AJMS
Asian Journal of Mathematical Sciences

RESEARCH ARTICLE

PREDICTING DIABETES USING DEEP LEARNING TECHNIQUES: A STUDY ON THE PIMA DATASET

[*] Emad Majeed Hameed, [1]Hardik Joshi

[*]Gujarat University, Ahmadabad, Gujarat, India & Middle Technical University, Baghdad, Iraq, [1]Gujarat University, Ahmadabad, Gujarat, India.

Email: emadhameed@gujaratuniversity.ac.in

Received: 12-04-2024; Revised: 28-04-2024; Accepted: 08-05-2024

ABSTRACT

Diabetes is one of the key reasons of growing death rates around the world. Diabetes is a medical condition that arises from chronic issues that influence carbohydrate metabolism and raise blood glucose levels. Scientific research is needed to diagnose diabetes early for prevention and treatment due to the growing rates of the disease. Researchers have recognized the value of classification models for disease prediction developed using machine learning and deep learning techniques. This study explores the efficacy of deep learning techniques Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and Multi-Layer Perceptron (MLP)—in forecasting diabetes using the Pima dataset. Preprocessing steps encompassed normalization, handling missing values, and outlier removal. The models were trained and evaluated, yielding noteworthy performance metrics. The CNN exhibited the highest accuracy of 0.77, while achieving precision, recall, and ROC-AUC scores of 0.69, 0.67, and 0.83, respectively. The LSTM and MLP models also demonstrated competitive results, achieving accuracies of 0.75 with similar precision, recall, and ROC-AUC values around 0.64-0.67 and 0.80-0.82, respectively. These findings highlight the potential of deep learning methodologies for predictive diabetes analysis and emphasize the significance of proper preprocessing techniques in enhancing model performance.

Keywords: Diabetes Prediction, CNN, MLP, LSTM, Pima Dataset.

INTRODUCTION

The pancreas plays a crucial role in generating insulin, an essential hormone that regulates the amount of glucose within the human bloodstream.. insulin, it is a vital and significant component in balancing the metabolism in the human body [1]. In cases such as insufficient generation of the insulin hormone in the human body and inability to use the produced insulin effectively, a chronic condition occurs. This disease is called Diabetes. Diabetes represents a significant health challenge significantly affecting human health, with an increasing prevalence worldwide. Its impact spans across multiple organs, leading to dysfunction and increased mortality rates. Predominantly affecting the nervous system, kidneys, heart, eyes, limbs, and vasculature [2], diabetes emphasizes the importance of timely identification and continuous surveillance to minimize its harmful impacts. Categorized into two primary types, which often identified as type 1 and type 2 diabetes, preventive strategies aim to delay its onset or mitigate its adverse effects. The onset of Type 1 diabetes manifests due to an autoimmune response wherein the body's immune system, typically tasked with combating pathogenic entities, erroneously targets pancreatic beta cells. This misguided immune reaction results in diminished or absent insulin secretion, leading to inadequate insulin levels within the body. The only solution for this type of diabetes is to inject the required amount of insulin into the patient's body as a supplement. However, When the body either stops producing enough insulin or becomes resistant to its effects, type 2 diabetes occurs. The etiology of type 2 diabetes mellitus implicates a multifactorial interplay involving genetic predisposition and environmental influences. The correlation between excess body weight and the onset of type 2 diabetes exhibits a robust association. Frequent indications of diabetes encompass polyuria, polydipsia, polyphagia, abrupt weight reduction (typically observed in type 1 diabetes), fatigue, increased body weight (commonly associated with type 2 diabetes), delayed healing, blurred vision, pruritus, irritability, genital candidiasis, partial paralysis, muscle rigidity, and alopecia [3,4]. With appropriate diagnosis and treatment, individuals with diabetes can lead a lifestyle comparable to that of non-diabetic individuals, experiencing a level of normalcy in their daily lives. Various methods exist to diagnose diabetes, including A1c testing, random blood glucose assessment, fasting blood glucose examination, and the oral glucose tolerance test [5]. Identification of diabetes relying solely on one parameter may result in misdiagnosis and erroneous decision-making. Hence, the imperative lies in amalgamating various parameters in order to achieve an effective diagnostic approach for diabetes. Optimizing the identification and management of diabetes can be enhanced through the utilization of diverse data sets encompassing parameters like glucose levels, body mass index (BMI), familial history of diabetes, blood pressure readings, age, pregnancy status, and skin thickness measurements. This comprehensive approach facilitates a more effective diagnosis and treatment regimen for diabetes mellitus.

The timely identification and diagnosis of the ailment rely significantly on the expertise and clinical acumen of the medical practitioner. The health sector generates a great deal of data on health services, but this data is not used effectively in undetected cases. Human decisions pose significant risks in the early detection of diseases due to their reliance on healthcare specialists' subjective observations and judgments, which may lack consistent accuracy [6]. Therefore, various advanced mechanisms and software-based programs are considered necessary for automatic diagnosis and early detection of diseases with better accuracy. Today, using machine learning in the medical fields enables analyzing many advanced and complex datasets. The advancements in the area of machine learning methodologies facilitates in the processing of huge amounts of data and extraction of the underlying data model that supports decision-making [7]. Early detection of diseases may benefit from applying of machine learning, which has become increasingly common in recent years. Deep learning techniques, which are a branch of machine learning, have been the focus of numerous research studies. In this work, the prediction of diabetes is performed using different deep learning techniques; long-short-term memory networks (LSTM), Deep Multilayer Perceptron (MLP), and Convolutional Neural Network (CNN). The results may be promising.

The following parts cover the rest of the contents of this work.:
The studies in the literature are examined in section 2. Section 3 of the paper introduces both the material utilized and the suggested methodology. In section 4, experiments and results obtained for diabetes detection are discussed. In section 5, the findings of the research are presented.

RELATED WORK

Numerous investigations within the scientific literature explore the utilization of machine learning and artificial intelligence methodologies for the automated identification, diagnosis, and autonomous management of diabetes. Bala, M.K., et al. [8] used a Deep Neural Network (DNN) classifier, an unsupervised learning approach, for efficient prediction on the Pima diabetes dataset. The model achieved good accuracy compared to other current methods with a performance of 98.16%.

A Deep Neural Network (DNN) was utilized by Ashiquzzaman et al [9]. The Multilayer Perceptron, General Regression Neural Network, and Radial Basis Function make up the architecture of DNN. The Pima Indian dataset was used as the foundation for evaluating the method. The authors reported an accuracy rate of 88.41%.

In work by Swapna et al. [10], two DL approaches were applied to enhance the performance of diabetes diagnosis. Electrocardiograms, a private dataset, have been utilized for estimating the effectiveness of CNN and CNN-LSTM. In order to divide the dataset into training and testing sets, the authors applied five-fold cross validation. The obtained performance of models was 90.9% and 95.1%, respectively.

Aslan, M. F., & Sabanci [11], K developed an innovative approach for the early diagnosis of diabetes that is based on deep learning. To harness the resilient depiction provided by CNN technique for the early identification of diabetes, the authors transformed the numerical data of the Pima diabetes dataset into images according to the importance of the features. Data augmentation is used for these images once each feature has been included to the image in proportion with its importance. The ResNet18 and ResNet50 CNN models were used for diabetes diagnosis after feeding them with the augmented image data. Support vector machines (SVM) is employed for the amalgamation and classify deep features of the ResNet models. According to the findings, the classification accuracy using the SVM/cubic model of 500 chosen features was 92.19%.

Long short-term memory was implemented by Massaro et al. [12] to build artificial records and classify the data within them. With cross validation, the LSTM-AR classification accuracy, which was reported as 89%, overcame both LSTM and MLP.

Alex et al. [13] developed a 1D CNN structure for diabetes diagnosis. By using outlier detection, missing values were filled in. They then used SMOTE technique to preprocess the data, deleting the data imbalance. They used the processed data in the 1D CNN structure, and got accuracy of 86.29%.

METHODOLOGY

3.1 Pima Diabetes Dataset:

The Pima Indians Diabetes Dataset holds significance in machine learning and comes from the National Institute of Diabetes and Digestive and Kidney Diseases. It focuses on Pima Indian women, aiming to forecast diabetes presence based on specific medical measurements. Key attributes encompass:

- Pregnancies: Frequency of pregnancies.
- Glucose: Blood sugar level post a glucose tolerance test.
- BloodPressure: Diastolic blood pressure.
- SkinThickness: Thickness of triceps skinfold.
- Insulin: Insulin levels after a 2-hour period.
- BMI: Body mass index.
- DiabetesPedigreeFunction: Indicates diabetes history among relatives.
- Age: Individual's age.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   SkinThickness             768 non-null    int64
 4   Insulin                   768 non-null    int64
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64
 8   Outcome                   768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

Fig. 1. Statistical Information and data type of Pima dataset features

3.1 The Deep Learning Techniques:

3.2.1. Convolutional Neural Network (CNN)

CNNs belong to the category of multi-layered, feed-forward structures within artificial neural network architectures. that have been successfully applied in image analysis and computer vision, especially image object recognition. CNNs are inspired by the neurobiological processes seen in the human brain. These specialized neural network models have demonstrated remarkable efficacy, particularly in domains like image recognition and classification The Convolutional Neural Network (CNN) represents a mathematical framework characterized by a hierarchical arrangement of three fundamental layer types, namely convolutional, pooling, and fully connected layers [16]. Within this architecture, convolutional and pooling layers play a pivotal role in feature acquisition, while the fully connected layers facilitate the transmission of these extracted features towards the conclusive output for classification purposes.

Convolutions play a vital role in facilitating neural networks' comprehension of image pixels as numeric data. These convolution layers serve the purpose of translating images into numerical representations, enabling subsequent pattern extraction by the neural network. Within the convolutional layer, filters traverse the input image, employing the convolution formula denoted as Equation 1 to transform the visual data into interpretable numerical values.. The 'M' given in the equation represents the feature mapping, and the 'w' denotes the dimension (x,y) of convolution kernel [16].

$$M(i, j) = (R * w)(i, j) = \sum_{x} \sum_{y} R(i - x, j - y) w(x, y) \qquad (1)$$

In convolutional neural networks (CNNs), a prevalent practice involves the incorporation of a nonlinear or activation layer subsequent to each convolutional layer. The main purpose of this layer is to imbue the system with the capability to transform linear operations into nonlinear computations. The Rectified Linear Unit (ReLU) layer employs the function f(x) = max(0,x) across incoming data points. Primarily, this layer functions by nullifying negative activations, setting them to zero. Its implementation augments the nonlinear characteristics of the model and the overall neural network, while preserving the convolution layer's underlying features. The formula for ReLu is provided in Equation 2 [17].

$$\mathrm{ReLU}(X) = \max(x,0) \tag{2}$$

The pooling layer, a fundamental component within a CNN, is typically integrated subsequent to the convolutional operation and Rectified Linear Unit (ReLu) activation function. Its primary purpose is the minimization of both network parameters and computational load. The pooling layer works independently of each feature map. The most frequently strategy applied in pooling is maximum pooling. Figure 2 shows an example of maximum pooling. After creating the features that are subsampled by the convolution and pooling layers, these features are connected to the fully connected layer. In the realm of neural network architecture, features exhibit associations with one or multiple fully connected layers. In this context, each input element establishes connections with an output, with each neuron possessing adjustable weights. The final fully connected layer conventionally comprises a matching count of output nodes corresponding to the classification categories. Typically, classification tasks are executed within this layer. While various classifiers are applicable within this last layer, the Softmax function given in 3 is commonly employed. The Softmax equation is crucial in constraining the output neuron values within the (0,1) range, facilitating their interpretation as probabilities.

$$\sigma(z)_j = \frac{e^{zj}}{\sum_{k=1}^{K} e^{zk}} \quad , for\ j = 1, \dots \dots, K \tag{3}$$

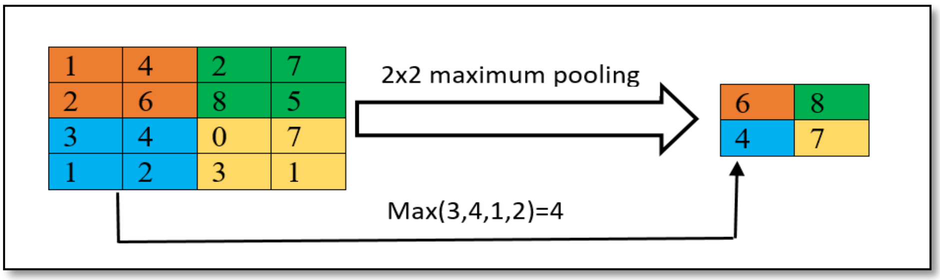

Fig.2. Maximum pooling

Artificial neural networks (ANNs) have a wide range of uses in machine learning applications. Various ANN models have been developed over time in response to various needs. Some ANNs are designed specifically to deal with various data formats. The term "LSTM" refers to Long Short Term Memory networks, a particular kind of Recurrent Neural Networks (RNN). In its hidden layer, the LSTM neural network has a sophisticated design known as the LSTM cell. The LSTM model is arranged as a chain structure [18]. The LSTM cell depictedin Figure 3 has three gates, the input gate, the forget gate, and the output gate, which manage the information transfer through the cell and neural network.

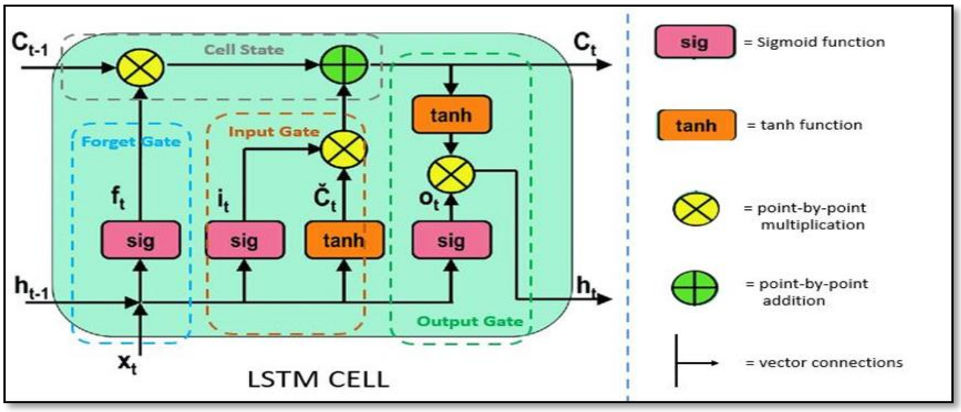

Fig.3. LSTM model

The forget gates determine how much information is forgotten and how much is transmitted to next level. These gates employ the sigmoid function, yielding outputs within the range of 0 to 1, where a value of 0 signifies complete inhibition of information transmission, while a value of 1 indicates full passage. Subsequently, the determination of what information merits storage is executed through the utilization of the sigmoid function at the input gate. Following this, the tanh function generates a vector of prospective values, symbolized as $\check{C}t$. These two mechanisms are then harmonized, culminating in the computation of the updated state information for the memory cell. The system's output is then computed. These operations can be mathematically stated as [18]:

$$f_t = \sigma\left(W_f\,[h_{t-1}, x_t] + b_f\right) \quad (4)$$
$$i_t = \sigma(W_i\,.\,[h_{t-1}, x_t] + b_i) \quad (5)$$
$$\check{C}_t = tanh(W_c.\,[h_{t-1}, x_t] + b_c) \quad (6)$$
$$C_t = f_t * C_{t-1} + i_t * \check{C}_t \quad (7)$$
$$o_t = \sigma(W_0.\,[h_{t-1}, x_t] + b_0) \quad (8)$$
$$h_t = o_t * tanh(C_t) \quad (9)$$

Here, σ is the sigmoid function and W_f and b_f are the weight matrices and bias of the forget gate, respectively. C_{t-1}, C_t are the cell states at times t-1 and t. The sigmoid layer decides which parts of the cell state will reach the output. Next, the output of the sigmoid gate (O_t) is multiplied by the new value generated from the cell state (C_t) by the tanh layer. W_0 and b_0 are the weight matrices and bias of the output gate, respectively.

3.2.2 Multi-Layer Perceptron Algorithm (MLP)

The MLP is a type of feedforward neural network that involves multiple intermediary layers positioned between the input and output layers. Each neuron within these layers is linked to every neuron in the surrounding layers. Figure 4 illustrates the configuration of an artificial neuron. This neuron computes the weighted sum of inputs, incorporates a threshold, and applies an activation function to determine the output [19], [20].

$$S = \sum x_i w_i + w_0 \quad (10)$$
$$Y = f(s) \quad (11)$$

The widely-used activation function, referred to as the sigmoid, is defined by Equation 9. The neural network's performance relies on this function's nonlinearity, which also helps scale the output within the [0-1] range.

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (12)$$

zϵR and $\boldsymbol{\sigma}$(z)ϵ(0,1)

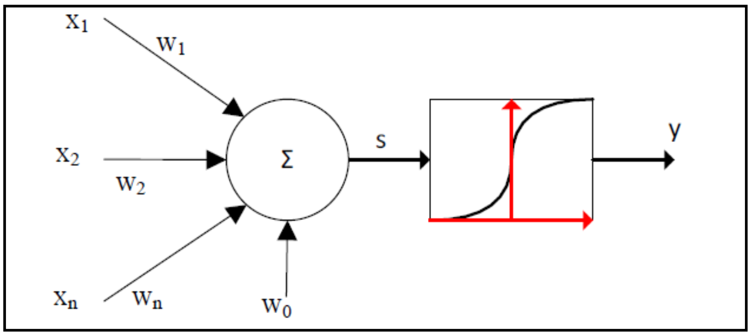

Fig.4. Artificial neuron

RESULTS

The experiment utilized a Pima dataset consisting of 9 features and 768 instances. The deep learning techniques employed for this experiment were CNN, LSTM, MLP; implemented using python tensorflow libraries. Data preprocessing steps involved filling missing values, remove outliers, and normalization. The dataset was divided into 80% training and 20% testing sets.

The structure of CNN implemented using the Keras library with a sequential model is:

1. **Input Layer**:
    - Input shape: **(X_train.shape[1], 1)** denotes the input shape for the data, where **X_train.shape[1]** stands for the number of features for each sample and **1** denotes the number of channels (here, it's a 1D convolution).
2. **Convolutional Layer**:
    - **Conv1D (32, 3, activation='relu')**:
        - 32 filters are applied.
        - Each filter has a width of 3.
        - Rectified Linear Unit (ReLU) activation function is used.
3. **Pooling Layer**:
    - **MaxPooling1D(2)**:
        - conducts a pool size of two for maximum pooling.
        - decrease the dimensionality of the data by taking the maximum value within each 2-element window.
4. **Flatten Layer**:
    - **Flatten()**:
        - The output generated by the preceding layer is transformed into a one-dimensional array, facilitating its input into the subsequent Dense layers.
5. **Dense Layers**:
    - **Dense (64, activation='relu')**:
        - A densely connected layer comprising 64 neurons and employing Rectified Linear Unit (ReLU) activation.
    - **Dense (1, activation='sigmoid')**:
        - The output layer is configured with a singular neuron employing a sigmoid activation function, a convention commonly applied in binary classification tasks.

The structure of LSTM implemented using the Keras library with a sequential model is:

1. **Sequential Model**: This is a linear stack of layers in Keras, where layers are added one by one.
2. **LSTM Layer**:

- **lstm_model.add(LSTM(64,input_shape=(X_train_lstm.shape[1], X_train_lstm.shape[2])))**
- This line adds an LSTM layer with 64 units (or cells).
- The **input_shape** parameter specifies the input shape of the data. In this case, it expects input sequences with dimensions **(X_train_lstm.shape[1], X_train_lstm.shape[2])**.
- **X_train_lstm.shape[1]** refers to the length of each input sequence, and **X_train_lstm.shape[2]** refers to the number of features in each time step of the sequence.

3. **Dense Layer**:

- **lstm_model.add(Dense(1, activation='sigmoid'))**
- After the LSTM layer, a Dense layer with 1 unit is added.
- The activation function used here is **'sigmoid'**, which is common for binary classification tasks as it compress the output within a range of 0 to 1, indicating the probability of membership to one class.

4. **Compilation**:

- **lstm_model.compile(loss='binary_crossentropy',optimizer='adam', metrics=['accuracy'])**
- Compilation configures the learning process of the model.
- **loss='binary_crossentropy'** defines the loss function employed for optimization in binary classification problems.
- **'adam'** is the optimizer being applied, which is an adaptive learning rate optimization algorithm.
- **metrics=['accuracy']** designates the evaluation score to be applied in training, in this case, accuracy.

The structure of MLP implemented using the Keras library with a sequential model is:

1. **Input Layer:** It takes input data with a shape determined by the number of features in **X.shape[1]**.

2. **Hidden Layers:**

- **Dense Layer 1:** Consists of 64 neurons (units) using the ReLU (Rectified Linear Unit) activation function.
- **Dense Layer 2:** Consists of 32 neurons with the ReLU activation function.

3. **Output Layer:**

- **Dense Layer 3 (Output):** Comprises a single neuron using the sigmoid activation function, typical for binary classification tasks.

4. **Model Compilation:**

- **Optimizer:** Uses the Adam optimizer, a highly effective algorithm employed in the training of neural networks**.**
- **Loss Function:** Employs binary cross-entropy, often used for binary classification tasks.
- **Metrics:** Evaluates model performance based on accuracy during training.

The evaluation of the models was conducted utilizing the following metrics [21], [22]:

1. Accuracy**:**
Accuracy denotes the ratio of correctly classified samples out of the overall samples.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{13}$$

2. Precision**:**
Precision quantifies the accuracy of positive predictions. t estimates the percentage of true positive predictions over all predictions that are positive.

$$Precision = \frac{TP}{TP + FP} \tag{14}$$

3. Recall**:**
The percentage of true positives that were accurately estimated is determined using recall. It establishes if the model can recognize every relevant case.

$$Recall = \frac{TP}{TP + FN} \tag{15}$$

4. ROC-AUC (Receiver Operating Characteristic - Area Under the Curve):

ROC-AUC evaluates the balance between true positive rate (sensitivity) and false positive rate (specificity) at different thresholds. A higher AUC value indicates a more efficient model. A perfect classifier would have a curve that reaches the top left corner (0 false positive rate, 1 true positive rate), indicating maximum performance [23], [24].

Table.1 showcases the Evaluations for performance (Accuracy, Precision, Recall, ROC-AUC) of various deep learning models (CNN, LSTM, MLP) on the Pima dataset.

Table I. the performance metrics of deep learning techniques

Model	Accuracy	Precision	Recall	ROC-AUC
CNN	0.77	0.69	0.67	0.83
LSTM	0.75	0.64	0.67	0.82
MLP	0.75	0.64	0.67	0.80

In the evaluation of our models, ROC curves function as a robust measure of their classification performance. The ROC-AUC is a vital measure reflecting the models' capability to differentiate across classes. We present the ROC-AUC curves (Figure. 5) for our three models: CNN, LSTM, and MLP, based on their respective performance metrics.
The CNN model demonstrated a commendable ROC-AUC score of 0.83, showcasing its strong discriminative capability in distinguishing between classes, as illustrated in the corresponding ROC curve.
Similarly, the LSTM model exhibited a competitive ROC-AUC score of 0.82, indicating its proficiency in classification tasks, as depicted in its ROC curve.
Moreover, the MLP model displayed a respectable ROC-AUC score of 0.80, indicating its reasonable discriminative performance, visualized through its ROC curve.
These ROC-AUC curves provide a graphic illustration of the models' trade-offs between true positive rate and false positive rate, underlining the CNN's stronger discriminatory power compared to the LSTM and MLP models in our classification task.

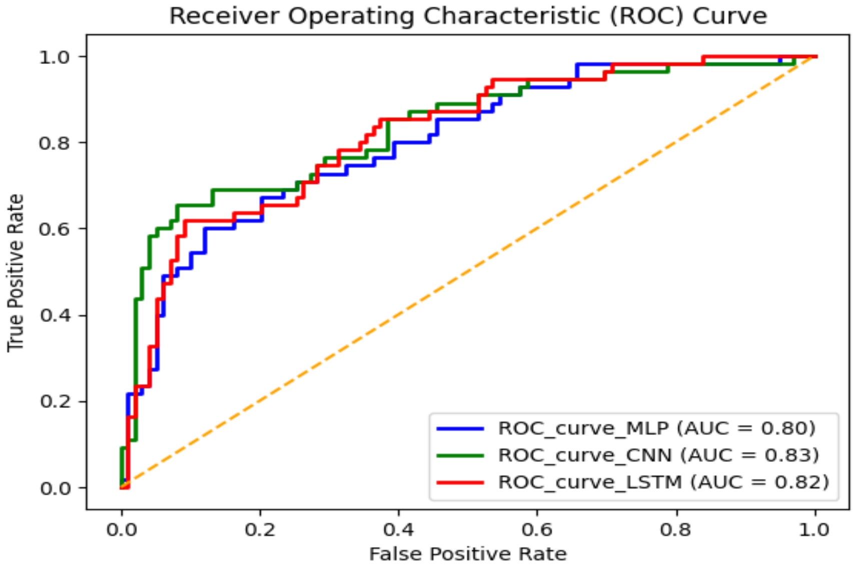

Fig.5. ROC_AUC of deep learning techniques

The following figures illustrates the progression of accuracy and loss plots across the different epochs for the models used in this study.

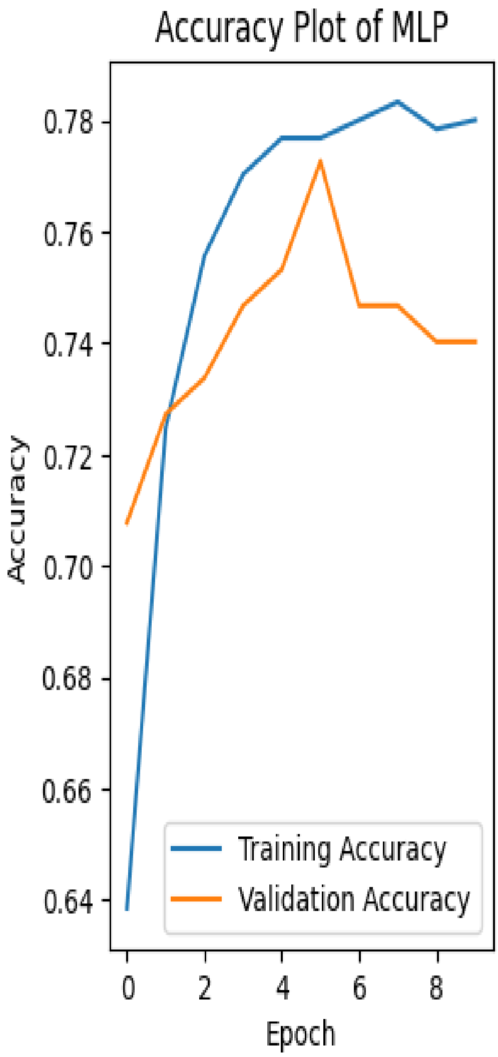

Fig.6. Accuracy plot of MLP

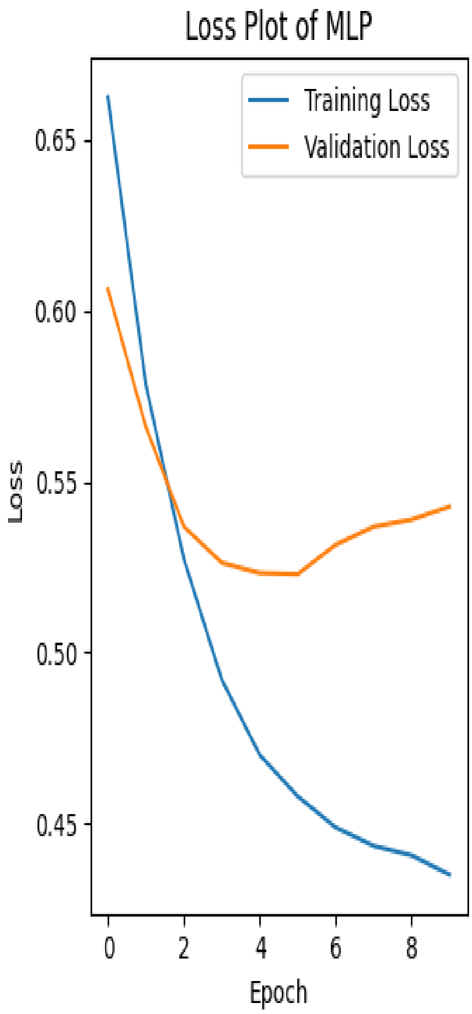

Fig.7. Loss plot of MLP

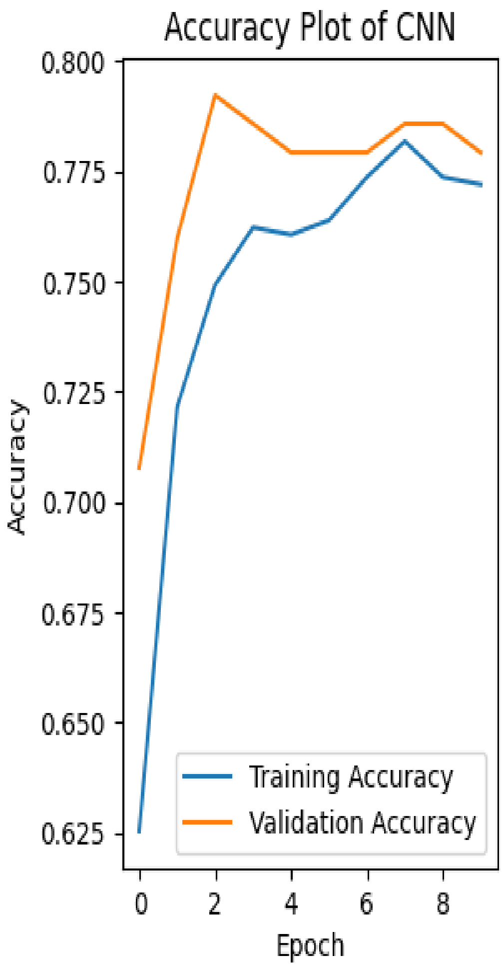

Fig.8. Accuracy plot of CNN MLP

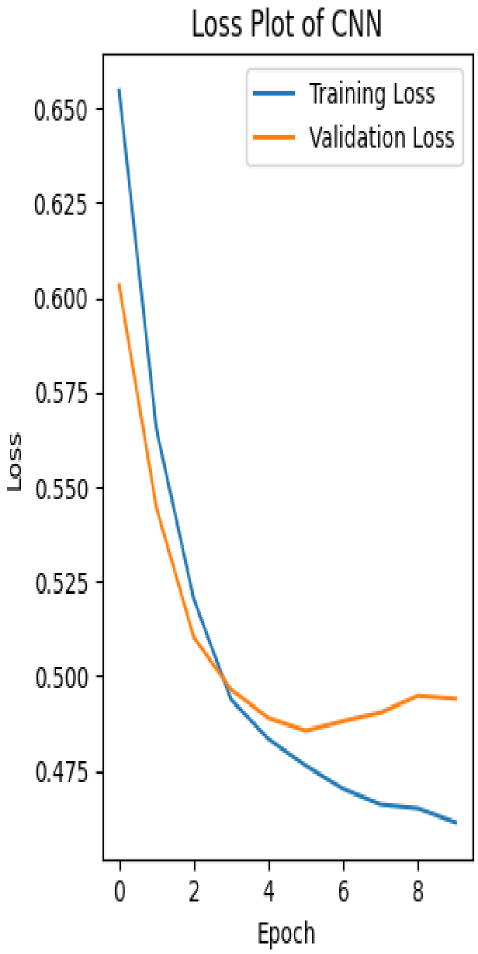

Fig.9. Loss plot of CNN

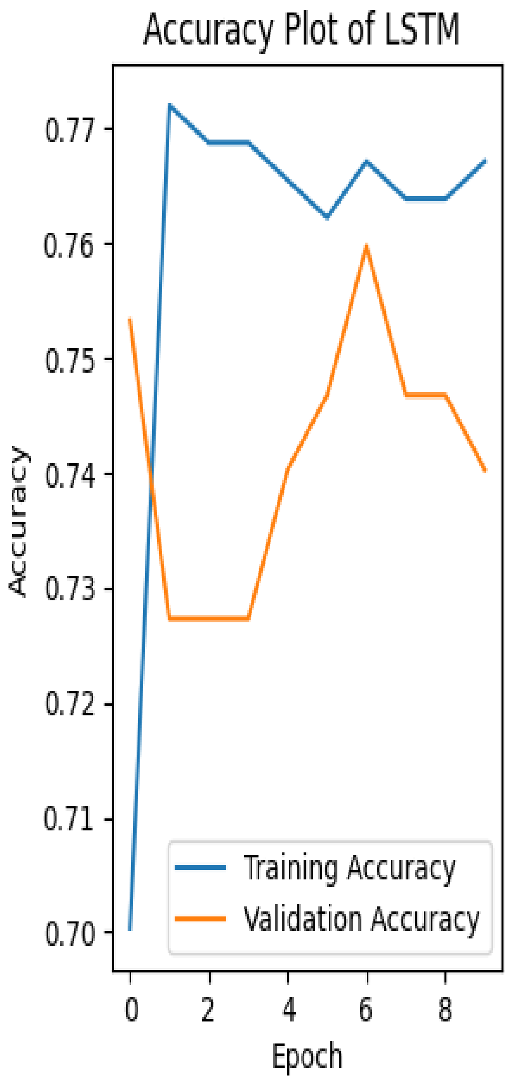

Fig.10. Accuracy plot of LSTM MLP

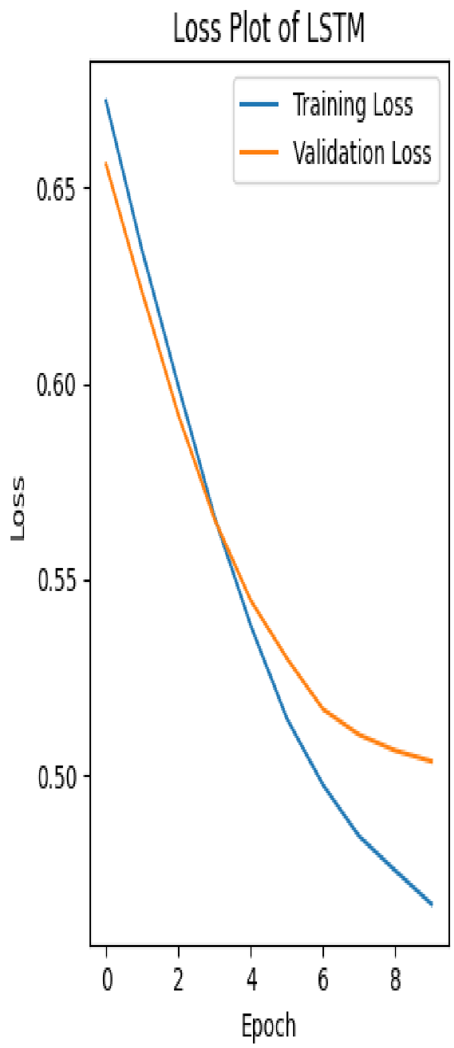

Fig.11. Loss plot of LSTM

DISCUSSION

The study compared three different models: CNN, LSTM, and MLP, based on their performance metrics - accuracy, precision, recall, and ROC-AUC score. The results indicate that CNN achieved the highest accuracy of 0.77 and the highest ROC-AUC score of 0.83 among the three models. LSTM and MLP both had similar accuracy and ROC-AUC scores, around 0.75 and 0.80 respectively. Precision scores were consistent among LSTM and MLP at 0.64, slightly lower than CNN's precision of 0.69. Recall scores were consistent between LSTM and MLP at 0.67, with CNN slightly lower at 0.67.
It is noticeable in this study that CNN appears to be the best performer overall, showcasing higher accuracy, precision, and ROC-AUC compared to LSTM and MLP. While LSTM and MLP exhibit

similar performance in most metrics, they lag slightly behind CNN in accuracy and ROC-AUC. Precision and recall are relatively consistent across the models, with CNN showing a slight advantage in precision.
The experiment involved training the models over various epochs using a pima dataset. The model achievement was evaluated based on its training and validation accuracy at each epoch. Figures 6-11 show the accuracy and loss plots of CNN, LSTM, and MLP.
When prioritizing accuracy and overall model performance, the CNN model might be the most suitable choice among these three for the given task or dataset.

CONCLUSION

In conclusion, this study explored the effectiveness of various deep learning techniques in predicting diabetes using the Pima dataset. The preprocessing steps of normalization, handling missing values, and outlier removal were integral in preparing the data for analysis. Through rigorous experimentation, three distinct models—CNN, LSTM, and MLP—were evaluated depending on accuracy, precision, recall, and ROC-AUC scores.
The results showcased promising predictive capabilities across all models, with the CNN achieving the highest accuracy of 77%, closely followed by LSTM and MLP, both attaining 75%. While the accuracy was notable, the precision, recall, and ROC-AUC scores indicated a fairly consistent performance among the models, with minor deviations.
The CNN model exhibited slightly superior precision and recall metrics compared to LSTM and MLP, with a commendable ROC-AUC score of 0.83, showcasing its robustness in discriminating between diabetic and non-diabetic instances. LSTM and MLP closely trailed behind with respectable ROC-AUC scores of 0.82 and 0.80, respectively.

These findings underscore the potential of deep learning methodologies in predicting diabetes from the Pima dataset. However, further research and fine-tuning of these models could enhance their performance, potentially leading to more accurate and reliable predictive tools in diagnosing diabetes. Overall, this paper contributes useful understanding of application of deep learning techniques for diabetes prediction, opening the door for additional advances in healthcare and predictive analytics.

REFERENCES

1. Naz H, Ahuja S. "Deep learning approach for diabetes prediction using PIMA Indian dataset." J Diabetes Metab Disord 2020;19(4):391–403. https://doi.org/10.1007/s40200-020-00520-5.

2. Islam MF, Ferdousi R, Rahman S, Bushra HY. "Likelihood prediction of diabetes at an early stage using data mining techniques." In Computer Vision and Machine Intelligence in Medical Image Analysis, pp. 113-125, Springer, Singapore, 2020.

3. "The 6 Different Types of Diabetes." (2024, January 4). The diabetic journey. https://thediabeticjourney.com/the-6-different-types-of-diabetes.

4. "Statistics About Diabetes." American Diabetes Association, 2024, January 4. https://www.diabetes.org.

5. Ramachandran A. "Know the signs and symptoms of diabetes." Indian J Med Res 2014;140:579–81.

6. Palaniappan S, Awang R. "Intelligent heart disease prediction system using data mining techniques." 2008 IEEE/ACS Int Conf Comput Syst Appl 2008. https://doi.org/10.1109/aiccsa.2008.4493524.

7. Goldenberg R, Punthakee Z. "Definition, classification and diagnosis of diabetes, prediabetes and metabolic syndrome." Canadian journal of diabetes 2013;37:S8-S11.

8. Bala Manoj Kumar P, Srinivasa Perumal R, Nadesh RK, Arivuselvan K. "Type 2: Diabetes mellitus prediction using Deep Neural Networks classifier." International Journal of Cognitive Computing in Engineering 2020;1:55–61.

9. Ashiquzzaman A, Kawsar Tushar A, Rashedul Islam MD, Shon D, Kichang LM, Jeong-Ho P, Dong-Sun L, Jongmyon K. "Reduction of overfitting in diabetes prediction using deep learning neural network." In IT Convergence and Security; Lecture Notes in Electrical Engineering, Springer: Singapore, 2017; Volume 449.

10. Swapna G, Soman KP, Vinayakumar R. "Automated detection of diabetes using CNN and CNN-LSTM network and heart rate signals." Procedia Comput. Sci. 2018;132:1253–1262.

11. Aslan MF, Sabanci K. "A novel proposal for deep learning-based diabetes prediction: Converting clinical data to image data." Diagnostics 2023;13(4):796.

12. Massaro A, Maritati V, Giannone D, Convertini D, Galiano A. "LSTM DSS Automatism and Dataset Optimization for Diabetes Prediction." Appl. Sci. 2019;9:3532.

13. Alex SA, Nayahi J, Shine H, Gopirekha V. "Deep convolutional neural network for diabetes mellitus prediction." Neural Comput. Appl. 2022;34:1319–1327.

14. "Pima Indians Diabetes Database." Accessed: Dec. 22, 2023. [Online]. Available: https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database

15. Emad Majeed Hameed, Hardik Joshi. "Current Diabetes Classification And Prediction Models Using Intelligent Techniques: Review." In Full Text Book of Minar Congress 6, Turkey: Rimar Academy, 2022, pp. 20–50. doi: http://dx.doi.org/10.47832/MinarCongress6.

16. Massaro A, Maritati V, Giannone D, Convertini D, Galiano A. "LSTM DSS automatism and dataset optimization for diabetes prediction." Applied Sciences 2019;9(17):3532.

17. Ciresan DC, Meier U, Masci J, Gambardella LM, Schmid-Huber JF. "High-performance convolutional neural networks for image classification." In Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, 2011, pp. 1237-1242.

18. Madan P, Singh V, Chaudhari V, Albagory Y, Dumka A, Singh R, Gehlot A, Rashid M, Alshamrani SS, Alghamdi AS. "An Optimization-Based Diabetes Prediction Model Using CNN and Bi-Directional LSTM in Real-Time Environment." Appl. Sci. 2022;12:3989. https://doi.org/10.3390/app12083989

19. Kumar Y. "Analysis of Bayes, Neural Network and Tree Classifier of Classification Technique in Data Mining using WEKA." In Computer Science & Information Technology (CS & IT), Academy & Industry Research Collaboration Center (AIRCC), May 2012, pp. 359–369. doi: 10.5121/csit.2012.2236.

20. Morariu D, Cretulescu R, Breazu M. "The Weka Multilayer Perceptron Classifier." Int. J. Adv. Stat. ITC Econ. Life Sci. 2017;7(1):Art. no. 1.

21. Hameed EM, Hussein IS, Altameemi HG, Kadhim QK. "Liver Disease Detection and Prediction Using SVM Techniques." In 2022 3rd Information Technology To Enhance e-learning and Other Application (IT-ELA), 2022, pp. 61-66.

22. Hastie T, et al. "The Elements of Statistical Learning: Data Mining, Inference, and Prediction." Springer, 2009.

23. Fawcett T. "An introduction to ROC analysis." Pattern Recognition Letters 2006;27(8):861-874.

24. Hanley JA, McNeil BJ. "The meaning and use of the area under a receiver operating characteristic (ROC) curve." 1982.