DIAGNOSTYKA, 2024, Vol. 25, No. 1

e-ISSN 2449-5220
DOI: 10.29354/diag/183633

NOVEL DIABETES CLASSIFICATION APPROACH BASED ON CNN-LSTM: ENHANCED PERFORMANCE AND ACCURACY

Yassine AYAT *, **Wiame BENZEKRI**, **Ali EL MOUSSATI**, **Ismail MIR**, **Mohammed BENZAOUIA**, **Abdelaziz EL AOUNI**

Energy, Embedded System, and Data Processing Laboratory, National School of Applied Sciences Oujda (ENSAO), Mohammed First University (UMP), Oujda, 60000, Morocco
* Corresponding author, e-mail: yassine.ayat@ump.ac.ma

Abstract

This paper deals with the development of an approach for diabetes classification harnessing Convolutional-Neural-network (CNN) and a Long-Short-Term-Memory (LSTM) model. The proposed method harnesses the strengths of LSTM and CNN architectures to effectively capture sequential patterns and extract meaningful features from the input data. A comprehensive dataset containing relevant features for diabetes patients is used to train and evaluate the classifiers. Evaluation metrics such as kappa score, F1-score, accuracy, precision, and recall are employed in ordre to assess the performance of each model. The results demonstrate that the CNN-LSTM model outperforms other models, including Logistic Regression, Random Forest, SVM, and KNN, achieving an impressive accuracy of 97%. These findings shed light on the effectiveness of the proposed approach in accurately classifying diabetes, resulting in significant advancement in diabetes diagnosis and treatment and opening up exciting possibilities for personalized healthcare.

Keywords: Diabetes, diabetes classification, dataset balancing, combined model, personalized healthcare.

List of Symbols/Acronyms

AB - AdaBoost
ACC – Accuracy
AMMLP- Artificial Metaplasticity On Multilayer Perceptron
AUC - Area Under the Curve
BMI - Body Mass Index
BP – Blood Pressure
DPF – Diabetes-Pedigree-Function
DT - Decision Tree
ELM - Extreme Learning Machine
FM - Fowlkes-Mallows Index
GPC - Gaussian Process Classification
G – Glucose
HPM - Hybrid prediction model
J48 - C4.5 Decision Tree
KNN - k-Nearest Neighbors
LDA - Linear Discriminant Analysis
MCC - Matthews Correlation Coefficient
MLP - Multilayer Perceptron
NB - Naive Bayes
NLR - Negative Likelihood Ratio
NPV - Negative Predictive Value
NN - Neural Network
PLR - Positive Likelihood Ratio
PPV - Positive Predictive Value
PDD- Pima Indian Diabetes Dataset
Pr - Pregnancies
QDA - Quadratic Discriminant Analysis
RF - Random Forest
SVM - Support Vector Machine
ST – Skin Thickness
Sp - Specificity (True Negative Rate)
XB - XGBoost
ENRC Egyptian National Research Centre

Received 2023-07-30; Accepted 2024-02-02; Available online 2024-02-11

1. INTRODUCTION

Diabetes is a chronic disease characterized by inadequate insulin production or utilization, posing a significant global health challenge. Its prevalence has been steadily increasing over the years, reaching alarming numbers. In 2021, the worldwide diabetic population reached a staggering 537 million, with projections estimating 643 million by 2030 and a concerning 784 million by 2045. This rise in cases has profound implications for public health, mortality rates, and healthcare systems worldwide [1].

In response to the complexities of diabetes management, there has been an increasing focus on utilizing advanced technologies and data-driven approaches. This interest has led to the exploration of machine learning algorithms and neural network models, which offer potential benefits in enhancing the accuracy and efficiency of diabetes diagnosis and treatment. These techniques have the ability to analyze extensive and complex datasets, extract significant patterns and insights, and create personalized systems for classifying and predicting diabetes [2-3].

Utilizing machine learning for the purpose of enhancing diabetes management holds promise for early diagnosis, risk prediction, treatment optimization, and the development of personalized interventions [4-6]. By continuously improving classification performance and leveraging the ongoing advancements in machine learning, researchers aim to enhance diabetes care and contribute to the global fight against this chronic disease.

Machine learning methodologies have captured the attention of varied research, investigating their application in diabetes classification and prediction. Polat and Güneş [7] suggested a sequential learning approach that merges Least Square-Support Vector Machine (LS-SVM) with Generalized Discriminant Analysis (GDA). They achieved an accuracy of 78.21% using LS-SVM and reported a classification accuracy of 79.16% using both GDA and LS-SVM with 10-fold cross-validation. Another study by same author [8] that integrates Principal Component Analysis (PCA) and the Adaptive Neuro-Fuzzy Inference System (ANFIS), the attained classification accuracy reaches 89.47%. Kannadasan et al. [9] developed a general regression neural network for diabetes diagnosis, attaining a classification accuracy of 80.21%. Additionally, a multilayer neural network-based approach achieved a classification accuracy of 77.08% [11]. Caliskan et al. [10] proposed a training strategy for a deep neural network classifier using the L-BFGS algorithm and evaluated it with various datasets, including the Pima Indian diabetes dataset. They reported a classification accuracy of 77.09% for the Pima Indian diabetes dataset.

Zhu et al. [12] introduce a data mining-driven model designed for the early detection and forecasting of diabetes. The model integrated PCA, k-means clustering, and logistic regression algorithms to improve both clustering and classification accuracy. In comparison to prior research, the experimental findings showed that integrating PCA improved the effectiveness of the k-means clustering algorithm, resulting in 25 more accurately categorized cases. Furthermore, the logistic regression classifier achieved a higher accuracy of 1.98% using the proposed model. Mercaldo et al. [13] presented a strategy for classifying diabetic patients based on WHO criteria, utilizing state-of-the-art machine learning algorithms. They evaluated real-world data and trained the model using six different classification approaches. The Hoeffding Tree method achieved a precision of 0.770 and a recall of 0.775 using the PIMA Indian community dataset in Phoenix, Arizona. Qawqzeh et al. [14] conducted a study on the classification of diabetes data using logistic regression. They employed a training dataset with 459 patients and a testing dataset with 128 patients, reporting a classification accuracy of 92% using logistic regression. However, a limitation of their study was the absence of a comparison with other diabetic prediction models, leading to a lack of validation for their proposed model. Tafa et al. [15] developed a diabetes prediction model through the integration of naïve Bayes and support vector machine algorithms. They divided the dataset into a 50% training set and a 50% testing set, achieving an accuracy of 97.6% with their ensemble model. However, the authors did not mention any preprocessing techniques for data filtering.

Hussain and Naaz [16] conducted a review comparing the accuracy of various machine learning techniques, such as random forest, Naïve Bayes, and neural networks. They evaluated these algorithms using the Matthews correlation coefficient.

In addition, Table 1 highlights a significant drawback that is consistently observed across all methods: the low classification performance. This finding emphasizes the need for further advancements in classification techniques to address this challenge effectively.

Within this research article, a new methodology is presented that combines Convolutional Neural Network (CNN) with Long Short-Term Memory (LSTM) models is developed and tested for the classification of the PIMA dataset. The classification performance of the proposed method is compared with other approaches, including Logistic Regression, Random Forest, Support Vector Machine (SVM), and k-Nearest Neighbors (KNN).

This paper presents the methodology and findings of our study. Section 2 presents the methodology of our study, detailing the dataset, preprocessing techniques, used algorithms, and evaluation metrics. In Section 3, the focus is on presenting and analyzing the results obtained, and a discussion of the findings. In closing, Section 5 summarizes the main results of the paper.

2. METHODOLOGY

The methodology employed in this research, as illustrated in Fig. 3., it involves several steps for analyzing the Pima Indian diabetes dataset. Preprocessing and balancing techniques have been applied to ensure data quality, including addressing missing values, outliers, and feature engineering. The dataset was divided into distinct training and testing subsets, facilitating the process of both training the model and conducting evaluations. K-fold cross-validation with k=6 is used for robust evaluation, and various machine-learning algorithms were selected for analysis. Performance metrics such as accuracy, precision, recall, F1-score, and kappa score were used to assess the models. Overall, the methodology ensured reliable findings on the effectiveness of the selected algorithms for the Pima Indian diabetes dataset.

2.1. Data preprocessing

Dataset

The used dataset is summarized in Table 2, it consists of 2000 records, each containing information on various features related to diabetes. Statistics that provide valuable insights into the dataset and sample of the dataset are shown in The mean values indicate the average levels of glucose, pregnancies, skin thickness, blood pressure, insulin, age, diabetes pedigree function and BMI (Body Mass Index). The standard deviation highlights the variability in the data, with higher values indicating a wider range of values for certain features such as insulin. The minimum and maximum values reveal the range covered by each feature, while the quartiles offer information on the distribution and spread of the data. These statistics collectively provide a comprehensive overview of the dataset, aiding in understanding its characteristics and potential patterns for diabetes classification.

Table 1. A literature review of existing machine learning in diabetes classifiers in diabetes

Ref.	dataset	CV	Evaluation	Algo.	ACC %
[17]	PIDD	none	ACC, and MCC	Pycaret	90
[18]	PIDD	5	Sp Sn AUC	DT RF MLP KNN AB NB XB	78.9
[19]	Luzhou and PIDD	5	ACC Sp MCC Sn	RF J48 NN	80.84
[20]	PIDD	5-10	Sp PPV Sn NPV ACC	LDA NB QDA GPC	81.97
[21]	NHA NES	2-5-10	ACC Sn FM NPV PPV AUC	DT NB RF AB	92.75
[22]	PIDD	-	ACC	RF NB	74.46
[23]	PIDD	-	NPV PLR NLR PPV DP ACC Sn Sp	NB DT KNN SVM RF	82.3
[24]	PIDD	-	ACC Sn Sp	MLP	77.5
[25]	PIDD	-	Recall ACC Sp	NB, SVM, and DT	76.3
[26]	ENRC	-	ACC	DT	84
[27]	PIDD	-	ACC	NN	--

Data Cleaning

In the data preprocessing phase, data cleaning stands as a crucial and indispensable step, aimed at improving the quality and reliability of the dataset. Several techniques can be employed to perform data cleaning effectively. handling missing values is crucial. Missing values can either be imputed using techniques like mean, median, or mode, or if the missingness is significant, the associated rows or columns can be eliminated.

To address the class imbalance in the dataset, where Class 0 has 1316 samples and Class 1 has 684 samples (see Fig. 1), it is important to balance the classes to avoid biased predictions. One approach to achieving class balance is through resampling techniques. Resampling involves either increasing the minority class samples (oversampling) or decreasing the majority class samples (under-sampling) (see Fig. 2).

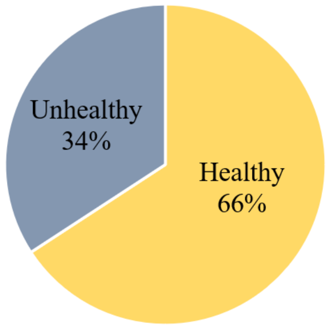

Fig. 1. Classification of dataset outcome

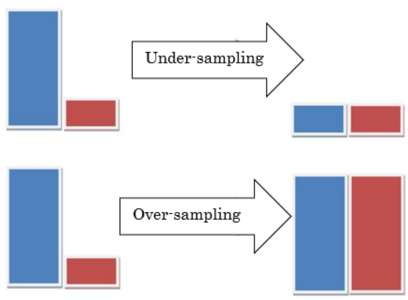

Fig. 2. Example of resampling

Hybrid learning technique for a dataset

In our study, the challenge of class imbalance in the dataset has been addressed by employing a hybrid technique called Balanced Bagging. This approach combines bagging, which leverages the power of ensemble learning, with resampling techniques. The aim is to enhance the precision of data classification within an imbalanced dataset. This can be achieved by creating an ensemble classifier consisting of decision trees trained on a balanced subset of the data. The resulting balanced dataset, obtained through the application of the Balanced Bagging technique, was then utilized for subsequent analysis and model training. This strategy effectively mitigated the impact of class imbalance and increased the robustness of our machine-learning models.

By analyzing Fig. 4 which represents the correlation matrix of the balanced dataset, we gained valuable insights into the relationships between the features. The correlation coefficients allowed us to identify potential associations and dependencies among the variables, providing essential information about the interplay between different features and their impact on the target variable.

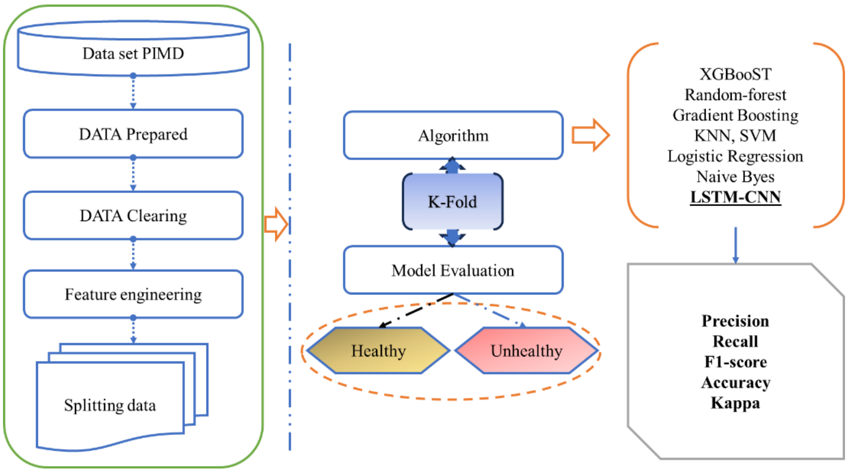

Fig. 3. Architectural design of employed models.

Table 2. Excerpt of the dataset

Pr	G	BP	ST	Insulin	BMI	DPF	age	Outcome Healthy/ Unhealthy
1	89	66	23	94	28.1	0.167	21	No
0	137	40	35	168	43.1	2.288	33	Yes
3	78	50	32	88	31	0.248	26	Yes
2	197	70	45	543	30.5	0.158	53	Yes
1	189	60	23	846	30.1	0.398	59	Yes
5	166	72	19	175	25.8	0.587	51	Yes
0	118	84	47	230	45.8	0.551	31	Yes
1	103	30	38	83	43.3	0.183	33	No
1	115	70	30	96	34.6	0.529	32	Yes
3	126	88	41	235	39.3	0.704	27	No

Table 3. Description statistical of features of the dataset

	Pr	G	BP	ST	Insulin	BMI	DPF	Age	Outcome Healthy/ Unhealthy
count	2 000	2000	2000	2000	2000	2000	2000	2000	2000
mean	3,70	121,18	69,15	20,94	80,25	32,19	0,47	33,09	0,34
std	3,31	32,07	19,19	16,10	111,18	8,15	0,32	11,79	0,47
min	0	0	0	0	0	0	0,08	21,00	0
25%	1,00	99,00	63,50	0	0	27,38	0,24	24,00	0
50%	3,00	117,00	72,00	23,00	40,00	32,30	0,38	29,00	0
75%	6,00	141,00	80,00	32,00	130,00	36,80	0,62	40,00	1,00
max	17,00	199,00	122,00	110,00	744,00	80,60	2,42	81,00	1,00

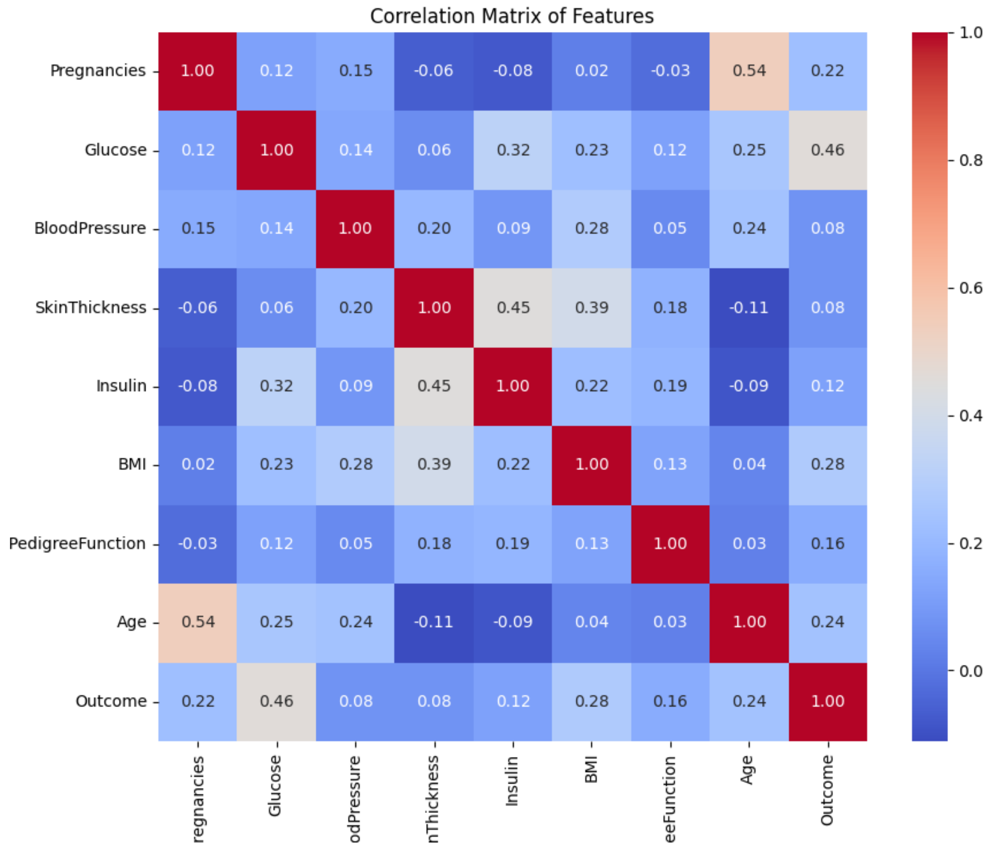

Fig. 4. Correlation matrix for the feature of the dataset

K-fold cross-validation

The K-fold C-V is a strategy frequently employed for classifier model selection and error estimation. In this paper, a k-fold cross-validation approach was used, as illustrated in Fig. 5, to split the PID dataset into multiple folds. In the inner loop. The K-1 folds were used for training the model and optimizing the hyperparameters through the grid search algorithm. In the outer loop, this process was repeated K times, with the best hyperparameters selected and the remaining fold used as the test data to evaluate the model's performance. To account for the imbalanced distribution of positive and negative samples in the PID dataset, stratified KCV was employed to maintain the original class proportions. The final performance metric was estimated using a specific equation (equation 1) [18].

$$M = \frac{1}{K} * \sum_{n=1}^{K} P_n \pm \sqrt{\sum_{n=1}^{K} (P_n - \bar{P})^2 * \frac{1}{K-1}} \quad (1)$$

The final performance metric (M) for the classifiers is determined based on the performance metric of each fold (P_n), where n ranges from 1 to K.

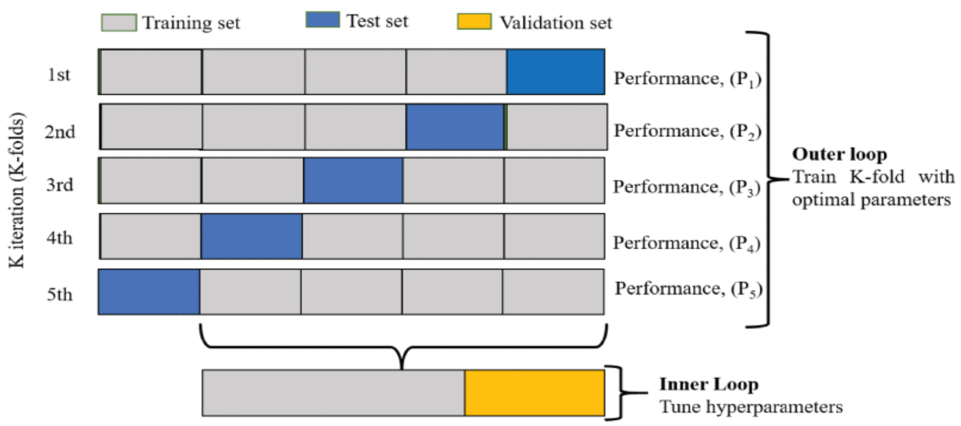

Fig. 5. K-fold Cross-validation for Hyperparameter Tuning and Evaluation in the PID Dataset

2.2. Data classification

Many algorithm classifications, such as Logistic Regression, Random Forest, SVM, coupled (LSTM & CNN), K-Nearest Neighbors, XG Boost, Gradient Boosting, and Naive Bayes, can be used for data classification. These algorithms employ various mathematical techniques and assumptions to determine the decision boundaries and make predictions based on the input features.

Logistic Regression Correlate the input features and the probability of a binary outcome in mathematical form, using the logistic function, also referred to as the sigmoid function (equation 2), is employed to convert a linear combination of input features into a value ranging from 0 to 1. This transformed value represents the predicted probability (see Fig. 6) [18-29].

$$y = \frac{e^{(b_0+b_1X)}}{1+e^{(b_0+b_1X)}} \qquad (2)$$

Where:
x = input value,
y = predicted output,
b_0 = bias or intercept term,
b_1 = coefficient for input (x),

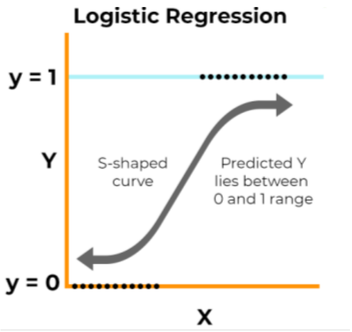

Fig. 6. Logistic Regression function

Random forest employs a collective learning technique by combining multiple decision trees to generate predictions (see Fig. 7). It is a versatile model applicable for both classification and regression tasks [28].

In a Random Forest, every decision tree is built using a randomized subset of the training data and a random subset of the input features. This randomness helps to create diversity among the trees, making them less prone to overfitting and improving the overall predictive performance.

The mathematical equation for a single decision tree is as follows (equation 3):

$$y = f(x) \qquad (3)$$

y: represents the predicted output or class label.
x: represents the input features.

XGBoost (eXtreme Gradient Boosting) is an highly optimized framework that has gained popularity in machine learning competitions and various real-world applications. It is based on the gradient boosting algorithm and is known for its efficiency, scalability, and performance [30].

The XGBoost algorithm aims to construct a resilient predictive model by amalgamating numerous less potent predictive models, frequently decision trees, in a cumulative manner. It iteratively builds decision trees and minimizes a specific objective function, incorporating both regularization techniques and gradient-based optimization [33].

The mathematical equation for XGBoost can be described as follows (equation 4):

$$y = \Sigma(b_s + \eta \,.\, \Sigma(w_t . p_t)) \qquad (4)$$

Where:
y represents the predicted output or class label,
b_s: $base\_score$ is the initial prediction made by the model,
η (eta) is the learning rate that controls the contribution of each tree to the final prediction,
w_t: $tree\_weight$ is the weight assigned to each individual decision tree,
p_t: $tree\_prediction$ represents the prediction made by an individual decision tree,

In Fig. 8, the training dataset is initially fed into classifier 1. The classifier predicts hyphens (-), indicated by the yellow background, and plus signs (+), indicated by the blue background.

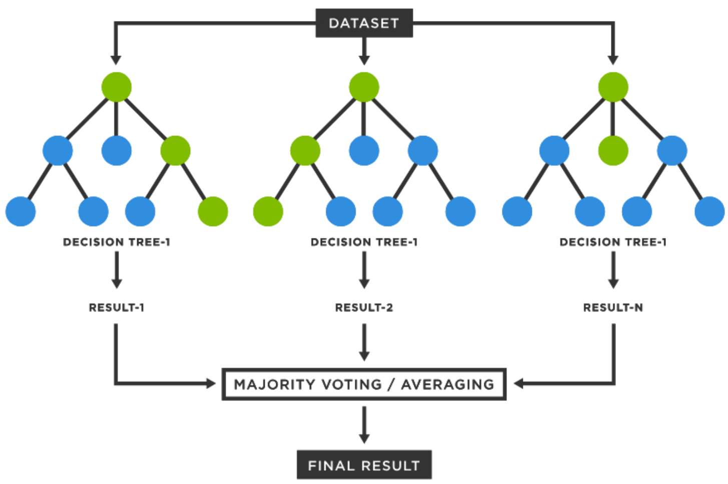

Fig. 7. Random Forest algorithm

However, classifier 1 makes two incorrect predictions of hyphens and one incorrect prediction of a plus sign, which are highlighted with circles. The weights assigned to these misclassified data points are then increased, and they are passed on to classifier 2.

Moving on to classifier 2, it correctly predicts the two hyphens that classifier 1 had initially misclassified. However, classifier 2 also introduces some new errors. This process continues with subsequent classifiers, each attempting to correct the errors made by the previous classifiers.

By the end of this iterative process, a final combined classifier is obtained, which successfully predicts all the data points correctly. This ensemble approach leverages the strengths of multiple classifiers to improve overall accuracy and performance.

Gradient Boosting is a versatile machine-learning technique utilized for both regression and classification tasks. It belongs to the family of ensemble methods and operates by combining multiple weak predictive models, often decision trees, to construct a powerful and accurate predictive model [32].

1. Initialization:

Assign initial weights to the training examples: w_i^0 , where i represents the index of the training example.

2. Boosting Round:

Fit the weak model to the training data: $M_{t(X)}$, where $M_{t(X)}$ represents the weak model at iteration t and X represents the training data.

Predict the values based on the weak model (equation 5):

$$F_{t(X)} = M_{t(X)} \quad (5)$$

Calculate the residuals (In the context of anticipated and real values) of the current model (equation 6):

$$r_i^t = y_i - F_{t(x_i)}, \quad (6)$$

where y_i represents the actual value of the i-th training example and x_i represents its features.

Update the weight (equation 7):

$$w_{i^{t+1}} = w_i^t * e^{\left(-l_r * \frac{\partial L\left(y_i, F_{t(x_i)}\right)}{\partial F_{t(x_i)}}\right)} \quad (7)$$

where L represents the loss function, l_r is the learning rate hyperparameter, and $\frac{\partial L\left(y_i, F_{t(x_i)}\right)}{\partial F_{t(x_i)}}$ is the derivative of the loss function concerning to the predictions.

3. Combine Weak Models:

Combine the weak models (equation 8) by taking a weighted average of their predictions, where the weights are determined by their performance on the training data:

$$F(X) = \sum_{t=1}^{T}\left(learning_{rate} * F_{t(X)}\right) \quad (8)$$

K-Nearest Neighbors (KNN) is a flexible algorithm used for both of classification and regression tasks. It finds the K closest labeled data points to a new unlabeled data point and predicts its label or value established upon the prevailing majority vote or average of the neighbors' labels or values. KNN requires selecting the value of K, calculating distances (equation 9), identifying neighbors, and assigning the new data point to the most common class or average value among its K nearest neighbors [34].

$$\mathrm{dist}(x, z) = \left(\sum_{r=1}^{d} |xr - zr|^{p}\right)^{\frac{1}{p}} \quad (9)$$

Minkowski distance between two data points x and z in a d-dimensional space. It calculates the distance by summing the absolute differences between the corresponding coordinates raised to the power of p, and then taking the p-th root of the sum. The Minkowski distance is a generalization of other distance metrics like $Euclidean\ distance\ (p = 2)$ and $Manhattan\ distance\ (p = 1)$.

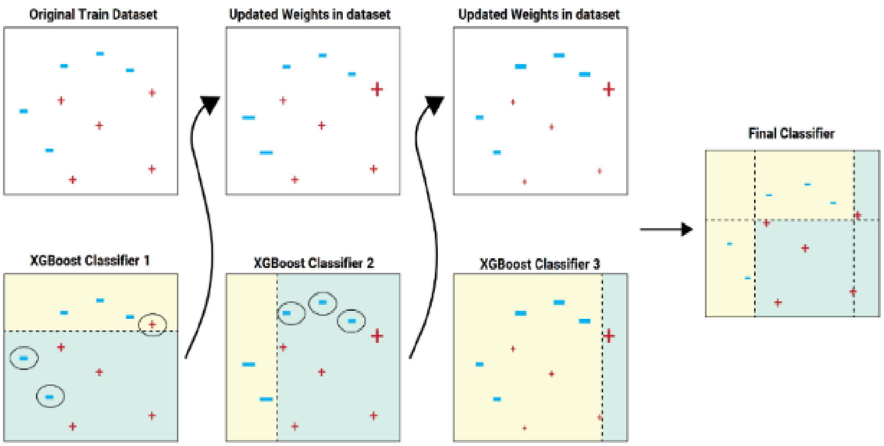

Fig. 8. XGBoost working algorithm

Support Vector Machines (SVM) are a versatile supervised machine learning algorithm capable of handling both classification and regression tasks. They excel in solving binary classification problems by effectively separating data into two distinct classes [29].

The main idea behind SVM (see Fig. 9) is to find the optimal hyperplane serves as the decision boundary that maximizes the margin between data points of different classes. The hyperplane is defined as the separator, while the margin in a support vector machine refers to the gap between the separating hyperplane that divides the classes and the closest data points from each class.

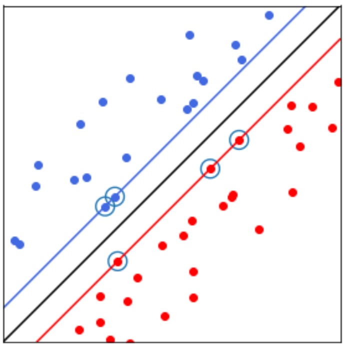

Fig. 9. Support Vector Machine (SVM) with Margin and Support Vectors in 2D Space

Given a training dataset with labeled examples (X, y), where X represents the feature vectors and y represents the corresponding class labels (+1 or -1), the SVM algorithm solves the following optimization problem:

Minimize (equation 10):

$$\left(\frac{1}{2}\right) * \left||w|\right|^2 + C * \Sigma_i^N \xi_i \tag{10}$$

Subject to (equation 11):

$$y_i * (w^T * x_i + b) \geq 1 - \xi_i, for\ all\ i \tag{11}$$

Where:

w is the weight vector perpendicular to the decision hyperplane

b is the bias term

ξ_i represents the slack variable for the i-th training example, allowing for misclassified examples or examples within the margin

C is the hyperparameter is responsible for balancing the trade-off between maximizing the margin and minimizing the classification error. A smaller C value leads to a larger margin but potentially more misclassifications, while a larger C value allows for fewer misclassifications but a smaller margin.

N is the total number of training examples. The objective function in the optimization problem is composed of two terms: the first term $\left[\left(\frac{1}{2}\right) * \left||w|\right|^2\right]$ represents the margin maximization, and the second term $C * \Sigma_i^N \xi_i$ represents the penalty for misclassifications or examples within the margin.

Naïve Bayes is a machine learning algorithm rooted in probability theory, utilizing Bayes' theorem as its foundational principle. It is commonly used for classification tasks. The algorithm assumes that features are conditionally independent given the class label, which is a naive assumption but simplifies the computation [22].

Overview of how Naive Bayes works:

Training Phase:

Calculate the prior probabilities (equation 12) of each class in the training dataset.

For each feature in the dataset, calculate the likelihood probabilities of that feature given each class.

Prediction Phase:

Given a new input instance, calculate the posterior probability of each class using Bayes' theorem.

The class with the highest posterior probability is assigned as the predicted class for the input instance.

Mathematically, Naive Bayes calculates the probability of a class label given the feature values using the following equation:

$$P_{c|f} = \left(P_c * \frac{P_{f|c}}{P_f}\right) \tag{12}$$

Where:

$P_{c|f}: P(class|features)$ is the posterior probability of the class given the features.

$P_c: P(class)$ is the prior probability of the class.

$P_{f|c}: P(features|class)$ is the likelihood probability of the features given the class.

$P_f: P(features)$ is the probability of the features.

2.3. Proposed method

1D CNN (Convolutional Neural Network).

The architecture is a variant of the traditional Convolutional Neural Network (CNN) that is specifically designed for processing one-dimensional sequential data. While the traditional CNN is primarily used for image-related tasks, the 1D CNN is commonly used for tasks involving time series analysis, natural language processing, and other sequential data analysis [37-38].

In the 1D CNN, the main idea is to apply filters to capture local patterns and extract relevant features from the input data. The filters slide across the input sequence, performing convolutions to produce feature maps. These feature maps represent the learned features from different positions of the input sequence. The 1D CNN architecture typically includes convolutional layers, pooling layers, and fully connected layers. The mathematical equation for the 1D CNN can be represented as follows (equation 13):

$$Output = [Conv(I\ .F) + B] \tag{13}$$

Where:

$I: Input$: The input sequence or data.

$F: Filter$: The filter or kernel applied to the input. It is a learnable parameter that captures specific patterns in the data.

$B: Bias$: The bias term added to the output of the convolution operation.

$Conv: Convolution$: The operation of applying the filter to the input, which involves element-wise multiplication and summation.

The 1D CNN applies multiple filters to capture different patterns in the input data, and the resulting feature maps are then processed further to extract relevant information for the task at hand (see Fig. 10) [35].

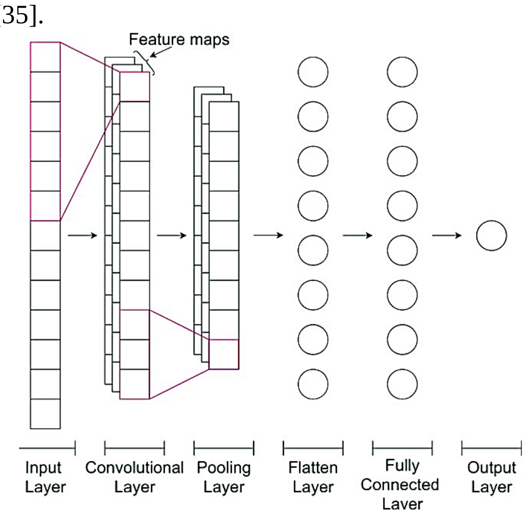

Fig. 10. One Dim-CNN architecture

LSTM (Long Short-Term Memory)

LSTM (Long_Short-Term_Memory) represents a sophisticated iteration of recurrent neural networks. Its key advantage lies in the integration of a memory cell, allowing it to retain and recall information over extended sequences [36]. The LSTM incorporates gating mechanisms (equations 14 and 15), including forget, input, and output gates, to control the flow of information within the cell. By leveraging these gates and mathematical equations, LSTM effectively models and retains long-term contextual information [39]. In classification tasks, LSTM shines in sequence modeling by understanding the relationships between elements in a sequence, enabling accurate predictions.

To perform classification, a classification layer is added on top of the LSTM layer, which can consist of dense layers and an appropriate activation function for the task. During training, the LSTM learns to minimize a specified loss function Through Backpropagation Through Time (BPTT) and updates its parameters accordingly.

LSTM is a powerful and versatile architecture for classification tasks involving sequential data, capturing long-term dependencies, and achieving accurate predictions based on the order and context of the input sequence.

Forget Gate:

$$f_t = sigmoid\left(W_f.\left[h_{\{t-1\}}, x_t\right] + b_f\right) \quad (14)$$

The forget gate decides which information to discard from the previous cell state $\left(h_{\{t-1\}}\right)$ and the current input (x_t).

Input Gate:

$$i_t = sigmoid\left(W_i.\left[h_{\{t-1\}}, x_t\right] + b_i\right) \quad (15)$$

$$\tilde{}\{C\}t = tanh(W_C.[h\{t-1\}, x_t] + b_C) \quad (16)$$

The input gate determines the new information to be stored in the cell state (equation 15, 16). It consists of two parts: the input gate (i_t) controls which values are updated (equation 17), and the candidate values $(\tilde{}\{C\}_t)$ represent new candidate cell state values.

Update Cell State:

$$C_t = f_t * C_{\{t-1\}} + i_t * \tilde{}\{C\}_t \quad (17)$$

The cell state (C_t) is updated by combining the old cell state $(C\_\{t-1\})$ with the candidate values scaled by the input gate.

Output Gate:

$$o_t = sigmoid\left(W_o.\left[h_{\{t-1\}}, x_t\right] + b_o\right) \quad (18)$$

$$h_t = o_t * tanh(C_t) \quad (19)$$

The output gate (o_t) (equation 18) controls which part of the cell state is output as the hidden state (h_t) (equation 19) of the LSTM model at the current time step.

In these equations, $W_f, W_i, W_C, W_o, b_f, b_i, b_C$, and b_o represent weight matrices and bias terms that are learned during the training process. $[h_{\{t-1\}}, x_t]$ denotes the concatenation of the previous hidden state $(h_{\{t-1\}})$ and the current input (x_t) **1D CNN and LSTM**

Combining these architectures can leverage the benefits of both approaches, allowing for a comprehensive analysis of both local and global patterns as well as long-term dependencies in the data [40].

By utilizing a combination of 1D CNN and LSTM (see Fig. 12), the classification model can effectively capture intricate patterns and dependencies in sequential data. The 1D CNN can serve as an initial feature extractor, capturing local patterns and generating higher-level representations [40]. The LSTM can then process the sequential information and leverage its memory cell to retain and utilize long-term dependencies. By integrating these architectures, the model can make accurate predictions by considering both local and global contextual information. This combined approach is particularly beneficial for classification tasks where both local patterns and long-term dependencies play a crucial role, such as sentiment analysis, activity recognition, and financial market prediction [41-42] [49-51].

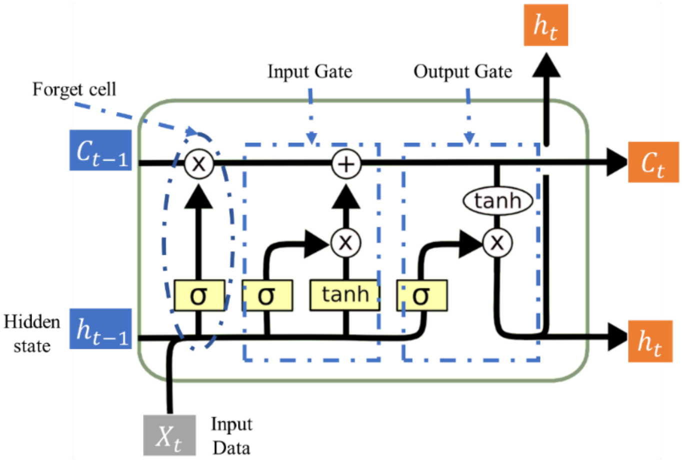

Fig. 11. LSTM architecture cell

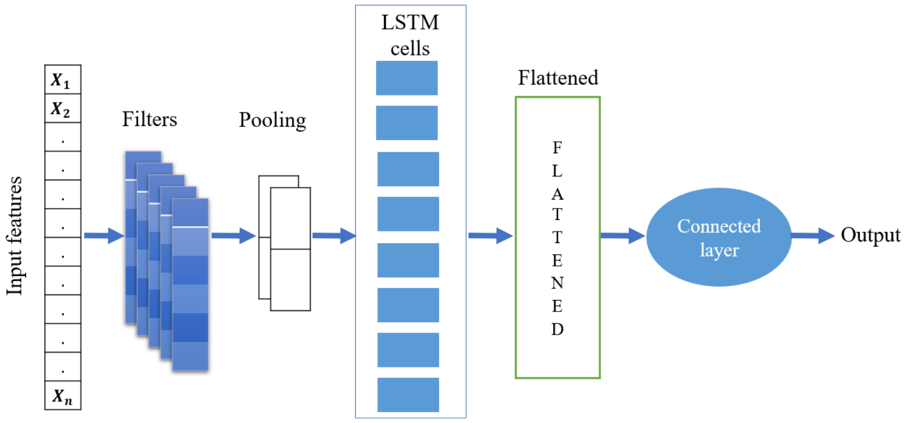

Fig. 12. Combined architecture of one-dimensional convolution network with Long Short-Term Memory

3. RESULTS AND EVALUATION

Within section, the outcomes derived from assessing diverse classifiers through various machine learning algorithms on the Pima dataset. The performance of each classifier is assessed based on several evaluation metrics, including accuracy (equation 20), precision (equation 21), recall (equation 22), and F1-score (equation 23) Kappa coefficient (equation 24), MCC, and ROC AUC. The classifiers were trained and tested using a balanced dataset.

The proposed combined LSTM and 1D-CNN model yielded impressive results when compared to the other algorithms employed in this study, thereby demonstrating its superiority and effectiveness (see Fig. 13). With a remarkable accuracy of 0.97, the model surpassed alternative approaches such as Logistic Regression (0.77), SVM (0.79), K-Nearest Neighbors (0.79), Naive Bayes (0.75), and Gradient Boosting (0.88). The precision score of 0.96 also outperformed the majority of the algorithms, with only Random Forest achieving a slightly higher value (0.99). Furthermore, the combined model exhibited a superior recall of 0.97 (see Fig. 16), surpassing Logistic Regression (0.58), SVM (0.63), and Naive Bayes (0.64). These compelling results were further validated by its impressive F1-score (Table 4) of 0.97, which exceeded most algorithms except for Random Forest (0.98).

$$Acc = \frac{TP+TN}{TP+TN+FP+FN} \tag{20}$$

$$Precision = \frac{TP}{TP+FP} \tag{21}$$

$$Sensitivity = \frac{TP}{TP+FN} \tag{22}$$

$$F - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{23}$$

where ,

TP represents True Positives, TN signifies True Negatives, FP corresponds to False Positives, and FN denotes False Negatives.

Table 4. F1-score algorithm

Model	F1-score
Naïve bayes	0.66
Logistic Regression	0.67
K-Nearest Neighbor	0.70
SVM	0.71
Gradient boosting	0.81
XGBoost	0.96
CNN+LSTM	0.97
Random Forest	0.98

In addition to the evaluation metrics mentioned previously, the Kappa coefficient (κ) was also calculated to assess the agreement between the predicted and actual classifications beyond chance.

The Kappa coefficient is calculated using the following equation:

$$\kappa = \frac{(Po - Pe)}{(1 - Pe)} \tag{24}$$

Where:

Po is the relative observed agreement, which is the proportion of instances on which the predicted and actual classifications agree,

Pe is the probability of agreement expected by chance,

The Kappa coefficient spans from -1 to 1, with 1 signifying perfect agreement, 0 denoting agreement similar to chance, and negative values indicating less agreement than chance.

The Kappa coefficient provides a robust measure of inter-rater agreement and helps assess the performance of the classifier beyond simple accuracy. It takes into account both the observed agreement and the expected agreement by chance, providing a more comprehensive evaluation of the model's performance. combined LSTM and 1D-CNN model consistently outshined its counterparts in terms of other crucial statistical measures. The Kappa coefficient present in Fig. 14, a robust measure of agreement beyond chance, yielded an outstanding value of 0.95, reaffirming the model's ability to make highly accurate predictions.

The Matthews correlation coefficient (MCC) (see Fig. 17) further bolstered the model's credibility, with a remarkable score of 0.95 indicating a strong correlation between predicted and actual classifications. Moreover, Fig. 15 shows the Receiver Operating Characteristic Area Under the Curve (ROC AUC) achieved a value of 0.97 highlighted the model's exceptional discrimination ability, positioning it among the top-performing algorithms, rivaled only by Random Forest (0.98) and XG Boost (0.97).

Table 5 presents a comprehensive comparison of various classification methods, along with their corresponding accuracies, based on the provided references. the proposed model stands out with an impressive accuracy of 97%, making it the top-performing method in this comparison. This achievement underscores the effectiveness and robustness of our proposed model in accurately classifying the given dataset. The results highlight the superiority of our approach over other well-known techniques such as K-means + Logistic, HPM, AMMLP, J48 (pruned), J48 (unpruned), and several others.

By meticulously analyzing these performance metrics, we gain a nuanced understanding of the strengths of our combined LSTM and 1D-CNN model, reaffirming its excellence in predictive accuracy, correlation, and discrimination ability. These detailed insights serve to highlight the superiority of our approach over a spectrum of well-established techniques.

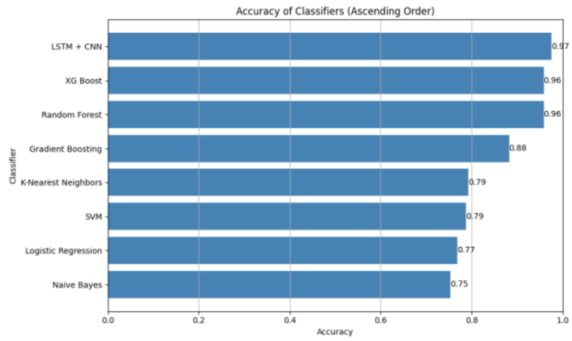

Fig. 13. Accuracy of all classifiers (Ascending Order)

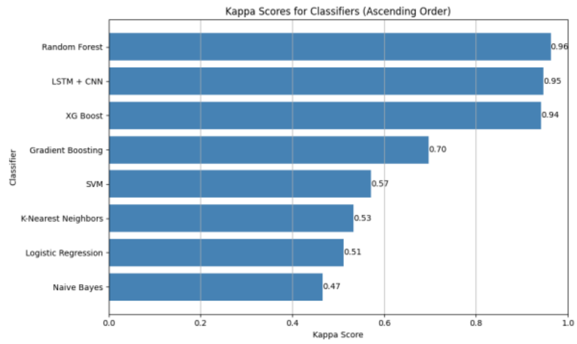

Fig. 14. Kappa sores for all models (Ascending Order)

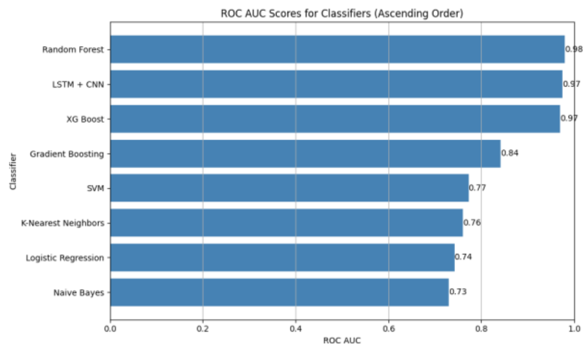

Fig. 15. ROC AUC score for all classifiers

Table 5. Accuracy analysis of classifier methods

Method	**Acc**	**Reference**
Proposed Model	97%	This paper
K-means+Logistic	95.42%	[48]
NaiveBay	74.9%	
BayesNet	74.7%	
HPM	92.38%	[43]
AMMLP	89.93%	[44]
J48 (pruned)	89.3%	[45]
J48 (unpruned)	86.6%	
MLP	81.9%	
Hybrid model	84.5%	[46]
ELM	75.72%	[47]

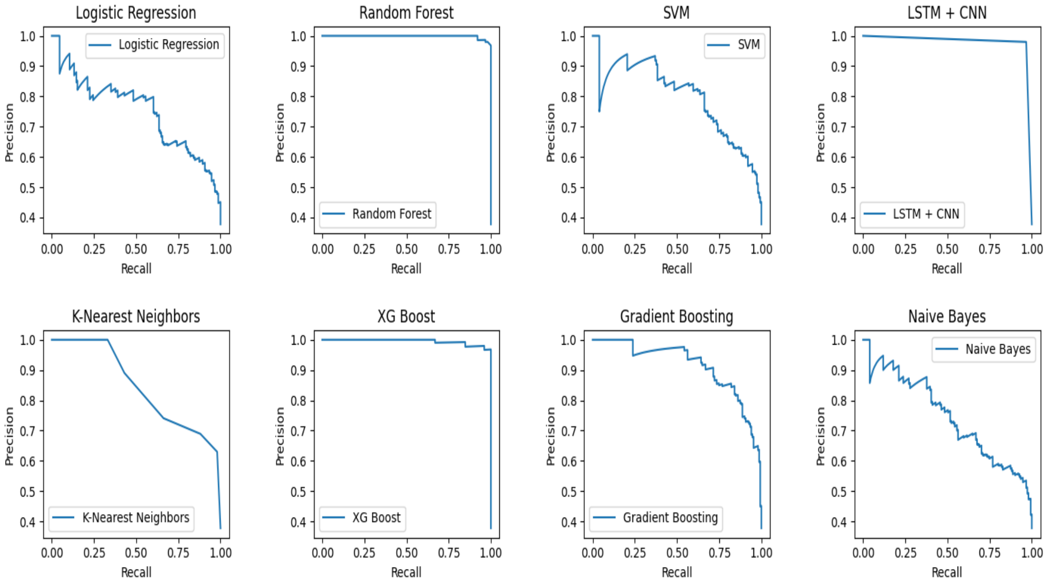

Fig. 16. Recall of all classifiers

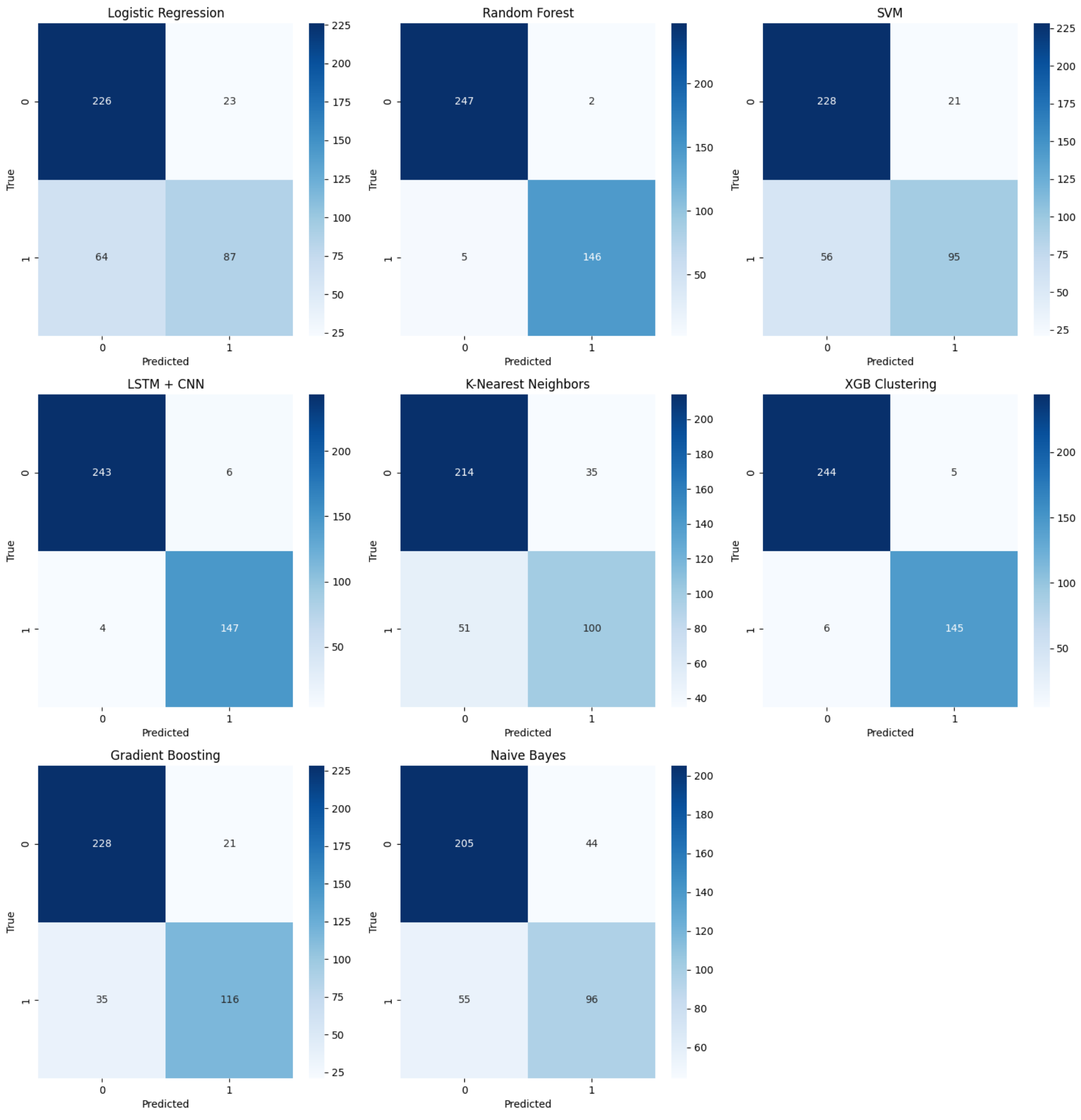

Fig. 17. Result of Matthews Correlation Coefficient (MCC) for all models

4. CONCLUSION

In conclusion, this study introduced a novel approach for diabetes classification using a combined CNN- LSTM model, which outperformed other classifiers. The model effectively captured sequential patterns and extracted meaningful features, resulting in an impressive accuracy of 97%. The successful application of this model signifies a significant advancement in diabetes diagnosis and treatment. The effective implementation of this paradigm represents a big step forward in diabetes diagnosis and treatment.

Looking ahead, integrating the developed model into a medical platform and extending its capabilities to predict other medical conditions hold immense potential for enhancing healthcare outcomes. Collaborations with healthcare institutions and research organizations will further validate the model's performance and applicability. Improving the model's interpretability and regularly updating it will ensure its long-term effectiveness. Implementing these future endeavors will culminate in a comprehensive medical platform that utilizes predictive models to enhance patient care, facilitate early detection, and support proactive interventions. By harnessing data-driven approaches, we can pave the way for personalized healthcare and contribute to the betterment of society's overall health.

Source of funding: *This research received no external funding.*

Author contributions: *research concept and design, A.E.M., Y,A., W.B., I.M., M.B., A.E.; Collection and/or assembly of data M.B.; Data analysis and interpretation, A.E.M.; Writing the article, Y.A.; Critical revision of the article, W.B., I.M., A.E.*

Declaration of competing interest: *The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.*

REFERENCES

1. Diabetes Statistics. Center for Diabetic Empowerment Education. https://ceed-diabete.org/fr/le-diabete/les-chiffres/.
2. Meneghetti L, Terzi M, Del Favero S, Susto GA, Cobelli C. Data-Driven Anomaly Recognition for Unsupervised Model-Free Fault Detection in Artificial Pancreas. IEEE Transactions on Control Systems Technology 2020; 28(1): 33–47. https://doi.org/10.1109/TCST.2018.2885963.
3. Warshaw H, Isaacs D, MacLeod J. The Reference Guide to Integrate Smart Insulin Pens Into Data-Driven Diabetes Care and Education Services. The Diabetes Educator 2020; 46(4_suppl): 3S-20S. https://doi.org/10.1177/0145721720930183.
4. Ellahham S. Artificial Intelligence: The Future for Diabetes Care. The American Journal of Medicine 2020; 133(8): 895–900. https://doi.org/10.1016/j.amjmed.2020.03.033.
5. Ibrahim MS, Saber S. Machine Learning and Predictive Analytics: Advancing Disease Prevention in Healthcare. Journal of Contemporary Healthcare Analytics 2023; 7(1), 53-71.
6. Behera A. Use of artificial intelligence for management and identification of complications in diabetes. Clinical Diabetology 2021; 10(2): 221–5. https://doi.org/10.5603/DK.a2021.0007.
7. Polat K, Güneş S, Arslan A. A cascade learning system for classification of diabetes disease: Generalized Discriminant Analysis and Least Square Support Vector Machine. Expert Systems with Applications 2008; 34(1): 482–7. https://doi.org/10.1016/j.eswa.2006.09.012.
8. Polat K, Güneş S. An expert system approach based on principal component analysis and adaptive neuro-fuzzy inference system to diagnosis of diabetes disease. Digital Signal Processing 2007; 17(4): 702–10. https://doi.org/10.1016/j.dsp.2006.09.005.
9. Kannadasan K, Edla DR, Kuppili V. Type 2 diabetes data classification using stacked autoencoders in deep neural networks. Clinical Epidemiology and Global Health 2019; 7(4): 530–5. https://doi.org/10.1016/j.cegh.2018.12.004.
10. Caliskan A, Yuksel ME, Badem H, Basturk A. Performance improvement of deep neural network classifiers by a simple training strategy. Engineering Applications of Artificial Intelligence 2018; 67: 14–23. https://doi.org/10.1016/j.engappai.2017.09.002.
11. Naz H, Ahuja S. Deep learning approach for diabetes prediction using PIMA Indian dataset. Journal of Diabetes & Metabolic Disorders 2020; 19(1): 391–403. https://doi.org/10.1007/s40200-020-00520-5.
12. Zhu C, Idemudia CU, Feng W. Improved logistic regression model for diabetes prediction by integrating PCA and K-means techniques. Informatics in Medicine Unlocked 2019; 17: 100179. https://doi.org/10.1016/j.imu.2019.100179.
13. Mercaldo F, Nardone V, Santone A. Diabetes Mellitus Affected Patients Classification and Diagnosis through Machine Learning Techniques. Procedia Computer Science 2017; 112: 2519–28. https://doi.org/10.1016/j.procs.2017.08.193.
14. Qawqzeh YK, Bajahzar S, Jemmali M, Otoom MM, Thaljaoui A. Classification of diabetes using photoplethysmogram (PPG) waveform analysis: logistic regression modeling. BioMed Research International 2020; Article ID 3764653.
15. Tafa Z, Pervetica N, Karahoda B. An intelligent system for diabetes prediction. 2015; 378–82. https://doi.org/10.1109/MECO.2015.7181948.
16. Hussain A, Naaz S. Prediction of Diabetes Mellitus: Comparative Study of Various Machine Learning Models. 2021; 103–15. https://doi.org/10.1007/978-981-15-5148-2_10.
17. Whig P, Gupta K, Jiwani N, Jupalle H, Kouser S, Alam N. A novel method for diabetes classification and prediction with Pycaret. Microsystem Technologies 2023; 29(10): 1479–87. https://doi.org/10.1007/s00542-023-05473-2.
18. Hasan MdK, Alam MdA, Das D, Hossain E, Hasan M. Diabetes Prediction Using Ensembling of Different Machine Learning Classifiers. IEEE Access 2020; 8: 76516–31. https://doi.org/10.1109/ACCESS.2020.2989857.
19. Mamuda M, Sathasivam S. Predicting the survival of diabetes using neural network. 2017; 1870: 040046. https://doi.org/10.1063/1.4995878.
20. Malasinghe LP, Ramzan N, Dahal K. Remote patient monitoring: a comprehensive study. Journal of Ambient Intelligence and Humanized Computing 2019; 10(1): 57–76. https://doi.org/10.1007/s12652-017-0598-x.

21. Maniruzzaman Md, Rahman MdJ, Ahammed B, Abedin MdM. Classification and prediction of diabetes disease using machine learning paradigm. Health Information Science and Systems 2020; 8(1): 7. https://doi.org/10.1007/s13755-019-0095-z.
22. Jackins V, Vimal S, Kaliappan M, Lee MY. AI-based smart prediction of clinical disease using random forest classifier and Naive Bayes. The Journal of Supercomputing 2021; 77(5): 5198–219. https://doi.org/10.1007/s11227-020-03481-x.
23. Sneha N, Gangil T. Analysis of diabetes mellitus for early prediction using optimal features selection. Journal of Big Data 2019; 6(1): 13. https://doi.org/10.1186/s40537-019-0175-6.
24. Mohapatra SK, Swain JK, Mohanty MN. Detection of Diabetes Using Multilayer Perceptron. International Conference on Intelligent Computing and Applications 2019; 109–16. https://doi.org/10.1007/978-981-13-2182-5_11.
25. Sisodia D, Sisodia DS. Prediction of Diabetes using Classification Algorithms. Procedia Computer Science 2018; 132: 1578–85. https://doi.org/10.1016/j.procs.2018.05.122.
26. Orabi KM, Kamal YM, Rabah TM. Early Predictive System for Diabetes Mellitus Disease. Proceedings of the Industrial Conference on Data Mining 2017; 420–427.
27. Alade OM, Sowunmi OY, Misra S, Maskeliūnas R, Damaševičius R. A Neural Network Based Expert System for the Diagnosis of Diabetes Mellitus. Information Technology Science 2018; 14–22. https://doi.org/10.1007/978-3-319-74980-8_2.
28. Kirasich K, Smith T, Sadler B. Random forest vs logistic regression: binary classification for heterogeneous datasets. SMU Data Science Review 2018; 1(3): 9.
29. Xu Y, Klein B, Li G, Gopaluni B. Evaluation of logistic regression and support vector machine approaches for XRF based particle sorting for a copper ore. Minerals Engineering 2023; 192: 108003. https://doi.org/10.1016/j.mineng.2023.108003.
30. Shehadeh A, Alshboul O, Al Mamlook RE, Hamedat O. Machine learning models for predicting the residual value of heavy construction equipment: An evaluation of modified decision tree, LightGBM, and XGBoost regression. Automation in Construction 2021; 129: 103827. https://doi.org/10.1016/j.autcon.2021.103827.
31. Basha SM, Rajput DS. Chapter 9 - Survey on Evaluating the Performance of Machine Learning Algorithms: Past Contributions and Future Roadmap. Deep Learning and Parallel Computing Environment for Bioengineering Systems 2019; 153–64. https://doi.org/10.1016/B978-0-12-816718-2.00016-6.
32. Taha AA, Malebary SJ. An Intelligent Approach to Credit Card Fraud Detection Using an Optimized Light Gradient Boosting Machine. IEEE Access 2020; 8: 25579–87. https://doi.org/10.1109/ACCESS.2020.2971354.
33. Demir S, Sahin EK. An investigation of feature selection methods for soil liquefaction prediction based on tree-based ensemble algorithms using AdaBoost, gradient boosting, and XGBoost. Neural Computing and Applications 2023; 35(4): 3173–90. https://doi.org/10.1007/s00521-022-07856-4.
34. Wang, Alex X, Chukova SS, Nguyen BP. Ensemble k-nearest neighbors based on centroid displacement. Information Sciences 2023; 629: 313-323.
35. Awad M, Khanna, R. Support vector machines for classification. Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers 2015; 39-66.
36. Nikhar S, Karandikar AM. Prediction of heart disease using machine learning algorithms. International Journal of Advanced Engineering, Management and Science 2016; 2(6): 239484.
37. Nam Y, Lee C. Cascaded Convolutional Neural Network Architecture for Speech Emotion Recognition in Noisy Conditions. Sensors 2021; 21(13): 4399. https://doi.org/10.3390/s21134399.
38. Soni A, Al-Sarayreh M, Reis MM, Brightwell G. Hyperspectral imaging and deep learning for quantification of Clostridium sporogenes spores in food products using 1D- convolutional neural networks and random forest model. Food Research International 2021; 147: 110577. https://doi.org/10.1016/j.foodres.2021.110577.
39. Chaerun NE, Yean-Der K. Comparative assessment to predict and forecast water-cooled chiller power consumption using machine learning and deep learning algorithms. Sustainability 2021; 13(2): 744.
40. Patel E, Kushwaha DS. A hybrid CNN-LSTM model for predicting server load in cloud computing. The Journal of Supercomputing 2022; 78(8): 1–30. https://doi.org/10.1007/s11227-021-04234-0.
41. Kim TY, Cho SB. Predicting residential energy consumption using CNN-LSTM neural networks. Energy 2019; 182: 72–81. https://doi.org/10.1016/j.energy.2019.05.230.
42. Zhu J, Chen H, Ye W. Classification of Human Activities Based on Radar Signals using 1D-CNN and LSTM. 2020 IEEE International Symposium on Circuits and Systems (ISCAS) 2020; 1–5. https://doi.org/10.1109/ISCAS45731.2020.9181233.
43. Patil BM, Joshi RC, Toshniwal D. Hybrid prediction model for Type-2 diabetic patients. Expert Systems with Applications 2010; 37(12): 8102–8. https://doi.org/10.1016/j.eswa.2010.05.078.
44. Marcano-Cedeño A, Torres J, Andina D. A prediction model to diabetes using artificial metaplasticity. International Work-Conference on the Interplay Between Natural and Artificial Computation. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.
45. Ahmad A, Mustapha A, Zahadi ED, Masah N, Yahaya NY. Comparison between Neural Networks against Decision Tree in Improving Prediction Accuracy for Diabetes Mellitus. Digital Information Processing and Communications 2011; 537–45. https://doi.org/10.1007/978-3-642-22389-1_47.
46. Kahramanli H, Allahverdi N. Design of a hybrid system for the diabetes and heart diseases. Expert Systems with Applications 2008; 35(1): 82–9. https://doi.org/10.1016/j.eswa.2007.06.004.
47. Priyadarshini R, Dash N, Mishra R. A Novel approach to predict diabetes mellitus using modified Extreme learning machine. 2014 International Conference on Electronics and Communication Systems (ICECS) 2014; 1–5. https://doi.org/10.1109/ECS.2014.6892740.
48. Wu H, Yang S, Huang Z, He J, Wang X. Type 2 diabetes mellitus prediction model based on data mining. Informatics in Medicine Unlocked 2018; 10: 100–7. https://doi.org/10.1016/j.imu.2017.12.006.
49. Yassine A, Ali EM, Ismail M. Telemedicine in the Era of Covid-19: Teleconsultation Architecture Platform. Proceedings of the 3rd International Conference on Electronic Engineering and Renewable Energy Systems 2023; 347–56. https://doi.org/10.1007/978-981-19-6223-3_38.
50. Ayat Y, El Moussati A, Benzaouia M, Mir I. New Topology of WSN for Smart Irrigation with Low Consumption and Long Range. In International Conference on Digital Technologies and Applications 2023; 221-231.
51. Ismail M, Anas B, Yassine A, Mohammed B. Improved control technique based on neural network for AC-Chopper of railway substations. 2022 2nd International Conference on Innovative Research in

Applied Science, Engineering and Technology (IRASET) 2022; 1-6.

Yassine AYAT obtained a Master's degree in Technical Science with a specialization in Embedded Systems and Robotics in 2020 from the Faculty of Technical Sciences in El Hociema, Morocco. He is currently in the process of developing his thesis, which commenced in 2020, focusing on the integration of IoT across various domains including medical, agriculture, domotics, computer electronics, embedded systems, internet of things, smart systems, and the design of electronic cards for 3D objects.
e-mail: yassine.ayat@ump.ac.ma.

Wiame BENZEKRI is a Ph.D. candidate in Computer Engineering at Mohammed First University in Oujda, Morocco, where she is conducting research in embedded system applications, artificial intelligence and Internet Of Things. She holds an engineering degree in computer engineering from the National School of Applied Sciences in Oujda, Morocco. She is currently employed at Mohammed First University as a computer engineer.
e-mail: w.benzekri@ump.ac.ma

Ali El MOUSSATI received his B.S. (1998) in E.E.A., M.S. (1999) in electronics and Ph.D. degrees (2004) in Microwave and microtechnologies from the University of Lille1, France. Currently, he is a professor in the Mohammed Premier University, National school of applied sciences of Oujda, Morocco. His main research interest is Radio over fiber communication, embedded system applications, artificial intelligence and Internet Of Things.
e-mail: a.elmoussati@ump.ac.ma.

Ismail MIR In the year 2020, he accomplished the attainment of a degree in Electrical and Electronic Engineering from the National School of Applied Science at Mohammed First University, Oujda, Morocco. His principal area of academic exploration centered around the realm of control strategies within the power grid and FACTS system. At present, Ismail is engrossed in the pursuit of a Doctorate in Electrical Engineering within the same institution and university.
e-mail: ismail.mir@ump.ac.ma.

Mohammed BENZAOUIA earned his Master's degree in Physics with a specialization in Mechanics and Energetics from the Faculty of Sciences in Oujda, Morocco, in 2019. Presently, he is pursuing a Ph.D. at the National School of Applied Sciences in Oujda. His research interests primarily focus on artificial intelligence applications across various domains, including embedded systems and power electronics.
e-mail: m.benzaouia@ump.ac.ma.

Abdelaziz EL AOUNI has graduated with a master on Optics and Materials 2021 delivered by the faculty of science Oujda, Morocco. Currently preparing his thesis started in 2021 the interest of his research: Application of Artificial Intelligence for Optimizing Energy Resource Management in a Microgrid.
e-mail: abdelaziz.elaouni@ump.ac.ma.