Machine Learning Based Diabetes Classification and Prediction

Data Set Line: https://www.kaggle.com/datasets/mathchi/diabetes-data-set

Table 1: Features' comparison of the proposed study vs. state-of-the-art studies.

Study	Diabetes classification	Diabetes prediction	Real-time healthcare data analysis	Performance measures
[15]	✓	✕	✕	Accuracy
[16]	✓	✕	✕	NA
[17]	✓	✕	✕	Accuracy
[18]	✓	✕	✕	NA
[19]	✓	✓	✕	Accuracy
[20]	✓	✕	✕	Accuracy
[21]	✓	✕	✕	NA
[22]	✕	✓	✕	Accuracy
[23]	✓	✓	✕	Accuracy
[24]	✕	✓	✕	Accuracy
[25]	✕	✓	✕	Accuracy
[26]	✕	✓	✕	Accuracy, correlation coefficient
[4]	✕	✕	✓	NA
[27–33]	✕	✕	✓	NA
[34]	✓	✕	✓	Accuracy
[8]	✓	✕	✓	Accuracy, standard deviation
Proposed	✓	✓	✓	Accuracy, Precision, Recall, RMSE, r

1. Proposed Diabetic Classification andPrediction System for Healthcare

The proposed diabetes classification and prediction system has exploited different machine learning algorithms. First, to classify diabetes, we utilized logistic regression, random forest, and MLP. Notably, we fine-tuned MLP for classification due to its promising performance in healthcare, specifically in diabetes prediction [20, 21, 35, 36]. The proposed MLP architecture and algorithm are shown in Figure 2 and Algorithm 1, respectively.

Second, we implement three widely used machine learning algorithms for diabetes prediction, i.e., moving averages, linear regression, and LSTM. Mainly, we optimized LSTM for crime prediction due to its outstanding performance in real-world applications, particularly in healthcare [53]. The implementation details of the proposed algorithms are as follows.

4.1. Diabetes Classification Techniques. For diabetic classification, we fine-tuned three widely used state-of-the-art techniques. Mainly, a comparative analysis is performed among the proposed techniques for classifying an individual in either of the diabetes categories. The details of the proposed diabetes techniques are as follows.

4.1.1. Logistic Regression. It is appropriate to use logistic regression when the dependent variable is binary [54], as we have to classify an individual in either type 1 or type 2 diabetes. Besides, it is used for predictive analysis and explains the relationship between a dependent variable and one or many independent variables, as shown in equation (1). Therefore, we used the sigmoid cost function as a hypothesis function ($h_\vartheta(x)$). The aim is to minimize cost function $J(\vartheta)$. It always results in classifying an example either in class 1 or class 2.

$$h_\vartheta(x) \frac{1}{1 + e^{-\vartheta_T x}},$$

$$J(\vartheta) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^i \log h_\vartheta(x^i) + (1-y^i)\log\left(1-h_\vartheta(x^i)\right)\right]. \quad (1)$$

4.1.2. Random Forest (RF). As its name implies, it is a collection of models that operate as an ensemble. The critical idea behind RF is the wisdom of the crowd, each model predicts a result, and in the end, the majority wins. It has been used in the literature for diabetic prediction and was found to be effective [55]. Given a set of training examples X $x_1, x_2, \ldots, x_m$ and their respective targets Y $y_1, y_2, \ldots, y_m$, RF classifier iterates B times by choosing samples with replacement by fitting a tree to the training examples. The training algorithm consists of the following steps depicted in equation (2).

(i) For b 1...B, sample with replacement n training examples from X and Y.

(ii) Train a classification tree f_b on X_b and Y_b.

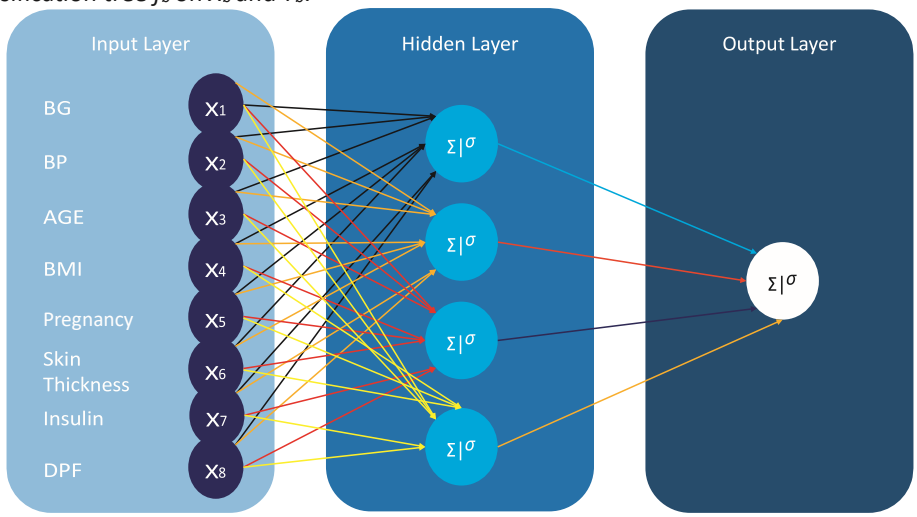

Figure 2: Proposed MLP architecture with eight variables as input for diabetes classification.

$$f = \frac{1}{B}\sum_{b=1}^{B} f_b(x'). \quad (2)$$

4.1.3. Multilayer Perceptron. For diabetes classification, we have fine-tuned multilayer perceptron in our experimental setup. It is a network where multiple layers are joined together to make a classification method, as shown in Figure 2. The building block of this model is perceptron, which is a linear combination of input and weights. We used a sigmoid unit as an activation function shown in Algorithm 1. The proposed algorithm consists of three main steps. First, weights are initialized and output is computed at the output layer (δ_k) using the sigmoid activation function. Second, the error is computed at hidden layers (δ_h) for all hidden units. Finally, in a backward manner, all network weights ($w_{i,j}$) are updated to reduce the network error. The detailed procedure is outlined in Algorithm 1 for diabetes classification.

Figure 2 shows the multilayer perceptron classification model architecture where eight neurons are used in the input layer because we have eight different variables. The middle
layer is the hidden layer where weights and input will be computed using a sigmoid unit. In the end, results will be computed at the output layer. Backpropagation is used for updating weights so that errors can be minimized for predicting class labels. For simplicity, only one hidden layer is shown in the architecture, which in reality is much denser.

Input data from the input layer are computed on the hidden layers with the input values and weights initialized. Every unit in the middle layer called the hidden layer takes the net input, applies activation function "sigmoid" on it, and transforms the massive data into a smaller range between 0 and 1. The calculation is functional for every middle layer. The same procedure is applied on the output layer, which leads to the results towards the prediction for diabetes.

4.2. Diabetes Prediction. It is more beneficial to identify the early symptoms of diabetes than to cure it after being diagnosed. Therefore, in this study, a diabetes prediction system is proposed where three state-of-the-art machine learning algorithms are exploited, and a comparative analysis is performed. The details of the proposed approaches are as follows.

4.2.1. Moving Averages. To predict diabetes, we used moving averages with the experimental setup due to its effectiveness in diabetes prediction for children [56]. It is

based on a calculation that analyzes data points by creating a series of averages of the subset of the data randomly. The moving average algorithm is based on the "forward shifting" mechanism. It excludes the first number from the series and includes the next value in the dataset, as shown in equation (3). The input values are calculated by averaging (P_{SM}) the train data at certain time stamps $P_M + P_M + ... P_{M-(n-1)}$. The algorithm used past observations as input and predicted future events.

$$P_{SM} = \frac{P_M + P_{M-1} + ... + P_{M-(n-1)}}{n} = \frac{1}{n} \sum_{i=0}^{n-1} P_{M-i} \quad (3)$$

4.2.2. Linear Regression. Second, a linear regression model is applied to the PIMA Indian dataset with the same experimental setup. We used this approach to model a relationship between a dependent variable, that is, outcome in our case, and one or more independent variables. The autonomous variable response affects a lot on the target/dependent variable, as shown in equation (4). We use a simplified hypothesis and cost function for multivariate linear regression, as there are eight different variables in our dataset [57]. We choose a very simplified hypothesis function ($h_\vartheta(x)$). The aim is to minimize cost function $J(\vartheta)$ by choosing the suitable weight ($\vartheta^T x$) parameters and minimizing sum of squared error (SSE).

$$J(\vartheta) = \frac{1}{m} \sum_{i=1}^{m} \left(h_\vartheta(x^{(i)}) - y^{(i)} \right)^2$$

$$h_\vartheta(x) = \vartheta^T x \quad (4)$$

$$\vartheta_0 x_0 + \vartheta_1 x_1 + \cdots + \vartheta_n x_n.$$

4.2.3. Long Short-Term Memory. For diabetic forecasting, we have calibrated the long short-term memory algorithm with our experimental setup. The proposed approach outperformed as compared to other state-of-the-art techniques implemented, as shown in Table 2. LSTM is based on recurrent neural network (RNN) architecture, and it has feedback connections that make it suitable for diabetes forecasting [58]. LSTM mainly consists of a cell, keep gate, write gate, and an output gate, as shown in Figure 3. The key behind using LSTM for this problem is that the cell remembers the patterns over a long period, and three portals help regulate the information flow in and out of the system. The details are presented in Algorithm 2.

Input to the algorithm is eight attributes enlisted in Table 3, measured from healthy and diabetic patients. The proposed LSTM-based diabetes prediction algorithm is trained with 80% of the data, and the remaining 20% is used for testing. We fine-tuned the prediction model by using a different number of LSTM units in the cell state. This finetuning helps to identify more prominent features in the dataset. These features will be kept in the cell state of the keep gate of the LSTM and will be given more weightage because they provide more insights to predict BG level. After that, we updated the network's weights by pointwise addition of the cell state and passed only those essential attributes for BG prediction. At this stage, we captured the dependencies between diabetes parameters and the output variable. Finally, the output gate updates the cell state and outputs/ forwards only those variables that can be mapped efficiently on the outcome variable.

The diabetes prediction algorithm consists of three fundamental steps. First, weights are initialized and a sigmoid unit is used in the forget/keep gate to decide which information should be retained from previous and current inputs (C_{t-1}, h_{t-1}, and x_t). The input/write gate

Input: No. of pregnancies, BG, BP, skin thickness, BMI, age, weight, HB
Output: A Trained Diabetes Classification Model
Method: Initialize all weights to a small random number
while (EER ≤ Threshold) **do**
  for all Training examples, **do**
    Input training example to the network & compute output;
  end for
  for all output unit k, **do**
    $\delta_k \leftarrow O_k(1 - O_k)(t_k - O_k)$
  end for
  for all hidden unit h **do**
    $\delta_h \leftarrow O_h(1 - O_h) \sum_{k \in \text{output}} w_{h,k}\delta_k$
  end for
  Update each network weight $w_{i,j}$;
    $w_{i,j} \leftarrow w_{i,j} + \Delta w_{i,j}$; $\Delta w_{i,j} = \eta \delta_j x_{i,j}$
End

Algorithm 1: Diabetes classification algorithm using MLP for healthcare.

takes the necessary information from the keep gate and uses a sigmoid unit which outputs a value between 0 and 1. Besides, a Tan_h unit is used to update the cell state C_t and combine both outputs to update the old cell state to the new cell state.

Finally, inputs are processed at the output gate and again a sigmoid unit is applied to decide which cell state should be output. Also, Tan_h is applied to the incoming cell state to push the output between 1 and − 1. If the output of the gate is 1, then the memory cell is still relevant to the required production and should be kept for future results. If the output of the gate is 0, the memory cell is not appropriate, so it should be erased. For the write gate, the suitable pattern and type of information will be determined written into the memory cell. The proposed LSTM model predicts the BG level (h_t) as output based on the patient's existing BG level (X_t).

2. Experimental Studies

The proposed diabetes classification and prediction algorithm is evaluated on a publicly available PIMA Indian Diabetes dataset (https://www.niddk.nih.gov/healthinformation/diabetes). Besides, a comparative analysis is performed with state-of-the-art algorithms. The experimental results show the supremacy of the proposed algorithm as compared to state-of-the-art algorithms. The details of the dataset, performance measures, and comparative analysis performed are described in the following sections.

Table 2: Performance comparison of classifiers in diabetes classification.

Algorithm	Accuracy (%)	Recall (%)	Precision (%)
Logistic regression	73.05	72.7	73
Random forest	77.4	75.7	76.9
Proposed fine-tuned MLP	86.083	85.1	86.6

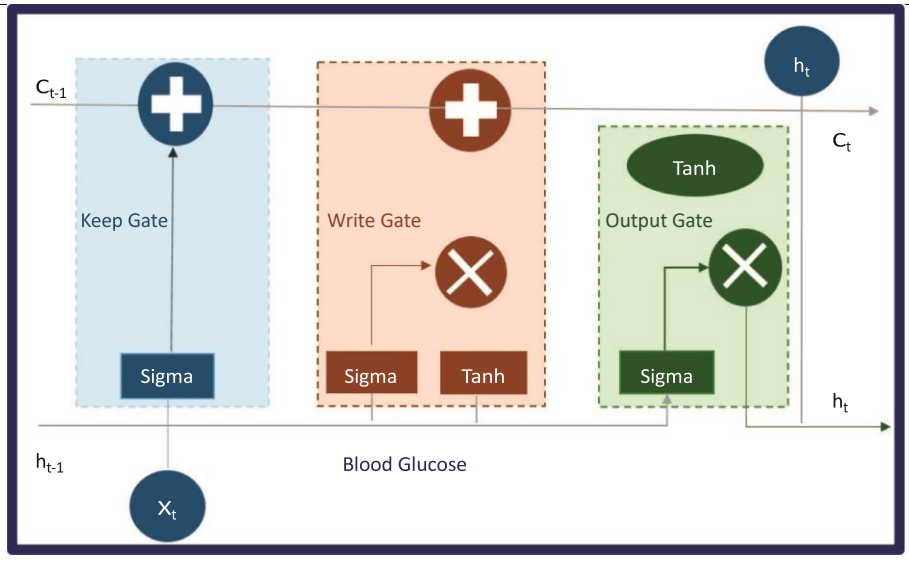

Figure 3: BG prediction using long short-term memory (LSTM) algorithm.

Input: Attributes (pregnancies, glucose, blood pressure, skin thickness, insulin, BMI, Diabetes Pedigree Function, age)
Output: BG Prediction Model
Method: Initialize all weights to a small random number
Keep Gate
while ((h_t and x_t) 1) **do**
end while
end
while
for all (Number in Cell State C_{t-1}), **do**
f_t $\sigma(W_f.[C_{t-1}, h_{t-1}, x_t] + b_f)$
end for
Write Gate
While ((C_t updating to C_t) **do**
for all (Cell in the Cell State) **do**
i_t $\sigma(W_i.[C_{t-1}, h_{t-1}, x_t] + b_i)$
C'_t $\tanh(W_c.[h_{t-1}, x_t] + b_c)$
C_t $f_t * C_{t-1} + t * C'_t$
end for Output Gate
while (Output Condition is not met) **do**
for all (Sigmoid and Output layer), **do**
o_t $\sigma(W_o.[C_t, h_{t-1}, x_t] + b_o)$
h_t $\tanh(o_t * \sigma_h(c_t)))$
end for
end

Algorithm 2: Diabetes prediction algorithm by exploiting LSTM for healthcare.

5.1. Dataset. This study used the PIMA Indian Diabetes using this dataset is to build an intelligent model that can (PID) dataset taken from the National Institute of Diabetes predict whether a person has diabetes or not, using some and Kidney Diseases center [59]. The primary objective of measurements included in the dataset. There are eight

Table 3: Description of variables in the dataset.

Attributes	Description	Mean	Std. deviation	Range
Pregnancies	No. of pregnancies	3.85	3.37	0–17
Glucose	2 hours of oral glucose tolerance test for plasma glucose concentration	121	32	0–199
Blood pressure	Blood pressure in mm Hg	69.1	19.3	0–122
Skin thickness	Skinfold thickness of triceps (mm)	20.5	15.9	0–99
Insulin	Two hours of serum insulin (mu U/ml)	79.8	115	0–846
BMI	Body mass index (weight in kg/(height in m)^2)	32	7.88	0–67
Diabetes Pedigree Function	Attribute used in diabetes prognosis	0.47	0.33	0.078–2.4
Age	Age (years)	33.2	11.8	21–81
Outcome	Class variable (0 or 1)	0.35	0.48	Y/N

medical predictor variables and one target variable in the dataset. Diabetes classification and prediction are a binary classification problem. The details of the variables are shown in Table 3.

The dataset consists of 768 records of different healthy and diabetic female patients of age greater than twenty-one, as shown in Figure 4. The feature value distribution is shown in Figure 5. The target variable outcome contains only two values, 0 and 1. The primary objective of using this dataset was to predict diabetes diagnostically. Whether a user has a chance of diabetes in the coming four years in women belongs to PIMA Indian. The dataset has a total of eight variables: glucose tolerance, no. of pregnancies, body mass index, blood pressure, age, insulin, and Diabetes Pedigree Function. All eight attributes shown in Table 3 are used for the training dataset in the classification model in this work.

5.2. Experimental Result and Discussion. This paper compares the proposed diabetes classification and prediction system with state-of-the-art techniques using the same experimental setup on the PIMA Indian dataset. The following sections highlighted the performance measure used and results attained for classification and

prediction, and a comparative analysis with baseline studies is presented.

5.2.1. Performance Metrics. Three widely used state-of-theart performance measures (Recall, Precision, and Accuracy) are used to evaluate the performance of proposed techniques, as shown in Table 4. TP shows a person does not have diabetes and identified as a nondiabetic patient, and TN shows a diabetic patient correctly identified as a diabetic patient. FN shows the patient has diabetes but is predicted as a healthy person. Moreover, FP shows the patient is a healthy person but predicted as a diabetic patient. The algorithm utilized 10-fold cross-validation for training and testing the classification and prediction model.

For diabetes prediction, the two most commonly used performance measures are the means correlation coefficient (r/Pearson R) and root mean square error (RMSE), as shown in Table 5. R is mainly used to measure the linear dependence strength among the two variables. One variable is for actual value, and another variable is for predicted values. RMSE generates a hint of the overall correctness of the estimate. There can be three values for correlation: 0 for no relation, 1 for positive correlation, and − 1 for the negative correlation. RMSE shows the difference between actual values and predicted values.

5.2.2. Attained Results of Diabetic Classification Technique. For diabetic classification, three state-of-the-art classifiers are evaluated on the PIMA dataset. The results illustrate that the fine-tuned MLP algorithm obtained the highest accuracy of 86.083% as compared to state-of-the-art systems, as shown in Table 2.

It is evident from the results that our proposed calibrated MLP model could be used for the effective classification of diabetes. The proposed classification approach can also be beneficial in the future with our proposed hypothetical system. Data of weight scales, blood pressure monitor, and blood glucometer will be collected through sensor devices such as BLE and input of user's demographic data (for example, date of birth, height, and age). The proposed MLP algorithm outperforms with 86.6% Precision, 85.1% Recall, and 86.083% Accuracy, as shown in Figure 6. These results are outstanding for decision-making with the proposed hypothetical system to determine patient diabetes, T1D or T2D.

We also have explored the dataset used in Andy Choens' study [27]. This dataset consists of records of only one patient. The information was recorded every five minutes. The collection of data was made by using a sensor device (a CGM device). This device allows the patient to store information about BG every five minutes. So, the recorded data by using this device are in massive amounts. Dataset was limited, and most data were noisy that can affect the accuracy of the proposed system, so we neglected it.

5.2.3. Achieved Results of Diabetic Prediction Techniques. For diabetic prediction, we implemented three state-of-theart algorithms, i.e., linear regression, moving averages, and LSTM. Notably, we fine-tuned LSTM and compared its performance with other algorithms. It is evident from Figure 7 and Table 6 that the LSTM outperformed as compared to other algorithms implemented in this study.

Table 2 shows the performance values of prediction models with RMSE and r evaluation measures. The proposed fine-tuned LSTM produced the highest accuracy, 87.26%, compared to linear regression and moving average. We can see in Table 6 that the correlation coefficient value is 0.999 using LSTM, − 0.071 for linear regression, and 0.710 for moving average, as shown in Figure 7.

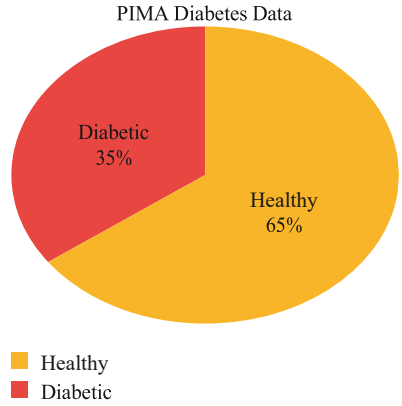

Figure 4: PIMA data distribution.

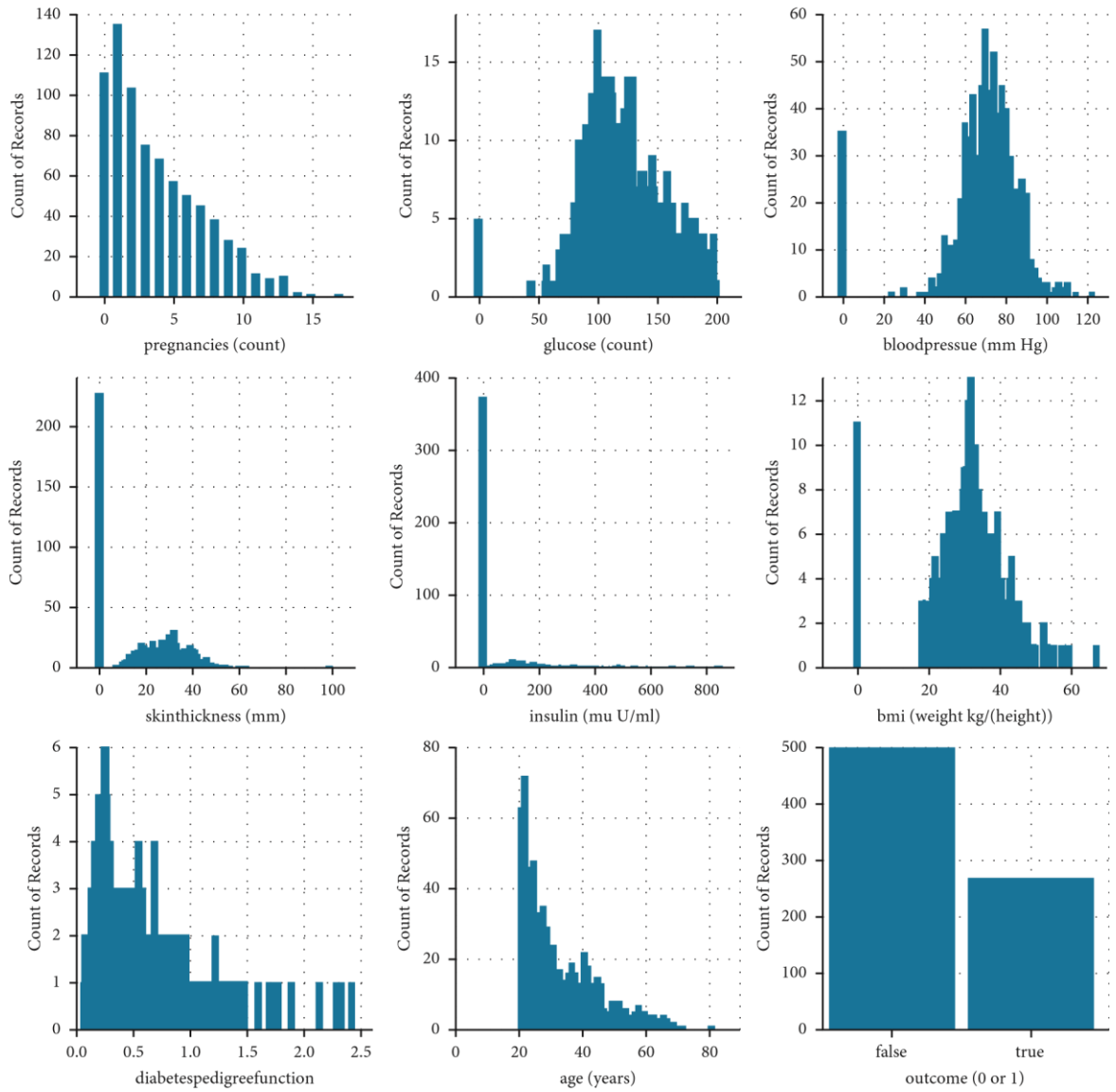

Figure 5: Dataset features' distribution visualization.

Table 4: Performance metrics for diabetes classification.

Performance metric	Formula
Recall	TP/(TP + FN)
Precision	TP/(TP + FP)
Accuracy	(TP + TN)/(TP + TN + FP + FN)

Table 5: Performance measure for diabetes prediction.

Performance metric	Formula
	□
r	$(n(\square XY) - (\square X)(\square Y))/ \quad [n\square X^2 - (\square X)^2][n\square Y^2 - (\square Y)^2]$
	$\square\ [\square_{Ni1}\ (y_{fi} - y_{f0})_2/N]$
Root mean square error (RMSE)	
Accuracy	(TP + TN)/(TP + TN + FP + FN)

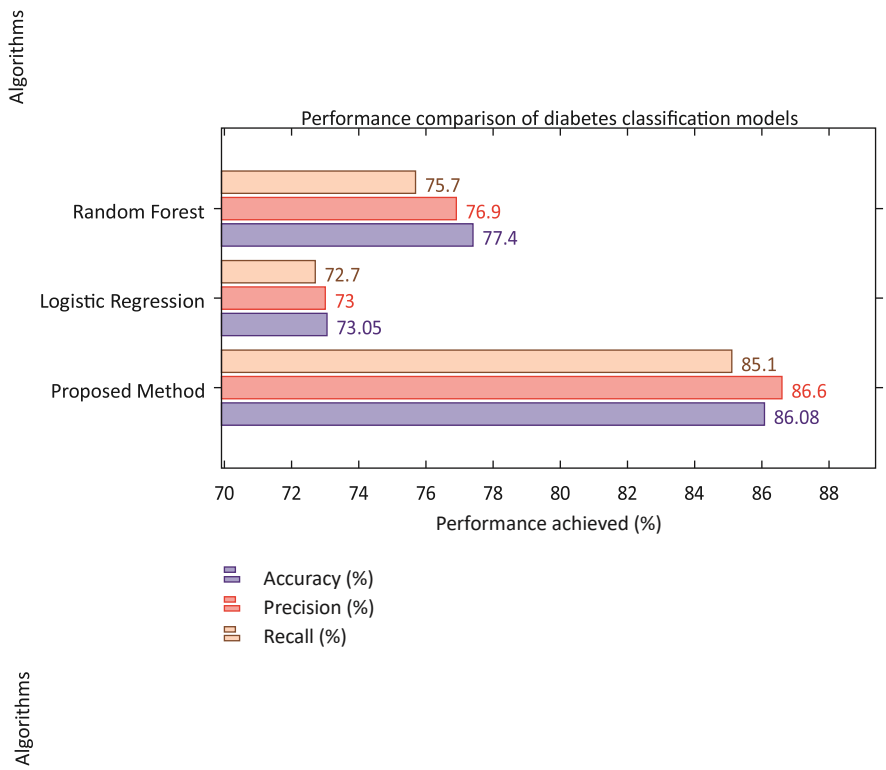

Figure 6: Performance comparison of classifiers.

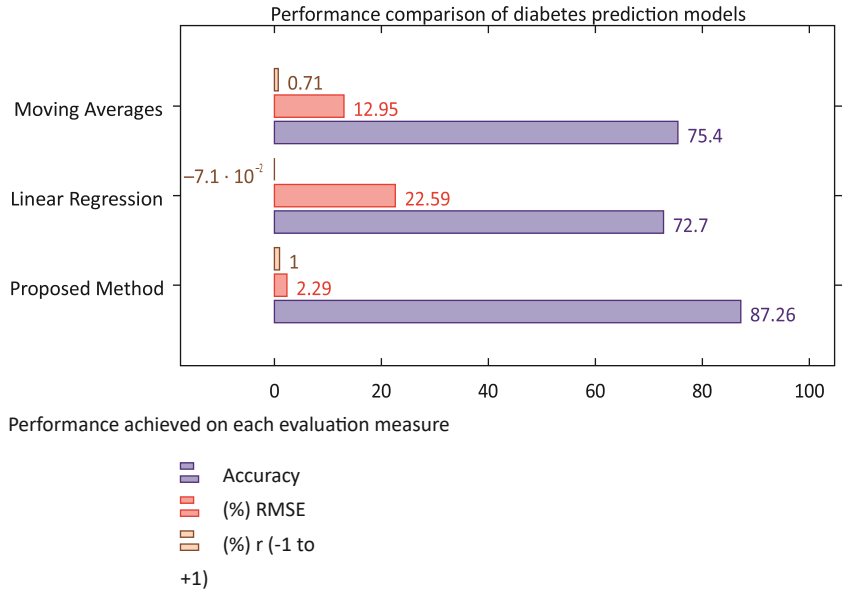

Figure 7: Performance comparison of forecasting model.

Table 6: Forecasting model comparison for BG.

Algorithm	*r*	RMSE	Accuracy
Moving average	0.71	42.946	75.4
Linear regression	− 0.071	82.592	72.7
Proposed fine-tuned LSTM	0.999	2.285	87.26

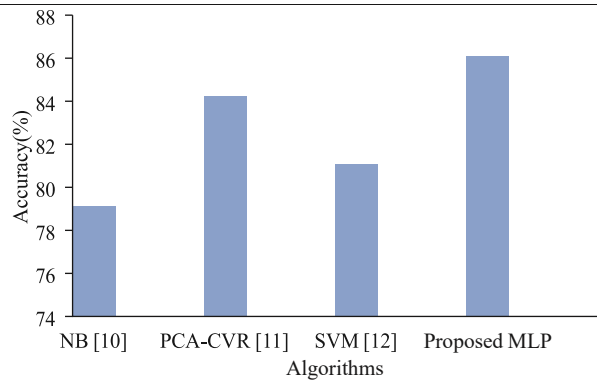

Figure 8: Proposed diabetes classification method vs. state-of-the-art techniques.

5.2.4. Comparison of the Proposed Method with Baseline Studies. Different baseline studies have been implemented and compared with the proposed system to verify the performance of the proposed diabetes classification and prediction system. Mainly, we focus on those studies that used the PIMA dataset.

First, we compare the state-of-the-art diabetes classification techniques with the proposed technique. All the baseline techniques [17–19] used the PIMA dataset and the same evaluation measures used in this study. In particular, the authors compared na¨ıve Bayes [17], PCA_CVR (classification via regression) [18], and SVM [19] with different machine learning techniques for diabetes classification. However, the proposed fine-tuned MLP-based diabetes classification technique outperformed as compared to baseline studies, as shown in Figure 8.

Several attempts have also been made in the literature for diabetic prediction due to its importance in real life. For this comparison, we have chosen the most recent and state-ofthe-art techniques. We compare the proposed system performance with the recent state-of-the-art systems [60–65], as shown in Figure 9 and Table 7. The proposed method outperformed as compared to state-of-the-art systems with an accuracy of 87.26%, all the compared systems evaluated on the PID with the same experimental setup.

3. Proposed Hypothetical IoT-Based DiabeticMonitoring System for Healthcare

This study has also proposed the architecture of a hypothetical diabetic monitoring system for diabetic patients. The proposed hypothetical system will enable a patient to control, monitor, and manage their chronic conditions in a better way at their homes. The monitoring system will store the health activities and create interaction between patients, smartphones, sensor medical devices, web servers, and medical teams by providing a platform having wireless communication devices, as shown in Figure 10. The central theme of the proposed healthcare monitoring system is the collection of data from sensors using wireless devices and transmitting to a remote server for diagnosis and treatment of diabetes. Knowledge-based data are stored. Rule-based procedures will be applied for the suggestions and treatment of diabetes, informing the patient about his current health condition, prediction, and recommendation of future changes in BG.

First, essential data about patient health will be collected from sensors such as BLE wireless devices. Data comprised weight, blood pressure, blood glucose, and heartbeat, along with some demographic information such as age, sex, name, and CNIC (Social Security Number). Some information is required in the application installed on the user's mobile and sensor data. All completed data in the application will be transferred to the real-time data processing system. On the other side, aggregate data will be stored in MongoDB for future processing. Analysis and prepossessing techniques are performed to extract rules from the knowledge base for the treatment and suggestions about the user. Results and treatment procedures will be sent to the monitoring system, and finally, the user can get the output by interacting with their android mobile phone. In the end, the patient will know about the health condition and risk prediction of diabetes based on the data transferred by their application and stored data from history about the user.

6.1. Tools and Technology for Implementation of Hypothetical System for Healthcare. The proposed structural design for hypothetical real-time processing and monitoring of diabetes is shown in Figure 11. The data from the user's mobile will be transmitted in the JavaScript Object Notation (JSON) format to the Application Program Interface (API) in any language. The data produced at this stage will be in the form of messages, which are then transferred to the Kafka application [27]. Kafka will store all the data and messages and deliver the required data and processed output to the endpoints that could be a web server, monitoring system, or a database for permanent storage. In Kafka, application data are stored in different brokers, which can cause latency issues. Therefore, within the system architecture, it is vital to consider processing the readings from the sensors closer to the place where data are acquired, e.g., on the smartphone.

The latency problem could be solved by placing sensors close to the place, such as a smartphone where data are sent and received.

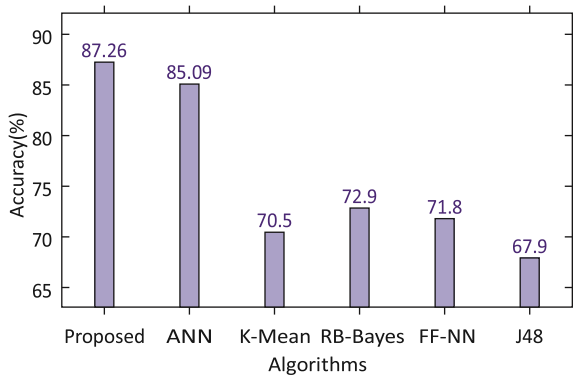

Figure 9: Proposed diabetes prediction method vs. state-of-the-art systems.

Table 7: Proposed prediction method vs. state-of-the-art systems.

Algorithm	Accuracy (%)
J48 [62]	67.9
K-mean [60]	70.5
Feed forward-neural network [63]	71.8
RB-Bayes [64]	72.9
Naive Bayes [65]	76.3
Artificial neural network [61]	85.09
Proposed method (LSTM)	87.26

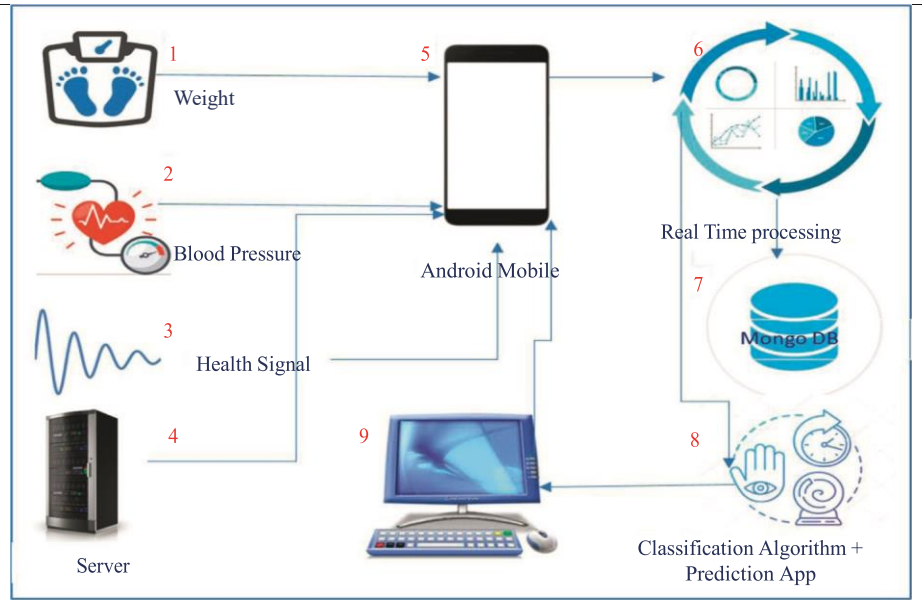

Figure 10: The proposed hypothetical architecture of the healthcare monitoring system.

This inclusion will make the overall network architecture compliant to the emerging Edge and Fog computing paradigms, whose importance in critical infrastructures such as hospitals is gaining momentum. It is essential to consider the Edge and Fog computation

Figure 11: Implementation level details of the proposed hypothetical system.

paradigm while sending and receiving data from smartphones to increase the performance of the hypothetical system. Edge computing utilizes sensors and mobile devices to process, compute, and store data locally rather than cloud computing. Besides, Fog computing places resources near data sources such as gateways to improve latency problems [9].

Apache Kafka will be used in real time as a delivery agent for messages in a platform that allows fault-tolerant, tall throughput, and low-latency publication. The vital signs' data collected by the patients are placed using the JSON format and then transmitted using wireless devices with the help of an android application having HTTP along with REST API for the confined remote server for the design [28]. Moreover, Node.js for web design will be used as a REST API to collect sensor data. Kafka application will receive it in the form of streams of records.

The sensor data that comes from the Kafka application is continuously generated and stored on the server. In the proposed system, the MongoDB NoSQL database will be used for data storage due to its efficiency in handling and processing real-world data [29]. The stored diabetes patient data can be input into our proposed diabetes classification and prediction techniques to get useful insights.