applied sciences

MDPI

Article

An Optimization-Based Diabetes Prediction Model Using CNN and Bi-Directional LSTM in Real-Time Environment

Parul Madan [1], Vijay Singh [1], Vaibhav Chaudhari [2], Yasser Albagory [3], Ankur Dumka [4], Rajesh Singh [5], Anita Gehlot [5], Mamoon Rashid [6,*], Sultan S. Alshamrani [7] and Ahmed Saeed AlGhamdi [3]

[1] Department of Computer Science and Engineering, Graphic Era Deemed to Be University, Dehradun 248002, India; parulmadan.cse@geu.ac.in (P.M.); vijaysingh.cse@geu.ac.in (V.S.)
[2] Department of Computer Science & Information Systems, BITS Pilani K.K. Birla Goa Campus, Sancoale 403726, India; chaudharivaibhav12@gmail.com
[3] Department of Computer Engineering, College of Computers and Information Technology, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia; y.albagory@tu.edu.sa (Y.A.); asjannah@tu.edu.sa (A.S.A.)
[4] Department of Computer Science & Engineering, Women Institute of Technology, Dehradun 248007, India; ankurdumka2@gmail.com
[5] Department of Research and Development, Uttaranchal Institute of Technology, Uttaranchal University, Dehradun 248007, India; rajeshsingh@uttaranchaluniversity.ac.in (R.S.); anita.ri@uttaranchaluniversity.ac.in (A.G.)
[6] Department of Computer Engineering, Faculty of Science and Technology, Vishwakarma University, Pune 411048, India
[7] Department of Information Technology, College of Computer and Information Technology, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia; susamash@tu.edu.sa
* Correspondence: mamoon.rashid@vupune.ac.in; Tel.: +91-7814346505

check for updates

Citation: Madan, P.; Singh, V.; Chaudhari, V.; Albagory, Y.; Dumka, A.; Singh, R.; Gehlot, A.; Rashid, M.; Alshamrani, S.S.; AlGhamdi, A.S. An Optimization-Based Diabetes Prediction Model Using CNN and Bi-Directional LSTM in Real-Time Environment. *Appl. Sci.* **2022**, *12*, 3989. https://doi.org/app12083989

Academic Editors: Andrea Barucci, Chiara Marzi and Marco Giannelli

Received: 10 March 2022
Accepted: 11 April 2022
Published: 14 April 2022

Featured Application: Diabetes is a common chronic disorder defined by excessive glucose levels in the blood. A good diagnosis of diabetes may make a person's life better; otherwise, it can cause kidney failure, major heart damage, and damage to the blood vessels and nerves. As a result, diabetes classification and diagnosis are vital tasks. By using our proposed methodology, clinicians may obtain complete information about their patients using real-time monitoring. To gain new insights, they can combine historical information with current data, making it easier for them to perform more thorough and comprehensive treatments than before, and they will be able to provide proactive care, which will help to improve health outcomes and reduce hospital re-admissions.

Abstract: Diabetes is a long-term illness caused by the inefficient use of insulin generated by the pancreas. If diabetes is detected at an early stage, patients can live their lives healthier. Unlike previously used analytical approaches, deep learning does not need feature extraction. In order to support this viewpoint, we developed a real-time monitoring hybrid deep learning-based model to detect and predict Type 2 diabetes mellitus using the publicly available PIMA Indian diabetes database. This study contributes in four ways. First, we perform a comparative study of different deep learning models. Based on experimental findings, we next suggested merging two models, CNN-Bi-LSTM, to detect (and predict) Type 2 diabetes. These findings demonstrate that CNN-Bi-LSTM surpasses the other deep learning methods in terms of accuracy (98%), sensitivity (97%), and specificity (98%), and it is 1.1% better compared to other existing state-of-the-art algorithms. Hence, our proposed model helps clinicians obtain complete information about their patients using real-time monitoring and can check real-time statistics about their vitals.

Keywords: deep learning; CNN; Bi-LSTM; PIDD; real-time; prediction model

1. Introduction

Diabetes is a prevalent chronic illness characterized by the presence of high glucose in the blood. A proper diagnosis of diabetes can make a person's life healthier; otherwise, it may cause kidney failure, serious damage to the heart, and may also affect the blood vessels and nerves [1]. There are three types of diabetes: Type 1, Type 2, and Gestational, which are found in the human body [2]. If our body does not utilize the insulin that our pancreas produces, then it is a severe explanation for Type 2 diabetes. People below 30 usually suffer from Type 1 diabetes, which cannot be treated with oral medicines. It requires additional insulin therapy. However, people of middle age and older who have type 2 diabetes may recover by living a healthy lifestyle and receiving proper checkups. However, gestational diabetes is a common kind of diabetes that affects women during pregnancy. High blood glucose levels may be caused by several hormones and increased insulin content during pregnancy [3].

There are some necessary diagnostics tests through which we can diagnose diabetes, such as A1c, random blood sugar, fasting blood sugar, and oral glucose tolerance tests, which need lots of parameters to predict diabetes properly. One cannot diagnose diabetes with one parameter, such as excessive consumption of vitamin E can alleviate A1c levels, while B9 and B12 can lower A1c levels. As a result, several criteria must be combined to diagnose diabetes accurately. There are many factors that can help to diagnose diabetes. These factors include glucose level and BMI, diabetes pedigree, blood pressure, age, pregnancy, skin thickness, and insulin, as referenced in Table 1.

Table 1. PIMA Diabetes real-time Dataset.

S.no	Parameters	Description of Parameters	Range
1.	Pregnancies	No. of times pregnant	0–17
2.	Glucose	Plasma glucose 2 h in an oral glucose tolerance test (mg/dl)	0–199
3.	Blood-pressure	Diastolic blood pressure (mm Hg)	0–122
4.	Skin Thickness	Skin fold thickness (mm)	0–99
5.	Insulin	2-h serum insulin (mu U/mL)	0–846
6.	BMI	Body mass index (weight in kg/(height in m)^2)	0–67.1
7.	Diabetes-Pedigree	Diabetes pedigree function (weight in kg/(height in m)^2)	0.08–2.42
8.	Age	Age (years)	21–81

1.1. Diabetes Prediction in Real-Time Environment

Intensive care of blood glucose levels helps in preventing and treating diabetic problems [4]. Innovative biosensors that may enable real-time monitoring of a patient's health, as well as recent advancements in information and communication technology (ICT), provide a new viewpoint on diabetes treatment. Diabetic patients can monitor their blood glucose levels by using self-monitoring blood glucose (SMBG) portable devices [5] or continuous glucose monitoring (CGM) sensors [6] to track glucose variations, because of which they will be able to respond quickly and with the necessary actions. The findings suggest that monitoring patients' glucose levels can help them control their disease and enhance their diabetes management performance [7]. The greatest option for improving diabetes care is glucose monitoring in a real-time system that includes sensors, a gateway (smartphone), and a cloud system [8]. It uses a smartphone as a gateway to acquire sensor data from a sensor node connected to the body [9]. Wireless technology is required for communication between the sensor node and the smartphone, as well as low-power operation for the sensor node, and the best choice for this is Bluetooth low energy (BLE) [10].

1.2. Motivation

There are 415 million people worldwide suffering from type 2 diabetes, and the major cause is an unhealthy lifestyle. According to WHO, 82% of deaths are due to noncommunicable disorders, and diabetes is one of them [1]. According to Vhaduri et al. [11], continuous glucose monitoring using personal health devices in diabetes care may benefit from the early detection of the disease. Medical recommendations advocate for early detection to identify risk-prone individuals and encourage patients to proactively self-monitor their lifestyle to reduce risk factors. Remote patient monitoring (RPM) may help decrease the alarming number of diabetes-related mortality by providing early detection and timely alerts to patients and medical practitioners. RPM lowers the need for routine examinations, allows continuous treatment efficacy to be measured, and allows for intervention strategies [12]. Through real-time monitoring, clinicians may obtain complete information about thei patients. To obtain new insights, they can combine historical information with current data, making it simpler for them to perform a more thorough and comprehensive treatment than before, and they are able to provide proactive care, which assists in improving health outcomes and minimizing hospital re-admissions. Additionally, patients themselves can monitor their vitals in real-time, such as blood pressure and heart rate. This not only motivates patients to regularize their habits, which contributes a lot to improving their health conditions and also helps clinicians to receive real-time statistics about their vitals.

1.3. Major Contributions

In this research, we have made a fourfold contribution:

- We made a comparison of several deep learning algorithms such as CNN, bi-directional long-short memory (Bi-LSTM), deep neural network (DNN) [13], and their combinations, CNN-LSTM and CNN-BiLSTM, for the detection and prediction of diabetes using the static PIMA Indian dataset (PIDD) [14];
- We used the best parameters to train our models. We ran a grid search algorithm that found the best values for parameters such as learning rate, epochs, optimizer, batch sizes, and hidden units;
- We split our dataset into test and training sets by using 10-fold cross-validation, and the precision of each model was improved. On the other hand, CNN-Bi-LSTM outperformed with 98% accuracy, 97% sensitivity, and 98% specificity;
- We proposed a framework to test our optimized models using a real-time dataset.

The remainder of the paper is structured in the following manner: The second section discusses related work; Section 3 describes the methodology, which defines the PIMA Indian dataset and preprocessing steps to filter data, the models used to diagnose patients with diabetes, and the proposed framework to diagnose diabetes in the real-time environment; the results and discussions are addressed in Section 4; and, finally, Section 5 draws conclusions and discusses future scope.

2. Related Work

In India, diabetes is an inescapable problem as over 70% of the grown-up populace suffers from diabetes. Different scientists attempted to detect and predict diabetes by applying various ML and data mining methodologies, and some applied deep learning and fuzzy logic [15]. Data mining approaches have supplanted old procedures because they are more accurate, precise, and predictive in their predictions. Furthermore, machine learning is an artificial intelligence system that learns correlations between nodes without the need for previous training [16]. The significant capacity of machine learning approaches to drive the prediction model without extensive training is connected to the mechanism that underlies these techniques. Methods such as data mining and machine learning assist in detecting information that is otherwise difficult to identify when utilizing a cutting-edge technique [17]. Some of the past related research work focuses on the detection and prediction of diabetes using PIDD [14,18–22]. Numerous ML methods, such as decision trees, RF, SVM, and Naive Bayes, have recently shown promising outcomes in various types of medical research. Zolfagri et al. [23] suggested a way to identify diabetes in female Pima Indian populations by using a neural network and support vector machine (SVM) models. Sanakal et al. [24] diagnosed diabetes using the prognosis of fuzzy c-means clustering and SVM. Sneha et al. [1] used PIDD and chose the best attributes that are essential for classification and excluded the remainder, such as plasma, glucose, postprandial, pregnancy, serum creatinine, and HBAIC attributes, from the dataset and applied different ML algorithms such as SVM [25], Naive Bayes, and RF. However, SVM outperformed the simplest with an accuracy of 77.37%. Karatsiolis et al. [26] suggested a region-based SVM method for the diagnosis of diabetes on the Pima Indian medical dataset. Similar to Karatsiolis, Kumari et al. [15] suggested SVM for the detection of Type 2 diabetes. Al et al. [27] applied a decision tree approach to identify type 2 diabetes.

The researchers Dey et al. [18] and Zou et al. [28] developed a web-based strategy to forecast diabetes using machine learning techniques on the PIMA Indian dataset. On the other hand, no single study has examined all the well-known supervised learning methods in a comprehensive manner. Sivastava et al. [20] used an ANN approach to predict diabetes using the PIMA Indian dataset. Saji et al. [29] developed a multilayer perceptron that was used to predict diabetes. Using an autotuned multilayer perceptron, Jahangir et al. [30] suggested an expert system to predict diabetes. Kannadasan et al. [31] proposed a DNN for the classification of Type 2 diabetes using stacked encoders for feature engineering, a softmax function for classification, and a backpropagation method for fine-tuning the network. Training of the model was performed with PIDD along with 786 patient records and eight features and achieved an accuracy of 86.26%. Apporva et al. [32] used the decision tree technique to predict type-2 diabetes using the PID dataset [33]. A comparison of performance using an SVM classifier showed that the decision tree successfully predicted type-2 diabetes. In summary, the above-discussed techniques have some pros and cons, which are as follows: For example, ML algorithms such as RF, decision trees, and SVM are helpful if we use them for classification problems, except for regression, where they may not be suitable for predicting training data beyond the range. Similarly, within the decision tree, if there is a little change in data, it may affect the entire structure of the model [31]. Furthermore, SVM faces minor issues with noisy data [34]. Therefore, these ML algorithms are suitable for classification problems. However, ANN and CNN are good at making predictions because, in backpropagation, these methods obtain good results when they use gradients to update the weights. However, they have some problems, such as vanishing gradient problems or exploding gradient problems, where the value of gradients (a value used to update the weights) decreases with backpropagation, so the value becomes too small and does not help much with learning. However, it is possible to overcome these limitations by applying an LSTM and GRU by using ReLU, which allows capturing the impact of the earliest given data. Moreover, by tuning the burden value during the training process, the vanishing gradient issue is usually avoided [35,36]. Our study used CNN-Bi-LSTM, where CNN is employed for feature extraction, and Bi-LSTM has a cell state memory during its training phase, which captures the impact of earlier stages. Besides, it has another peehole connection, which helps remove the vanishing gradient problem. Furthermore, Bi-LSTM can collect information in two ways: one from the past and one from the future, which helps more efficiently with the prediction of diabetes.

3. Methodology

In this part, we proposed a framework by describing several components of the framework, such as the dataset that was uploaded to cloud servers; preprocessing procedures that are required for data cleaning; description of models used for detection and prediction of Type 2 diabetes where we explained the implementation of three models using a static PIMA dataset such as CNN, CNN-LSTM, and CNN-Bi-LSTM. Here, CNN-Bi-LSTM is further explained through several phases, such as model training using static PIDD. Then the model is optimized using a grid search algorithm. Furthermore, training results were utilized for real-time testing, and lastly, the prediction process of CNN-Bi-LSTM was discussed [36].

3.1. Availability of Real-Time Dataset for Training and Testing

This section provides a comprehensive description of the real-time PIMA Indian dataset [14] (1UCI: https://archive.ics.uci.edu/ml/support/Diabetes (accessed on 15 January 2020)), which consists of 768 female patients [37] who were between the ages of 21 and 25 years. There are 268 diabetics among them, and the rest are healthier ones. The dataset consists of 8 vital parameters, and a complete overview of the dataset is given in Table 1, where we input parameters and their ranges given, such as the number of times a woman was pregnant and expressed in Figure 1.

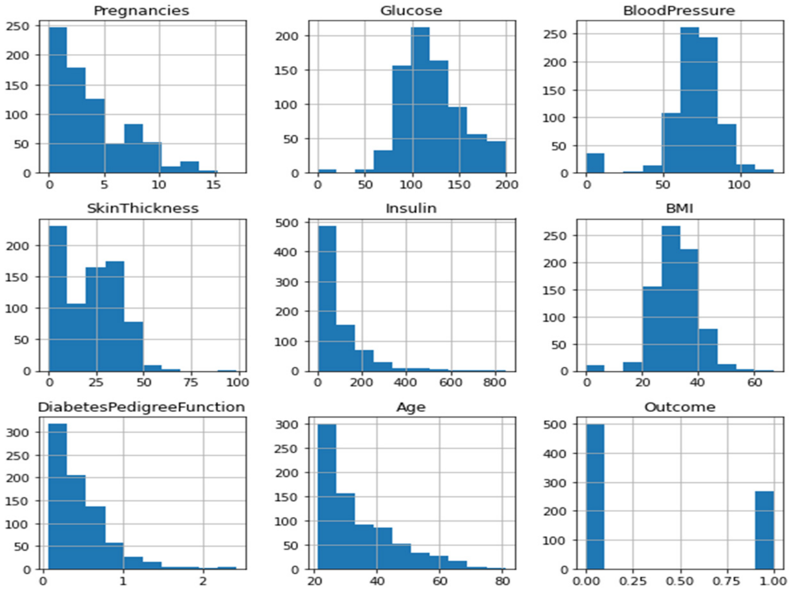


Figure 1. Representation of input parameter.

Remark 1. *In Table 1, range specifies a threshold value for, e.g., BMI (weight in kg)/(height in m)*^{2}.

Now we discuss the reasons for selecting these parameters for our model: There is a high possibility that glucose levels may increase during pregnancy, which can lead to diabetes-related complications [38]. One of the main reasons for diabetes mellitus is the presence of high glucose in the blood. Obesity is another cause of Type 2 diabetes. Diabetes is a genetic disease, so diabetes pedigree functions are critical in providing data. Additionally, imbalances in insulin may also cause diabetes mellitus in people who may consume insulin and suffer from skin-thickening problems. Lastly, as we age, the probability of diabetes increases, specifically after the age of 45. Therefore, it is evident that all these biological parameters play a significant role in measuring and correctly classifying diabetes mellitus.

3.2. Preprocessing of Real-Time Data

To a considerable degree, the accuracy of the data determines the results of the prediction. This indicates that preprocessing data plays a vital function in the model [39]. In this analysis, we picked some of the necessary methods to refine the initial dataset. Firstly, there are some incomplete and inaccurate dataset values due to mistakes or deregulation. These pointless values contributed to several deceptive experimental results, such as the diastolic blood pressure, systolic blood pressure, and body mass index values may not have been 0 in the initial dataset, implying that the true value was absent. We used the mean from the training data to replace all missing values and reduce irrelevant values shown in Table 2.

Second, elimination of outliers: any attribute that does not conform to the usual boundary is referred to as an outlier, as seen in Figure 2, and can be removed by using Equation (3):

$$Q_1 = a.quantile(0.25) \tag{1}$$

$$Q_3 = a.quantile(0.75) \tag{2}$$

$$IQR = Q_3 - Q_1 \tag{3}$$

where Equations (1) and (2) are the first and third quantile [38], respectively. *IQR* stands for interquartile range, and its values are shown in Table 3. All values that lie beyond this threshold (*IQR*) are termed outliers.

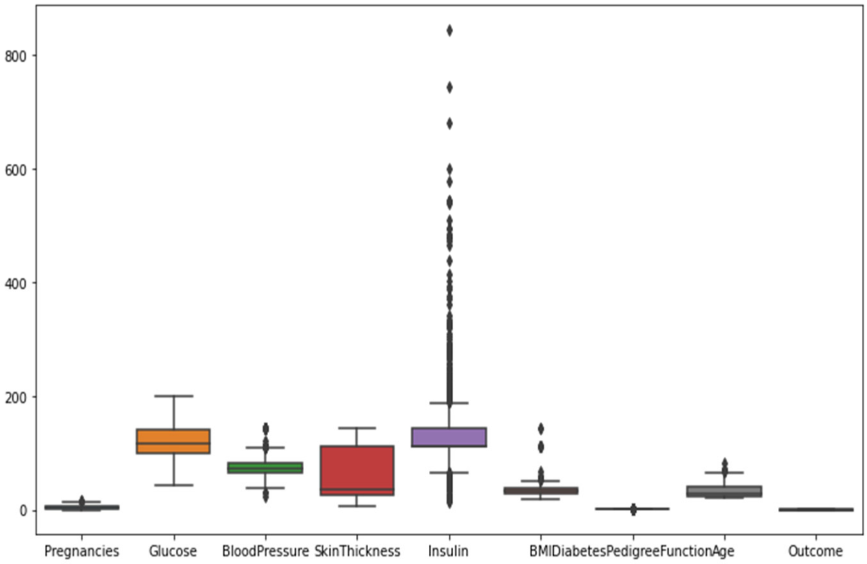


Figure 2. Box plot representation of data along with outliers.

Table 2. Missing values in PIMA dataset.

S.no	Attributes	Missing Values
1.	Pregnancies	0
2.	Glucose	5
3.	Blood-pressure	35
4.	Skin Thickness	227
5.	Insulin	374
6.	BMI	11
7.	Diabetes-Pedigree	0
8.	Age	0

Table 3. IQR values for PIMA Indian dataset.

S.no	Attributes	IQR Threshold Values
1.	Pregnancies	5
2.	Glucose	41.25
3.	Blood-pressure	18
4.	Skin Thickness	32
5.	Insulin	27.25
6.	BMI	9.3
7.	Diabetes-Pedigree	0.38
8.	Age	17.0
9.	Outcome	1

Filtered values are seen in Figure 3. The next step is to normalize data, such as bringing data into the range of 0 and 1 by adding normalized filters and calculating z-score values by using Equation (4) [40]. Where $\overline{a}$ is the mean or average value of the variable, a_i is input values, and s is the standard deviation of the variable. However, b_i is a new normal value. Table 4 shows mean, standard deviation, minimum, and maximum values of the PIAM dataset. This reduces the uncertainty of estimation and accelerates the process.

$$b_i = (a_i - \overline{a})/s \tag{4}$$

where

b_i : normalised value
a_i = input data
$\overline{a}$: input data average
s = input data standard deviation

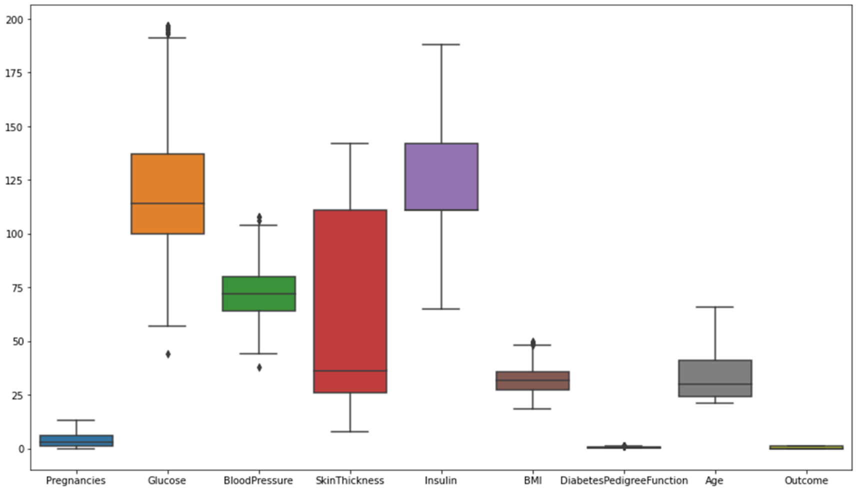


Figure 3. Box plot representation filtered data.

Table 4. Statistical analysis of PIMA Indian dataset.

S.no	Parameters	Mean	Standard Deviation	Minimum	Maximum
1.	Pregnancies	3.84	3.36	0	17
2.	Glucose	121.6	30.46	44	199
3.	Blood-pressure	74.8	16.68	24	142
4.	Skin Thickness	56.89	44.51	7	142
5.	Insulin	139.42	87.24	14	846
6.	BMI	33.63	12.22	18	142
7.	Diabetes-Pedigree	0.47	0.33	0	2.42
8.	Age	33.24	11.76	21	81
9.	Outcome	0.34	0.47	0	1

3.3. Feature Selection

Feature selection is the process of removing non-informative or redundant input characteristics from the dataset. Feature selection decreases the computational complexity of prediction algorithms. This minimizes prediction uncertainty and improves the model's overall efficacy.

The Chi-squared test is a non-parametric statistical technique used to examine the relationship between two variables [41]. The approach generates a number that quantifies the relationship between the input characteristics and the projected result. The greater the value, the stronger the connection between the input and output characteristics, and features with values less than the critical value are removed. As the Chi-squared approach operates on categorical data, the numerical values of the features in this dataset were discretized depending on their frequency of occurrence.

Extra trees apply several randomized decision trees to different subsets of the total dataset [42]. In the tree building process, the input variables and cut-off values are chosen at random to divide a node so that they are fully independent of the output variable. Each tree leads to a different model, which was trained with subsets of data, and the algorithm evaluates the relevance of the contributing features using a criterion known as the Gini index.

LASSO is an L1 regularization approach for feature selection that is used to facilitate dataset interpretation [43]. In this technique, regression analysis is used to estimate parameters and pick models at the same time, minimizing feature variability by lowering coefficients of noncorrelation characteristics to zero. Table 5 shows the essential characteristics chosen by each technique, along with their ranking measures. The Chi-squared test employs the chi-score, extra trees use the Gini index, and LASSO employs regression coefficients. The characteristics of glucose, insulin, BMI, and age were consistently scored as high in relevance and were chosen by each of the techniques used. After experimenting with the characteristics, it was discovered that removing the skin fold thickness and diabetes pedigree features enhanced the model's overall performance.

Table 5. PIMA Diabetes real-time Dataset.

S.no	Features	Chi-Squared Test	Extra Trees	LASSO
1.	No of Pregnancies	110.54	0.105	0.00
2.	**Glucose**	**1537.20**	**0.230**	**0.0065**
3.	Blood-pressure	54.26	0.089	0.0
4.	Skin Thickness	145.20	0.090	0.0
5.	**Insulin**	**6779.24**	**0.147**	**0.0004**
6.	**BMI**	**108.32**	**0.123**	**0.0006**
7.	**Age**	**189.30**	**0.128**	**0.0002**
8.	Diabetes-Pedigree	4.30	0.128	0.0

3.4. Data Augmentation

The synthetic minority oversampling approach (SMOTE) was utilized to eliminate biases in the produced models [44]. SMOTE is an oversampling approach that generates new samples from existing class samples to increase the number of minority class samples in the dataset. The method creates new minority class samples that are convex mixtures of two or more randomly selected neighboring data samples in the feature space rather than duplicates. A recent study showed that using SMOTE in clinical datasets improves model performance by decreasing the detrimental impact of unbalanced data.

3.5. Diabetes Prediction Models

This study aimed to develop a model for forecasting diabetes using CNN-Bi-LSTM that has not been used for diabetes classification and prediction. Recently, different approaches of deep learning, such as LSTM, CNN, and their derivatives, have been used for the classification of diabetes, although these methods achieve good accuracy in the predictions. However, they still face certain challenges, such as vanishing gradient problems and exploding gradient problems, that adversely affect the model's training. These drawbacks can be resolved by applying a combination of CNN and Bi-LSTM, which adjusts the weight value during the training phase to gather data results. This part clarifies the detailed architecture of CNN, CNN-LSTM, and CNN-Bi-LSTM over the PIDD [15] and then attempts to assess how well these models perform in terms of precision, sensitivity, and specificity.

3.5.1. Convolutional Neural Network

Here, in this section, we explained the role of the CNN by explaining the functionality of the different layers for the prediction of Type 2 diabetes mellitus. Initially, CNN was used mainly for image classification, but today, CNN can be applied in various domains.

Definition 1. *A CNN is a special kind of multi-layer perceptron identical to a traditional neural network where specific inputs are supplied to each neuron. These are self-learned neurons that learn from data with the assistance of weight and bias by conducting such operations as the dot product [23]. CNN is made of layers, namely: a convolutional layer, a maximum pooling layer, a flattening layer, and a fully connected layer. The goal of the convolutional layer is to learn the feature representation for the input data. It is the heart of the network and has local connections and weights for common features. In the first stage, input parameters are passed through the kernel and then outputs are sent via a nonlinear activation function ReLU, which does not activate all the neurons at the same time. It only activates those neurons which are in the range of 0 and 1. Then output neurons are passed through the pooling layer, which may be thought of as a fuzzy filter since it decreases the dimensionality of the features while increasing their robustness. Finally, the fully connected layer receives signals from the preceding layers and delivers them to each neuron in the system. The output layer, which is generally a softmax classifier, then does the classification. As shown in Figure 4, In our case, the PIDD consists of six features as input and one as target output, which consists of two values such as 0 and 1. Input features are described as where input parameters belong to a feature set; here, the outcome variable belongs to a class label, such as 1 specifies a diabetic and 0 specifies a healthier one.*

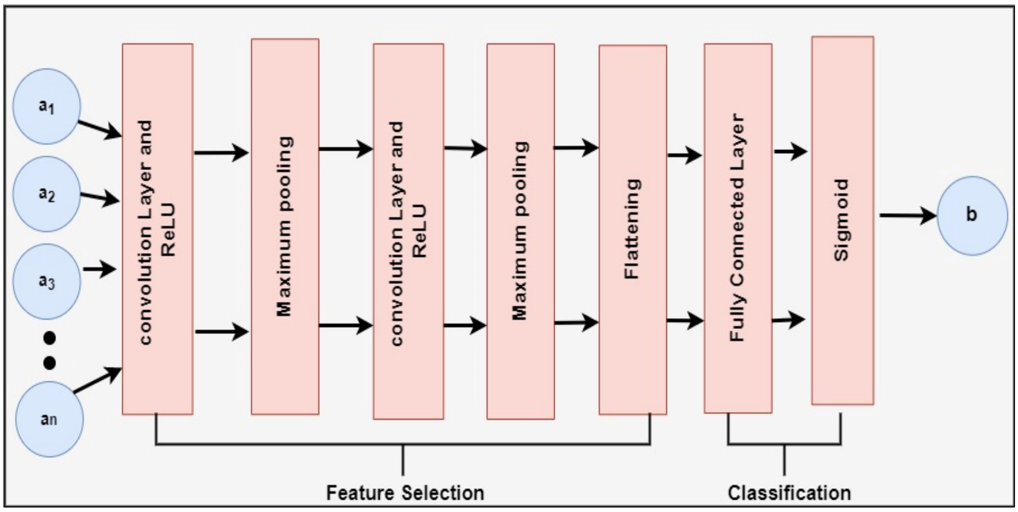


Figure 4. Architectural CNN Model for Diabetes Prediction.

In our proposed model, these input features were passed through convolutional 1D, where we applied batch normalization (BN) along with ReLU. Here, BN normalizes input features into batches, which minimizes the gradient saturation during covariate shift [45], and the ReLU activation function decreases the redundancy by allowing values that range between 0 and 1 to accelerate the speed. The complete process is mathematically explained in Equation (5) [46]

$$y(k) = f\left(\sum_{1}^{N} \widetilde{x}j(k) \times Wp(k) + bp(k)\right) \tag{5}$$

where

W_p : specifies weight
$\widetilde{x}$: Batch Normalization of input features
$f(.)$: represents the activation function ReLU

p : no of filters
b_p : bias term value ranges from 0–1

3.5.2. Architectural CNN-LSTM Model for Diabetes Prediction

Here, in this segment, we explained the working of the CNN-LSTM hybrid model by explaining the functionality of CNN and LSTM for the prediction of Type 2 diabetes mellitus. CNN and LSTM are deep learning models and are used for predictions. Here we use the CNN-LSTM [47] combination to classify Type 2 diabetes mellitus over the PIDD. Whereas CNN is used for feature engineering as it automatically selects the unseen features for model training, and LSTM is used for diabetes classification. The complete structure of CNN-LSTM is shown in Figure 5. Firstly, input features are passed through a convolutionary layer responsible for generating a feature map by striding of filters at one step. Next, we introduced non-linearity to the feature by using the ReLU function, which ranges from 0 to 1, i.e., it does not activate all neurons while simultaneously deactivating neurons whose values are less than zero. These function maps are then transferred into a batch normalization that regularizes their meaning and prevents over-fitting functions. Besides, these functions were transferred into the max-pooling layer used for the downsampling of the function diagram. Next, down-sampled features were passed through the flattening layer, which is responsible for translating these function matrices into 1D vectors, and were passed through the LSTM layers as inputs. LSTM is a particular type of RNN that uses cell state memory instead of primary neurons to manage the sequence classification. Eventually, these values were transferred into a classification layer that functions similarly to how the ANN works. Finally, it went into the sigmoid activation, responsible for the binary classification and predicted diabetes.

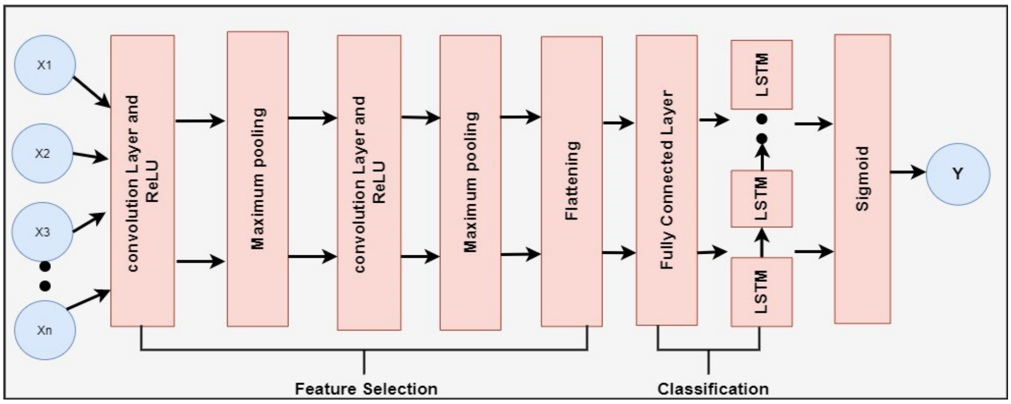


Figure 5. Architectural CNN-LSTM Model for Diabetes Prediction.

3.5.3. CNN-Bi-LSTM: A Real-Time Framework

In this study, we utilized the architecture of CNN-LSTM in a bidirectional way, which is designed for diabetes prediction in a real-time framework. Here, first, we performed the training process over the static dataset using optimized parameters by using the grid search hyperparameter optimization technique. Then, we cross-validated our model with a real-time scenario, and lastly, we performed a prediction process over the training dataset to predict diabetes. The complete working of the CNN-Bi-LSTM [48] architecture was discussed in three sections, such as training of the model using the PIDD dataset, optimizing the model with hyperparameter optimization, and lastly, prediction of diabetes through our proposed optimized CNN-Bi-LSTM model. Architectural representation is shown in Figure 6.

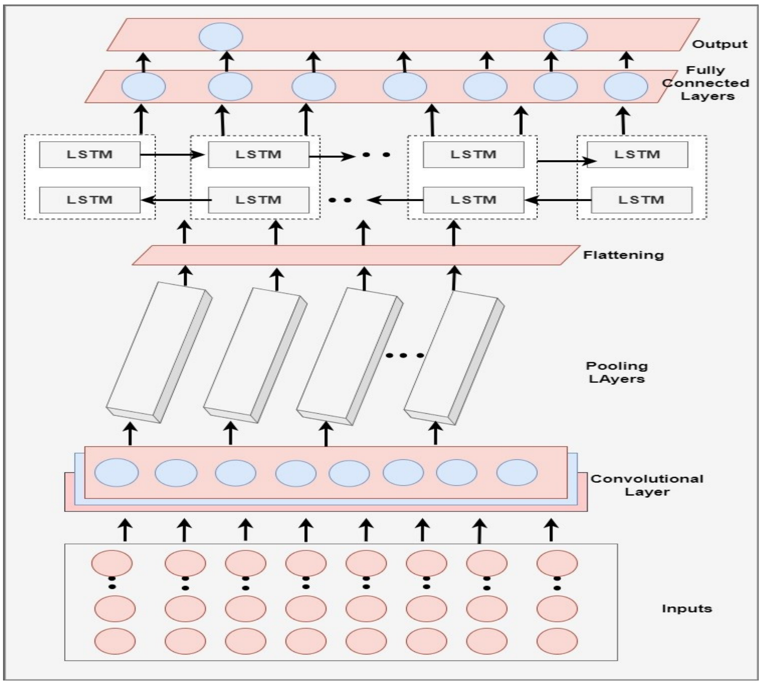


Figure 6. Architectural CNN-Bi-LSTM Model for Diabetes Prediction.

I. CNN-Bi-LSTM Training Process: The main steps are explained by the activity diagram shown in Figure 7.

- Input data: The necessary data for CNN-Bi-LSTM training must be entered;
- Preprocessing of input data: The z-score standardization approach was used to normalize input data since there was a substantial gap in input data to fully train the algorithm, as indicated in Equation (4);
- Initialization of network: Here, we initialized weights and biases for each layer of CNN-Bi-LSTM;

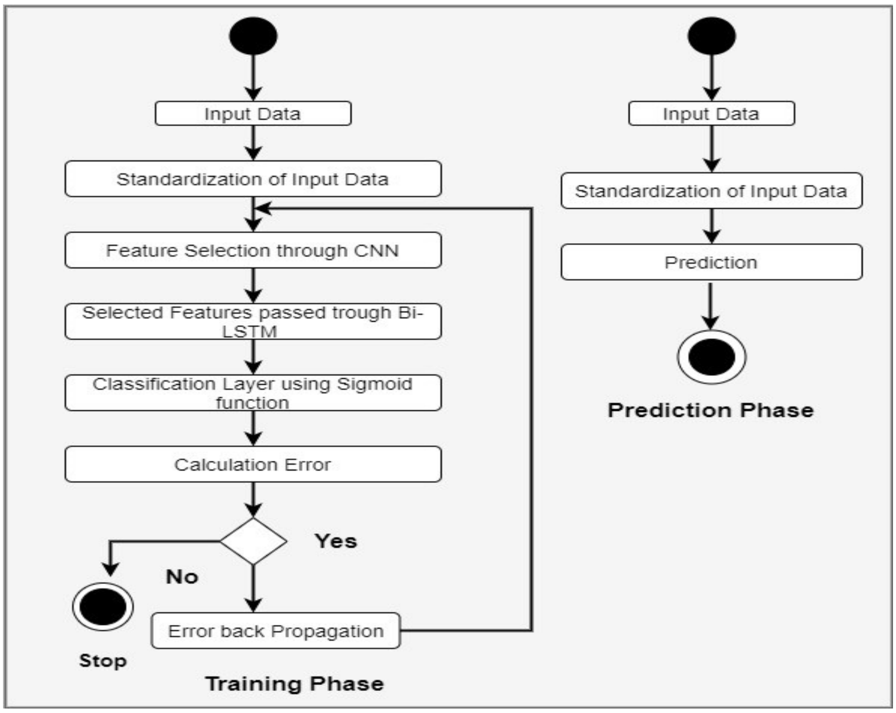

Input Data
Standardization of Input Data
Feature Selection through CNN
Selected Features passed trough Bi-LSTM
Classification Layer using Sigmoid function
Calculation Error
Yes
No
Stop
Error back Propagation
Training Phase
Input Data
Standardization of Input Data
Prediction
Prediction Phase

Figure 7. Activity diagram for CNN-Bi-LSTM Model.

- CNN Layer Calculation: Eight input features with an input shape of 6 × 1 were passed through a convolutionary layer, which is responsible for generating a feature map by striding filters of kernel size 1 at one step. We introduced non-linearity to the feature by using the ReLU function, which has a range of 0 and 1, i.e., it does not activate all neurons at the same time as it deactivates neurons whose values are less than zero. These function maps are then transferred into a batch normalization that regularises the meaning and prevents the over-fitting functions. Besides, these functions are transferred into the max-pooling layer, with pool size 1 being used for the downsampling of the function diagram. Down-sampled features are then passed through the flattening layer, which is responsible for translating these function matrices to 1D vectors.

Remark 2. *(Stride:) The stride is the number of pixels that have been moved across the input matrix. If we set the value of stride as 1, the filters shift one pixel at a time, whereas if we set the value of stride at 2, the filters move two pixels at a time. In this section, we use stride filters with kernel sizes of 1.*

- Bi-LSTM Estimation of the layer: Using Bi-LSTM [49,50], the output data of the CNN layer are determined. Bi-LSTM is a bidirectional RNN that consists of 32 hidden LSTM cells, so there are a total of 64 LSTM cells that have an additional peehole connection that prevents vanishing gradient problems and additional cell state memory that uses past and future knowledge to forecast output by using two separate hidden layers, such as forward state sequence, is represented by $\overrightarrow{h}_t$ as shown in Equation (6) [43], the backward state sequence $\overleftarrow{h}_t$ is shown in Equation (7) [43], and the output vector is represented by Equation (8)

$$\overrightarrow{h_t} = H\left(W_{ph}p_t + W_{hh}h_{t-1} + b_h\right), \tag{6}$$

$$\overleftarrow{h_t} = H\left(W_{ph}p_t + W_{hh}h_{t+1} \ + b_h\right), \tag{7}$$

where

h_t : hidden state at timestamp t
W_{ph} : weight matrix between input and hidden vector
p_t : input vector at timestamp t
W_{hh} : the weight vector between two hidden states
h_{t+1} : the hidden state vector at timestamp $t \ +1$
b_h: the bias vector for hidden state vectors

$$q_t = W_{pq}\overrightarrow{h}_t + W_{pq}\overleftarrow{h}_t + B_q, \tag{8}$$

As a result, output values q_t are derived by summing up weight matrices of input and output by performing a dot product by forwarding hidden layers $\overrightarrow{h_t}$ and backward hidden layers $\overleftarrow{h_t}$ and adding a constant bias B_q.

- Dropout: 15% dropout was applied, which is used to reduce the overfitting in neural networks by dropping some of the random nodes during the network training process;
- Output values were transferred into a sigmoid function used for binary classification, which determines whether or not the input instance is diabetic [51];
- Calculation Error: The cost function assesses how effectively the neural network is equipped by describing the difference between the provided testing sample and the expected performance. The optimizer function was used to decrease the cost function. A cross-entropy function, which comes in a variety of forms

and sizes, is commonly used in deep learning. Mathematically cost function φ is expressed as Equation (9) [35]:

$$\varphi = \frac{1}{m}\sum(a\ \log(b) + (1-a)\log(1-b)), \quad (9)$$

where

m : batch size
a : output resultant value
b : expected value

Experiments have shown that when Adam is used, the optimal point is reached fast. As a result, we employed the Adam optimizer method, which has a 1×10^{-4} learning rate.

- Evaluate if the prediction mechanism's final criterion has been achieved: Effective completion of cycles depends on two factors: weights should not exceed a certain threshold, and the estimated error rate is below a specified threshold. If at least one of the graduation standards is fulfilled, the training is finished. Otherwise, the instruction would be resumed;
- Back Propagation Error: The calculated error is spread in the opposite direction, the weight and bias of each layer are modified, and then the process goes back to stage (4) to start network training.

Hyperparameter Optimization: In this part, we presented the techniques for tuning the hyperparameter for model training in the best way possible to predict diabetes more accurately. It is a technique that helps us reduce the cost of the model by tuning hyperparameters that change the shape of the model to achieve the highest accuracy. This study applied a grid search algorithm to five deep learning models to train their parameters.

Definition 2. *(Grid Search algorithm) is often referred to as an exhaustive search across hyperparameters using permutation and combination. It returns the settings with maximum precision and accuracy during the validation process. We discovered that each of the five models performs well after analyzing them using a 10-fold cross-validation approach rather than dividing the dataset for training and testing. We also reported that the CNN-Bi-LSTM model outperformed four deep learning models in terms of accuracy, sensitivity, and specificity.*

II. CNN-Bi-LSTM Prediction Process: The prediction of the CNN-Bi-LSTM model is explained by the activity diagram shown in Figure 7;

- Input Data: The essential data for CNN-Bi-LSTM predictions must be entered;
- Preprocessing of input data: This is performed through standardization through Equation (4);
- Process of prediction: Standardized data are fed into the CNN-Bi-LSTM, which is then used to calculate the output value;
- Output Result: Recovered results are provided to complete the prediction process. The model summary of CNN-Bi-LSTM is represented in Table 6.

Table 6. Layered structure of CNN-Bi-LSTM.

S.no	Layers (Type)	Filters	Output Shape	Param
1.	Sequential model with input_shape (None, 1, 6)	-	-	-

2.	CONV1D	F = 64,
		Kernel = [1 1]
			(None, None, 1, 64)	576

3.	BatchNormalization	-	(None, None, 1, 64)	256
4.	MaxPooling	Filter_size = [1 1]	(None, None,1, 64)	0

Table 6. *Cont.*

S.no	Layers (Type)	Filters	Output Shape	Param
5.	Flattening	-	(None, None, 64)	0
6.	Bi-LSTM	-	(None, 128)	66,048
7.	Dropout	50%	(None, 128)	0
8.	Dense		(None, 1)	129
9.	Classification		-	-

3.6. Prototype Implementation and Testing of Proposed Model Using Real-Time Database

This section explains the prototype implementation of a real-time system that uses deep learning models to predict diabetes and provide assistance to users and medical experts. Because the focus of this work is on integrating a diabetes prediction model and leveraging user acquired data to make predictions for enabling lifestyle management, the proposed system is based on a cloud-based brokering framework that integrates multiple health cloud platforms and devices. The devices used in this work are the Samsung Note 8 smartphone an IOT health device gateway that integrates SPO2 sensors such as MAX30100, which is an integrated sensor used to sense SPO2, as well as BPM; an IOT sensor as Node 32; BP sensor; and Pulse sensor. Additionally, it includes AWS serverless API gateway, AWS storage, and AWS server. The working of the proposed framework is shown in Figure 8. Data are collected from medical device smartphones through AWS serverless gateways and transmitted to AWS cloud servers.

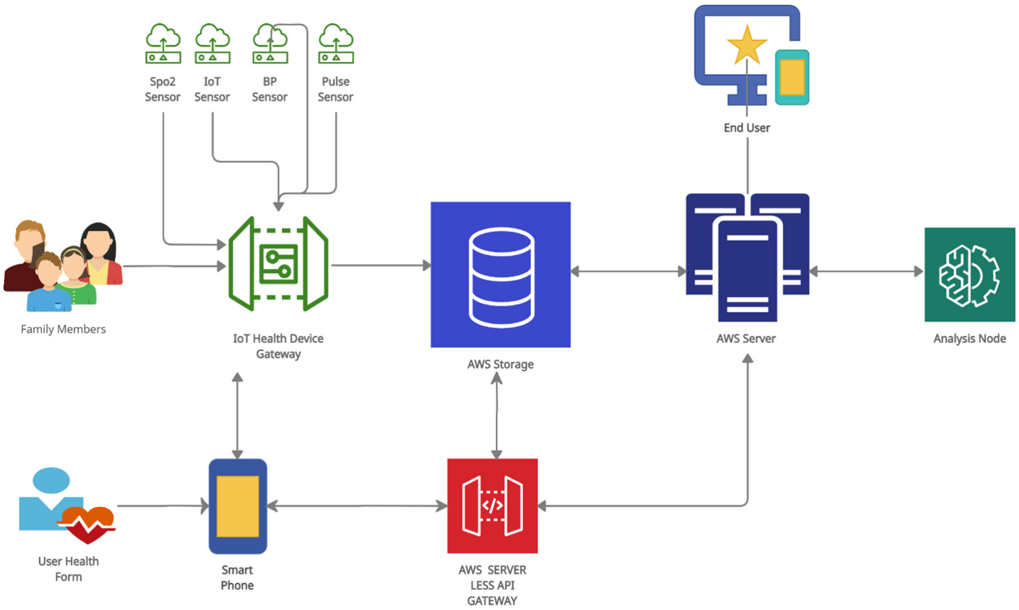


Figure 8. Data integration using real-time proposed framework.

3.6.1. Sensing of Real-Time Data through a Proposed Framework

This node contains gadgets that capture user data both actively and passively. The user first enters their static profile information. The IOT devices offer active collection, in which the user monitors their levels of glucose, blood pressure (BP), and SPO2. The smartwatch and smartphone allow passive collection; they automatically track their steps, caloric outflow, activity type, and duration. Additionally, one can upload its physical reports through smartphones. We utilized an ESP32 based gateway that has built-in Bluetooth and Wi-Fi. Our gateway takes data from all three sensors (Spo2, heartbeat, and blood pressure) and sends data to the AWS cloud using Wi-Fi, and receives data from our handheld device via Bluetooth. Our gateway provides independence to elderly users who do not use mobile phones. Signing into their respective mobile apps allows users to access this information. Figure 9 depicts the data gathered from each of the devices for a single volunteer individual in this study. The various device sensors on Android smartphones and smartwatches are used to automatically collect and aggregate activity data and present the fitness history to consumers via the mobile application. The Google vendor API monitors movements, activities, and heart rates, as well as calculating the calories burnt and steps taken by using gyroscope and accelerometer sensors on smartphones and smartwatches, as well as an added heart rate sensor on smartwatches.

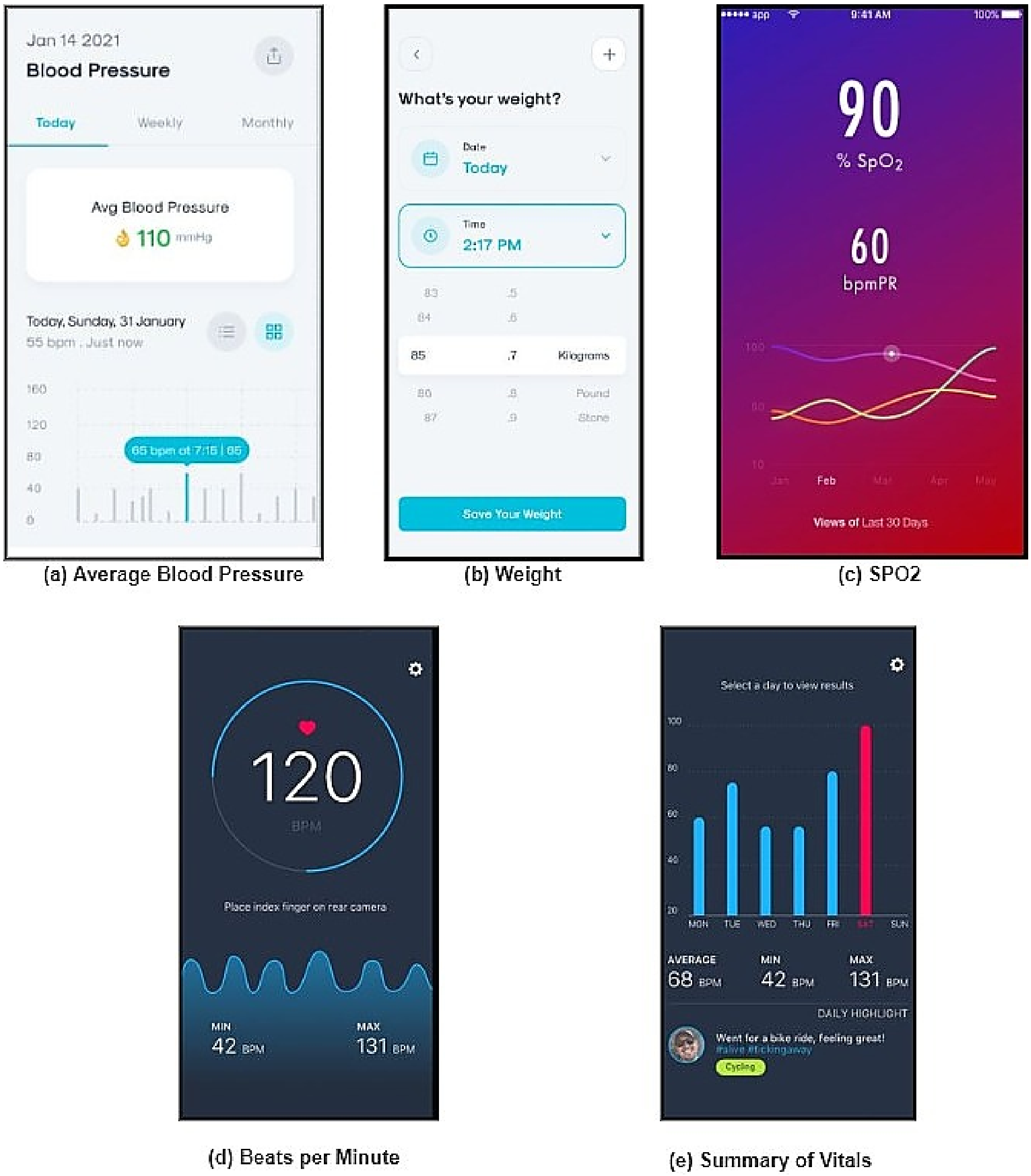


Figure 9. Visualization of real-time data through mobile application.

In the proposed framework AWS server retrieves patient information from the mobile app as well as other mobile applications, e.g., Google Fit, every hour by connecting to the AWS cloud using Server Less API gateway (AWS), and it requires only one-time credential authentication. For authentication purposes, we required the OAuth 2.0 protocol to connect with the mobile application API [12]. It is standard protocol for connection with the various devices online, desktop, and mobile application authorization. This is the security protocol that is utilized for both Google services and API data interaction. A one-time authentication strategy was used to reduce the intrusiveness of reminding medical professionals or caregivers to authenticate themselves with the online application every time they wish to monitor their data. When registering for the first time, the user connects to their mobile app account over the API to approve access to vendor cloud API privileges for the framework. The user was provided this one-time, two-step method through the mobile app, and they do not need to repeat it. If required, the user may include optional parameters such as skin fold thickness, number of pregnancies, and pre-existing medical concerns into the online application. The server obtains and stores the unique user authentication token for each vendor API. In order to keep the token continually updated, the server connects to the cloud API of the respective vendor and automatically renews the token before its expiry date. This is accomplished by configuring a server-side automated method to update tokens depending on the expiry settings of each vendor API token. AWS server then creates additional matrices depending on the other data that have been acquired, such as age, gender, height, and weight, among other things, that have been previously gathered. User data are aggregated on a daily basis by the server and ordered according to the time stamp of data collection, even if it occurs on the same day. This allows the server to sort user data chronologically. Additionally, metrics such as total daily energy expenditure (TDEE), basal metabolic rate (BMR), and body mass index (BMI) are generated to offer medical practitioners a more comprehensive monitoring capability. Maximum, average, and lowest heart rate values are also computed for the same parameter. In order to make it easier to adopt different prediction and monitoring strategies for accessible diabetic user lifestyle management, it is necessary to increase the number of parameters that can be extracted from the collected data.

3.6.2. Prediction of Diabetes Using Proposed Model Using Real-Time Dataset

We tested our optimized proposed model using a real-time dataset, where data were imported from the AWS server in the form of chunks, each of which was tested with our optimized CNN-Bi -LSTM model, and the results are updated after each instance. The entire method is demonstrated in Figure 10. referred to through Algorithm 1.

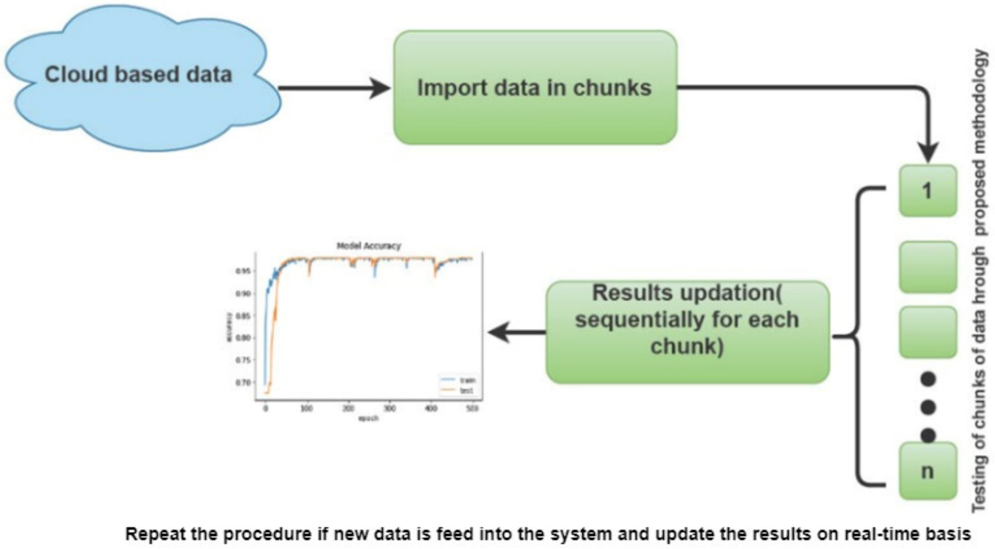

Cloud based data
Import data in chunks
1
n
Results updation(sequentially for each chunk)
Model Accuracy
accuracy
0.95
0.90
0.85
0.80
0.75
0.70
0
100
200
300
400
500
epoch
train
test
Testing of chunks of data hrough proposed methodology
Repeat the procedure if new data is feed into the system and update the results on real-time basis

Figure 10. Real-time data processing using optimized proposed model CNN-Bi-LSTM.

Algorithm 1: Algorithm to fetch real-time data

```
Require: $C_{size}\ , D_{size}, n, i, R_i, R_c$
initialization: $i = 0, R_i = 0$
Ensure: $n = \frac{D_{size}}{C_{size}}, n \geq 0$
  if $D_{size,} \geq C_{size}$ then
    while $n \geq 1$ do
      Step1: Import chunk $i$ of size $C_{size}$
      Step2: Perform testing using proposed model.
      Step3: Obtained result $R_c$ of chunk $i$.
      Step4: Update previous results $R_i$: $R_i \leftarrow R_i \diamond R_c$
        $i \leftarrow i + 1$
        $n \leftarrow n - 1$
    end while
  end if
$D_{size,} \leftarrow D_{size} - C_{size} = 0$
```

Notations used in Algorithm 1:
D_{size} : Size of database
C_{size} : Size of chunks
n : Number of chunks
R_i : Initial results of model
R_c: Results of the model at an instance i
$\diamond$: used for updating of previous results

In the above algorithm complete set of steps are as follows:

- Step1: Real-time dataset of size D_{size} is imported from the cloud server in the form of n chunks such as $n = \frac{D_{size,}}{C_{size}}$;
- Step2: We check if the size of the dataset is greater than or equal to chunk size. Then, each chunk i is tested over our optimized proposed model;
- Step3: For every instance, previous results R_i were updated with a new result R_c;
- Step4: Lastly, we updated the size of the dataset $D_{size,} \leftarrow D_{size} - C_{size}$. If it was still greater, then the chunk size algorithm starts from step1 again.

4. Experimental Results and Its Analysis in Real-Time Environment

In this section, the experiments were evaluated and analyzed over a real-time dataset PIDD [14] with a python environment with six essential critical parameters such as glucose, insulin, pregnancy, blood pressure, age, and BMI. In order to evaluate the effectiveness of the proposed framework, the results were compared with the similar and recent existing methods such as CNN [50], Bi-LSTM [52], DNN [53], and a combination of CNN-LSTM [54], and CNN-Bi-LSTM for the classification of Type 2 diabetes over PIDD [14]. In this section, five metrics were used to measure the overall success of our proposed model: accuracy (A), as seen in Equation (10) [25]; Recall (R), as in Equation (11) [55]; sensitivity (SN), as in Equation (12) [41]; and specificity (SP), as in Equation (13) [25], where the sensitivity of a model determines its capacity to classify patients who currently have a disease correctly; whereas the specificity of a model determines its capacity to classify disease-free patients correctly. The precision of the model defines the number of patients accurately described by the model. However, the ratio of the number of patients properly classified by the model is called accuracy. Formulas are stated as follows:

$$A = \frac{TP + TN}{Total\ no\ of\ samples} \tag{10}$$

$$R = \frac{TP}{TP + FN} \tag{11}$$

$$SN = \frac{TP}{TP + FN} \tag{12}$$

$$SP = \frac{TN}{TN + FP} \tag{13}$$

where true positive (*TP*) is defined as the number of positive patients who were scored as positive. True negative (*TN*) is the number of negative patients predicted to be negative. False-positive (*FP*) is the percentage of patients who are classified as positive but are actually negative. Negative (*FN*) is used to calculate the percentage of positive patients negatively observed.

4.1. Real-Time Qualitative Analysis

For real-time qualitative evaluation, we used grid search algorithms to find the top three mean test scores, such as 90.38, 85.58, and 79.4, that helped to achieve the highest accuracy, as seen in Table 7. By using the highest test score of 90.38, we were able to achieve the best hyperparameters used for training models, such as learning rate of 0.01, epochs as 250, batch-size as 32, kernel-size as 1, hidden-units as 32, regularisation dropout as 0.05, and optimizer as (referenced in Table 8). Where the following parameters are stated as:

- Learning Rate: states how many weights are modified in the loss gradient model;
- Batch Size: Specified are run through the model at any particular moment;
- Epochs: Defines the number of times the machine learning model is performed on the same dataset;
- Dropout: Is a method of regularization to reduce the issue of overfitting by dropping some of the random nodes during the training phase and improving generalization error in NN.

Table 7. Optimized Hyper parameters.

Learning Rate	Batch Size	Hidden Units	Epochs	Mean Test Score
0.01	32	32	250	90.38
0.03	64	64	300	85.58
0.05	128	128	500	79.49
0.09	64	64	300	89.0

Table 8. Optimized Parameters used for training.

S.no	Parameters	Values
1.	Regularization (dropouts)	0.05
2.	Loss-function	Binary-crossentropy
3.	Optimizer	Adam
4.	Metrics	accuracy
5.	Learning Rate	0.01
6.	Batch-size	32
7.	Epochs	250

Remark 3. *Adam is a stochastic gradient optimizer for the training of deep learning models, and it is a combination of the best features of AdaGrad and RmsProp, so it can solve problems with low gradients or a lot of background noise.*

4.2. Real-Time Quantitative Analysis

For quantitative evaluation, we conducted a comparative analysis of deep learning models such as CNN [56], DNN [54], Bi-LSTM [55], CNN-LSTM [50], and CNN-Bi-LSTM, which are trained over a static PIDD dataset by splitting it into two portions, 70% for the training dataset and 30% for the testing dataset, by using hyperparameters such as kernel size as 1; the number of filters as 64; batch-size as 32; regularization dropout as 0.05; optimizer as; maximum pool size as 1; loss-method as binary cross-entropy; epsilon as 1−08; decay as 0.0; epochs as 250 with 32 hidden units and found CNN-LSTM [52] with 90%, CNN [51,55,57,58] has an accuracy of 82%, Bi-LSTM [1] with 85%, DNN [1] with 87%, and, ultimately, CNN-Bi-LSTM outperforms and achieves an accuracy of 88.37% as shown in Table 9. Although the accuracy of all models is reasonable, they suffer from under-fitting and over-fitting problems, and they emerge when models have learned less than or more than 250 epochs. The over-fitting problem model tends to memorize data and cannot generalize new data, while the under-fitting model does poorly in testing but can generalize new data.

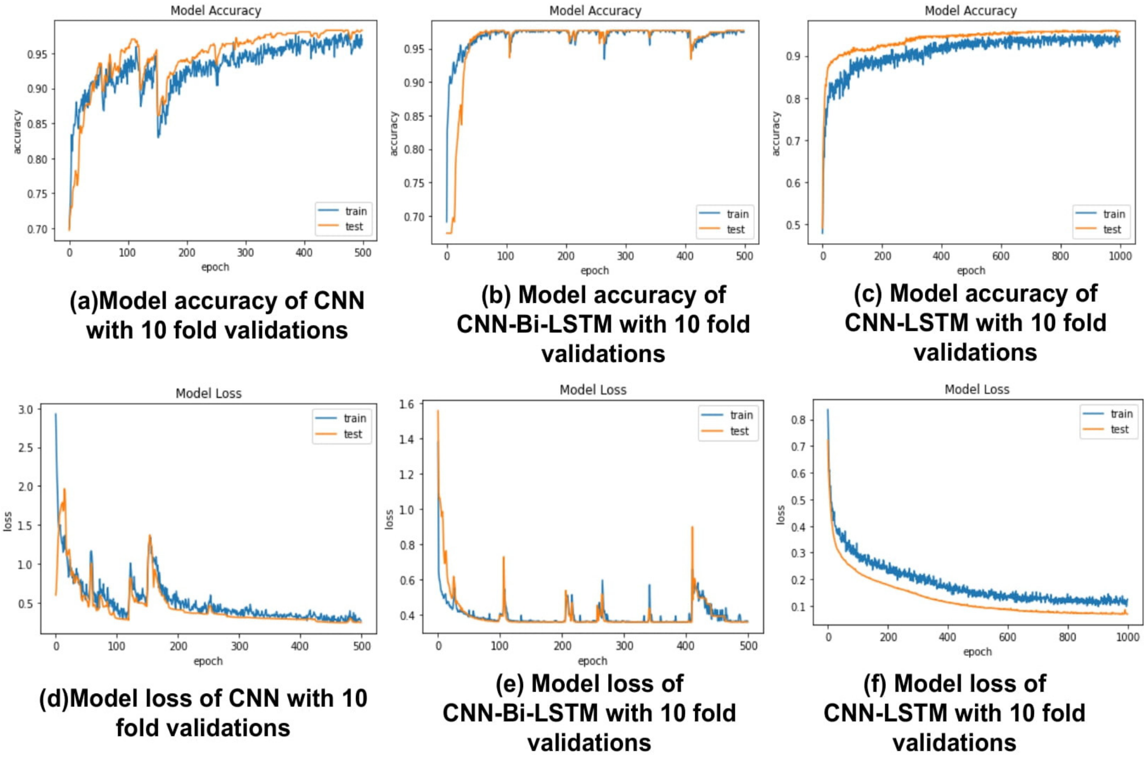
Table 9. Comparative study of various deep learning algorithms for classification of siabetes by dividing dataset into testing and training [59,60].

S.no	Accuracy Measures	CNN-LSTM [26]	CNN [27]	Bi-LSTM [1]	Dense-NN [1]	Proposed Method
1.	Accuracy	90	82	85	87	88.37
2.	Precision	0.84	0.72	0.82	0.84	0.85
3.	Recall	0.79	0.77	0.70	0.75	0.79
4.	F1-score	0.83	0.74	0.75	0.79	0.81
5.	Cohens Kappa Score	0.76	0.61	0.64	0.70	0.73
6.	ROC-Accuracy	0.89	0.87	0.87	0.91	0.90

To remove the over-fitting and under-fitting problems, we trained our models at 250 epochs and over-optimized parameters using 10-fold cross-validation, and the accuracy of each model was increased, as shown in Table 10. Besides, the accuracy of CNN-LSTM was increased to 93%, CNN was increased to 96%, BI-LSTM was also increased to 95%, and DNN was increased to 90%. However, our CNN-Bi-LSTM model outperformed compared to other models and achieved the highest accuracy of 98.85%, as shown in Figure 11, with a sensitivity of 97% and specificity of 98%. Thus, after the discussion, we can infer that CNN-Bi-LSTM is better relative to other deep learning models in terms of accuracy, sensitivity, specificity, precision, and recall. Therefore, we utilized the CNN-Bi-LSTM model in a real-time setting to classify diabetic patients more accurately as well as to monitor their vitals on a real-time basis. Additionally, we have validated our proposed model with different scenarios. First, the proposed model is validated without imputations where the values for precision~0.83, recall~0.88, and F1-score~0.85 for the outcome 0. However, for the outcome 1 values of precision~0.83, recall~0.88, F1-score~0.85 and accuracy~80% as shown in Table 11. Secondly, the proposed model is validated without removing outliers, where the values for precision~0.81, recall~0.87, and F1-score~0.83 for the outcome 0. However, for the outcome 1 values of precision~0.72, recall~0.61, F1-score~0.70 and accuracy~79% as shown in Table 12. Hence, missing values and outliers affect the performance of the model. Therefore, it is essential to preprocess data before training the proposed model.

Table 10. A Comparative Study of various Deep Learning Algorithms for the Classification of Diabetes using 10-fold Cross-Validation.

S.no	Accuracy Measures	CNN-LSTM [26]	CNN [27]	Bi-LSTM [1]	Dense-NN [1]	Proposed Method
1.	Accuracy	93	96	95	90	98.85
2.	Precision	0.93	0.97	0.97	0.92	0.98
3.	Recall	0.93	0.96	0.86	0.87	0.94
4.	F1-score	0.92	0.96	0.92	0.90	0.96
5.	Cohens Kappa Score	0.95	0.95	0.89	0.89	0.94
6.	ROC-Accuracy	0.96	0.96	0.99	0.95	0.96



(a)Model accuracy of CNN with 10 fold validations

(b) Model accuracy of CNN-Bi-LSTM with 10 fold validations

(c) Model accuracy of CNN-LSTM with 10 fold validations

(d)Model loss of CNN with 10 fold validations

(e) Model loss of CNN-Bi-LSTM with 10 fold validations

(f) Model loss of CNN-LSTM with 10 fold validations

Figure 11. (**a**) Model accuracy of CNN with 10-fold cross-validation. (**b**) Model accuracy of optimized proposed model (CNN-Bi-LSTM). (**c**) Model accuracy of CNN-LSTM model. (**d**) Model loss of CNN. (**e**) Model loss of proposed model. (**f**) Model loss of CNN-LSTM.

Table 11. Validation of proposed model with K-fold cross-validation without filling missing values.

	Precision	**Recall**	**F1-Score**
0	0.83	0.88	0.85
1	0.75	0.65	0.70
Accuracy			80%

Table 12. Validation of proposed model with K-fold cross-validation without removing outlier values.

	Precision	**Recall**	**F1-Score**
0	0.81	0.87	0.83
1	0.72	0.61	0.70
Accuracy			79%

The performance of different models can be visualized through a graph, as shown in Figure 12. When compared to other current approaches, it is seen that a testing dataset is adequately fitted to the training model in CNN-Bi-LSTM with very few distortions. Hence, our suggested framework is more accurate and capable of properly classifying diabetic patients. Additionally, we have tested various CNN-Bi-LSTM findings over-optimized hyperparameters with different mean test scores (reference from Table 7), and the rest of the results of our proposed methodology with different mean test scores are presented in Table 13. It is clearly seen that our model outperformed, with the highest mean test score at 90 (Reference from Table 10). Lastly, comparisons were made between various state-of-the-art algorithms and the proposed model in terms of accuracy, as shown in Table 14, and it was found that CNN-Bi-LSTM outperformed in terms of accuracy (referenced from Table 10) [61,62]. After all these conversations, we may infer that the CNN-Bi-LSTM model is more accurate in identifying diabetes patients than the other four models as well as state-of-the-art algorithms. As a result of the fact that CNN-Bi-LSTM combines the power of both CNN and Bi-LSTM, where CNN is used for feature extraction, and Bi-LSTM has additional peehole connections that prevent vanishing gradient problems as well as additional cell state memory that uses past and future knowledge to forecast the output by using two separate hidden layers.

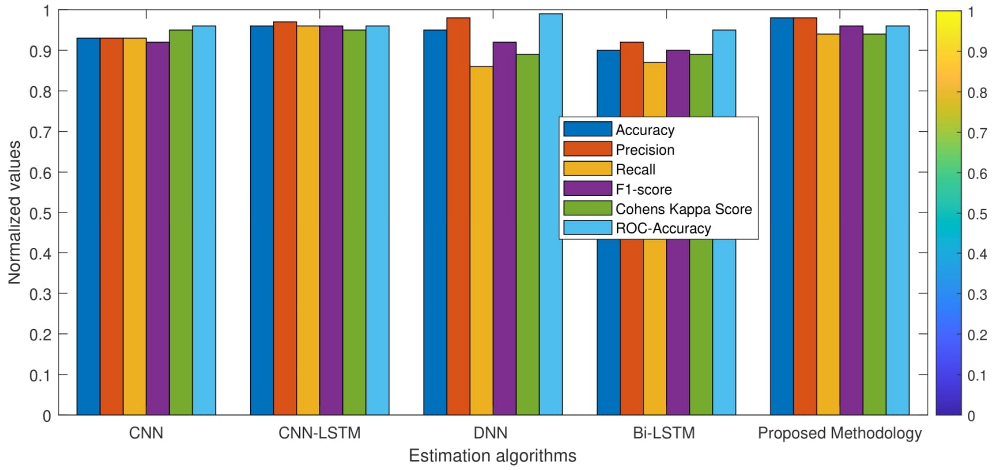


Figure 12. Performance analysis of optimized proposed methodology with respect to existing deep learning algorithms.

Table 13. Results of the proposed architecture with different Mean Test Scores using Grid Search Optimization.

S.no	Accuracy Measures	Mean-Test Score 85.58%	Mean-Test Score 79.49%	Mean-Test Score 89%
1.	Accuracy	91	78	67
2.	Precision	0.87	0.67	0
3.	Recall	0.82	0.67	0
4.	F1-Score	0.85	0.67	0
5.	Cohens Kappa Score	0.78	0.51	0
6.	ROC-Accuracy	0.92	0.83	0.28
7.	Sensitivity	91%	84%	2
8.	Specificity	0.87%	67%	0

Table 14. Comparison of our model with other state-of-the-art algorithms in terms of accuracy mentioned in related work.

S.no	Author	Dataset	Validation Technique	Algorithm Used	Accuracy
1.	Ashiquzzaman (2017) et al. [3]	PIMA Dataset	0.1 split validation	Deep learning architecture	88.41%
2.	Kumari et al. [15]	PIMA Dataset	Divide hyperplane	SVM	77%
3.	K.Kannadasan (2019) et al. [31]	PIMA Dataset	70% Training, 30% Testing	DNN with auto-encoders	86%
4.	Massaro (2019) et al. [36]	PIMA Dataset	The training and validation sets are split in an 80/20 ratio	LSTM	86%
5.	Patil et al. [41]	PIMA Dataset	K-fold cross-validation	PCA, K-Means Algorithm	73%
6.	Gill and Mittal (2016) et al. [54]	PIMA Dataset	70% Training, 30% Testing	SVM and NN	96.09%
7.	Yuvaraj andSriPreethaa (2019) et al. [57]	PIMA Dataset	70% Training, 30% Testing	RF	94%

Table 14. *Cont.*

S.no	Author	Dataset	Validation Technique	Algorithm Used	Accuracy
8.	Naz and Ahuja (2017) et al. [63]	PIMA Dataset	The training and validation sets are split in an 80/20 ratio.	ANN	90.34%
9.	Kemal Polat et al. [64]	PIMA Dataset	K-fold cross-validation	neuro-fuzzy-inference system (ANFIS)	89.47%
10.	Haritha et al. [65]	PIMA Dataset	70% Training, 30% Testing	Firefly and Cuckoo Search Algorithms	80%
11.	Mohammad et al. [66]	PIMA Dataset	70% Training, 30% Testing	NN with Genetic Algorithm	86.78%
12.	Chen et al. [67]	PIMA Dataset	10 fold cross validation	K-means and DT	91.23%
13.	Proposed Method	PIMA Dataset	K-fold cross-validation	CNN-BI-LSTM	98.85%

In order to check the performance of deep learning models, we additionally performed a statistical Student t-test between CNN-LSTM and CNN_Bi-LSTM. We used the variance estimate by checking the dependency between the dataset and computing the *p*-value. In the null hypothesis, we assumed that there was no statistical difference between the performances of the models. However, the alternative hypothesis we have considered is that there is a potential difference between the performance of models. If the *p*-value is less than the significant value, then we rejected the null hypothesis and assumed that there was no significant difference between the performance of deep learning models. If the *p*-value was higher, then we rejected the hypothesis and considered the alternative hypothesis. In order to calculate the *p*-value, we calculated the mean of the difference between the results of two classifiers at every iteration step as we are using K-fold cross-validation by using the formula: $\overline{d} = \frac{1}{n}\sum_1^n d_i$. Then we calculated the variance of the difference, through $\overline{\sigma}^2 = \frac{\sum_1^n \left(d_i - \overline{d}\right)^2}{n-1}$ where n is the total number of data points. Then we computed the data points used for training, i.e., n_1 and data points for testing n_2. Then we computed the mod of variance, $\overline{\sigma}^2{}_{Mod} = \left(\frac{1}{n_1} + \frac{n_1}{n_2}\right)\overline{\sigma}^2$ and finally, we calculate the time statics through $\frac{\overline{d}}{\overline{\sigma}_{Mod}}$. The calculation of the *p*-value is shown in Figure 13, where it can be easily seen that the *p*-value is approximately 1.84%, and the significance value is 5%. As the *p*-value is less than the significance value, we rejected the null hypothesis and assume that the performance of the proposed model is different from others and better in terms of accuracy as compared to other models (referenced from Table 10).

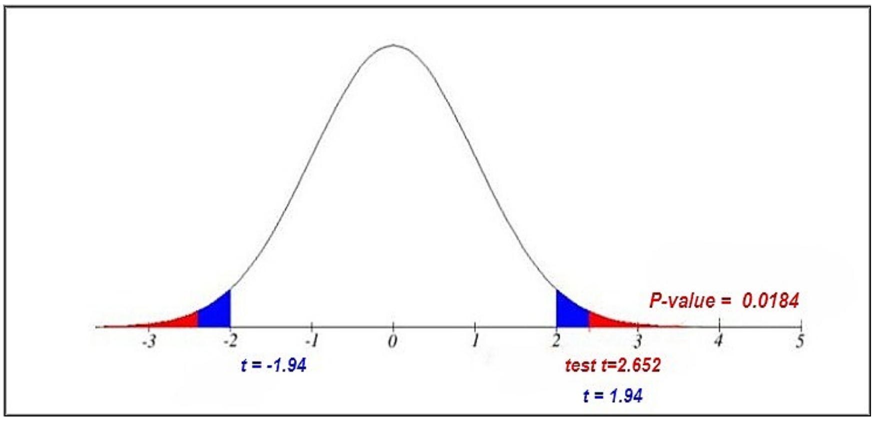


Figure 13. Paired *t*-test to evaluate the performance of the proposed model.

5. Conclusions

Diabetes is one of the prolonged diseases triggered by the unbalanced release of insulin, which becomes apparent when blood glucose levels are above average levels. In this study, five different models, such as CNN, DNN, CNN-LSTM, Bi-LSTM, and CNN-Bi-LSTM, are used to identify diabetic patients over static PIDD. The CNN-Bi-LSTM is used for the first time in this study to perform well in the multiple classification and prediction problems, which makes it unique. These models are applied to the dataset defined in two ways: training data are kept separate from testing data. Furthermore, ten-fold cross-validation methods are applied to measure the accuracy of models. Furthermore, hyperparameter optimization is achieved using a grid search algorithm to track the maximum values of the parameters. After an experimental analysis, we found that each of the five models works well when using a 10-fold cross-validation approach instead of splitting the data set for training and testing. Our analysis showed that the CNN-Bi-LSTM model outperformed all deep learning models in terms of accuracy of 98.85%, sensitivity of 97%, and specificity of 98%. Lastly, we proposed a framework that demonstrates the evaluation of our CNN-Bi-LSTM model over a real-time scenario, which helps the clinicians to keep complete information about the patients and check real-time statistics about their vitals. In the future, we can generate a dashboard to visualize the summary of the vitals to the practitioners as well as to the patients.

Author Contributions: Conceptualization, P.M.; methodology, V.C.; validation, R.S. and M.R.; formal analysis, A.G. and A.D.; writing—original draft preparation, P.M.; writing—review and editing, M.R., S.S.A. and A.S.A.; supervision, V.S., Y.A. and R.S.; funding acquisition, Y.A. All authors have read and agreed to the published version of the manuscript.

Funding: Taif University Researchers Supporting Project number (TURSP-2020/161), Taif University, Taif, Saudi Arabia.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data used in this article will be made available on request to the corresponding author.

Acknowledgments: The authors would like to thank Taif University Researchers Supporting Project number (TURSP-2020/161), Taif University, Taif, Saudi Arabia for supporting this work.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations: PIMA Indian dataset (PIDD), convolutional neural network (CNN), dense neural network (DNN), long short-term memory (LSTM), support vector machine (SVM), neuro-fuzzy-inference system (ANFIS), random forest (RF), sensitivity (SN), specificity (SP), recall (R), batch-normalization (BN), interquartile range (IQR), continuous glucose monitoring (CGM), self-monitoring blood glucose (SMBG), Information and communication technology (ICT), Bluetooth low energy (BLE).

References

1. Sneha, N.; Gangil, T. Analysis of diabetes mellitus for early prediction using optimal features selection. *J. Big Data* **2019**, *6*, 13. [CrossRef]
2. Allam, F.; Nossai, Z.; Gomma, H.; Ibrahim, I.; Abdelsalam, M. A recurrent neural network approach for predicting glucose concentration in type-1 diabetic patients. In *Engineering Applications of Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 254–259.
3. Ashiquzzaman, A.; Tushar, A.K.; Islam, M.; Shon, D.; Im, K.; Park, J.H.; Lim, D.S.; Kim, J. Reduction of overfitting in diabetes prediction using deep learning neural network. In *IT Convergence and Security 2017*; Springer: Singapore, 2018; pp. 35–43.
4. Metzger, B.E.; Coustan, D.R.; Trimble, E.R. Hyperglycemia and adverse pregnancy outcomes. *N. Engl. J. Med.* **2008**, *358*, 1991–2002.
5. Care, D. Medical care in diabetes 2018. *Diabet Care* **2018**, *41*, S105–S118.
6. Bruen, D.; Delaney, C.; Florea, L.; Diamond, D. Glucose sensing for diabetes monitoring: Recent developments. *Sensors* **2017**, *17*, 1866. [CrossRef]
7. Acciaroli, G.; Vettoretti, M.; Facchinetti, A.; Sparacino, G. Calibration of minimally invasive continuous glucose monitoring sensors: State-of-the-art and current perspectives. *Biosensors* **2018**, *8*, 24. [CrossRef]
8. Torres, I.; Baena, M.G.; Cayon, M.; Ortego-Rojo, J.; AguilarDiosdado, M. Use of sensors in the treatment and follow-up of patients with diabetes mellitus. *Sensors* **2010**, *10*, 7404–7420. [CrossRef]

9. Rodríguez-Rodríguez, I.; Zamora-Izquierdo, M.Á.; Rodríguez, J.V. Towards an ict-based platform for type 1 diabetes mellitus management. *Appl. Sci.* **2018**, *8*, 511. [CrossRef]
10. Nieminen, J.; Gomez, C.; Isomaki, M.; Savolainen, T.; Patil, B.; Shelby, Z.; Xi, M.; Oller, J. Networking solutions for connecting bluetooth low energy enabled machines to the internet of things. *IEEE Netw.* **2014**, *28*, 83–90. [CrossRef]
11. Vhaduri, S.; Prioleau, T. Adherence to personal health devices: A case study in diabetes management. In Proceedings of the 14th EAI International Conference on Pervasive Computing Technologies for Healthcare, Atlanta, GA, USA, 18–20 May 2020; ACM Press: New York, NY, USA, 2020; pp. 62–72.
12. Specifications—Samsung Galaxy Note8. The Official Samsung Galaxy Site. Available online: https://www.samsung.com/global/galaxy/galaxy-note8/specs/ (accessed on 7 July 2020).
13. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
14. Gomez, C.; Oller, J.; Paradells, J. Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology. *Sensors* **2012**, *12*, 11734–11753. [CrossRef]
15. Kumari, V.A.; Chitra, R. Classification of diabetes disease using support vector machine. *Int. J. Eng. Res. Appl.* **2013**, *3*, 1797–1801.
16. Craven, M.W.; Shavlik, J.W. Using neural networks for data mining. *Future Gener. Comput. Syst.* **1997**, *13*, 211–229. [CrossRef]
17. Radhimeenakshi, S. Classification and prediction of heart disease risk using data mining techniques of support vector machine and artificial neural networks. In Proceedings of the 2016 International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 16–18 March 2016; pp. 3107–3111.
18. Aslan, M.F.; Unlersen, M.F.; Sabanci, K.; Durdu, A. CNN-based transfer learning–BiLSTM network: A novel approach for COVID-19 infection detection. *Appl. Soft Comput.* **2021**, *98*, 106912. [CrossRef]
19. Dey, S.K.; Hossain, A.; Rahman, M.M. Implementation of a web application to predict diabetes disease: An approach using machine learning algorithm. In Proceedings of the 2018 21st International Conference of Computer and Information Technology (ICCIT), Dhaka, Bangladesh, 21–23 December 2018; pp. 1–5.
20. Srivastava, S.; Sharma, L.; Sharma, V.; Kumar, A.; Darbari, H. Prediction of diabetes using artificial neural network approach. In *Engineering Vibration, Communication and Information Processing*; Springer: Singapore, 2019; pp. 679–687.
21. Radha, P.; Srinivasan, B. Predicting diabetes by cosequencing the various data mining classification techniques. *Int. J. Innov. Sci. Eng. Technol.* **2014**, *1*, 334–339.
22. Karegowda, A.G.; Manjunath, A.S.; Jayaram, M.A. Application of genetic algorithm optimized neural network connection weights for medical diagnosis of pima Indians diabetes. *Int. J. Soft Comput.* **2011**, *2*, 15–23. [CrossRef]
23. Zolfaghari, R. Diagnosis of diabetes in female population of pima indian heritage with ensemble of bp neural network and svm. *Int. J. Comput. Eng. Manag.* **2012**, *15*, 2230–7893.
24. Sanakal, R.; Jayakumari, T. Prognosis of diabetes using data mining approach-fuzzy c means clustering and support vector machine. *Int. J. Comput. Trends Technol.* **2014**, *11*, 94–98. [CrossRef]
25. Zhang, Y. Support vector machine classification algorithm and its application. In Proceedings of the International Conference on Information Computing and Applications, Chengdu, China, 14–16 September 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 179–186.
26. Karatsiolis, S.; Schizas, C.N. Region based support vector machine algorithm for medical diagnosis on pima indian diabetes dataset. In Proceedings of the 12th International Conference on Bioinformatics & Bioengineering (BIBE), Larnaca, Cyprus, 11–13 November 2012; pp. 139–1442.
27. Jarullah, A.; Asma, A. Decision tree discovery for the diagnosis of type II diabetes. In Proceedings of the 2011 International Conference on Innovations in Information Technology, Abu Dhabi, United Arab Emirates, 25–27 April 2011; pp. 303–307.
28. Zou, Q.; Qu, K.; Luo, Y.; Yin, D.; Ju, Y.; Tang, H. Predicting diabetes mellitus with machine learning techniques. *Front. Genet.* **2018**, *9*, 515. [CrossRef]
29. Saji, S.A.; Balachandran, K. Performance analysis of training algorithms of multilayer perceptrons in diabetes prediction. In Proceedings of the International Conference on Advances in Computer Engineering and Applications, Ghaziabad, India, 19–20 March 2015; pp. 201–206.
30. Jahangir, M.; Afzal, H.; Ahmed, M.; Khurshid, K.; Nawaz, R. An expert system for diabetes prediction using auto tuned multi-layer perceptron. In Proceedings of the Intelligent Systems Conference (IntelliSys), London, UK, 7–8 September 2017; pp. 722–728.
31. Kannadasan, K.; Edla, D.R.; Kuppili, V. Type 2 diabetes data classification using stacked autoencoders in deep neural networks. *Clin. Epidemiol. Glob. Health* **2019**, *7*, 530–535. [CrossRef]
32. Apoorva, S.; Aditya, S.K.; Snigdha, P.; Darshini, P.; Sanjay, H.A. Prediction of diabetes mellitus type-2 using machine learning. In Proceedings of the International Conference on Computational Vision and Bio Inspired Computing, Coimbatore, India, 19–20 November 2019; Springer: Cham, Germany, 2019; pp. 364–370.
33. Kamble, A.K.; Manza, R.R.; Rajput, Y.M. Review on diagnosis of diabetes in pima indians. *Int. J. Comput. Appl.* **2016**, *975*, 8887.
34. Kotsiantis, S.B. Decision trees: A recent overview. *Artif. Intell. Rev.* **2013**, *39*, 261–283. [CrossRef]
35. Swapna, G.; Vinayakumar, R.; Soman, K. Diabetes detection using deep learning algorithms. *ICT Express* **2018**, *4*, 243–246.
36. Yamashita, R.; Nishio, M.; Do, R.K.G.; Togashi, K. Convolutional neural networks: An overview and application in radiology. *Insights Imaging* **2018**, *9*, 611–629. [CrossRef]

37. Massaro, A.; Maritati, V.; Giannone, D.; Convertini, D.; Galiano, A. LSTM DSS automatism and dataset optimization for diabetes prediction. *Appl. Sci.* **2019**, *9*, 3532. [CrossRef]
38. Taspinar, Y.S.; Cinar, I.; Koklu, M. Classification by a stacking model using CNN features for COVID-19 infection diagnosis. *J. X-ray Sci. Technol.* **2021**, *30*, 73–88. [CrossRef]
39. Rahman, M.; Islam, D.; Mukti, R.J.; Saha, I. A deep learning approach based on convolutional lstm for detecting diabetes. *Comput. Biol. Chem.* **2020**, *88*, 107329. [CrossRef]
40. Rahman, M.M.; Roy, C.K.; Kula, R.G. Predicting usefulness of code review comments using textual features and developer experience. In Proceedings of the 2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR), Buenos Aires, Argentina, 20–21 May 2017; pp. 215–226.
41. Shetty, D.; Rit, K.; Shaikh, S.; Patil, N. March. Diabetes disease prediction using data mining. In Proceedings of the 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), Coimbatore, India, 17–18 March 2017; pp. 1–5.
42. Liu, H.; Setiono, R. Chi2: Feature selection and discretization of numeric attributes. In Proceedings of the 7th IEEE International Conference on Tools with Artificial Intelligence, Herndon, VA, USA, 5–8 November 1995; IEEE: Piscataway, NJ, USA, 1995; pp. 388–391.
43. Geurts, P.; Ernst, D.; Wehenkel, L. Extremely randomized trees. *Mach. Learn.* **2006**, *63*, 3–42. [CrossRef]
44. Alghamdi, M.; Al-Mallah, M.; Keteyian, S.; Brawner, C.; Ehrman, J.; Sakr, S. Predicting diabetes mellitus using SMOTE and ensemble machine learning approach: The Henry Ford Exercise Testing (FIT) project. *PLoS ONE* **2017**, *12*, e017980. [CrossRef]
45. Hemanth, D.J.; Deperlioglu, O.; Kose, U. An enhanced diabetic retinopathy detection and classification approach using deep convolutional neural network. *Neural Comput. Appl.* **2020**, *32*, 707–721. [CrossRef]
46. Budreviciute, A.; Damiati, S.; Sabir, D.K.; Onder, K.; Schuller-Goetzburg, P.; Plakys, G.; Katileviciute, A.; Khoja, S.; Kodzius, R. Management and prevention strategies for non-communicable diseases (NCDs) and their risk factors. *Front. Public Health* **2020**, 788. [CrossRef]
47. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning, PMLR, Lille, France, 1 June 2015; pp. 448–456.
48. Livieris, I.E.; Pintelas, E.; Pintelas, P. A cnn–lstm model for gold price time-series forecasting. *Neural Comput. Appl.* **2020**, *32*, 17351–17360. [CrossRef]
49. Sun, Q.; Jankovic, M.V.; Bally, L.; Mougiakakou, S.G. Predicting blood glucose with an lstm and bi-lstm based deep neural network. In Proceedings of the 2018 14th Symposium on Neural Networks and Applications (NEUREL), Belgrade, Serbia, 20–21 November 2018; pp. 1–5.
50. Orabi, K.M.; Kamal, Y.M.; Rabah, T.M. Early predictive system for diabetes mellitus disease. In Proceedings of the Industrial Conference on Data Mining, New York, NY, USA, 13–17 July 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 420–427.
51. Rahman, M.; Siddiqui, F.H. An optimized abstractive text summarization model using peephole convolutional LSTM. *Symmetry* **2019**, *11*, 1290. [CrossRef]
52. Singh, T.; Vishwakarma, D.K. A deeply coupled convnet for human activity recognition using dynamic and rgb images. *Neural Comput. Appl.* **2021**, *33*, 469–485. [CrossRef]
53. Rathor, S.; Agrawal, S. A robust model for domain recognition of acoustic communication using bidirectional lstm and deep neural network. *Neural Comput. Appl.* **2021**, *33*, 11223–11232. [CrossRef]
54. Tama, B.A.; Lee, S. Comments on "stacking ensemble based deep neural networks modeling for effective epileptic seizure detection". *Expert Syst. Appl.* **2021**, *184*, 115488. [CrossRef]
55. Temurtas, H.; Yumusak, N.; Temurtas, F. A comparative study on diabetes disease diagnosis using neural networks. *Expert Syst. Appl.* **2009**, *36*, 8610–8615. [CrossRef]
56. Gill, N.S.; Mittal, P. A computational hybrid model with two level classification using svm and neural network for predicting the diabetes disease. *J. Theor. Appl. Inf. Technol.* **2016**, *87*, 1–10.
57. Yuvaraj, N.; SriPreethaa, K. Diabetes prediction in healthcare systems using machine learning algorithms on hadoop cluster. *Clust. Comput.* **2019**, *22*, 1–9. [CrossRef]
58. Swapna, G.; Kp, S.; Vinayakumar, R. Automated detection of diabetes using CNN and CNN-LSTM network and heart rate signals. *Procedia Comput. Sci.* **2018**, *132*, 1253–1262.
59. Lee, K.W.; Ching, S.M.; Ramachandran, V.; Yee, A.; Hoo, F.K.; Chia, Y.C.; Wan Sulaiman, W.A.; Suppiah, S.; Mohamed, M.H.; Veettil, S.K. Prevalence and risk factors of gestational diabetes mellitus in Asia: A systematic review and meta-analysis. *BMC Pregnancy Childbirth* **2018**, *18*, 1–20. [CrossRef]
60. Christobel, Y.A.; Sivaprakasam, P. A new classwise k nearest neighbor (CKNN) method for the classification of diabetes dataset. *Int. J. Eng. Adv. Technol.* **2013**, *2*, 396–400.
61. George, G.; Lal, A.M.; Gayathri, P.; Mahendran, N. Comparative study of machine learning algorithms on prediction of diabetes mellitus disease. *J. Comput. Theor. Nanosci.* **2020**, *17*, 201–205. [CrossRef]
62. Sivanesan, R.; Dhivya, K.D.R. A review on diabetes mellitus diagnoses using classification on Pima Indian diabetes data set. *Int. J. Adv. Res. Comput. Sci. Manag. Stud.* **2017**, *5*, 12–17.
63. Naz, H.; Ahuja, S. Deep learning approach for diabetes prediction using PIMA Indian dataset. *J. Diabetes Metab. Disord.* **2020**, *19*, 391–403. [CrossRef]

64. Polat, K.; Güneş, S. An expert system approach based on principal component analysis and adaptive neuro-fuzzy inference system to diagnosis of diabetes disease. *Digit. Signal Process.* **2007**, *17*, 702–710. [CrossRef]
65. Haritha, R.; Babu, D.S.; Sammulal, P. A Hybrid Approach for Prediction of Type-1 and Type-2 Diabetes using Firefly and Cuckoo Search Algorithms. *Int. J. Appl. Eng. Res.* **2018**, *13*, 896–907.
66. Mohammad, S.; Dadgar, H.; Kaardaan, M. A Hybrid Method of Feature Selection and Neural Network with Genetic Algorithm to Predict Diabetes. *Int. J. Mechatron. Electr. Comput. Technol.* **2017**, *7*, 3397–3404.
67. Chen, W.; Chen, S.; Zhang, H.; Wu, T. A hybrid prediction model for type 2 diabetes using K-means and decision tree. In Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS, Beijing, China, 24–26 November 2017; pp. 386–390. [CrossRef]