Multi-class multi-level classification algorithm for skin lesions classification using machine learning techniques," *Expert Systems with Applications*, vol. 141, Article ID 112961, 2019.
- [15] J. S. Almeida, P. P. Rebouças Filho, T. Carneiro et al., "Detecting Parkinson's disease with sustained phonation and speech signals using machine learning techniques," *Pattern Recognition Letters*, vol. 125, pp. 55–62, 2019.
- [16] S. B. Belhaouari, H. R. Al-Absi, and K. Kassoul, "Breast cancer classification using cluster k-nearest neighbor," in *Proceedings of the AIP Conference Proceedings*, vol. 1482, no. 1, pp. 382–

- 385, Universiti Teknologi PETRONAS, Tronoh, Perak, Malaysia, 27 September 2012.
- [17] H. R. Al-Absi, S. B. Belhaouari, and S. Sulaiman, "A computer aided system for breast cancer detection and diagnosis," in *Proceedings of the 2014 International Conference on Computer and Information Sciences (ICCOINS)*, pp. 1–4, IEEE, Kuala Lumpur, Malaysia, 3 June 2014.
- [18] S. B. Belhaouari, "Fast and accuracy control chart pattern recognition using a new cluster-k-nearest neighbor," *Journals of World Academy of Science, Engineering and Technology*, vol. 30, 2009.
- [19] S. B. Belhaouari, S. Ahmed, and S. Mansour, "Optimized K-means algorithm," *Mathematical Problems in Engineering*, vol. 2014, Article ID 506480, 14 pages, 2014.
- [20] N. Yuvaraj and K. R. SriPreethaa, "Diabetes prediction in healthcare systems using machine learning algorithms on Hadoop cluster," *Cluster Computing*, vol. 22, no. S1, pp. 1–9, 2017.
- [21] A. Ramezankhani, O. Pournik, J. Shahrabi, D. Khalili, F. Azizi, and F. Hadaegh, "Applying decision tree for identification of a low risk population for type 2 diabetes. Tehran Lipid and Glucose Study," *Diabetes Research and Clinical Practice*, vol. 105, no. 3, pp. 391–398, 2014.
- [22] R. Ahuja, S. C. Sharma, and M. Ali, "A diabetic disease prediction model based on classification algorithms," *Annals of Emerging Technologies in Computing*, vol. 3, no. 3, pp. 44–52, 2019.
- [23] A. Jakka and R. J. Vakula, "Performance evaluation of machine learning models for diabetes prediction," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 11, pp. 1976–1980, 2019.
- [24] J. J. Khanam and S. Y. Foo, *A Comparison of Machine Learning Algorithms for Diabetes Prediction*, The Korean Institute of Communications and Information Sciences, Korea, 2021.
- [25] H. Temurtas, N. Yumusak, and F. Temurtas, "A comparative study on diabetes disease diagnosis using neural networks," *Expert Systems with Applications*, vol. 36, no. 4, pp. 8610–8615, 2009.
- [26] T. Roopesh, A. Srinivasulu, and K. S. Kannan, "Prediction of diabetes disease using data mining and deep learning techniques," *EasyChair*, vol. 1608, 2019.
- [27] 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.
- [28] A. Choudhury and D. Gupta, "A survey on medical diagnosis of diabetes using machine learning techniques," *Advances in Intelligent Systems and Computing*, vol. 740, pp. 67–78, 2019.
- [29] A. Mujumdar and V. vaidehi, "Diabetes prediction using machine learning algorithms," *Procedia Computer Science*, vol. 165, pp. 292–299, 2019.
- [30] M. A. Sarwar, K. Nasir, W. Hamid, and M. A. Shah, "Prediction of diabetes using machine learning algorithms in healthcare," in *Proceedings of the 24th International Conference on Automation & Computing*, pp. 1–6, IEEE, Newcastle Upon Tyne, UK, 6 September 2018.
- [31] N. Li, J. Tuo, Y. Wang, and M. Wang, "Prediction of blood glucose concentration for type 1 diabetes based on echo state networks embedded with incremental learning," *Neurocomputing*, vol. 378, pp. 248–259, 2019.
- [32] N. An, H. Ding, J. Yang, R. Au, and T. F. A. Ang, "Deep ensemble learning for Alzheimer's disease classification," *Journal of Biomedical Informatics*, vol. 105, p. 103411, Article ID 103411, 2020.
- [33] G. Swapna, R. Vinayakumar, and K. P. Soman, "Diabetes detection using deep learning algorithms," *The Korean Institute of Communications and Information Sciences*, vol. 4, no. 4, pp. 243–246, 2018.
- [34] Y. Benhammou, B. Achchab, F. Herrera, and S. Tabik, "BreakHis based breast cancer automatic diagnosis using deep learning: taxonomy, survey and insights," *Neurocomputing*, vol. 375, pp. 9–24, 2020.
- [35] D. Schmidt, C. A. d. Costa, R. d. R. Righi, and B. Eskofier, "DeepSigns: a predictive model based on Deep Learning for the early detection of patient health deterioration Denise Bandeira da Silva," *Expert Systems with Applications*, vol. 165, Article ID 113905, 2021.
- [36] S. B. Belhaouari and A. Islam, "Deep learning in healthcare," *Deep Learning in Healthcare. Multiple Perspectives on Artificial Intelligence in Healthcare*, Springer, Cham, pp. 155–168, 2021.
- [37] A. S. A. Rahman, S. B. Belhaouari, A. Bouzerdoum, H. Baali, T. Alam, and A. M. Eldaraa, "Breast mass tumor classification using deep learning," in *Proceedings of the 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies, ICloT*, pp. 271–276, IEEE, Doha, Qatar, 2 February 2020.
- [38] <https://www.kaggle.com/johndasilva/diabetes?select=diabetes.csv>.
- [39] 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, pp. 40–46, 2021.
- [40] D. W. Hosmer and S. Lemeshow, *Applied Logistic Regression*, Vol. 398, John Wiley & Sons, Hoboken, New Jersey, US, 2013.
- [41] M. Schumacher, R. Robner, and W. Vach, "Neural networks and logistic regression: Part I," *Computational Statistics & Data Analysis*, vol. 21, no. 16, pp. 661–682, 1996.
- [42] R. D. Joshi and C. K. Dhakal, "Predicting type 2 diabetes using logistic regression and machine learning approaches," *International Journal of Environmental Research and Public Health*, vol. 18, no. 14, p. 7346, 2021.
- [43] L. Zhang, Y. Wang, M. Niu, C. Wang, and Z. Wang, "Machine learning for characterizing risk of type 2 diabetes mellitus in a rural Chinese population: the Henan Rural Cohort Study," *Scientific Reports*, vol. 10, no. 1, pp. 4406–4410, 2020.
- [44] B. Attallah, Y. Brik, Y. Chahir, M. Djeriou, and A. Boudjelal, "Fusing palmprint, finger-knuckle-print for Bi-modal recognition system based on LBP and BSIF," in *Proceedings of the 6th International Conference on Image and Signal Processing and their Applications (ISPA)*, pp. 1–5, IEEE, Mostaganem, Algeria, 24 November 2019.
- [45] T. Chen and C. Guestrin, "XGBoost: a scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, ACM, San Francisco California USA, 13 August 2016.
- [46] J. H. Friedan, "Greedy function approximation: a gradient boosting machine," *Annals of Statistics*, vol. 29, pp. 1189–1232, 2001.
- [47] A. I. A. Osman, A. N. Ahmed, M. F. Chow, Y. F. Huang, and A. El-Shafie, "Extreme gradient boosting (Xgboost) model to predict the groundwater levels in Selangor Malaysia," *Ain Shams Engineering Journal*, vol. 12, no. 2, pp. 1545–1556, 2020.

- [48] C. Xu, X. Liu, H. Wang et al., "A study of predicting irradiation-induced transition temperature shift for RPV steels with XGBoost modeling," *Nuclear Engineering and Technology*, vol. 53, 2021.
- [49] K. Budholiya, S. K. Shrivastava, and V. Sharma, "An optimized XGBoost based diagnostic system for effective prediction of heart disease," *Journal of King Saud University – Computer and Information Sciences*, 2020, In press.
- [50] G. N. Dimitrakopoulos, A. G. Vrahatis, V. Plagiannakos, and K. Sgarbas, "Pathway analysis using XGBoost classification in Biomedical Data," in *Proceedings of the 10<sup>th</sup> Hellenic Conference on Artificial Intelligence*, pp. 1–6, ACM, Patras Greece, 9 July 2018.
- [51] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE transactions on systems, man, and cybernetics*, vol. 21, no. 3, pp. 660–674, 1991.
- [52] Y. K. Qawqzeh, M. M. Otoom, F. Al-Fayez et al., "A proposed decision tree classifier for atherosclerosis prediction and classification," *IJCSNS*, vol. 19, no. 12, p. 197, 2019.
- [53] G. Xu, M. Liu, Z. Jiang, D. Söffker, and W. Shen, "Bearing fault diagnosis method based on deep convolutional neural network and random forest ensemble learning," *Sensors*, vol. 19, no. 5, p. 1088, 2019.
- [54] J. L. Speiser, M. E. Miller, J. Tooze, and E. Ip, "A comparison of random forest variable selection methods for classification prediction modeling," *Expert Systems with Applications*, vol. 134, pp. 93–101, 2019.
- [55] K. Yunchuan and T. Yu, "A deep neural network model using random forest to extract feature representation for gene expression data classification," *Scientific Reports*, vol. 8, no. 1, pp. 1–9, 2018.
- [56] T. R. Gadekallu, N. Khare, S. Bhattacharya, S. Singh, P. K. Reddy-Maddikunta, and G. Srivastava, "Deep neural networks to predict diabetic retinopathy," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–14, 2020.
- [57] A. Liaqat, R. Atiqur, A. Khan, M. Zhou, J. Azhir, and J. Ali-Khan, "An automated diagnostic system for heart disease prediction based on  $\chi^2$  statistical model and optimally configured deep neural network," *IEEE Access*, vol. 7, pp. 34938–34945, 2019.
- [58] A. W. K. Al-Faqeh, A. Zerguine, M. A. Al-Bulayhi, A. H. Al-Sleem, and A. S. Al-Rabiah, "Credit card fraud detection via integrated account and transaction submodules," *Arabian Journal for Science and Engineering*, vol. 46, pp. 1–9, 2021.
- [59] A. K. Dwivedi, "Analysis of computational intelligence techniques for diabetes mellitus prediction," *Neural Computing & Applications*, vol. 30, no. 12, pp. 3837–3845, 2017.
- [60] V. Alcalá-Rmz, L. A. Zanella-Calzada, C. E. Galván-Tejada et al., "Identification of diabetic patients through clinical and para-clinical features in Mexico: an approach using deep neural networks," *International Journal of Environmental Research and Public Health*, vol. 16, no. 3, p. 381, 2019.
- [61] T. V. Bandos, G. C. Valls, and E. S. Olivas, "Statistical criteria for early-stopping of support vector machines," *Neurocomputing*, vol. 70, no. 13–15, pp. 2588–2592, 2007.
- [62] S. Ramraj, N. U. R. Sunil, and S. Banerjee, "Experimenting XGBoost algorithm for prediction and classification of different datasets," *International Journal of Control Theory and Applications*, vol. 9, pp. 651–662, 2016.
- [63] F. Ahmad, N. A. M. Isa, Z. Hussain, and M. K. Osman, "Intelligent medical disease diagnosis using improved hybrid genetic algorithm - multilayer Perceptron network," *Journal of Medical Systems*, vol. 37, no. 2, pp. 1–8, 2013.
- [64] S. Malik, S. Harous, and H. El-Sayed, "Comparative analysis of machine learning algorithms for early prediction of diabetes mellitus in women," *International Symposium on Modelling and Implementation of Complex Systems*, Springer, Cham, pp. 95–106, 2020.