Diabetes Prediction using Deep Learning Model

ABSTRACT: A fully automatic detection system for diabetes is presented in this work. The proposed detection system includes the pre-processing of features, training, and testing stages. In the first step, data is normalized by using the Z-score method. A sequential deep learning technique is applied for the prediction of diabetes which includes training and testing of the model. A publicly available PIMA Indian diabetes dataset is used for the experiment in this work. The results of the proposed model have compared with state of art machine learning techniques also. The highest 96.108%, 96.06%, 93%, 98%, 95%, 94% training accuracy, testing accuracy, sensitivity, specificity, precision, and F1-score, is obtained, respectively by the proposed deep learning model. The proposed model outperforms the state of art machine learning techniques.

Keywords: Diabetes, Deep Learning, PIMA, Machine Learning

1 Introduction

Diabetes is a disease that occurs either when the pancreas does not produce enough insulin or when the body cannot effectively use the produced insulin. Insulin is a hormone that regulates blood sugar. Hyperglycaemia, or raised blood sugar, is a common effect of uncontrolled diabetes and after some time, it leads to serious damage to many of the body's systems, especially the nerves and blood vessels. The World Health Organization (WHO) evaluated the worldwide regularity of diabetes among grown-ups more than 18 years old as 8.5% in 2014. A total of 72.96 million instances of diabetes are observed in the grown-up populace of India [1]. The commonness in urban regions ranges approximately 10.9% and 14.2%. The pervasiveness in provincial India is 3.0-7.8% among the populace matured 20 years or more and higher predominance among people matured more than 50 years. Diabetes can be dominated by improving lifestyle and switching to a healthy diet.

Diabetes is a financially exhausting health condition, individuals with diabetes bring about normal clinical expenses of $13,700 every year, and $7,900 of that is attributed to diabetes [2]. In the on-going diabetes in America review led by Health Union, 74% and 32% of overview respondents have a yearly family unit income below $75K, and $30K, respectively. A portion of the expenses of clinical supplies and specialist visits are secured by protection. Thirty-nine percent of review members noted they get bunch inclusion through their boss or their life partner's manager.

Medicare covers 35% of the respondents. However, there are as yet significant cash-based costs for the individual living with type 2 diabetes. Diabetes is a health condition that causes a financial burden on people with no stable background. This paper provides the automatic prediction of diabetes based on the features of a person whether the person is suffering from diabetes or not. The deep learning technique has been utilized for the training and testing of the model and results are analysed.

This work is structured in the following sections. The literature review is presented in Section 2. The proposed methodology is described in Section 3. Experimental results are discussed in Section 4. Conclusions are presented in Section 5.

2 Related Work

Various algorithms are already implemented by different authors. Calisir et al. developed an automatic diabetes diagnosis system based on the Linear Discriminant Analysis (LDA) technique[3]. The highest 89.74% classification accuracy is achieved by applying The Morlet wavelet support vector machine classifier. Zou et al. have applied decision tree, random forest, and neural network techniques for the prediction of diabetes mellitus. The highest 80.84% accuracy is achieved by using principal component analysis and minimum redundancy maximum relevance (mRMR) for dimensionality reduction [4]. Tigga et al. used Logistic regression, K-Nearest Neighbour, Support vector machine, Naïve Bayes, decision tree and, Random forest for the classification of diabetic and non-diabetic [5]. The highest 90% accuracy is achieved by applying the Random Forest classifier. Sisodia et al. applied decision tree, naïve Bayes, and SVM for prediction of diabetes. The highest 76.30% accuracy is achieved by using the Naïve Bayes classifier [6]. Wu et al. obtained 95.42% accuracy by utilizing improved k-NN and logistic regression techniques to predict Type 2 diabetes mellitus [7]. Meng et al. achieved 73.23% and 77.87% classification accuracy by applying artificial neural network and decision tree (C.5) model, respectively[8]. Choubey et al. used Genetic algorithm and radial basis function based neural network techniques for feature selection and diabetes classification. The highest 76.087% classification accuracy is achieved on Pima Indian Diabetes Dataset (PIDD) [9]. Haung et al. obtained the highest 95% accuracy on Ulster Community and Hospitals Trust

(UCHT) data set by applying Naïve Bayes, IB1, and decision tree classifier for diabetes prediction [10]. Perveen et al. applied Naïve Bayes and decision tree machine learning techniques for diabetes supervised prediction achieved 81% and 80% true positive rate, respectively [11]. The different machine learning algorithm is applied by various authors in existing work. In this work, deep learning algorithm is applied for the prediction of diabetes to improve the accuracy of the prediction result.

3 Methodology

The block diagram of the proposed model for automatic diabetic prediction is shown in Fig. 1. It includes preprocessing of input data followed by training and testing of the deep learning model. Each step is briefly outlined in subsequent subsections.

3.1 Raw Data Input

The PIMA diabetes dataset has been used for the validation of the proposed model [12]. This dataset is originally collected from the National Institute of Diabetes, Digestive, and Kidney Diseases which consists of various attributes collected from 768 people.

The dataset includes 8 independent variables which are pregnancies, plasma glucose concentration in 2 hours in an oral glucose tolerance test, diastolic blood pressure (mm Hg), Triceps skinfold thickness (mm), 2-Hour serum insulin (mu U/ml), Body mass index (weight in kg/ (height in m) ^2), and diabetes pedigree function. Sample records of the PIMA dataset are listed in Table-1. A brief description of each feature of the dataset is presented as:

Pregnancies: During pregnancy, the placenta makes hormones that cause glucose to develop in the blood. Typically, the pancreas can convey enough insulin to deal with it. However, sometimes the body cannot make enough insulin or quits utilizing insulin as it should, the glucose levels rise, and results in gestational diabetes.

Glucose: Diabetes is an issue with the human body that causes glucose levels to ascend higher than ordinary. This is additionally called hyper-glycemia. Sudden rise and fall of glucose level can be subjected to Diabetes.

Blood Pressure (BP): Type 2 diabetes attributable due to impedance from insulin which is a hormone in the human body that utilizes glucose for energy. Over the long run, diabetes harms the little veins in the human body and makes the walls of the veins solidify which builds pressure and prompts hypertension or High Blood Pressure. So blood pressure and diabetes are correlated.

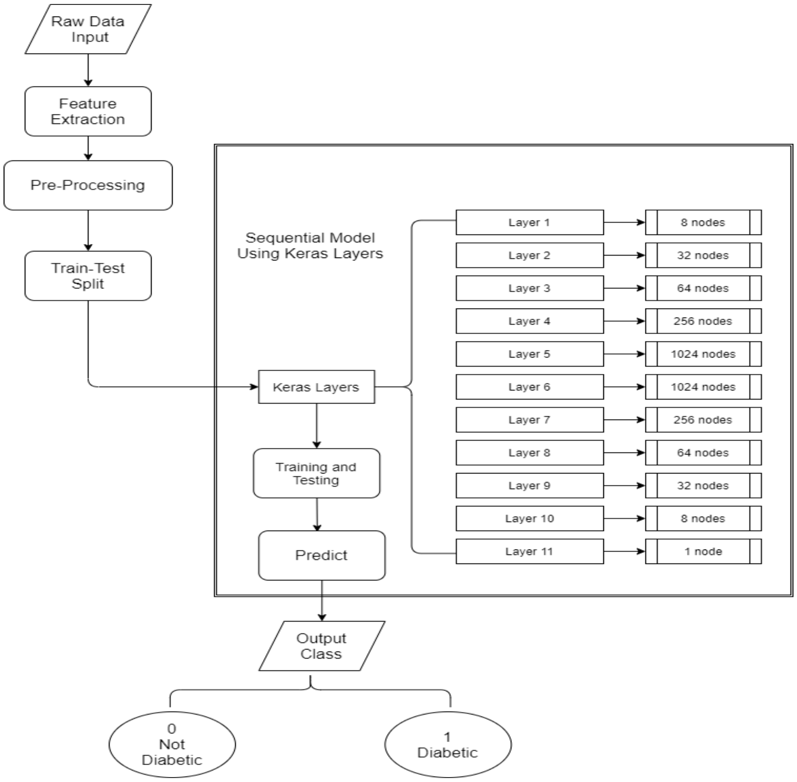


Fig. 1. Block Diagram of Automatic Diabetic Prediction Model

Skin Thickness (ST): Skin thickness is dictated by collagen content and expanded in insulin-subordinate diabetes mellitus (IDDM). Skin thickness of triceps skinfold is correlated with diabetes.

Insulin Level: Wrecked insulin-delivering cells also create insulin. Insulin should be incumbent to move glucose into cells all through the body. The subsequent insulin lack leaves an an excessive amount of sugar in the blood and insufficient in the cells for energy and thus causes diabetes.

Body Mass Index (BMI): Overweight (BMI > 25) burdens the internal parts of individual cells. Insulin obstruction and high centralizations of the sugar glucose in the blood are definite indications of diabetes.

Diabetes Pedigree Function (DPF): It is a function that scores the probability of having diabetes depending on family ancestry and genetics.

Age: There is a high danger for the development of type 2 diabetes because of the joined impacts of expanding insulin opposition and debilitated pancreatic islet work with aging

Table 1. Samples of PIMA Data set

S no.	Pregnancies	Glucose	BP	ST	Insulin Level	BMI	DPF	Age	Out-come
1	6	148	72	35	0	33.6	0.627	50	1
2	1	85	66	29	0	26.6	0.351	31	0
3	8	183	64	0	0	23.3	0.672	32	1
4	1	89	66	23	94	28.1	0.167	21	0
5	0	137	40	35	168	43.1	2.288	33	1
6	5	116	74	0	0	25.6	0.201	30	0
7	3	78	50	32	88	31	0.248	26	1
8	10	115	0	0	0	35.3	0.134	29	0
9	2	197	70	45	543	30.5	0.158	53	1
10	8	125	96	0	0	0	0.232	54	1

3.2 Pre-Processing

Data Standardization is a process of tuning and rescaling features in such a manner that the resulting attribute has 0 mean and the standard deviation of 1. The dataset is normalized by using the Z-score method to ensure its uniform-ness and it is defined as:

$$z = \frac{(x-u)}{S^1} \tag{1}$$

Here, *u* and *s* represent the mean and standard deviation of the individual features, respectively.

3.3 Deep Learning Model

The basic sequential model which is defined as a network of Dense Layers is used for training of deep learning framework. Keras deep learning framework is used in this work for the prediction of diabetes. This model comprises many layers and sustains the balance of all the layers.

3.3.1 Keras Layers (for Neural Network)

Keras is an open-source deep learning framework for python which is based on the minimal structure. It provides a clean and easy way to create deep learning models based on TensorFlow [13].

Keras Layers are the fundamental units of Neural Networks. A layer comprises a tensor-input and tensor-output computation function and a state, which is stored in TensorFlow variables. Tensors are multidimensional arrays with a uniform data type. All tensors are immutable.

Tensors contain floats, integers, complex numbers, and strings. However, there are specialized types of tensors that can handle different shapes: (i) Ragged tensors (ii) Sparse tensors. Basic math including addition, element-wise multiplication, and matrix multiplication can be done on tensors. Models can be defined, saved, and restored to perform machine learning training and testing. The model can be defined as: (i) A function that computes something on tensors (a forward pass) and (ii) Some variables that can be amended in response to training. Most models are made of layers. Layers are functions with a known mathematical structure that can be reused and have trainable variables. In TensorFlow, most high-level implementations of layers and models, such as Keras or Sonnet, are built on the same foundational class.

A Keras layer requires the shape of the input (input_shape) to understand the structure of the input data, an initializer to set the weight for each. The constraints parameter restricts and specifies the range in which the weight of input data to be generated. The regularizer optimizes the layer and the model by dynamically applying the penalties on the weights during the optimization process. Input Shape needs to be provided to the first layer of the network. The output of the previous layer becomes the input of the next layer for subsequent layers of the network [14]. The input parameters set for the Keras model are given as:

- The first parameter represents the number of units (neurons).
- ***input_shape*** represents the shape of the input data.
- ***kernel_initializer*** is resolute as a uniform function.
- ***kernel_regularizer*** is established as none which represents **the regularizer** to be used.

- ***kernel_constraint*** represents the constraint to be used and its value is set as a ***MaxNorm*** function.
- ***activation*** represents activation to be used and its values is fixed as Relu function.

This model consists of a total of 11 layers which include 9 hidden layers. Out of 9 hidden layers, the first 10 layers include Rectified Linear Activation with 8, 32, 64, 256, 1024, 1024, 256, 64, 32, and 8 nodes, respectively. The last layer consists of a single node with Sigmoid Activation which classifies output as diabetes and non-diabetes.

3.3.1.1 Keras Sequential Model

It is the basic sequential model that can be defined as a network of Dense Layers, which is used for deep learning using Keras [14]. It helps us in making a model consisting of many layers and sustaining the balance of all the layers. A Sequential model is appropriate for a plain stack of layers where each layer has exactly one input tensor and one output tensor. A Sequential model is not appropriate when: (i) The model has multiple inputs or multiple outputs, (ii) Any of the layers has multiple inputs or multiple outputs. (iii)Layer Sharing is done, (iv) Need of non-linear topology (e.g., a residual connection, a multi-branch model).

3.3.1.2 Keras Functional Model

The Keras functional API is a way to create models that are more flexible than the Keras Sequential API [14]. The functional API can handle models with non-linear topology, shared layers, and even multiple inputs or outputs. The functional API is a way to build graphs of layers. In the functional API, models are created by specifying their inputs and outputs in a graph of layers. That means that a single graph of layers can be used to generate multiple models. The functional API makes it easy to manipulate multiple inputs and outputs.

4. Experimental Results and Discussion

Training and testing are performed by using the Train-Test split method which divides the data arrays into two subsets i.e. for testing and training. The dataset has been divided as 25% and 75% testing and training set, respectively. Keras Layers for Neural Network has been used to predict if the person has diabetes or not. The model is trained with binary cross-entropy loss function with stochastic gradient descent optimizer and accuracy in metrics. Stochastic gradient descent (SGD) performs a parameter update for *each* training example $x^{(i)}$ and label $y^{(i)}$ defined as:

$$\theta = \theta - \eta \nabla_\theta J(\theta; x^i;\ y^i) \tag{2}$$

The binary cross-entropy loss function is used for binary classification which is defined as:

$$H_p(q) = -1/N(\sum_{i=1}^{N}(y_i\,.log\;(p(y_i) + (1-y_i).log\;(1-p(y_i\,) \qquad (3)$$

here, y_i represents the predicted output. The training and testing set is compiled with 220 epochs with batch size and verbose as 1. The performance of the proposed deep learning framework and state-of-art machine learning is measured in terms of sensitivity, specificity, accuracy, precision, and F1-score[15]. All the performance matrices are computed using a confusion matrix which is shown in Table 2. For two output class, the predicted outcome can be categorized as false negative when the person is diabetic though the model predicted as non-diabetic, false positive when the person is non-diabetic and the model predicted as diabetic, true negative when the person is non-diabetic and the model predicted as non-diabetic and true positive when the person is diabetic and the model predicts as diabetic. The model predicts as non-diabetic or outcome is predicted as 0 and the model predicts as diabetic or outcome is predicted as 1.

Table 2. Confusion Matrix

Predicted output Actual Output	True	False
True	True Positive(TP)	False Positive(FP)
False	False Negative(FN)	True Negative(TN)

The sensitivity, specificity, accuracy, precision, recall, and F1- score are defined as:

Sensitivity: $$\frac{TP}{TP+FN} \qquad (4)$$

Specificity: $$\frac{TN}{TN+FP} \qquad (5)$$

Precision: $$\frac{TP}{TP+FP} \qquad (6)$$

Accuracy: $$\frac{TP+TN}{TP+TN+FP+FN} \qquad (7)$$

F1- score: $$\frac{2*TP}{2*TP+FP+FN} \qquad (8)$$

The experiment has been done with various numbers of dense layers. Total of six deep learning models are created with different dense layers which are presented in Table 2. Results obtained with each model are analyzed in Table 3. The highest value of 96.108%, 96.06%, 93%, 98%, 95%, 94% training accuracy, testing, sensitivity, specificity, precision, and F1-score, respectively, is obtained, by the proposed deep learning model. Training model accuracy and loss of each model are presented in Fig. 2.

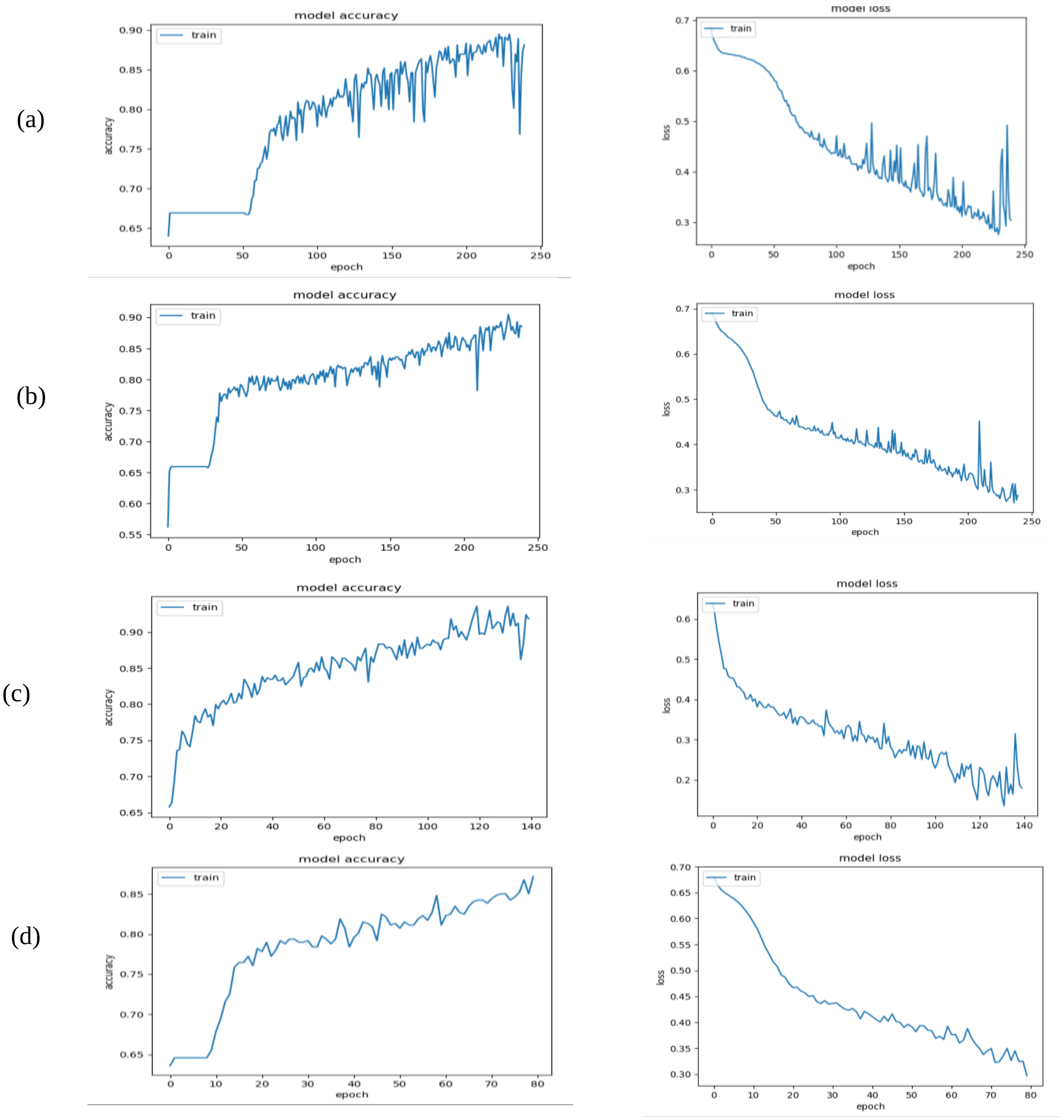
Table 2. Input Parameters of the Deep Learning Model

(Number of Input nodes=8 and Number of Output node=1)

Model Number	**Type**	**Number of Dense Layers**	**Epochs**	**Batch Size**
1.	Functional	8/32/64/128/512/1024/1024/512/128/64/32/8	140	27
2.	Functional	8/32/64/256/1024/1024/256/64/32/8	45	5
3.	Functional	8/32/64/128/512/1024/512/128/64/32/8	15	10
4.	Sequential	-8/32/64/256/256 /1024/1024/256/256/32/8	232	18
5.	Sequential	8/32/64/256/1024 /256/64/32/8-	1670	227
6.	Sequential	8/32/64/256/1024 /1024/256/64/32/8-	1670	215

Table 3. Results obtained with Proposed Deep Learning Models

Model Number	**Training Accuracy (%)**	**Testing Accuracy (%)**	**Sensitivity (%)**	**Specificity (%)**	**Precision (%)**	**F1 Score (%)**
1	88.91	71.65	50	85	68	58
2	91.25	72.83	56	83	65	60
3	93.39	66.14	52	75	55	53
4	83.07	72.05	41	88	64	50
5	92.80	73.23	72	74	58	65
6	**96.11**	**96.06**	**93**	**98**	**95**	**94**


(a)
model accuracy
train
accuracy
0.65 0.70 0.75 0.80 0.85 0.90
epoch
0 50 100 150 200 250
model loss
train
loss
0.3 0.4 0.5 0.6 0.7
epoch
0 50 100 150 200 250
(b)
model accuracy
train
accuracy
0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90
epoch
0 50 100 150 200 250
model loss
train
loss
0.3 0.4 0.5 0.6 0.7
epoch
0 50 100 150 200 250
(c)
model accuracy
train
accuracy
0.65 0.70 0.75 0.80 0.85 0.90
epoch
0 20 40 60 80 100 120 140
model loss
train
loss
0.2 0.3 0.4 0.5 0.6
epoch
0 20 40 60 80 100 120 140
(d)
model accuracy
train
accuracy
0.65 0.70 0.75 0.80 0.85
epoch
0 10 20 30 40 50 60 70 80
model loss
train
loss
0.30 0.35 0.40 0.45 0.50 0.55 0.60 0.65 0.70
epoch
0 10 20 30 40 50 60 70 80

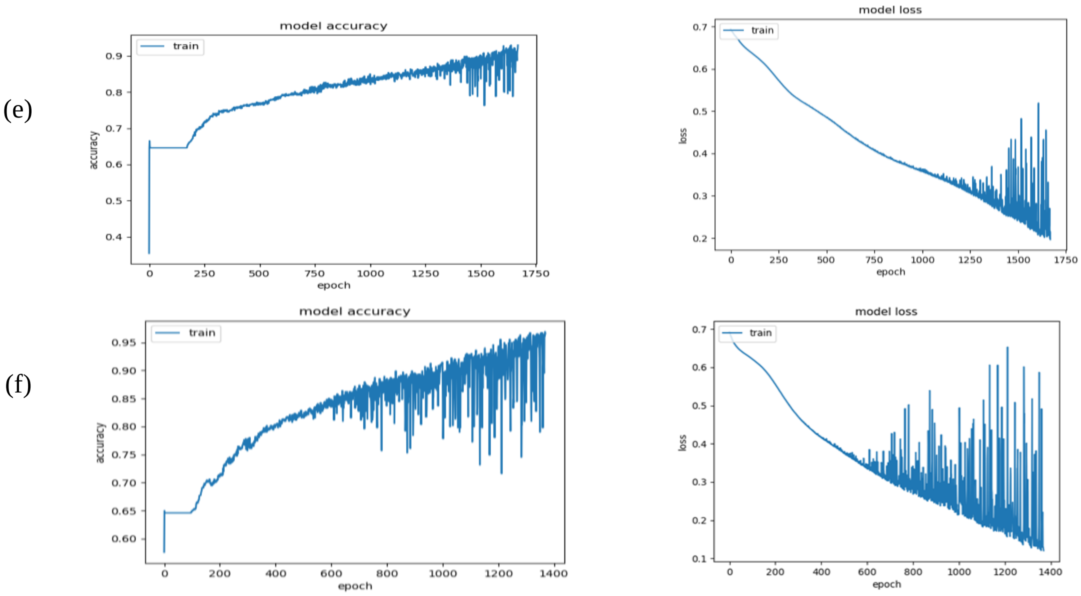


Fig. 2 Training accuracy and Loss obtained of defined (a) Model1 (b) Model2 (c) Model3 (d) Model 4 (e) Model 5 (f) Model6

4.1 Comparison of Proposed Model with State of art Machine Learning Techniques

Performance of the deep learning model is also compared with the state-of-art machine learning techniques such as Logistic Regression, Random Forest, Support Vector Machine, and KNN (with K=3, 4,5, 6) [15] is shown in Table 4. It can be observed that the performance of the deep learning model is better than the state-of-art machine learning techniques.

5. Conclusions

This endeavour presents an automatic prediction of diabetes using a deep learning approach. The experiment has been on the publicly available PIMA dataset. Data normalization has been done by using the Z-score technique before applying the input data set for training and testing of the deep learning model. The various machine learning techniques such as logistic regression, support vector machine, k-NN, Random Forest are also applied for comparison aligns with the proposed deep learning model.

It is empirical that the deep learning approach performed better than the state-of-art machine learning approaches. In future, the proposed model will be tested on a new dataset of diabetes. This model will be deployed in website for the prediction of diabetes.

Table 4. Comparison of the Performance of Deep Learning Model with state-of-art Machine Learning Techniques

Model Type	Training Accuracy (%)	Testing Accuracy (%)	Sensitivity (%)	Specificity (%)	Precision (%)	F1 Score (%)
Logistic Regression	79.38	72.44	65	75	65	57
Random Forest	100	79.53	63	89	76	69
Support Vector Machine	78.82	72.40	49	86	67	57
KNN - 3	85.02	70.47	50	81	57	53
KNN - 4	81.52	70.47	37	88	60	46
KNN - 5	82.10	69.69	50	80	56	53
KNN - 6	75.68	75.59	44	92	73	55
Sequential model	**96.108**	**96.06**	**93**	**98**	**95**	**94**

References

1. JULIO V., SANTIAGO, J. E. DAVIS, FISHER, F. : Hemoglobin A1c Levels in a Diabetes Detection Program. The Journal of Clinical Endocrinology & Metabolism, 47, 3, 578–-580,(1978).

2. Carr, D. B., and Steven, G..: Gestational diabetes: detection, management, and implications. Clinical Diabetes, 16, 1, 4+. (1998).

3. Çalişir, D., Doğantekin,E.: An automatic diabetes diagnosis system based on LDA-Wavelet Support Vector Machine Classifier, Expert Systems with Applications, 38, 7, 8311-8315 (2011).

4. Zou, Q., Qu, K., Luo, Y., Yin, D., Ju, Y., Tang, H. : Predicting diabetes mellitus with machine learning techniques Frontier in Genetics, 9, 515(2018).

5. Tigga N.P., Garg S. : Predicting Type 2 Diabetes Using Logistic Regression. In: Nath V., Mandal J.K. (eds) Proceedings of the Fourth International Conference on Microelectronics, Computing and Communication Systems. Lecture Notes in Electrical Engineering, 673, (2021)

6. Sisodia, D., Sisodia, D. S, : Prediction of Diabetes using Classification Algorithms, Procedia Computer Science, 132, 1578-1585 (2018).

7. Wu, H., Yang, S , Huang Z, He, J. , Wang, X , : Type 2 diabetes mellitus prediction model based on data mining, Informatics in Medicine Unlocked, 10, 100-107(2018).

8. Meng,, X., Huang, Y., Rao, D., Zhang, Q., Liu, Q.: Comparison of three data mining models for predicting diabetes or prediabetes by risk factors, The Kaohsiung Journal of Medical Sciences, 29, 2, 93-99(2013).

9. Choubey, D.K., Paul, S.: GA_RBF NN: a classification system for diabetes. International Journal of Biomedical Engineering and Technology 23, 1, 71-93(2017).

10. Huang, Y., McCullagh, P., Black, N., Harper, R. Feature selection and classification model construction on type 2 diabetic patients' data. Artificial intelligence in medicine 41 , 3, 251-262(2007) .
11. Perveen, S., Shahbaz, M., Keshavjee, K., Guergachi, A : Metabolic Syndrome and Development of Diabetes Mellitus: Predictive Modeling Based on Machine Learning Techniques." IEEE Access 7, 1365-1375. (2019).

12. http://networkrepository.com/pima-indians-diabetes.php
13. https://www.tensorflow.org/guide/intro_to_modules
14. https://keras.io/guides/sequential_model/
15. Duda, R. O., Hart, P. E.,, Stork, D. G. (2001). Pattern Classification. New York: Wiley. ISBN: 978-0-471-05669-0