AI-Powered Platform for Comprehensive Diabetes Management

Yousra Beldjebel[1], Meftah Zouai[1], Ahmed Aloui[1] and Ilyes Naidji[1]

[1]*Department of Computer Science, Mohamed Khider University, Biskra, Algeria*

Abstract

This paper introduces an innovative AI-powered platform designed to enhance comprehensive diabetes management. The platform leverages advanced machine learning (ML) and deep learning (DL) algorithms to significantly improve the processes of diagnosis, continuous monitoring, and overall patient care. By utilizing a substantial dataset obtained from a Taipei Municipal medical center, the platform integrates a range of AI techniques, such as Logistic Regression, Decision Trees, Random Forest, K-Nearest Neighbor (KNN), Support Vector Machines (SVM), and Long Short-Term Memory (LSTM) networks. These algorithms work in tandem to provide accurate predictions and personalized insights into patient health. Key pre-processing steps ensure high data quality, including handling missing values, assessing the relevance of attributes, and balancing the dataset using the Synthetic Minority Over-sampling Technique (SMOTE). These measures enhance the robustness of the models, resulting in improved prediction accuracy and model performance. Notably, the Random Forest model emerged as a standout performer, achieving an impressive accuracy rate of 92.78%, significantly advancing the accuracy, sensitivity, and specificity of diabetes prediction. The platform is built with a scalable software architecture, complemented by an intuitive user interface that caters to a variety of clinical applications, making it a valuable tool for healthcare providers. This study highlights the transformative potential of AI in revolutionizing diabetes care, empowering clinicians to make informed decisions, and creating personalized treatment plans. Future research aims to expand the diversity of datasets, further refine the AI models, and incorporate real-time patient feedback to optimize the platform's effectiveness.

Keywords

LinFiSim, Smart Home Simulation, Internet of Things (IoT), Java, JavaFX, Home Automation,

1. Introduction

Diabetes mellitus is a chronic metabolic disorder characterized by elevated blood sugar levels, which, if inadequately managed, can result in severe health complications such as cardiovascular disease, neuropathy, nephropathy, and retinopathy[1]. The prevalence of diabetes is steadily increasing worldwide, posing a significant public health challenge. According to the International Diabetes Federation, approximately 463 million adults were living with diabetes in 2019, with this number projected to rise to 700 million by 2045. This growing burden necessitates innovative approaches to improve the diagnosis, monitoring, and management of diabetes[2].

The management of diabetes involves multiple components, including early diagnosis, continuous monitoring of blood glucose levels, lifestyle modifications, and personalized treatment regimens[3]. Traditional methods of diabetes management rely heavily on manual monitoring and periodic clinical visits, which can be cumbersome and less effective in providing real-time feedback[4, 5, 6]. The advent of artificial intelligence (AI) and machine learning (ML) technologies[7, 8, 9] has opened new avenues for improving diabetes care by enabling more accurate predictions, continuous monitoring, and personalized treatment strategies.

AI technologies, particularly ML and deep learning (DL)[10], have shown immense potential in revolutionizing healthcare. These technologies can analyze large datasets to uncover hidden patterns, predict outcomes, and provide actionable insights. In the context of diabetes management, AI can enhance various aspects such as early diagnosis through predictive modeling, real-time monitoring using wearable devices, and personalized treatment plans based on patient-specific data[11, 12].

Recent studies have demonstrated the effectiveness of AI in diabetes diagnosis and monitoring. For instance, ML algorithms have been used to analyze patient data and predict the onset of diabetes with high accuracy. DL models, such as convolutional neural networks (CNNs)[13] and long short-term memory (LSTM) networks[14, 15], have been applied to continuous glucose monitoring systems to provide real-time predictions of blood glucose levels. These advancements highlight the potential of AI to improve clinical outcomes and patient quality of life[16, 17].

This paper aims to present a comprehensive AI-powered platform for diabetes management that integrates various ML and DL algorithms to enhance diagnosis, monitoring, and overall management. The specific objectives of the study are as follows:

SYSTEM 2025: 11th Sapienza Yearly Symposium of Technology, Engineering and Mathematics. Rome, June 4-6, 2025

beldjebelyousra@gmail.com (Y. Beldjebel); meftah.zouai@univ-biskra.dz (M. Zouai); a.aloui@univ-biskra.dz (A. Aloui); ilyes.naidji@univ-biskra.dz (I. Naidji)
0000-0003-0950-2667 (M. Zouai); 0000-0003-2623-5118 (A. Aloui); 0000-0001-8747-0766 (I. Naidji)

1. **Diagnosis:** To evaluate and compare the performance of different ML algorithms in predicting diabetes using a comprehensive dataset.
2. **Monitoring:** To develop and assess DL models for real-time blood glucose level prediction.
3. **Platform Development:** To design a scalable and user-friendly software architecture that integrates the AI models and supports clinical application.
4. **Evaluation:** To analyze the empirical findings in terms of accuracy, sensitivity, and specificity, and to discuss the implications for diabetes care and future research directions.

The subsequent sections of this paper will provide a detailed literature review, describe the methodology used in developing the platform, present the results and findings, explore the technical aspects of the software architecture, discuss the implications and future research directions, and conclude with the key takeaways from the study.

2. Literature Review

This section provides a brief overview of related work in the field of AI-driven diabetes management.

2.1. Diagnosis of diabetes

Chandrashekar D. K. et al. [1] conducted a study on the prediction of gestational diabetes utilizing the PIMA Indian dataset from the UCI Machine Learning Repository, which comprises 8 features. The objective of the research was to evaluate the efficacy of several machine learning algorithms in predicting the onset of gestational diabetes in female patients. The algorithms tested included Naive Bayes (NB), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), K-Means Clustering, Artificial Neural Networks (ANN), and Random Forest (RF).

The study reported varying degrees of accuracy for each algorithm. The Artificial Neural Network (ANN) achieved an accuracy of 72%, Support Vector Machine (SVM) attained 79%, K-Means Clustering and K-Nearest Neighbors (KNN) both reached 77%, Random Forest (RF) showed 80%, and Naive Bayes (NB) achieved the highest accuracy at 82%. This research highlights the significant potential of machine learning techniques in improving the prediction and early detection of gestational diabetes, offering valuable insights for developing more efficient diagnostic tools.

Thotad et al. [2], in their study, analyze machine learning-based classifiers to diagnose diabetes in India using data from the Indian Demographic and Health Survey (2019–21). The study demonstrates that the Random Forest algorithm achieved remarkable accuracy, with a classification accuracy of 95.35% after Principal Component Analysis (PCA) and 96.5% before PCA. Prior to using PCA, XGBoost achieved 95.33% accuracy, while SVM (RBF) obtained 74.83%. After applying PCA, SVM (RBF) maintained an accuracy of 74.14%, and XGBoost's accuracy slightly decreased to 93.33%. These findings indicate the reliable performance of the Random Forest model in diagnosing diabetes.

Navya Pratyusha Miriyala et al. [18] suggested a diagnostic analysis of diabetes mellitus using a machine learning approach. The study utilized the Pima Indians Diabetes Dataset (PIDD) to train six different machine learning (ML) algorithms, including Naïve Bayes, KNN, Random Forest, Logistic Regression, Decision Tree, and eXtreme Gradient Boosting (XGBoost). According to the observed experimental data, the Decision Tree algorithm delivered an accuracy of 85.3%, while XGBoost provided the best accuracy at 88.2%. The study suggests that future work could focus on handling the sampling strategy to balance the data, as there is a slight imbalance present.

Jobeda JK et al. [11] suggested a comparison of machine learning algorithms for diabetes prediction using the Pima Indian Diabetes (PID) dataset, which contains data on 768 patients. They used seven different machine learning algorithms, including Decision Tree (DT), K-Nearest Neighbors (KNN), Random Forest (RF), Naïve Bayes (NB), AdaBoost (AB), Logistic Regression (LR), and Support Vector Machine (SVM). Every model offered an accuracy of at least 70%, with LR and SVM providing approximately 77-78% accuracy for both train/test split and K-fold cross-validation methods. Additionally, they tested a neural network (NN) model with varying hidden layers (1, 2, 3) and epochs (200, 400, 800). The best accuracy, achieved by the NN with two hidden layers and 400 epochs, was 88.6%.

Sireesha et al. [3] proposed implementing a model to detect diabetes using machine learning classifiers to achieve high accuracy with the Pima Indian Diabetes Dataset. They applied several classification algorithms, including K-Nearest Neighbor (KNN), Decision Tree (DT), Random Forest (RF), AdaBoost, Naive Bayes, and XGBoost. The results showed that the Decision Tree Classifier achieved 85.2% accuracy, the XGBoost Classifier achieved 88.8% accuracy, the KNN Classifier achieved 86.2% accuracy, the Random Forest Classifier achieved 88.1% accuracy, the AdaBoost Classifier achieved 87.7% accuracy, and the Naive Bayes Classifier achieved 80.7% accuracy. Consequently, the study concluded that the XGBoost Classifier is the best among all the classifiers mentioned.

Zhu et al. [19] recently conducted a comprehensive review of how deep learning is being utilized in diabetes care. They categorized their findings into three main areas: diagnosing diabetes, monitoring blood sugar levels, and identifying complications associated with the disease. The review included 40 studies that compared deep learning models with traditional machine learning methods, and found that deep learning models generally outperformed the traditional approaches.

The review also examined how continuous glucose monitoring and artificial pancreas devices could aid in diabetes management. However, it highlighted the challenges of dealing with significant fluctuations in blood sugar levels and maintaining them within target ranges. The authors discussed various deep learning architectures, such as Deep Multilayer Perceptrons (DMLPs), Convolutional Neural Networks (CNNs)[20, 21**?**], and Recurrent Neural Networks (RNNs), that have been used in diabetes research. They noted that these models excel at handling complex data but face issues such as limited data for training and the interpretability of their predictions[**?**].

The authors concluded that future advancements in deep learning have the potential to significantly improve diabetes management strategies.

Rahman et al. [7] introduced an innovative method for detecting diabetes using a Convolutional Long Short-term Memory (Conv-LSTM) model. This study was the first to apply this type of model for diabetes detection.

The researchers utilized the Pima Indians Diabetes Database (PIDD) to test their Conv-LSTM model against three other well-known models: CNN-LSTM, Traditional LSTM (T-LSTM), and Convolutional Neural Network (CNN)[22]. They employed the Boruta method to identify the most significant features in the data, such as age, blood pressure, insulin, glucose, and BMI. The Conv-LSTM model achieved the highest performance with an accuracy of 97.26% when tested with cross-validation, outperforming the other models and previous techniques.

The study underscores the importance of using advanced methods and feature selection techniques for diabetes prediction. The Conv-LSTM model addressed several issues inherent in other LSTM models, such as the vanishing gradient problem and challenges related to temporal data changes.

Swapna G. et al. [23] present a methodology for the classification of diabetic and normal HRV signals using deep learning architectures. They employed a combination of convolutional neural networks (CNN) and long short-term memory (LSTM) networks applied to HRV data, achieving an accuracy of 95.1%. The study further improves upon this methodology by incorporating a support vector machine (SVM) for classification, which increased the accuracy to 95.7%.

2.2. Monitoring of diabets

Rabbi et al. [24] performed a groundbreaking study using a novel approach for blood glucose prediction by employing a deep recurrent neural network (RNN) model coupled with long short-term memory (LSTM) stacking-based Kalman smoothing to address sensor failures. The goal of this method was to establish ground truth by comparing fingerprint blood glucose readings with expected continuous glucose monitoring (CGM) values. To evaluate the model, their study utilized the OhioT1DM dataset, which includes eight weeks of data from five T1D patients. The proposed method outperformed previous approaches, achieving root mean squared errors (RMSE) of 6.45 mg/dL and 17.24 mg/dL for the 30-minute and 60-minute prediction ranges, respectively.

Hatice Vildan Dudukcu et al. [25] proposed a method for blood glucose prediction using deep neural networks with weighted decision level fusion, leveraging patients' past BG data to address the challenge of accurately forecasting BG levels for diabetic patients. The authors employed three neural network architectures: Long Short-Term Memory (LSTM), WaveNet, and Gated Recurrent Units (GRU). They combined these models to enhance prediction accuracy by fusing the outputs of these networks.

The study utilized the OhioT1DM dataset, which includes blood glucose history from 12 diabetic patients, and evaluated the performance of the models over 30, 45, and 60-minute prediction horizons. The results demonstrated that the fusion of the three models yielded the best results for short-term blood glucose prediction, with RMSE values of 21.90 mg/dL for 30 minutes, 29.12 mg/dL for 45 minutes, and 35.10 mg/dL for 60 minutes.

Martinsson et al. [16] employed long short-term memory (LSTM) networks, a variant of RNNs that effectively capture temporal dependencies in sequential data. Their model processes historical blood glucose measurements to predict future levels, requiring no additional feature engineering or complex data preprocessing. The study demonstrated that the model performs comparably to state-of-the-art methods on the OhioT1DM dataset, using metrics such as root-mean-squared error (RMSE) and the blood glucose-specific surveillance error grid (SEG) to evaluate performance. Furthermore, by incorporating a variance estimation method, the model generates a confidence measure in the form of a univariate Gaussian distribution for every prediction. This feature enhances the interpretability and reliability of the forecasts, allowing users to know when to exercise caution based on predicted accuracy. Because this method is computationally efficient, it can be used on devices with low computing capacity, such as cell phones and CGM devices.

3. Methodology

To develop a comprehensive diabetes management platform. We used a combination of AI techniques:

3.1. Diagnosis of diabetes

In the diagnosis of diabetes we used several Machine learning and deep learning algorithms as follows:

3.1.1. Logistic Regression (LR)

The statistics branch is where the LR models were obtained. This approach has been modified for problem statements including binary classification. The primary goal of LR is to determine the coefficient values. The value is converted to 0–1 by +e LR. The LR model determines whether to anticipate a given data instance of the class as 0 or 1. This method can be used to solve issues if there are several plausible explanations for our predictions .Standard function of lr is shown as follow [26]:

$$h_\theta(X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}} \tag{1}$$

Equation 1 represents the logistic decision for the projected data. The data label X represents the constants as β_1 and β_0.

3.1.2. Decision Trees (DTs)

DTs form a tree structure by determining thresholds for input features. The classifier creates judgment rules to forecast the target class or value [27].

3.1.3. Random Forest

RF is a supervised learning system. The RF classifier consists of many decision trees for different subjects from the provided dataset. To boost forecast accuracy, the algorithm takes the average of subsets from each tree. Instead of depending on a single decision tree, RF uses the majority vote from all trees to forecast the result. Each node in the decision tree answers a query about the data [26].

3.1.4. K-Nearest Neighbor (K-NN)

KNN is a popular machine learning algorithm that uses the Supervised Learning approach. According to Brownlee (2016b), K-NN is commonly used for regression and classification. The K-NN method compares the similarities between new and current cases/data. The new case is allocated to the most similar category from the available possibilities [28].

3.1.5. Support Vector Machines (SVM)

SVM is non-parametric algorithms that solve regression and classification problems with linear and non-linear functions. These functions assign input feature vectors to an n-dimensional space known as the feature space [27].

3.1.6. Artificial neural networks

ANNs mimic real neural networks by connecting their artificial neurons in a manner akin to that of the brain network. The brain, or neural network, is made up of connections between these cells, also known as neurons [29] Information enters a biological neuron by its dendrite, is processed by the soma, and then is transferred via an axon [30]. When it comes to artificial neurons, they are simply mathematical models (functions). This model comprises three simple sets of rules: multiplication, summation, and activation. Artificial neuron inputs are weighted, ensuring that each value is considered.is multiplied by individual weight. The sum function in the middle of an artificial neural network adds all weighted inputs and biases. At the exit of an artificial neuron, the total of previously weighted inputs and bias is passed through the activation function, also known as the transfer function.

3.1.7. Convolutional Neural Networks

Convolutional Neural Networks (CNN) are a powerful class of deep learning models that are widely applied to various tasks, including object detection, speech recognition, computer vision [**?**], classification imaging [**? ?**] and bioinformatics. CNN is a feed forward neural network that leverages convolutional structures to extract features from data. Unlike traditional methods, CNN automatically learns and recognizes features in data without the need for manual feature extraction by humans. The design of CNN is inspired by visual perception. The main components of a CNN include a convolutional layer, a pooling layer, and a fully connected layer [31].

3.1.8. Dataset

This study used outpatient examination data from a Taipei Municipal medical center, with 15,000 women aged 20-80 as samples. These ladies were hospitalized between 2018 and 2020, as well as 2021 and 2022, with or without a diabetes diagnosis. The study looked at eight patient parameters, including number of pregnancies, plasma glucose level, diastolic blood pressure, sebum thickness, insulin level, BMI, diabetes pedigree function, and age where the patients with diabetic are 5000 and the healthy patients count is 10000. Initial inspection shows an imbalance in the dataset, with more non-diabetic instances than diabetic ones (Figure 1).

3.1.9. Data Pre-processing

Data pre-processing is a critical step to ensure the effectiveness of AI techniques. Structured data is essential for accurate modeling and prediction.

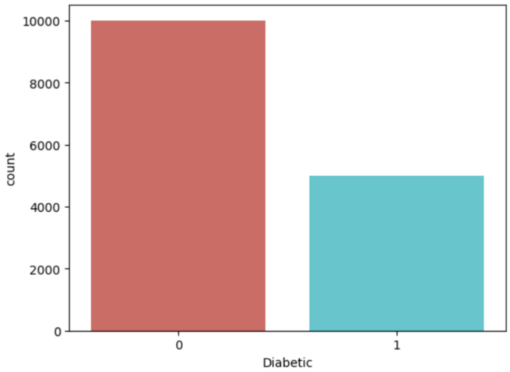

Figure 1: Diabetes and non-diabetes dataset count .

	Selected Features	Description of Selected Features	Range
1	Pregnancies	Number of times a participant is pregnant	0–17
2	PlasmaGlucose	Plasma glucose concentration a 2 h in an oral glucose tolerance test	0–199
3	DiastolicBloodPressure	Diastolic blood pressure (mm Hg), when blood exerts into arteries between heart	0–122
4	TricepsThickness	Triceps skinfold thickness (mm), concluded by the collagen content	0–99
5	SerumInsulin	2-Hour serum insulin (mu U/ml)	0–846
6	BMI	Body mass index (weight in kg/(height in m)^2)	0–67.1
7	DiabetesPedigree	An appealing attribute used in diabetes prognosis	0.078–2.42
8	Age	Age of participants	21–81
9	Diabetic (Outcome)	Diabetes class variable, Yes represents the patient is diabetic and No represents the patient is not diabetic	Yes/No

Table 1
Description of PIMA Indian dataset attributes [7].

1. **Handling Missing Values:** We checked for and addressed any missing values in the dataset by either eliminating rows/columns with missing data or imputing them using statistical methods. In our dataset, there were no missing values, as shown in Figure 2.
2. **Determining Attribute Relevance:** The relevance of each attribute was assessed using Pearson's correlation coefficient, illustrated in Figure 3. This method calculates a correlation coefficient between -1 and 1 to quantify the relationship between input and output properties. A coefficient value above 0.5 or below -0.5 indicates a substantial correlation, while a value close to zero indicates no correlation [32].

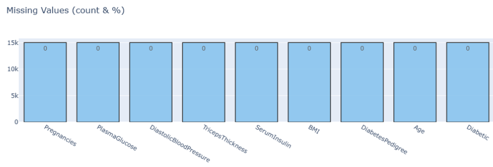

Figure 2: Missing values.

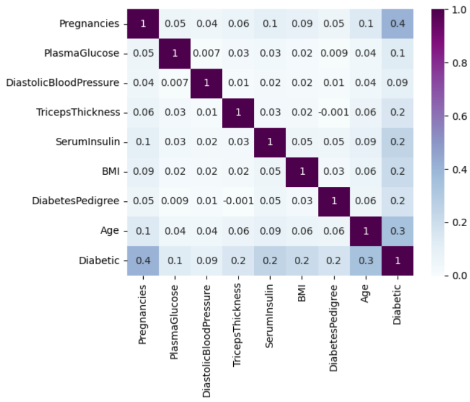

Figure 3: Correlation heatmap.

3. **Balancing the Dataset:** To balance the dataset, we employed the Synthetic Minority Oversampling Technique (SMOTE). This technique ensures an equal number of diabetic and non-diabetic data points, preventing bias caused by under-sampling. SMOTE generates synthetic samples that are close to the existing minority class samples, as demonstrated in our dataset [2].

3.2. Monitoring of diabets

3.2.1. AI Background

An improved version of recurrent neural networks (RNN) called long short-term memory (LSTM) deals with the problem of storing long-term dependencies. The LSTM was first presented by in 1997.The current input in an LSTM network at a given moment in time step and the output from the preceding time step are supplied to the Long Short-Term Memory (LSTM) unit, which produces an output that is forwarded to the subsequent time step. For categorization purposes, the last hidden layer of the last time step—and occasionally all hidden layers—are frequently used [31]. An LSTM network's general architecture is shown in figure 4:

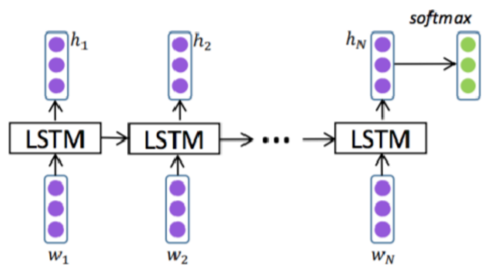

Figure 4: The high- level architecture of LSTM model [31].

By including three gated units—a forget gate, input gate, and output gate—that allow for effective control over the memory of previous states, LSTM circumvents the vanishing gradient problem in RNN .Based on the current input and the prior internal state, the input gate determines how to update the internal state. How much of the prior internal state should be lost is decided by the forget gate. Lastly, the output gate controls how much the internal state affects the system [31].

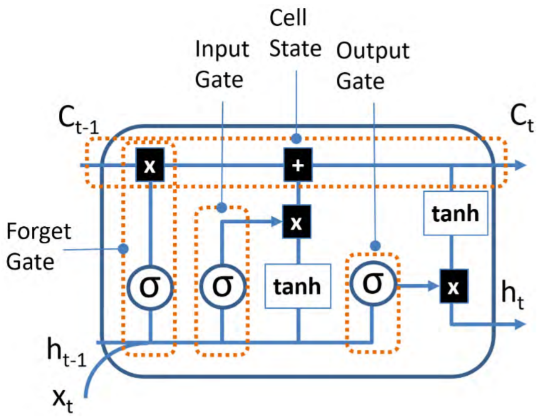

Figure 5: LSTM block with memory cell and gates [33].

Figure 5 shows the input, output, and cell values as C, x, and h. In other words, $t-1$ is from the previous LSTM block (or from time $t-1$), while t indicates the current block. Subscript t indicates the time step value. The hyperbolic tangent function is denoted by tanh, and the sigmoid function is represented by σ. Elements are summed elementwise by operator $+$ and multiplied elementwise by operator $\times$.

The equations below describe how the gates are computed [33].

$$f_t = \sigma(W_f x_t + w_f h_{t-1} + b_f) \tag{2}$$

$$i_t = \sigma(W_i x_t + w_i h_{t-1} + b_i) \tag{3}$$

$$o_t = \sigma(W_o x_t + w_o h_{t-1} + b_o) \tag{4}$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \sigma_c(W_c x_t + w_c h_{t-1} + b_c) \tag{5}$$

$$h_t = o_t \otimes \tanh(c_t) \tag{6}$$

where the input and output gate vectors are denoted by the letters f, i, and o, respectively. W , w, b and $\otimes$ represent weights of input,weights of recurrent output, bias and element-wise multiplication respectively.

3.2.2. Dataset

We trained and evaluated our method on the Ohio T1DM dataset, which is developed to advance research in blood glucose levels. This data was gathered over eight weeks from 12 individuals with type 1 diabetes. Each participant supplied self-reported life events, insulin delivery records, physiological sensor metrics, and continuous glucose monitoring (CGM) data, all of which were anonymized by a random ID. The dataset facilitates research on machine learning with the goal of improving blood glucose level prediction accuracy, which is important for managing diabetes and developing artificial pancreas devices. The dataset contains extensive data points: CGM readings every 5 minutes, blood glucose levels from finger sticks, insulin doses (bolus and basal), self-reported meal times with carbohydrate estimates, exercise, sleep, work, stress, and illness records, along with physiological data from fitness bands. The first cohort used Basis Peak fitness bands, while the second cohort used Empatica Embrace bands, providing detailed metrics such as heart rate, skin temperature, galvanic skin response, and step count. To determine the ideal attribute set for the BG prediction model, we test each of these attributes individually. The quantity of training and test examples for every patient is displayed in Table 2 [24].

3.2.3. Pre-processing data

The preprocessing process consists of several key steps, as illustrated in Figure 6.

Loading Data Each XML file is parsed to create an XML tree, which forms the basis for data extraction.

Patient ID	Gender	Training examples	Test examples
559	Female	10796	2514
563	Male	12124	2570
570	Male	10982	2745
575	Female	11866	2590
588	Female	12640	2791
591	Female	10847	2760

Table 2
Patient data with training and test examples of six patients .

Rounding Timestamps Each timestamp in the collection is rounded to the nearest defined period (in this example, 120 minutes).

Data Extraction The data extraction step depends on the study:

- Study 1: Extracting glucose level.
- Study 2: Extracting glucose level and carbohydrates to analyze their effect on glucose levels.
- Study 3: Extracting glucose level, carbohydrates, and steps.
- Study 4: Extracting glucose level, carbohydrates, and quality of sleep (1 for Poor, 2 for Fair, 3 for Good).
- Study 5: Extracting glucose level, carbohydrates, and intensity of exercise (on a scale of 1 to 10, with 10 being the most physically active).
- Study 6: Extracting glucose level, carbohydrates, quality of sleep, and intensity of exercise.

Merging Data The extracted data are merged into a single DataFrame indexed by the rounded timestamps.

Handling Missing Values Missing values for extracted data expected glucose level are filled with -1, indicating nothing was recorded at those times. Rows with missing glucose levels are dropped to maintain data integrity.

Removing Duplicates Data is grouped by timestamps, and the maximum value for each group is retained to ensure each time span has a distinct entry.

Loading and Splitting Data The integrated data for each patient is loaded, with timestamps converted to datetime objects. The data is split into training (80%) and testing (20]%) sets and then consolidated into two comprehensive data frames: integrated_train_data and integrated_test_data.

Data Scaling The data is normalized using MinMaxScaler to ensure all features are on a similar scale.

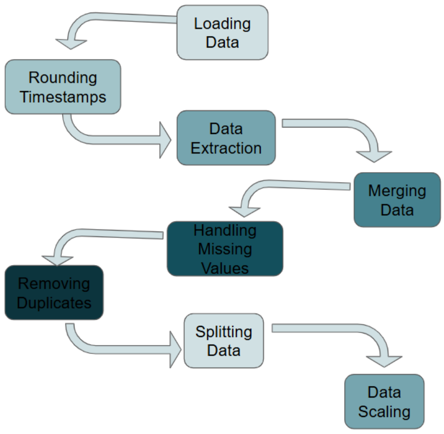

Figure 6: preprocessing steps .

4. Results and Findings

4.1. Diagnosis of diabetes

4.1.1. Performance criteria

Model Evaluation Metrics The following measures were utilized to assess the suggested model. When making predictions about occurrences, there will be four categories of outcomes[34]:

True Positives (TP): Someone with diabetes who was anticipated to develop diabetes.

False Positives (FP): A person who did not have diabetes was projected to have it.

False Negatives (FN): Someone with diabetes was not expected to have diabetes.

True Negatives (TN): A person without diabetes was not expected to have diabetes.

Accuracy (Acc.,) refers to the overall performance of a classifier and its ability to properly predict data [18].as this formula 7:

$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{7}$$

Sensitivity (SN): The metrics describe the classifier's positive results , as follows:

$$\text{Sensitivity (Recall)} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{8}$$

Specificity (Sp.) refers to the negative result discovered by the classifier and is expressed as:

$$\text{Specificity (Sp.)} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{9}$$

Precision (Pr.,) is the ratio of total positive findings to projected positive results , represented as:

$$\text{Precision (Pr.)} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{10}$$

The F1-Score (F1.,) represents the precision and recall harmonic mean, with a range of [0,1]. The F1-Score indicates classifier robustness, with the mathematical expression :

$$F1 = 2 \times \frac{1}{\frac{1}{Recall} + \frac{1}{Precision}} \tag{11}$$

4.1.2. Machine learning results

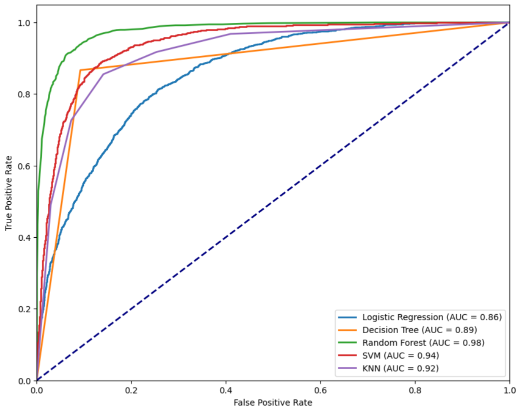

Figure 7: ROC curve figure of five machine-learning models.

The ROC curve plot provides a visual comparison of the performance of the five machine learning models. The Area Under the Curve (AUC) measures the model's ability to discriminate between classes, with a higher AUC indicating better performance. As shown in Figure 7 , the logistic regression model (blue curve) has a moderate AUC, indicating adequate but not optimal performance. The decision tree model (orange curve) performs slightly better with an AUC of 0.89. The random forest model (green curve) demonstrates the best performance with an AUC of 0.98, indicating excellent classification power. The support vector machine (SVM) model (red curve) also performs well with an AUC of 0.94. The K-Nearest Neighbor (KNN) model (purple curve) shows good performance with an AUC of 0.92, although it is slightly less efficient than the random forest and SVM models

Based on the evaluation metrics in Figure 8, the Random Forest classifier stands out as the best-performing model for diabetes prediction in this study achieved the highest accuracy (92.78%) .

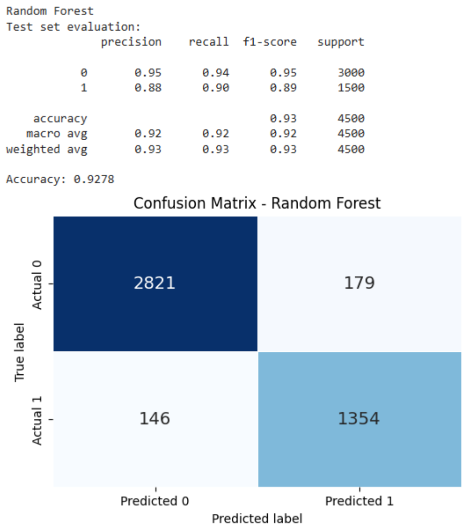

Figure 8: Random Forest .

To ensure accurate evaluation, we conduct two separate investigations using different cross-validation methodologies on the classifier that has shown high accuracy in previous study (Random forest): 10-fold cross-validation and 5-fold cross-validation. We also employ GridSearchCV during hyper-parameter tuning to determine the optimal model parameters.
The Random Forest classifier performed well with both 5-fold and 10-fold cross-validation, achieving overall accuracies of 92.84% and 92.67%, respectively, as shown in the Figure ?? and ??. This performance is comparable to the initial evaluation without cross-validation, which had an accuracy of 92.78% .

4.1.3. Deep learning results

Artificial Neural Network The following table 3 provides a detailed overview of the architecture and training details of the Artificial Neural Network (ANN) employed in this study.

- **Results Before Enhancement Strategies** The initial performance of the ANN was evaluated using the original architecture and training setup. The plots in Figures 9 and 10 indicates that the model performs well and learns efficiently. Both the training and validation accuracy curves indicate a consistent rise, beginning at 0.75 and attaining 0.93 after 50 epochs. The validation accuracy closely tracks the training accuracy, indicating high generalization with minimal overfitting. The

	Details
Building the Neural Network	- Four dense layers with ReLU activation: 64, 32, 16, and 8 neurons respectively. - L2 regularization (0.01) applied to each dense layer. - Batch normalization after the first dense layer. - Dropout (0.2) after each dense layer to prevent overfitting. - Additional dense layer with 4 neurons. - Output layer: Single neuron with sigmoid activation.
Model Compilation	- Loss function: Binary cross-entropy. - Optimizer: Adam optimizer.
Callbacks	- Model checkpoints to save the best model during training. - Early stopping to halt training when performance plateaus.
Enhancement Strategies	- Increased model complexity (add layers). - Implemented k-fold cross-validation (10-fold and 5-fold).

Table 3
Summary of Model Architecture and Training Details

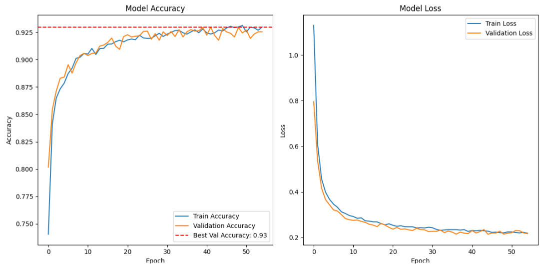

Figure 9: Model Plot .

```
Confusion Matrix:
[[1913  101]
 [ 114  872]]

Classification Report:
              precision    recall  f1-score   support

           0       0.94      0.95      0.95      2014
           1       0.90      0.88      0.89       986

    accuracy                           0.93      3000
   macro avg       0.92      0.92      0.92      3000
weighted avg       0.93      0.93      0.93      3000
```

Figure 10: Confusion matrix and Classification report .

red dashed line at 0.93 represents the best validation accuracy attained, and the model's accuracy plateaus around this value after about 20 epochs, suggesting convergence. The loss curves for both training and validation data fall significantly in the early epochs before stabilizing at low values, showing effective learning and minimal overfitting. Overall, the closely aligned training and validation curves for both accuracy and loss show that the model generalizes effectively to unknown data.

- **Results After Enhancement Strategies** Following the implementation of enhancement strategies, the performance of the ANN was re-evaluated.
  Based on the plot in Figure 11 The validation accuracy curve, although plateauing after a certain epoch (around 30), still remains high throughout. This suggests the model has achieved a good level of performance on unseen data. Even though it might not be significantly improving after that point, it's maintaining a strong performance overall. From the classification report in Figure 12 we observe that the Precision is high for both classes, at 0.94 for class 0 and 0.92 for class 1.Recall (how many of the actual positive cases did the model predict correctly) is also high for both classes, at 0.96 for class 0 and 0.88 for class 1. This means that the model is good at not missing actual positive cases.finally the Accuracy is 0.94, which is also high. This means that the model is performing well overall.

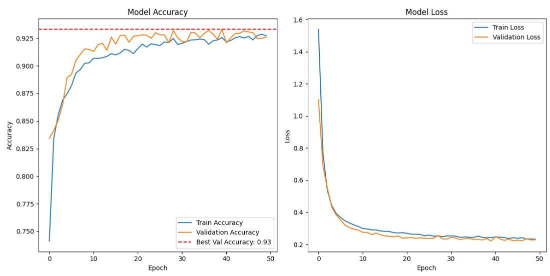

Figure 11: 5-fold cross-validation plot .

Based on the plots in Figure 17, the validation accuracy (orange curve) reaches a high value of

```
Confusion Matrix:
[[1916   87]
 [ 107  890]]

Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.96      0.95      2003
           1       0.91      0.89      0.90       997

    accuracy                           0.94      3000
   macro avg       0.93      0.92      0.93      3000
weighted avg       0.94      0.94      0.94      3000
```

Figure 12: Confusion matrix and Classification report
.

around 0.93, which is a positive sign. The training loss and validation loss generally decrease over time, which is what you expect as the model learns. which suggests the model is generalizing well to unseen data. Based on confusion matrix and classification report the model performs well in classifying occurrences, as evidenced by its 93% overall accuracy. It shows good recall (0.94) and precision (0.95) for class 0, indicating that true negatives can be identified effectively, however there are some misclassifications as class 1. Class 1 precision (0.89) and recall (0.92) are marginally poorer, suggesting that some cases were incorrectly classified as class 0. Overall, the model works well in both classes; however, it could be even more accurate if it could be optimized to accurately categorize instances of class 1.

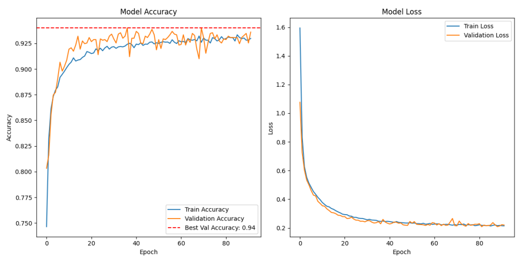

Figure 13: 10-fold cross-validation plot.

CNN model The table 4 summarizes these essential steps for building a CNN architecture, illustrating how each component contributes to the construction and training of the CNN model.

The plot in Figure 15 shows the accuracy of the model over 100 epochs for both training and validation datasets.The validation accuracy is consistently higher than the training accuracy, indicating that the model is

```
Confusion Matrix:
[[923  58]
 [ 44 475]]

Classification Report:
            | precision    recall  f1-score   support

           0       0.95      0.94      0.95       981
           1       0.89      0.92      0.90       519

    accuracy                           0.93      1500
   macro avg       0.92      0.93      0.93      1500
weighted avg       0.93      0.93      0.93      1500
```

Figure 14: Confusion matrix and Classification report
.

	Details
CNN Layers	• **Conv2D Layers:** – Layer 1: 64 filters – Layer 2: 64 filters – Layer 3: 32 filters • **BatchNormalization:** Normalize activations. • **MaxPooling2D:** Reduce spatial dimensions. • **Dropout:** Apply dropout regularization. • **Flatten Layer:** Convert 2D features into a 1D vector. • **Dense Layers:** – Layer 1: 256 units – Layer 2: 128 units – Output Layer: 2 units for binary classification
Model Compilation	• Loss Function: Categorical cross-entropy. • Optimizer: Adam.
Training Strategies	• Early Stopping: Stop training when validation loss stops improving. • Learning Rate Reduction: Reduce learning rate on validation loss plateau.

Table 4
Structured CNN Architecture Steps

performing well on unseen data. In the right plot, both training and validation losses decrease rapidly, indicating effective learning. The close alignment of training and validation losses suggests the model generalizes well and does not suffer from significant overfitting.The confusion matrix (Figure 16) demonstrates that the model has a large number of right predictions for both the negative and positive classes, indicating good performance.

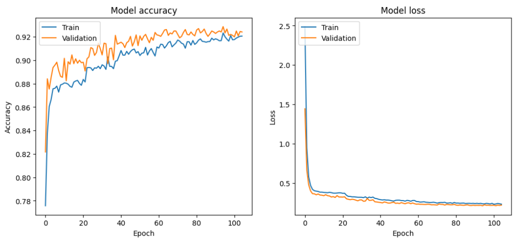

Figure 15: CNN plot.

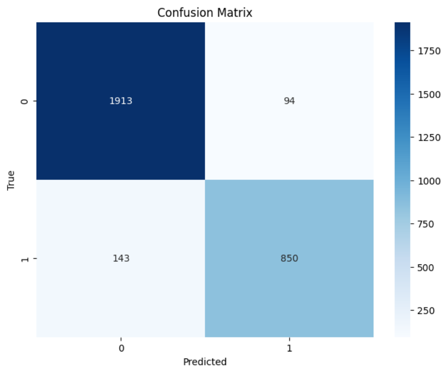

Figure 16: Confusion matrix .

```
===================
Classification Report
              precision    recall  f1-score   support

     Class 0       0.93      0.95      0.94      2007
     Class 1       0.90      0.86      0.88       993

    accuracy                           0.92      3000
   macro avg       0.92      0.90      0.91      3000
weighted avg       0.92      0.92      0.92      3000
```

Figure 17: Classification report.

4.2. Monitoring of diabets

4.2.1. Performance criteria

We utilize two standard performance metrics: root-mean-square error (RMSE) and mean absolute error (MAE). Let x_i be the actual value, $\hat{x}_i$ the predicted value, and n the sample size.

$$\textbf{Root-Mean-Square Error (RMSE)} : \text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2} \tag{12}$$

$$\textbf{Mean Absolute Error (MAE)} : \text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|x_i - \hat{x}_i| \tag{13}$$

RMSE and MAE values range from 0 to $+\infty$, with improved performance resulting from values closer to 0.

4.2.2. Lstm model

Here in table 5 is a summary of the compilation, training, and evaluation details for the LSTM model:

	Details
Compiling the Model	- **Optimizer:** Adam - **Learning Rate:** 0.001 - **Loss Function:** Mean Squared Error (MSE)
Training the Model	- **Epochs:** 100 - **Batch Size:** 32 - **Callbacks:** - Early Stopping: Monitors validation loss, patience=10 - Model Checkpoint: Saves best model based on validation loss to `best_model_g.keras`
Evaluating the Model	- **Metrics:** - Root Mean Squared Error (RMSE) - Mean Absolute Error (MAE)

Table 5
Summary of Model Compilation, Training, and Evaluation

based in figure 24 in all studeis the validation loss generally remains lower than the training loss after the initial epochs, indicating good generalization performance. The consistent patterns across different studies suggest robustness in the model training process.

Based on Table 6 provided for Mean Absolute Error (mae) and Root Mean Squared Error (rmse) for each study, here's a comparison of models performance. Generally, as more features are added to the model, the MAE tends to decrease, indicating improved accuracy in predicting blood glucose levels. Models including sleep quality and exercise intensity show slightly lower MAE values compared to those with fewer features.

Similarly, RMSE decreases as more features are incorporated, suggesting better overall predictive performance. Models with more features show lower RMSE values, indicating more accurate predictions.

Including additional features such as carbs, steps, sleep quality, and exercise intensity consistently improves prediction accuracy (lower MAE and RMSE). However, the differences between models with more features compared to those with fewer are relatively small but generally consistent.

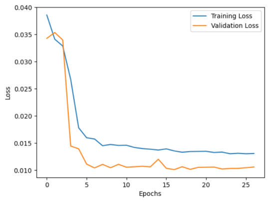

Figure 18: Study 1

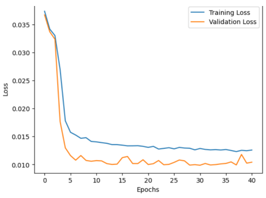

Figure 19: Study 2

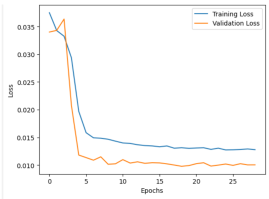

Figure 20: Study 3

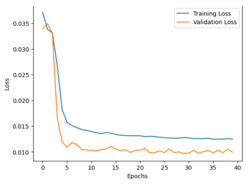

Figure 21: Study 4

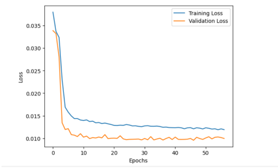

Figure 22: Study 5

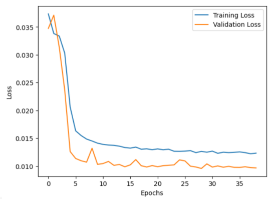

Figure 23: Study 6

Figure 24: Training loss and validation loss.

Study	Factors Analyzed	MAE	RMSE
study1	glucose level	0.07	0.10
study2	glucose level, carbs	0.0755	0.1019
study3	glucose level, carbs, steps	0.07321	0.10018
study4	glucose level, carbs, quality of sleep (1 for Poor; 2 for Fair; 3 for Good)	0.07258	0.09957
study5	glucose level, carbs, intensity of exercise (1 to 10, with 10 the most physically active)	0.07274	0.09993
study6	glucose level, carbs, quality of sleep, intensity of exercise	0.07	0.10

Table 6
Summary of Studies and Evaluation Metrics

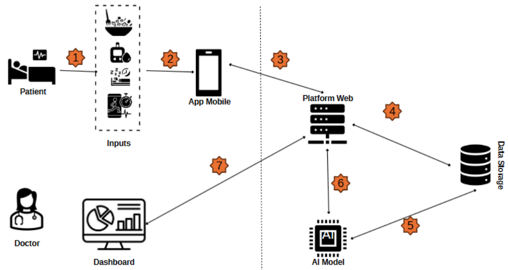

Figure 25: Diabetes monitoring management system.

5. Technical Exploration

This section discusses the prototype implementation of the AI-powered platform for comprehensive diabetes management, focusing on its architecture, key components, and functionalities that are important for ensuring the platform's scalability, usability and effectiveness in real-world clinical settings.

5.1. Software Architecture

The software architecture of the AI-powered diabetes management platform consists of several key components that collaborate to collect, process, store, and analyze data, providing valuable insights and supporting clinical decisions. The general architecture is illustrated in Figure 25 and includes the following steps:

1. **Data Collection:** Patients manually gather essential health data, such as blood glucose levels, carbohydrate intake, quality of sleep, and exercise intensity.
2. **User Interaction:** Patients use a mobile application to enter the collected health data.
3. **Data Transfer:** The mobile application serves as the primary interface for patients to input data, which is then transferred to the web platform for further processing, storage, and analysis. Patients can view their health metrics and receive personalized recommendations..
4. **Data Storage:** The data storage layer securely stores all collected data in a database.
5. **Data Preparation:** The platform formats the new input data to align with the requirements of the AI model.
6. **Analysis and Prediction:** The AI model analyzes the prepared data to generate insights, which are then presented to users.
7. **Informed Decision-Making:** The dashboard provides healthcare providers with detailed health data and trends for their patients. Doctors can use the platform to fill out forms for diagnosis and monitoring. It also offers AI-generated

insights to assist with clinical decisions, which may include changes to prescriptions, lifestyle advice, or scheduling further appointments

5.2. User Interface Design

The foremost aim of the UI design is to create a clean and easy-to-use enjoy for users. This method making the interface simple to navigate, ensuring all functions are smooth to discover, and displaying data in reality. By focusing on user-friendly layout, the platform objectives to improve consumer engagement and satisfaction. The following figures offer an overview of the platform's person interface:

6. Discussion

This study assessed various machine learning methods for diabetes management. Using 5-fold cross-validation, the second DNN architecture achieved the highest accuracy of 94%, demonstrating the effectiveness of deep learning techniques for diabetes prediction. Compared to our results, other studies have shown lower accuracies for most algorithms, with their Random Forest model achieving only 80% accuracy, a notable difference from our 93%. This discrepancy can be attributed to differences in feature selection, data preparation, or hyperparameter optimization methods.

In monitoring results, models incorporating additional relevant features beyond glucose levels exhibited slightly better predictive performance in terms of MAE and RMSE. However, the differences between models were relatively minor, suggesting diminishing returns as more features are added.

It is important to consider that differences in datasets used across studies can significantly impact results. Variations in data characteristics, such as sample size, demographics, and data quality, may influence machine learning model performance. While our study shows promising results, future research should focus on refining models, exploring advanced feature engineering methodologies, and validating these strategies across diverse datasets to ensure robustness and generalizability. Future work may also involve developing new AI models for predicting diabetes-related complications and risk assessments and integrating wearable devices for real-time monitoring to enhance analytics capabilities and improve prediction accuracy.

7. Conclusion

In summary, this study demonstrates the significant potential of AI-powered platforms in transforming diabetes management. By leveraging advanced machine learning and deep learning algorithms, the proposed system effectively predicts diabetes onset, monitors glucose levels, and assists healthcare professionals in providing personalized care.

The performance evaluation indicates that the DNN achieved a validation accuracy of 94%, showcasing its robustness and generalization capabilities. Enhancement strategies, including increased model complexity and k-fold cross-validation, further improved the model's performance, ensuring minimal overfitting and high precision. Similarly, the LSTM model demonstrated a strong ability to predict blood glucose levels, with validation losses indicating good generalization to unseen data.

Moreover, the inclusion of user-friendly interfaces for healthcare professionals and patients ensures that the platform is accessible and practical for everyday use. This fosters better communication between patients and doctors, streamlining the management of health records, prescriptions, and medical analyses.

Future work will focus on expanding dataset diversity, refining AI models, and incorporating real-time patient feedback to further optimize the platform, ultimately improving clinical decision-making and personalizing treatment plans for diabetes care.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT, Grammarly in order to: Grammar and spelling check, Paraphrase and reword. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] R. A. S. K. S. G. M. Chandrashekar D. K, Imran Pasha C, Diabetes prediction using machine learning algorithms, Indian Scientific Journal Of Research In Engineering And Management 07 (2023). doi:10.55041/ijsrem17771.

[2] P. N. Thotad, A machine learning-based diagnosis and prediction of diabetes mellitus disease (2023).

[3] P. Sireesha, M. K. Prakash, D. Sumathi, Implementing a model to detect diabetes prediction using machine learning classifiers, Journal of Algebraic Statistics 13 (2022) 558–566.

[4] C. Napoli, V. Ponzi, A. Puglisi, S. Russo, I. Tibermacine, et al., Exploiting robots as healthcare resources for epidemics management and support caregivers, in: CEUR Workshop Proceedings, volume 3686, CEUR-WS, 2024, pp. 1–10.

[5] B. Nail, M. A. Atoussi, S. Saadi, I. E. Tibermacine, C. Napoli, Real-time synchronisation of multiple

fractional-order chaotic systems: an application study in secure communication, Fractal and Fractional 8 (2024) 104.
[6] I. Naidji, A. Tibermacine, W. Guettala, I. E. Tibermacine, et al., Semi-mind controlled robots based on reinforcement learning for indoor application., in: ICYRIME, 2023, pp. 51–59.
[7] M. Rahman, D. Islam, R. J. Mukti, I. Saha, A deep learning approach based on convolutional lstm for detecting diabetes, Computational biology and chemistry 88 (2020) 107329.
[8] I. E. Tibermacine, A. Tibermacine, W. Guettala, C. Napoli, S. Russo, Enhancing sentiment analysis on seed-iv dataset with vision transformers: A comparative study, in: Proceedings of the 2023 11th international conference on information technology: IoT and smart city, 2023, pp. 238–246.
[9] S. Bouchelaghem, I. E. Tibermacine, M. Balsi, M. Moroni, C. Napoli, Cross-domain machine learning approaches using hyperspectral imaging for plastics litter detection, in: 2024 IEEE Mediterranean and Middle-East Geoscience and Remote Sensing Symposium (M2GARSS), IEEE, 2024, pp. 36–40.
[10] N. Boutarfaia, S. Russo, A. Tibermacine, I. E. Tibermacine, Deep learning for eeg-based motor imagery classification: Towards enhanced human-machine interaction and assistive robotics, life 2 (2023) 4.
[11] J. J. Khanam, S. Y. Foo, A comparison of machine learning algorithms for diabetes prediction, Ict Express 7 (2021) 432–439.
[12] A. Tibermacine, I. E. Tibermacine, M. Zouai, A. Rabehi, Eeg classification using contrastive learning and riemannian tangent space representations, in: 2024 International Conference on Telecommunications and Intelligent Systems (ICTIS), IEEE, 2024, pp. 1–7.
[13] S. Russo, I. E. Tibermacine, A. Tibermacine, D. Chebana, A. Nahili, J. Starczewscki, C. Napoli, Analyzing eeg patterns in young adults exposed to different acrophobia levels: a vr study, Frontiers in Human Neuroscience 18 (2024) 1348154.
[14] S. Russo, S. Ahmed, I. E. Tibermacine, C. Napoli, Enhancing eeg signal reconstruction in cross-domain adaptation using cyclegan, in: 2024 International Conference on Telecommunications and Intelligent Systems (ICTIS), IEEE, 2024, pp. 1–8.
[15] A. Tibermacine, D. Akrour, R. Khamar, I. E. Tibermacine, A. Rabehi, Comparative analysis of svm and cnn classifiers for eeg signal classification in response to different auditory stimuli, in: 2024 International Conference on Telecommunications and Intelligent Systems (ICTIS), IEEE, 2024, pp. 1–8.
[16] J. Martinsson, A. Schliep, B. Eliasson, O. Mogren, Blood glucose prediction with variance estimation using recurrent neural networks, Journal of Healthcare Informatics Research 4 (2020) 1–18.
[17] S. eddine Boukredine, E. Mehallel, A. Boualleg, O. Baitiche, A. Rabehi, M. Guermoui, A. Douara, I. E. Tibermacine, Enhanced performance of microstrip antenna arrays through concave modifications and cut-corner techniques, ITEGAM-JETIA 11 (2025) 65–71.
[18] N. P. Miriyala, R. L. Kottapalli, G. P. Miriyala, G. Lorenzini, C. Ganteda, V. A. Bhogapurapu, et al., Diagnostic analysis of diabetes mellitus using machine learning approach, Revue d'Intelligence Artificielle 36 (2022) 347–352.
[19] T. Zhu, K. Li, P. Herrero, P. Georgiou, Deep learning for diabetes: a systematic review, IEEE Journal of Biomedical and Health Informatics 25 (2020) 2744–2757.
[20] A. TIBERMACINE, W. GUETTALA, I. E. TIBERMACINE, Efficient one-stage deep learning for text detection in scene images., Electrotehnica, Electronica, Automatica 72 (2024).
[21] B. Nail, B. Djaidir, I. E. Tibermacine, C. Napoli, N. Haidour, R. Abdelaziz, Gas turbine vibration monitoring based on real data and neuro-fuzzy system, Diagnostyka 25 (2024).
[22] B. Ladjal, I. E. Tibermacine, M. Bechouat, M. Sedraoui, C. Napoli, A. Rabehi, D. Lalmi, Hybrid models for direct normal irradiance forecasting: A case study of ghardaia zone (algeria), Natural Hazards 120 (2024) 14703–14725.
[23] G. Swapna, R. Vinayakumar, K. Soman, Diabetes detection using deep learning algorithms, ICT express 4 (2018) 243–246.
[24] M. F. Rabby, Y. Tu, M. I. Hossen, I. Lee, A. S. Maida, X. Hei, Stacked lstm based deep recurrent neural network with kalman smoothing for blood glucose prediction, BMC Medical Informatics and Decision Making 21 (2021) 1–15.
[25] H. V. Dudukcu, M. Taskiran, T. Yildirim, Blood glucose prediction with deep neural networks using weighted decision level fusion, Biocybernetics and Biomedical Engineering 41 (2021) 1208–1223.
[26] R. Krishnamoorthi, S. Joshi, H. Z. Almarzouki, P. K. Shukla, A. Rizwan, C. Kalpana, B. Tiwari, et al., A novel diabetes healthcare disease prediction framework using machine learning techniques, Journal of Healthcare Engineering 2022 (2022).
[27] L. Fregoso-Aparicio, J. Noguez, L. Montesinos, J. A. García-García, Machine learning and deep learning predictive models for type 2 diabetes: a systematic review, Diabetology & metabolic syndrome 13 (2021) 148.
[28] N. Ahmed, R. Ahammed, M. M. Islam, M. A. Uddin, A. Akhter, M. A. Talukder, B. K. Paul, Machine learning based diabetes prediction and development of smart web application, International Journal of

Cognitive Computing in Engineering 2 (2021) 229–241.
[29] K. Shiruru, An introduction to artificial neural network, International Journal of Advance Research and Innovative Ideas in Education 1 (2016) 27–30.
[30] A. Krenker, J. Bešter, A. Kos, Introduction to the artificial neural networks, Artificial Neural Networks: Methodological Advances and Biomedical Applications. InTech (2011) 1–18.
[31] F. M. Shiri, T. Perumal, N. Mustapha, R. Mohamed, A comprehensive overview and comparative analysis on deep learning models: Cnn, rnn, lstm, gru, arXiv preprint arXiv:2305.17473 (2023).
[32] L. P. Nguyen, D. D. Tung, D. T. Nguyen, H. N. Le, T. Q. Tran, T. V. Binh, D. T. N. Pham, The utilization of machine learning algorithms for assisting physicians in the diagnosis of diabetes, Diagnostics 13 (2023) 2087.
[33] A. Shrestha, A. Mahmood, Review of deep learning algorithms and architectures, IEEE access 7 (2019) 53040–53065.
[34] C.-Y. Chou, D.-Y. Hsu, C.-H. Chou, Predicting the onset of diabetes with machine learning methods, Journal of Personalized Medicine 13 (2023) 406.