ISSN 2303-4521
Periodicals of Engineering and Natural Sciences *Original Research*
Vol. 10, No. 6, December 2022, pp.156-167

Enhanced the prediction approach of diabetes using an autoencoder with regularization and deep neural network

Hussein A. Ismael[1], Nabeel H. Al-A'araji[1], Baheja Khudair Shukur[2]
[1] College of Information Technology, University of Babylon, Iraq
[2] College of Computer Science & Information Technology, University of Kerbala, Iraq

ABSTRACT

Diabetes mellitus is considered one of the foremost common and extreme diseases worldwide. A precise and early diagnosis of diabetes is essential to avoid complications and is of crucial importance to the medical care that patients get. To achieve that, we need to develop a model to predict diabetes. There are many prediction models, but they suffer from some problems such as the accuracy of prediction being poor and the time complexity. The prediction process is highly dependent on important features. So, in this paper, we proposed a new model called (CAER-DNN) that depends on an unsupervised technique for generating newly important features and a deep neural network for the prediction process. The unsupervised technique is called complete autoencoder with regularization techniques (CAER) that uses to reconstruct the original features (newly learned features). It is focused too much on training the most important learned features and misses out on less important features. Thus, improving the performance of the prediction process. These important features are used as input to the deep neural network for the prediction of diabetes. Our model is applied to two sets of data including Pima Indian and Mendeley diabetic datasets. Based on the 10-fold cross-validation technique Pima Indian dataset achieves high performance in evaluation measures (f1-score 97.38%, accuracy, recall 97.25%, specificity 97.59%, precision 97.53%,). While the Mendeley diabetes dataset achieved high performance in evaluation measures (f1-score 94.51%, accuracy 98.48, recall 91.74%, accuracy-balance 98.21%, precision 98.21%) based on the holdout technique. compared with other existing machine learning and deep learning techniques our model outperformed existing techniques.

Keywords: Deep learning, Autoencoder, Diabetic Mellitus, Machine Learning, Deep neural network

Corresponding Author:

Hussein A. Ismael
College of Information Technology
University of Babylon
Babil, Iraq
E-mail: husseinyessari@uobabylon.edu.iq, husseinyessari@hotmail.com

1. Introduction

Diabetes Mellitus (DM) is one of the most common diseases that occurs when the production of insulin in the body of a human is or when the body human is not able to utilize the produced insulin appropriately, and as a result, this leads to high blood sugar[1]. The number of people with diabetes increased from 108 million in 1980 to 451 million in 2017 according to the International Diabetes Federation. The danger that a person with diabetes will pose if it is not diagnosed early leads to major complications such as blindness, heart attack, lower limb amputation sometimes even death. Therefore, early diagnosis helps control the disease and choose the best treatments to avoid complications. Thus, as a result of the spread of information technology and data science,

which in turn introduced many effective methods for diagnosing diabetes[2], [3]. With the increment in the power of computational, capacity, and memory computers are having been utilized to perform a wide range of complex assignments with salient exactness[2]. ML techniques permit computers to perform complex assignments. Two fields that will benefit from the application of ML techniques within the field of medicine are prediction and diagnosis. This incorporates a potential for the determination of high danger for emergencies of medical, such as setbacks or any other disease case. ML methods offer enormous potential for enhancing medical research and clinical care[4], [5]. DL is considered one of the most common ML techniques that use a Deep Neural Network (DNN). It can resolve issues that are difficult for conventional artificial intelligence to handle. Deep learning is much suited to handle information or data heterogeneous to discover and acquire knowledge. It currently outperforms compared with traditional ML techniques, becomes more scalable, and the high precision can be investigated by using the huge training dataset or increase the size of neural networks (more hidden layers). Many studies highlight the importance of this technique in understanding complex-data, predicting medical images, classifying texts, and others[6], [7].

Diabetes prediction tasks have faced some challenges, including the approaches used to predict diabetes must be effective and give high performance and accuracy, the reason is that the decision is related to the human health condition, so requires high accuracy. And time complexity. This paper is used unsupervised and deep learning techniques to enhance the accuracy and performance of our model. And uses the early stopping technique and batch normalization layer in our model to reduce the complexity time of the training process.
The following are the major contributions of the study:

- Develop a new model based on complete autoencoding with regularization for a fine-tuning and deep neural network for the prediction process.
- The performance of diabetes prediction is improved. Our model achieved the highest f1-score up to (98.90%), (and 98.48%) for the Pima Indian dataset and the Mendeley diabetes dataset respectively.

The main structure of the paper includes the following: after the introduction in section 1, section 2 shows the related works. In section 3 the methodology that includes datasets, preprocessing, autoencoder, tuning of hyperparameters, and DNN classifier. Section 4 presents the experimental study and section 5 shows the results and discussions. Finally, the conclusion section.

2. Related works

Many research projects use artificial intelligence (AI) techniques to predict diabetes. It can be summarized as follow (i) constructing and developing approaches based on ML techniques, (ii) constructing and developing approaches based on DL techniques.
The study of Ebru and Tuncay [8] proposed new approaches for predicting diabetes. First, (ANN-GA) approach depends on a genetic algorithm and Artificial Neural Networks (ANN). The second, (CART-GA) approach is based on classification and regression tree. This study used GA to enhance accuracy and CART-GA gives high accuracy. Authors Mani et al. [1] suggested a new model for diagnosing diabetes. In this work, there are two phases. First, used an ensemble model that depends on two classifiers Decision Tree (DT), and Logistic Regression (LR), which trains independently. Second, using ANN classifier that trains the output of the first phase to enhance the final decision. It achieved 83% accuracy. While research by Joy and Ledisi [9] developed a model to predict the type of diabetes (type I or type II) based on the ANN classifier. In this study, the ANN consists of an input layer with 18 neurons that represent the symptoms of diabetes, one hidden layer with two neurons, and an output layer with two outputs representing the type of diabetes. They used a private dataset consisting of 100 samples collected from diabetes patients in a survey. Safial and Milon [10] proposed a strategy that depends on the DNN model for predicting of diabetes. In this study, DNN consists of four hidden layers with (12,16,16 and 14) neurons respectively. They used 10-fold and 5-fold cross-validation techniques to evaluate the model. Victor et al. [11] applied a new E-Diagnosis system for predicting type II diabetes. It relies on ML techniques that are implemented on the internet of things for the medical environment. This study, used

Random Forest (RF), Naïve Bayes (NB), and J48 Decision Tree (J48DT) as classifiers to train and test the Pima Indian dataset (PID). Then choose the best one in terms of accuracy and performance. They prove the NB classifier with feature selection achieved high performance. While RF achieved high with more features. Huma and Sachin [12] compared the performance of ML techniques and Deep Learning for the diagnosis of diabetes. In this study, use DNN, ANN, NB, and Decision Tree (DT). The study proved the DNN outperformed the other techniques. Himanshu et al. [13] proposed two techniques include DL and Quantum Machine Learning (QML). They used MLP to train and test a dataset that consists of five hidden layers with (16, 32, 8, and 2) neurons respectively. QML used the advantage of quantum computers to solve the prediction diabetes. It has been demonstrated to be capable of efficiently resolving these issues with current technology by calculating several states concurrently. The authors proved the DL techniques achieved high performance as compared with QML. Maher et al. [14] suggest an approach to the diagnosis of diabetes relies on Adaptive Neuro-Fuzzy Inference System (ANFIS). This approach consists of several phases including, preprocessing phase, classification phase, and evaluation. The first phase includes normalization, outlier detection, and imputation. Second. Apply ANFIS classifier to predict the disease. Finally, evaluate the model. The authors proved the model achieved the highest performance in accuracy.

3. Methodology

The methodology includes several phases as follows: (i) datasets, (ii) pre-processing, (iii) complete autoencoder with regularization techniques, and (iv) deep neural network classifier. Figure 1 shows the diagram of the proposed model.

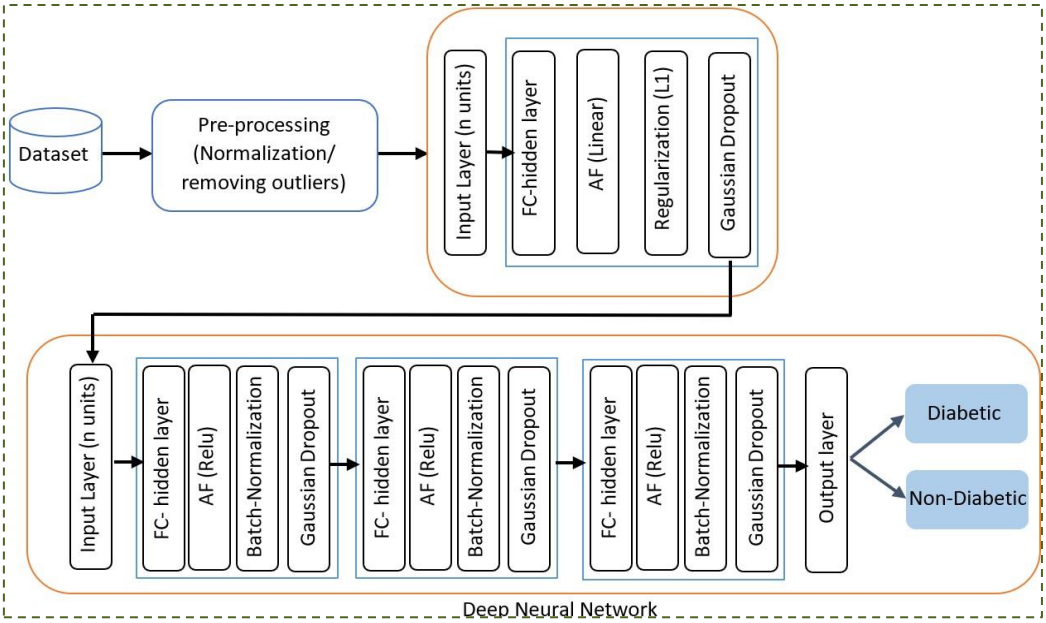

Figure 1. Diagram of the proposed model

3.1. Datasets

In this section, the classification as a problem related to diabetes was posed for patients who provided some medical information. Two datasets are used in this study. The first dataset in [15] is titled the Pima Indian dataset (PID) from the Machine learning Repository dataset (UCI). It is considered one of the most important datasets for diabetics. It has a sample of 768 females with ages not less than 21 years old, where 500 samples belong to females who do not suffer from the disease and 268 samples suffer from diabetes. Each item of the dataset has 8 features as input and one class label as output. Table 1 briefly shows the details of the PID features. Second, the dataset in [16] Madeley diabetes dataset (MDD) that collects from the society of Iraq. It contains 1000 samples with three classes covered includes (diabetes patients, healthy, and predicted diabetes patients). The Medical City Hospital laboratory and the Specializes Center for Endocrinology and Diabetes-Al-Kindy

Teaching Hospital provided the data (MCHL). It has laboratory testing and medical data. The dataset features include (Blood Sugar Levels (BSL), Ratio of Creatinine (Cr), Age, sex, Body Mass Index (BMI), cholesterol are all factors (Chol), and lipid status during fasting, which includes VLDL, LDL, HDL, TG, and an HBA1C. Table 2 briefly shows the details of MDD features.

Table 1. Descript features of PID

Features	Description	Type
Pregnancies	Pregnancies Number with a range (of 0-17)	Numeric
Skin Thickness	The thickness of triceps skin-fold in (mm) with a range (of 0-99)	Numeric
Glucose	the two-hour plasma glucose level in an oral glucose tolerance test with a range (of 0–199)	Numeric
Diastolic Blood pressure	Diastolic blood pressure gauges how much pressure is present in the arteries between heartbeats when the heart is rested. with a range (of 0-122)	Numeric
BMI	Index of mass of the body (weight in kg/ power (Hight in m,2)) with a range (of 0-67.1)	Numeric
Serum Insulin	two-hours serum insulin in (mu U/ml) with a range (of 0-846)	Numeric
Diabetes pedigree Function	An engaging feature that helps diagnose the diabetic with a range (of 0.078–2.42)	Numeric
Age	Patient age with a range (of 21-81)	Numeric

Table 2. Descript attributes of MDD

Features	Description	Type
Urea	Urea in mg/dl, with a range (of 0.5- 38.9)	Numeric
Gender	Male or Female	Categorical
Age	Patient age with a range (of 20- 79)	Numeric
CR	In mol/l with a range (of 48- 80)	Numeric
BMI	Index of mass of the body (weight in kg/ power (Hight in m,2)) with a range (of 19-47)	Numeric
LDL	In mmol/l with a range (of 0.3-9.9)	Numeric
VLDL	In mmol/l with a range (of 0.1- 35)	Numeric
HDL	In mmol/l with a range (of 0.2- 9.9)	Numeric
Chol	In mmol/l with a range (of 0.0- 10.3)	Numeric
TG	In mmol/l with a range (of 0.3-13.8)	Numeric
HBA1C	In mmol/l with a range (of 0.9- 16)	Numeric

3.2. Preprocessing

It is one of the most important phases that help improve the performance of the classifier. In this phase, two techniques will be applied, including the following:

3.2.1. Data-Normalization

It is one of the most important preprocessing techniques that is applied to the numerical attributes that precede the process of classification or clustering. The objective of normalization is to convert the numerical attribute values to a consistent scale without deform variations in their value ranges. It is applied when the attribute values are not in the same range[17], [18]. In this paper, applied Min-Max Normalization (MMN) that makes attribute values in the range [0,1] or [-1, 1]. Equation (1) shows the MMN as follows [14]:

$$MMN(v) = \frac{v - min\,(v)}{max(v) - min\,(v)} \quad (1)$$

Where (v) refers to the value of the given attribute, min(v) and max(v) refer to the minimum and the maximum value in the given attribute respectively.

3.2.2. Removing-outliers

In data analysis, removing outliers is an important part. They give unimportant statistical results, so understanding or even eliminating outliers helps improve model performance[19]. In this study, Standard Deviation Method (SDM) is applied as a statistical method to determine the outliers. by calculating the lower and upper boundaries of distribution through, taking $\lambda = 3$ standard deviations from the mean of the data as follows: [17].

- Calculate the mean ($\bar{x}$) as in (2), SD as in (3) of the data.

$$\bar{x} = \frac{\sum x}{n} \quad (2)$$

$$\sigma = \sqrt{\frac{\sum (x - \bar{x})^2}{n}} \quad (3)$$

- Calculate the upper (up) as in (4), lower (lp) as in (5) boundaries:

$$lp = \bar{x} - \lambda\sigma \quad (4)$$

$$up = \bar{x} + \lambda\sigma \quad (5)$$

- Identify outliers (ol) as in (6).

$$ol = x \text{ if } (x > up \text{ or } x < lp) \quad (6)$$

After that, it will be removed ol from the data

3.3. Complete autoencoder with regularization technique (CAER)

Autoencoder is considered one of the most important unsupervised techniques, meaning that no labels are present. It is a more difficult problem for the agent of Artificial Intelligent (AI) since it is less well-specified than the supervised learning problem. However, if dealt with properly, it becomes more powerful. It contributes to the improvement of supervised techniques through the pretraining process that allowed the production of a good representation of the original data, thus it helps the portion supervised to solve the specific task in the best condition. Recently, there are different types of autoencoders such as Denoising autoencoder, sparse autoencoder, variational autoencoder, complete autoencoder, and under-complete autoencoder [20], [21].

In this study, the complete autoencoder is used. Where the number of neurons in the input layer is equal to the number of neurons in the encoder layer. CAER consists of three layers including an input layer, a hidden layer (encoder layer), and an output layer (decoder layer). The encoder and encoder functions are unbounded linear activation functions for both the hidden and an output layer. The encoder function is applied to reconstruct the original data and generate newly learned observations. The decoder function is applied to reconstruct the newly learned observations to the original format.

Regularization techniques are used to improve the performance of CAER, avoid overfitting problems and get more important features. Where it uses two types of regularization, including Gaussian dropout and L1-regularization. Gaussian dropout is used to select random neurons from the hidden layer for training with a probability of up to (30%). L1-regularization is applied by adding a penalty term to the loss function. So, the weight parameters will be reduced by pushing the weights toward zero[22]. Other hyperparameters applied in the autoencoder are the number of epochs equal to (100), and the batch-size is 32. Table 3 shows the architecture of the CAER fine-tuning. Figure 2 shows the diagram of the autoencoder (CAER).

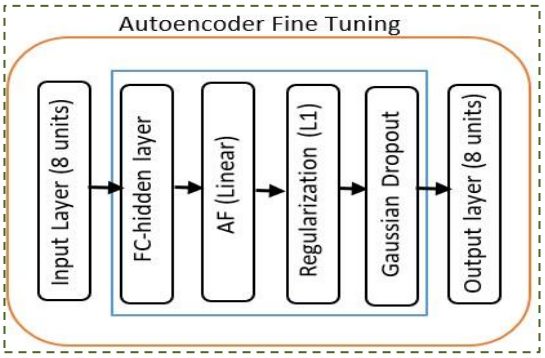

Figure 2. The CAER diagram

Table 3. Descript an architecture of a complete autoencoder model with layers no. neurons in each layer and no. of parameters for two datasets

Layer name	Output shape	Activation function	No. parameters
Input layer	(None, n)	None	0
Dense layer	(None, n)	linear	(n× n) + n
Gaussian dropout layer	(None, n)	linear	0
Output layer	(None, n)	linear	(n× n) + n

Where (n) refers to the number of attributes for any dataset.

3.4. Deep Neural Network (DNN) classifier

DL is a popular technique in machine learning which uses deep neural networks. It is multiplied layers neural network that contains two or more hidden-layer[6]. In this phase, DNN will be used as a classifier that contains several layers. Table 4 shows the architecture of the DNN classifier. As briefly can be classified the layers of DNN are as follow:

Input layer: this layer is used to specify an input value corresponding to the number of attributes, where the PID dataset has (8) attributes and the MDD dataset has (11) attributes. The data in this layer has been pre-processed by applying normalization and outliers' removal techniques. Note that the preprocessing process is done on the data before passing it to the input layer to improve performance.

Dense layer: In this classifier, each neuron in the dense layer receives the output from all the neurons in the layer before it. In this classifier used three dense layers, and each dense layer has 2048 neurons.

Relu Activation Function (Relu): It is a nonlinear activation function that comes after each dense layer in this classifier. Equation (7) is performed by calculating and maximum function, as follows: [20]

$$Relu(v) = Max\ (0, v), \text{ where v is the input vector} \quad (7)$$

Batch-Normalization layer (BNL): In this classifier, apply BNL after the activation function with a dense layer. it accelerates the training and lowers the generalization error[21].

Gaussian dropout layer (GDL): Apply GDL to select random neurons based on gaussian at each dense layer to reduce the overfitting issue [20]. It used three CDL with a probability of up to (30%).

Sigmoid activation function (Sigmoid): It is nonlinear AF; it looks like S-shape values range between 1 and 0. In this classifier, is used to predict the binary classes. Equation (8) is referred to this function as follows: [16]

$$Sigmoid(v)=\frac{1}{1+e^{-v}} \quad (8)$$

SoftMax activation function (SoftmaxAF): this classifier is used it to predict the multiply-class. It is nonlinear AF. Equation (9) is referred to this function as follows:[23]

$$SoftmaxAF(v_j)=\frac{e^{v_j}}{\sum_{i=1}^{M} e^{v_i}} \quad (9)$$

Where, v represents in-vector. e^{v_j} represents function of standard exponential for in-vector. M represents the number of classes. e^{v_i} represents function of standard exponential for out-vector.

3.4.1. Tuning hyperparameters for DNN Classifier

Hyperparameters are parameters that assign to certain values before begins the training process. Unlike ML, the DL is full of hyperparameters. In DL, the optimization of hyperparameters has a significant effect on improving the performance of the model. The GridSearchCV method is used to tune our DNN model. It is a class in the Keras library in python language to tune hyperparameters. Which specific the number of hidden layers and neurons at each hidden layer (i.e., determine the structure of the DNN classifier). The Learning Rate (LR) determines how the DNN classifier is trained. The tuning value of LR is equal to (0.001). Batch Size (BS) is equal to (32), which refers to the number of the sub-trainings dataset that will pass to the DNN classifier and then update the parameters (weights and biases). Adam optimizer optimizes the parameters of the DNN classifier. The gaussian dropout ratio is a simple and effective regularization method. The tuning value of the gaussian dropout ratio is (0.3). It used Binary Cross Entropy (BCE) loss function for PID and Categorical Cross-Entropy (CCE) loss function for the MDD dataset.

It is very possible to increase the training time and have overfitting when the number of epochs for the training dataset is large. To avoid this, the Early-Stop method is used to tune the number of epochs. Where the training process of the DNN classifier will terminate stop as soon as the performance of the model is stopped enhancing using holdout validation.

Table 4. Descript an architecture of the DNN model with layers no. neurons in each layer and no. of parameters for PID

Layer name	Output shape	Activation function	No. parameters
Input layer	(None, 8)	None	0
Dense layer	(None, 2048)	Relu	18432
Batch-normalization layer	(None, 2048)	Relu	8192
Gaussian dropout layer	(None, 2048)	Relu	0
Dense layer	(None, 2048)	Relu	4196352
Batch-normalization layer	(None, 2048)	Relu	8192
Gaussian dropout layer	(None, 2048)	Relu	0
Dense layer	(None, 2048)	Relu	4196352
Batch-normalization layer	(None, 2048)	Relu	8192
Gaussian dropout layer	(None, 2048)	Relu	0
Output layer	(None, 1)	sigmoid	2049

4. Experimental Study

In this study, an 8 GB NVIDIA GeForce RTX 3070Ti GPU and 16 GB DDR5 RAM system with Keras Tensorflow 2.4.0 was used for the implementation. 10-fold cross-validation has been used as a performance measure to evaluate our proposed model. The metrics used to determine the performance model include accuracy as in (14), recall as in (10), precision as in (12), specificity as in (11), and f1-score as in (13). The following equations refer to these metrics [24]:

$$Recall = \frac{TP}{TP+FN} \quad (10)$$

$$Specificity = \frac{TN}{FP+TN} \quad (11)$$

$$Precision = \frac{TP}{TP+FP} \quad (12)$$

$$F1\text{-}Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (13)$$

$$Accuracy = \frac{TP+FN}{TP+TN+FP+FN} \quad (14)$$

Given the above equations, when an item is predicted to belong to a class and really does, this is known as a True-Positive (TP). When an object is expected to not belong to a class and really does not, it is said to be True-Negative (TN). When an item is assumed to belong to a class when, in fact, it does not, this is known as a false positive (FP). When an object is predicted to not belong to a class when it actually does, this is known as a false negative (FN)[24].

5. Results and discussion

In this study, results were carried out by utilizing the CAER-DNN model. The performance of our model with two datasets including MDD and PID is presented in Table 5. It has been noticed our model with a PID dataset achieved high performance compared with an MDD dataset in terms of all evaluation measures. In addition, our model outperformed existing models in terms of f1-score 10-fold cross-validation. Table 6 shows how our model with PID dataset outperforms all other existing models in terms of performance measures. The proposed model is evaluated against the most recent diabetes prediction models include (deep neural network[10], CART-GA, ANN-GA[8], Feedforward neural network [9], Artificial Neural Network+ Logistic Regression+DecisionTree[1], Random forest, J48 decision tree, Naïve Bayes, Naïve Bayes with feature selection three factors, and Naïve Bayes with feature selection five factors [25], DL, QML[13], DL [12], and ANFIS [14]). In terms of performance metrics, it showed that the suggested model performed better than these models. High-performance scores were attended in accuracy, specificity, precession, recall, and f1-score (98.90%, 99.00%, 98.14%, 98.66%, and 98.39%) respectively, as shown in Figure 3. The model deep neural network in [10] achieved low scores in accuracy, specificity, recall, and f1-score (97.27%, 96.27%, 97.80%, and 98.0%) respectively compared with our model. The CART-GA and ANN-GA models in [8] achieved low scores in accuracy, specificity, precision, recall, and f1-score (93.42%, 88.89%, 94.00%, 95.92%, and 94.95%) and (81.82%, 69.70%, 80.00%, 90.91%, and 85.11%) respectively compared with our model. The model in [1] achieved low scores in accuracy, (83.08%) compared with our model. All models suggests in [25] achieved low scores in accuracy, specificity, precision, recall, and f1-score compared with our model. The model DL in [12] also, achieved low scores in accuracy, precision, recall, and f1-score (98.07%, 95.22%, 98.46%, and 96.81%) respectively. Table 7 shows how the proposed model with PID dataset outperforms all other existing models in terms of performance measures. Our model outperformed all models, which included Multinomial logistic Regression, Decision Tree, Random Forest, Stochastic, gradient Boosting, and Naïve Bayes [26] as shown in Figure 4. The performance of the proposed model outperformed these models in all evaluation measures including accuracy, accuracy Balance, recall, and f1-score except precision measure (98.85%) with the Stochastic gradient Boosting model that achieved a high score compared with our model. Multinomial logistic Regression achieved low scores with accuracy, accuracy Balance, precision, recall, and f1-score (86.70%, 78.10%, 70.00%, 70.00%, and 70.00%) respectively. Decision Tree achieved low scores with accuracy, accuracy Balance, precision, recall, and f1-score (95.07%, 82.58%, 98.12%, 74.62%, and .67%) respectively. Random Forest achieved low scores with accuracy, accuracy Balance, precision, recall, and f1-score (90.64%, 84.45%, 75.00%, 78.00%, and 76.40%) respectively. Naïve Bayes achieved low scores with accuracy, accuracy Balance, precision, recall, and f1-score (90.64%, 84.45%, 75.00%, 78.00%, and 76.40%) respectively.

Additionally, the model has excellent generalization, dependability, and unbiasing performance because of the adoption of the 10-fold cross-validation procedure.

Table 5. The model with different datasets using 10-fold cross-validation

Model with dataset	Accuracy	Specificity	Precision	Recall	F1-score
CAER-DNN with MDD	96.24%	-	91.65%	91.46%	**90.96%**
CAER-DNN with PID	98.90%	99.00%	98.14%	98.66%	**98.39%**

Table 6. Compression of the model with others (PID dataset)

Model with dataset	Accuracy	Specificity	Precision	Recall	F1-score
deep neural network[10]	97.27%	96.27%	-	97.80%	98.0%
CART-GA[8]	93.42%	88.89%	94.00%	95.92%	94.95%
ANN-GA[8]	81.82%	69.70%	80.00%	90.91%	85.11%
Artificial Neural Networks [9]	90.00%	-	-	-	-
Artificial Neural Network + Logistic Regression + Decision Tree[1]	83.08%	-	-	-	-
Random forest[25]	79.57%	75.00%	89.40%	81.33%	85.17%
J48 decision tree[25]	74.78%	59.63%	70.86%	88.43%	78.68%
Naïve Bayes[25]	78.67%	63.29%	81.88%	86.75%	84.24%
Naïve Bayes with feature selection three factors [25]	79.13%	62.03%	81.60%	88.08%	84.71%
Naïve Bayes with feature selection five factors [25]	77.83%	62.03%	81.25%	86.09%	83.60%
QML[13]	86.00%	86.00%	74.00%	85.00%	79.00%
DL[13]	95.00%	95.00%	90.00%	95.00%	93.00%
DL[12]	98.07%	-	95.22%	98.46%	96.81%
ANFIS[14]	92.00%	84.00%	-	97.00%	-
Our model CAER-DNN	**98.90%**	**99.00%**	**98.14%**	**98.66%**	**98.39%**

Table 7. Compression of the model with others (PID dataset)

Model with dataset	Accuracy	Accuracy Balance[13]	Precision	Recall	F1-score
Multinomial logistic Regression[26]	86.70%	78.10%	70.00%	70.00%	70.00%
Decision Tree[26]	95.07%	82.58%	98.12%	74.62%	89.67%
Random Forest[26]	90.64%	84.45%	75.00%	78.00%	76.40%
Stochastic gradient Boosting[26]	97.04%	88.00%	**98.85%**	81.00%	89.00%
Naïve Bayes[26]	93.10%	80.40%	89.00%	71.86%	79.50%
Our model CAER-DNN	**98.48%**	**98.21%**	**98.21%**	**91.74%**	**94.51%**

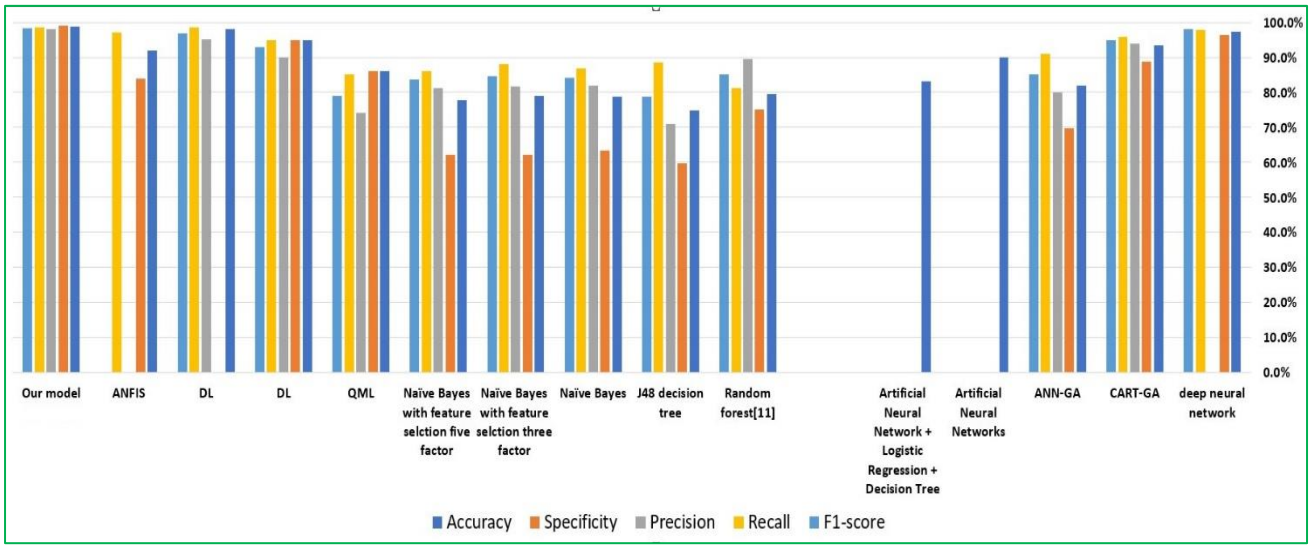

Figure 3. Comparison of our results to others by evaluation of performance measurements with (PID)

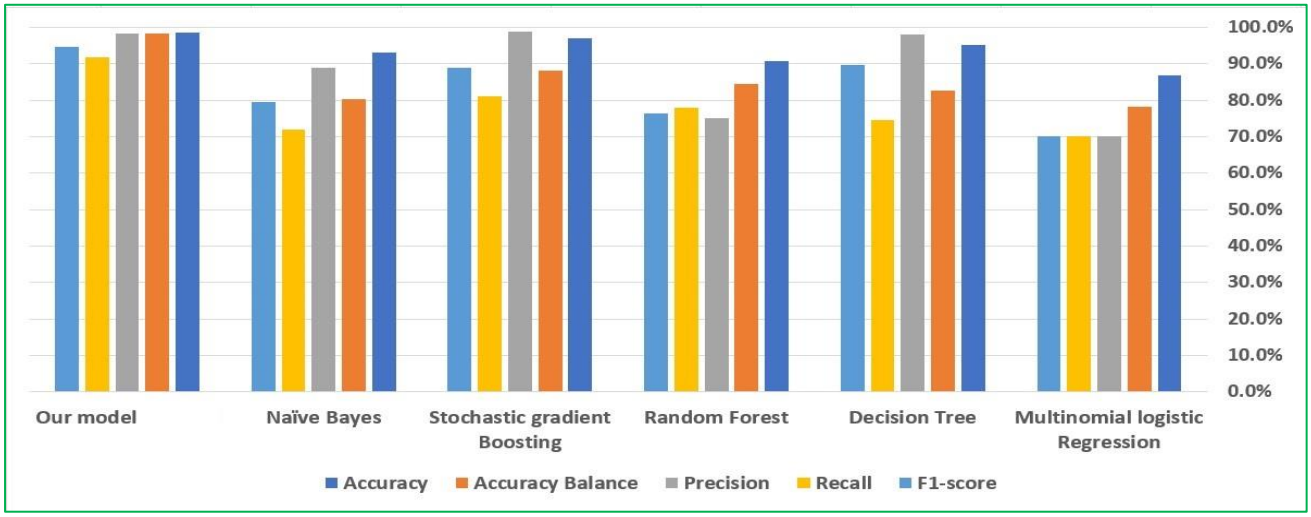

Figure 4. Comparison of our results to others by evaluation of performance measurements with (MDD)

6. Conclusions

By means of this work, a new CAER-DNN model is proposed. which is specified by the complete autoencoder with regularization technique to generate newly important features and new DNN architecture for prediction. Two datasets have been applied to our proposed model. In the first dataset (PID), our model achieved positive results in the prediction of diabetic patients. Moreover, the empirical study findings showed an improved performance of our model for predicting diabetes patients compared to other models that have been published before. It has been demonstrated that our model yields a high accuracy metric with an f1-score metric of up to 98.90% and 98.39% respectively. In the second dataset (MDD), our model also achieved high accuracy matric and f1-score metric of up to 98.48% and 94.51% respectively. This is due to the use the complete autoencoder with a regularization technique that produced good features with the DNN classifier. Finally, the advantage of these techniques has fared better than any competing, earlier approaches.

Declaration of competing interest

The authors declare that they have no known financial or non-financial competing interests in any material discussed in this paper.

Funding information

No funding was received from any financial organization to conduct this research.

References

[1] M. Abedini, A. Bijari, and T. Banirostam, "Classification of Pima Indian Diabetes Dataset using Ensemble of Decision Tree , Logistic Regression and Neural Network," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 9, no. 7, pp. 7–10, 2020.

[2] J. Chaki, S. Thillai Ganesh, S. K. Cidham, and S. Ananda Theertan, "Machine learning and artificial intelligence based Diabetes Mellitus detection and self-management: A systematic review," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 6, pp. 3204–3225, 2020.

[3] M. Alehegn, "Analysis and Prediction of Diabetes Mellitus using Machine Learning Algorithm," *Int. J. Pure Appl. Math.*, vol. 118, no. 9, pp. 871–878, 2018.

[4] V. Sankaravadivel and S. Thalavaipillai, "Symptoms based endometriosis prediction using machine learning," *Bull. Electr. Eng. Informatics*, vol. 10, no. 6, pp. 3102–3109, 2021.

[5] J. A. M. Sidey-Gibbons and C. J. Sidey-Gibbons, "Machine learning in medicine: a practical introduction," *BMC Med. Res. Methodol.*, vol. 19, no. 1, pp. 1–18, 2019.

[6] M. Bakator, "Deep Learning and Medical Diagnosis : A Review of Literature," *Multimodal Technol. Interact.*, vol. 2, no. 3, p. 47, 2018.

[7] Z. Ebrahimi, M. Loni, M. Daneshtalab, and A. Gharehbaghi, "A Review on Deep Learning Methods for ECG Arrhythmia Classification," *Expert Syst. with Appl. X*, vol. 7, p. 100033, 2020.

[8] E. Pekel Özmen and T. Özcan, "Diagnosis of diabetes mellitus using artificial neural network and classification and regression tree optimized with genetic algorithm," *J. Forecast.*, vol. 39, no. 4, pp. 661–670, 2020.

[9] J. O. Orukwo and L. G. Kabari, "Diagnosing Diabetes Using Artificial Neural Networks," *Eur. J. Eng. Res. Sci.*, vol. 5, no. 2, pp. 221–224, 2020.

[10] S. Islam Ayon and M. Milon Islam, "Diabetes Prediction: A Deep Learning Approach," *Int. J. Inf. Eng. Electron. Bus.*, vol. 11, no. 2, pp. 21–27, 2019.

[11] V. Chang, J. Bailey, Q. A. Xu, and Z. Sun, "Pima Indians diabetes mellitus classification based on machine learning (ML) algorithms," *Neural Comput. Appl.*, pp. 1–17, 2022.

[12] H. Naz and S. Ahuja, "Deep learning approach for diabetes prediction using PIMA Indian dataset," *J. Diabetes Metab. Disord.*, vol. 19, no. 1, pp. 391–403, 2020.

[13] H. Gupta, H. Varshney, T. K. Sharma, N. Pachauri, and O. P. Verma, "Comparative performance analysis of quantum machine learning with deep learning for diabetes prediction," *Complex Intell. Syst.*, vol. 8, no. 4, pp. 3073–3087, 2022.

[14] M. T. Alasaady, T. Noranis, M. Aris, N. M. Sharef, and H. Hamdan, "A proposed approach for diabetes diagnosis using neuro-fuzzy technique," vol. 11, no. 6, pp. 3590–3597, 2022.

[15] Dua D and Karra Taniskidou, "UCI Machine Learning Repository," *University of California, Irvine, School of Information and Computer Sciences*, 2017.

[16] A. Rashid, "Diabetes Dataset," vol. 1, 2020.

[17] C. Fan, M. Chen, X. Wang, J. Wang, and B. Huang, "A Review on Data Preprocessing Techniques Toward Ef fi cient and Reliable Knowledge Discovery From Building Operational Data," vol. 9, no. March, pp. 1–17, 2021.

[18] A. Performance, E. Alshdaifat, D. Alshdaifat, A. Alsarhan, F. Hussein, and S. Moh, "The Effect of Preprocessing Techniques, Applied to Numeric Features, on Classification Algorithms' Performance," *Data*, vol. 6, no. 2, p. 11, 2021.

[19] I. F. Ilyas and X. Chu, *Data Cleaning*, First Edit. ACM Books, 2019.

[20] A. M. Alhassan and W. M. N. W. Zainon, "Brain tumor classification in magnetic resonance image using hard swish-based RELU activation function-convolutional neural network," *Neural Comput. Appl.*, vol. 33, no. 15, pp. 9075–9087, 2021.

[21] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *International conference on machine learning*, Jun. 2015, pp. 448–456.

[22] A. G. Khachaturyan and G. A. Shatalov, "Dropout: A Simple Way to Prevent Neural Networks from Overfittin," *J. Mach. Learn. Res.*, vol. 31, no. 1, pp. 1929–1958, 2014.

[23] I. Kouretas and V. Paliouras, "Simplified Hardware Implementation of the Softmax Activation

Function," in *2019 8th International Conference on Modern Circuits and Systems Technologies (MOCAST)*, May. 2019, pp. 1–4.

[24] H. A. Ismael, N. H. Al-A'araji, and B. K. Shukur, "An Enhanced Diabetic Foot Ulcer Classification Approach Using GLCM and Deep Convolution Neural Network An Enhanced Diabetic Foot Ulcer Classification Approach Using GLCM and Deep Convolution Neural Network," *Karbala Int. J. Mod. Sci.*, vol. 8, no. 4, pp. 1–10, 2022.

[25] V. Chang, J. Bailey, Q. A. Xu, and Z. Sun, "Pima Indians diabetes mellitus classification based on machine learning (ML) algorithms," *Neural Comput. Appl.*, vol. 0123456789,pp. 1-17, 2022.

[26] M. R. Rajput and S. S. Khedgikar, "Diabetes prediction and analysis using medical attributes: A Machine learning approach," *J. Xi'an Univ. Archit. Technol.*, vol. 14, no. 1, pp. 98–103, 2022.