http://doi.org/10.35784/iapgos.7835 received: 30.05.2025 | revised: 05.12.2025 | accepted: 15.12.2025 | available online: 20.12.2025

INFORMATICS AND MEASUREMENT IN HEALTHCARE: DEEP LEARNING FOR DIABETIC PATIENT READMISSION PREDICTION

Shiva Saffari, Mahdi Bahaghighat
Imam Khomeini International University, Computer Engineering Department, Qazvin, Iran

Abstract. *Approximately 460 million individuals were living with diabetes globally in 2023. This study explores and contrasts methods for forecasting hospital readmissions among diabetic patients by integrating traditional approaches with modern deep learning frameworks. In this work, a variety of deep learning architectures – including recurrent models like LSTM and GRU, as well as CNNs and Autoencoders – are examined along with conventional machine learning approaches. Four essential metrics – accuracy, precision, recall, and F1-score – were employed to measure and compare the effectiveness of different models. The results revealed that deep neural network methods significantly outperformed classical machine learning algorithms. Among traditional methods, the Decision Tree achieved the highest effectiveness. However, the LSTM network demonstrated superior performance, achieving scores of 0.74 for accuracy, 0.73 for precision, 0.74 for recall, and 0.73 for the F1-score. Additionally, the GRU and Vanilla LSTM models exhibited performance close to the best model, indicating that recurrent networks are more suitable for this problem than traditional methods.*

Keywords: diabetes, readmission prediction, neural networks, machine learning, deep learning

INFORMATYKA I POMIARY W OPIECE ZDROWOTNEJ: GŁĘBOKIE UCZENIE SIĘ W CELU PRZEWIDYWANIA PONOWNYCH HOSPITALIZACJI PACJENTÓW Z CUKRZYCĄ

Streszczenie. *W 2023 r. na całym świecie na cukrzycę cierpiało około 460 milionów osób. Niniejszy artykuł analizuje i porównuje metody prognozowania ponownych hospitalizacji pacjentów z cukrzycą poprzez połączenie tradycyjnych podejść z nowoczesnymi frameworkami głębokiego uczenia się. W ramach niniejszej pracy przeanalizowano różne architektury głębokiego uczenia się – w tym modele rekurencyjne, takie jak LSTM i GRU, a także CNN i autoenkodery – wraz z konwencjonalnymi podejściami do uczenia maszynowego. Do pomiaru i porównania skuteczności różnych modeli wykorzystano cztery podstawowe wskaźniki – dokładność, precyzję, czułość i F1-score. Wyniki wykazały, że metody głębokich sieci neuronowych znacznie przewyższały klasyczne algorytmy uczenia maszynowego. Spośród metod tradycyjnych najwyższą skuteczność osiągnęło drzewo decyzyjne. Jednak sieć LSTM wykazała się lepszą wydajnością, osiągając wyniki 0,74 dla dokładności, 0,73 dla precyzji, 0,74 dla czułości i 0,73 dla F1-score. Ponadto modele GRU i Vanilla LSTM wykazały wydajność zbliżoną do najlepszego modelu, co wskazuje, że sieci rekurencyjne są bardziej odpowiednie dla tego problemu niż metody tradycyjne.*

Słowa kluczowe: cukrzyca, uczenie maszynowe, uczenie głębokie, sieci neuronowe, głębokie modele

Introduction

Diabetes mellitus is a multifactorial metabolic disorder marked by sustained hyperglycemia caused by inadequate insulin secretion [17]. It is one of the most significant global public health challenges, drawing immense attention from researchers and clinicians due to the rising number of patients and the high cost of care. According to a 2023 report by the WHO, more than 460 million individuals globally were living with diabetes, and this number continues to rise [33]. In 2017, global estimates placed the diabetic population at around 425 million, with projections suggesting a rise to 629 million by 2045. This grim trend is driven primarily by accelerating rates of obesity and unhealthy lifestyles characterized by poor diets and physical inactivity. Moreover, these lifestyle habits are strongly influenced by broader social factors, such as global dietary changes – a phenomenon known as the "nutrition transition" [12].

Many challenges exist in managing diabetes; among them, the most significant is the frequent hospital readmission of patients due to ineffective disease management after discharge. In this context, hospital readmission refers to the situation in which a patient, after being discharged from care, returns for an additional round of treatment. Common reasons for these readmissions include symptomatic relapses, noncompliance with prescribed treatment regimens, and a lack of proper follow-up care – all of which can adversely affect healthcare outcomes and inflate medical costs. Therefore, predicting the probability of readmission among diabetic patients within 30 days post-discharge is crucial [25]. Such predictions can facilitate more efficient treatment strategies, reduce hospital costs, and improve overall patient care [11, 32].

Predicting hospital readmission in diabetic patients is commonly approached through supervised learning techniques. These approaches rely on various patient-related features, including demographic characteristics, clinical history, and lab test outcomes, to estimate readmission probability [9]. In recent years, conventional algorithms such as logistic regression and SVM, alongside more advanced deep architectures like Convolutional and Recurrent Neural Networks, have also been employed to improve prediction performance. Despite their potential, these algorithms face practical challenges, including limited accuracy and difficulties in handling complex datasets.

Prior research suggests that failing to adopt advanced models – and neglecting modern techniques for data integration, hybrid modeling, and data optimization – can result in reduced accuracy and performance. Moreover, deep learning approaches, which utilize complex, multi-layered neural architectures, can learn layered feature abstractions directly from raw input data. In contrast, conventional ML techniques typically depend on handcrafted input features, which may not be as effective at detecting intricate patterns [34].

This paper aims to develop better prediction of diabetic patient hospital readmission through designing and tuning deep neural network models. In order to do this, we have trained and compared a large number of classification models – from traditional classification techniques, including logistic regression, decision tree algorithms, and SVM, to more complex ones (such as CNNs, LSTMs, hybrid CNN–LSTM models, GRUs, vanilla LSTM networks, and even autoencoders). Our study reached a significant level of accuracy over previous work by performing an extensive analysis of multi-dimensional datasets of persons with diabetes.

The paper first provides an overview of previous studies in Section 1, and then presents a comprehensive discussion on the proposed methodology and implementation details in Section 2. Section 3 presents a thorough explanation of the evaluation criteria, while Section 4 discusses the results obtained by the proposed models along with comparisons to earlier research. Lastly, the final section summarizes and presents propositions for future research.

1. Related works

Many of the articles discussed in this section have used the same dataset that we employed in our study to predict hospital readmissions among diabetic patients (see Section 2.1). This dataset contains clinical records from diabetic patients at over 130 hospitals in the United States and is widely used in research on hospital readmission prediction and healthcare system performance improvement, owing to its significance and standing in the medical community.

According to reference [4], several computational algorithms were assessed to estimate the risk of hospital re-entry among individuals suffering from diabetes. This research explores how various algorithms perform – logistic regression, SVM, tree-based ensembles like random forest, and gradient boosting models such as LightGBM [5] – after applying a set of preprocessing techniques. According to the results, LightGBM achieved slightly superior performance compared to the other models. Nevertheless, the research presents certain shortcomings, including limited consideration of influential variables, exclusive reliance on the AUC metric, and a lack of comprehensive feature analysis for enhancing predictive accuracy.

Another study also applied machine learning methods to predict hospital readmissions for diabetic patients. The findings in [3] suggest that the random forest algorithm achieved superior performance compared to the other evaluated models. Additionally, cost analysis showed that these models could contribute to significant financial savings in the management of diabetic patients. One drawback of this research lies in its exclusive reliance on the AUPRC metric for assessing model performance. This narrow focus may limit the depth of evaluation, as it does not take into account other critical measures like overall accuracy, recall, or the F1-score.

[29] examines the issue of predicting hospital readmissions for diabetic patients within 30 days after discharge. Given the rising treatment costs and complications associated with diabetes, this study seeks to enhance health management and reduce costs by accurately predicting patient readmissions. The study utilized feature reduction methods, including raw sets, Boruta and recursive feature elimination and found that recursive feature elimination delivered the best performance in improving prediction accuracy. Evaluation outcomes revealed that the C4.5 decision tree was the least accurate model, achieving only 59.67% accuracy with the Boruta feature selection method. Conversely, using the SVM algorithm in combination with a feature selection technique based on recursive elimination yielded the highest classification accuracy, which reached 63.97% [29].

Another study attempted to forecast the probability of readmission of hospitalised diabetic patients with several data mining models. Among all the models evaluated in this study, the SVM algorithm showed the highest overall performance, achieving an accuracy score of 63.38%. Other techniques used to improve the accuracy of predictions include mix of different techniques for feature reduction and data re-classification [22].

A different investigation examined a range of supervised learning methods – such as k-Nearest Neighbors (KNN), logistic models, probabilistic classifiers like naive Bayes, tree-based algorithms, ensemble strategies like random forests, and boosting techniques – and found that boosting models yielded the highest predictive accuracy. The accuracy of this model reached approximately 65% in the testing phase; however, the recall and precision were reported as 57% and 18%, respectively, highlighting significant challenges in accurately identifying patients at risk of readmission [13].

In a different study, researchers evaluated the performance of four classification algorithms: naive Bayes, random forest, logistic regression, and a multilayer perceptron neural network. Without feature reduction results of Multilayer Perceptron were 61.91% and Logistic Regression gave even slightly better result of 61.97%. This system Multilayer Perceptron achieved accuracy of 62.14% when feature selection was applied. In this study, it was noted that feature selection may affect prediction accuracy but not significantly in this case [1].

Building on previous research, this study leverages advanced deep learning techniques and a systematic model design to achieve high accuracy in analyzing diabetic patient data. Moreover, the study avoids relying on just one evaluation metric by applying a comprehensive assessment framework that includes precision, recall, F1-score, and accuracy. This broader perspective enhances the understanding of model performance and emphasizes how the proposed method improves upon earlier approaches.

2. Methodology

This research addresses the challenge of forecasting hospital reentries among individuals with diabetes, a critical issue faced by modern healthcare systems. Readmission of diabetic patients causes higher healthcare costs, poor quality of life for patients and additional pressure on health systems. Hence, detecting patients that have a higher risk of readmission, which allows optimizing treatment management and reducing costs. Having multifactorial nature, the complex nature of patient data along with the need of efficient ways to analyze this data is one of the biggest challenges of this issue.

2.1. Dataset

For this research, the dataset selected was the "Diabetes 130-US Hospitals for years 1999-2008" dataset [8] from the UCI Machine Learning Repository. The information about diabetic patients [4] in this dataset is varied and the sources of this data are among the most credible, so it has received the most references in the context of relevant scientific communities [3–5]. By analyzing these data, we wish to predict the 30 day readmissions to a hospital and identify patterns associated with readmissions so that we can predict them better. There are 101,766 records and 50 different features including patient ID, race, gender, age, weight, type of admission, time in hospital, number of medical tests, medication, and diagnosis related information.

2.2. Data preprocessing

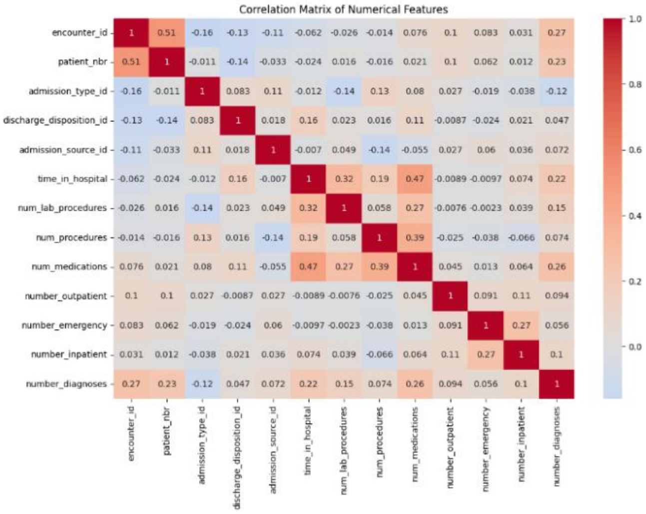
The dataset was loaded and checked for the number of samples, columns, missing values and datatypes. As per the analysis we have 13 numerical features and 37 categorical features in the dataset. Table 1 summarizes the numerical variables used in the study, while Table 2 displays a selection of the categorical attributes.

Table 1. summary of the dataset's numerical attributes

Feature Name	Attribute Definition
encounter_id	Identifier for each hospital visit
patient_nbr	Identifier for the patient
admission_type_id	Type of admission
discharge_disposition_id	Type of discharge
admission_source_id	Source of admission
time_in_hospital	Length of stay (days)
num_lab_procedures	Count of laboratory tests conducted
num_procedures	Count of clinical procedures
num_medications	Number of medications administered
number_outpatient	Count of ambulatory care encounters
number_emergency	Frequency of emergency room admissions
number_inpatient	Frequency of patient hospitalizations
number_diagnoses	Total diagnostic codes assigned to the patient

Table 2. Summary of the dataset's categorical attributes

Feature Name	Attribute Definition
race	Patient's race
gender	Patient's gender
age	Grouped age category of the patient
weight	Patient's weight
payer_code	Payer code
medical_specialty	Related medical specialty
diag_1	Primary diagnosis
diag_2	Secondary diagnosis
diag_3	Tertiary diagnosis
max_glu_serum	Maximum glucose serum level
A1Cresult	Hemoglobin A1C test result
Various medications (e.g., metformin, insulin)	Medication usage status (No, Up, Down, Steady)
change	Change in medications (Yes/No)
diabetesMed	Use of diabetes medications (Yes/No)
readmitted	Patient readmission status (NO, <30, >30)



	encounter_id	patient_nbr	admission_type_id	discharge_disposition_id	admission_source_id	time_in_hospital	num_lab_procedures	num_procedures	num_medications	number_outpatient	number_emergency	number_inpatient	number_diagnoses
encounter_id	1	0.51	-0.16	-0.13	-0.11	-0.062	-0.026	-0.014	0.076	0.1	0.083	0.031	0.27
patient_nbr	0.51	1	-0.011	-0.14	-0.033	-0.024	0.016	-0.016	0.021	0.1	0.062	0.012	0.23
admission_type_id	-0.16	-0.011	1	0.083	0.11	-0.012	-0.14	0.13	0.08	0.027	-0.019	-0.038	-0.12
discharge_disposition_id	-0.13	-0.14	0.083	1	0.018	0.16	0.023	0.016	0.11	-0.0087	-0.024	0.021	0.047
admission_source_id	-0.11	-0.033	0.11	0.018	1	-0.007	0.049	-0.14	-0.055	0.027	0.06	0.036	0.072
time_in_hospital	-0.062	-0.024	-0.012	0.16	-0.007	1	0.32	0.19	0.47	-0.0089	-0.0097	0.074	0.22
num_lab_procedures	-0.026	0.016	-0.14	0.023	0.049	0.32	1	0.058	0.27	-0.0076	-0.0023	0.039	0.15
num_procedures	-0.014	-0.016	0.13	0.016	-0.14	0.19	0.058	1	0.39	-0.025	-0.038	-0.066	0.074
num_medications	0.076	0.021	0.08	0.11	-0.055	0.47	0.27	0.39	1	0.045	0.013	0.064	0.26
number_outpatient	0.1	0.1	0.027	-0.0087	0.027	-0.0089	-0.0076	-0.025	0.045	1	0.091	0.11	0.094
number_emergency	0.083	0.062	-0.019	-0.024	0.06	-0.0097	-0.0023	-0.038	0.013	0.091	1	0.27	0.056
number_inpatient	0.031	0.012	-0.038	0.021	0.036	0.074	0.039	-0.066	0.064	0.11	0.27	1	0.1
number_diagnoses	0.27	0.23	-0.12	0.047	0.072	0.22	0.15	0.074	0.26	0.094	0.056	0.1	1

Fig. 1. Heatmap of feature correlations

Figure 1 illustrates a heatmap depicting the relationships among the dataset's numerical variables. Each cell displays the Pearson correlation coefficient between a pair of features. Coefficients near 1 or -1 reflect strong positive or negative relationships, respectively, whereas values close to 0 suggest minimal or no correlation. The analysis highlights several notable correlations, including a relatively strong correlation between patient_nbr and encounter_id (0.512) and a correlation between num_medications and time_in_hospital (0.466). Additionally, other significant correlations are observed, such as between num_medications and num_procedures (0.386) and between num_lab_procedures and time_in_hospital (0.318). Such insights reveal interdependencies among variables, which can inform more effective feature selection strategies and contribute to dimensionality reduction efforts in model development.

In this study, the target variable is the 'readmitted' attribute, which reflects if and when a patient returns to the hospital – categorized as within 30 days, after 30 days, or not readmitted at all. Analyzing the distribution of values in the target column revealed a significant class imbalance. Specifically, 53.9% of patients fall into the "no readmission" (NO) category, 34.9% correspond to the ">30 days readmission" group, while only 11.2% belong to the "<30 days readmission" category (see Figure 2). This uneven distribution highlights the need for data balancing techniques for enhancing model effectiveness. presents a bar chart illustrating how target classes are distributed, with evident differences in sample counts between the classes. Such an imbalance in class distribution may adversely affect both the predictive accuracy and the generalization capability of the models, thus requiring careful handling during model development. Since the primary goal is to predict readmission within 30 days, this becomes a binary classification problem, where values less than 30 days are considered class 1 and other values are considered class 0.

To address missing values in continuous variables, the median was used as a replacement, given its resilience to the influence of outliers. Categorical fields were encoded numerically through label encoding to facilitate compatibility with machine learning models. Next, all numerical features were standardized by adjusting them to have a zero mean and a standard deviation of one – a step that enhances model stability during training and improves prediction accuracy. The data was partitioned into three distinct portions: 60% allocated for model learning, 20% used during the validation phase and for tuning hyperparameters, and the remaining 20% set aside for test phase. To address class imbalance within the data, the SMOTE algorithm (Synthetic Minority Over-sampling Technique) was applied. This approach generates artificial samples for the underrepresented class, which helps balance the dataset and enhances the model's ability to detect meaningful structures. This preprocessing step prepared the data for modeling and improved its quality.

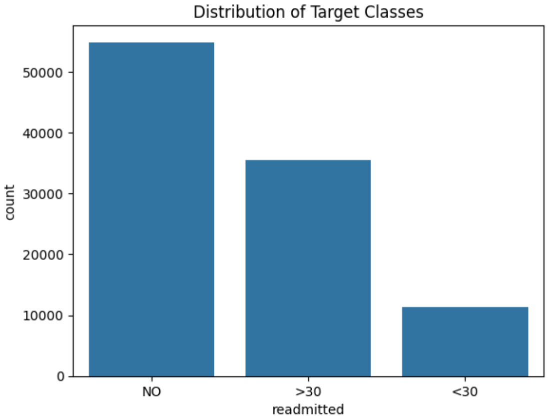


Fig. 2. Distribution chart of target column values

2.3. Models

This study utilized several classical machine learning techniques – such as logistic regression, decision tree algorithms, and support vector machines – to assess their performance in comparison with more complex architectures, including CNNs and LSTM networks, among other deep learning approaches.

Logistic regression

In medical and epidemiological studies, logistic regression remains a widely adopted approach for modeling outcome variables [20]. Unlike linear regression, which produces a continuous range of output values, logistic regression utilizes the sigmoid function to map predictions to a probability score between 0 and 1. This number signifies the likelihood of a specific class being predicted. The benefits of this model are its simplicity, interpretability, and effectiveness when data are linear and separable [24]. The core parameters of the model, along with their respective values and brief descriptions, are detailed in Table 3 below.

Table 3. Parameter setup for implementing the logistic regression algorithm

Parameter	Value Used	Parameter Info
solver	'lbfgs'	Optimization algorithm
max_iter	1000	Maximum number of iterations
C	0.1	Regularization strength parameter
penalty	'l2'	Type of regularization
class_weight	'balanced'	Adjusts class weights to handle class imbalance

Decision tree

Decision tree models offer a simple yet intuitive approach to classification by mapping possible outcomes through a hierarchical structure of conditions. This model consists of nodes, where each node selects a feature from the data and compares it to a specific value. This process continues until the model divides the data into final categories [31]. The advantages of decision trees include their simplicity of interpretation and the lack of need for complex preprocessing. On the downside, this model can overfit when working with large datasets. Table 4 presents the key parameters of this model, along with the default values and relevant explanations.

Table 4. Parameters and settings of the decision tree model

Parameter	Value Used	Parameter Info
criterion	'gini'	Criterion for splitting the nodes
max_depth	5	Limits how many levels the tree can grow downward from the root node
min_samples_split	2	Smallest number of samples needed to divide a non-leaf (internal) node
min_samples_leaf	20	Minimum required samples for a node to be considered a final (leaf) node
class_weight	'balanced'	Adjust class weights to address class imbalance
random_state	42	Set the initial value for reproducible results

Support Vector Machine (SVM)

Support Vector Machines (SVMs) belong to the category of supervised learning techniques and are applicable to both classification and regression problems, although they are more commonly employed for classification tasks [15]. SVM works by determining the hyperplane that best separates the two classes by maximizing the margin between their closest data points [14]. This property of the algorithm allows to effectively handle complex and non-linear problems; thus, it can be effectively applied in the domains of pattern recognition and image analysis [10]. The main parameters of this model with default values and descriptions are given in Table 5.

Table 5. Parameters and settings for the support vector machine model

Parameter	Value Used	Parameter Info
C	1	Penalty factor controlling the decision boundary's smoothness
kernel	'rbf'	Kernel function that transforms input data into feature space
gamma	'scale'	Parameter related to the kernel function
random_state	42	Initial value for reproducible results
probability	True	Enables class probability estimation
class_weight	'balanced'	Adjusts class weights to address class imbalance

Convolutional Neural Networks (CNNs)

CNNs are a class of neural architecture used in deep learning, specifically developed to process structured data arranged in a grid-like format, such as images. Their key advantage is their capability to autonomously learn informative representations from raw image data through deep learning methods [27]. CNNs are particularly effective in applications like categorizing images, detecting objects, and recognizing visual patterns, owing to their strength in capturing spatial dependencies within data [2]. In this research, despite working with non-image data, CNNs were employed to extract hidden patterns and complex relationships between features. This decision was made due to the effectiveness of these networks in identifying connections across different dimensions of the data.

The architecture of this model utilizes several convolutional layers and feature analysis layers to extract complex features. The proposed architecture is composed of three sequential convolutional blocks. The initial block integrates two convolutional layers, each with 64 filters, and applies batch normalization and max pooling after every layer. Dropout is incorporated to reduce the risk of overfitting. In the subsequent block, the number of filters is doubled to 128, enabling the extraction of more abstract and complex features. The third block further increases filter depth to 256, and integrates a global average pooling layer to reduce dimensionality. Finally, several dense layers are added following the convolutional components to further refine extracted features and enhance the model’s predictive accuracy. These layers have 256 and 128 neurons and L2 regularization is used to prevent overfitting. To finalize the network architecture, an output layer composed of fully connected neurons and governed by a sigmoid activation function is integrated, enabling the model to produce probabilistic outcomes for binary classification tasks.

Long Short-Term Memory (LSTM)

This model has two main concepts – cell state and gates. The cell state also keeps the information across the sequence and it is the memory of the network [6]. In contrast, Gateways manages the cell status information. These essentially separate networks of neurons decide which the information that will remain in the state of the cell [21]. They must also learn to select what knowledge to retain or forget. LSTM cells contain three primary gates – input, forget, and output – each of which is responsible for selectively managing the flow of information over time. Based on this structure, the activation of the input gate can be expressed approximately as follows [16]:

$$i_t = \sigma(\mathrm{W_i}[h_{t-1}.x_t] + \mathrm{b_i}) \tag{1}$$

$$C_t = \tanh(\mathrm{W_c}\,.\,[h_{t-1}.x_t] + \mathrm{b_c}) \tag{2}$$

Consider i_t to be the input gate, the prior hidden state (h_{t-1}), and the current input (x_t); the new value is then calculated accordingly [16]:

$$C_t = f_t * C_{t-1} + i_t * C_t \tag{3}$$

where C_{t-1} is the previous cell state, f_t is the forget gate, i_t is the input gate, and C_t is the updated cell state at time step t.

The architecture implemented in this work includes an LSTM layer composed of 128 units. This architecture is particularly advantageous when working with sequential inputs, as it enables the model to effectively learn temporal dynamics and retain long-range contextual relationships.

Since the neurons in this layer are sequentially connected, it effectively transfers prior time-step information to the current processing step. The hyperbolic tangent (tanh) activation function is used in this layer to normalize input values within the range of -1 to 1. o_t and h_t can be obtained according to Equations (4) and (5) [16]:

$$o_t = \sigma(\mathrm{W_o}[h_{t-1}.x_t] + \mathrm{b_o}) \tag{4}$$

$$h_t = o_t * \tanh(C_t) \tag{5}$$

where o_t is the output gate activation, h_t denotes the hidden state, and C_t represents the cell state.

LSTM networks utilize the Sigmoid activation function within their gate structures to control how information is retained and transmitted. It is similar to the hyperbolic tangent, but the values are within a scale of 0 to 1. That is useful for data updates or for forgetting. The f_t is computed using Equation (6) [16]:

$$f_t = \sigma(\mathrm{W_f}.\,[h_{t-1}.x_t] + \mathrm{b_f}) \tag{6}$$

where b_f represents the bias term, while f_t denotes the activation of the forget gate.

The architecture of this model is three recurrent layers of LSTM network with 128 units, 256 units and 128 units respectively. There are batch normalisation layers after each of the recurrent layers and a dropout layer at the end to avoid overfitting. After the recurrent components, the architecture includes dense layers containing 128 and 64 neurons, each employing ReLU activation. The network concludes with a densely connected output layer employing a sigmoid activation, designed to generate probability estimates corresponding to each class in a binary decision task. This model's proposed architecture is displayed in Figure 3.

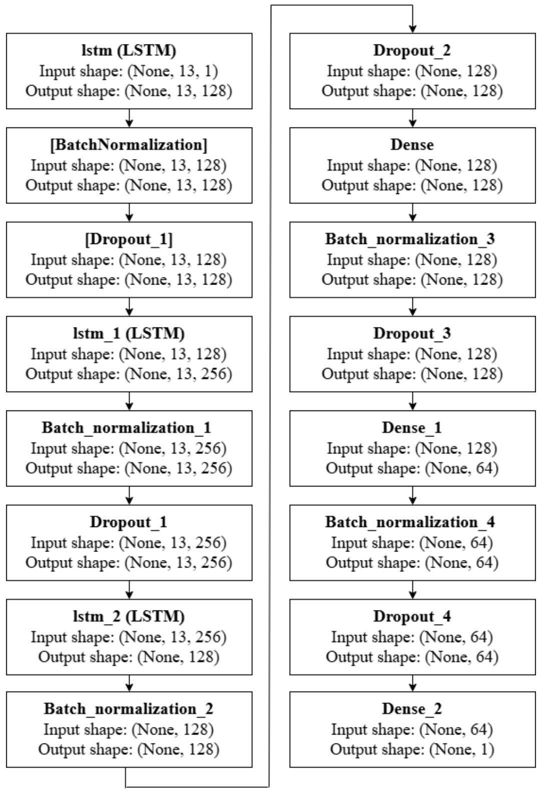


Fig. 3. Architecture overview utilizing LSTM model

Vanilla LSTM

The Vanilla LSTM architecture includes a stack of three LSTM layers comprising 128, 256, and 128 units, respectively. To enhance model generalization and stability, batch normalization and dropout techniques are applied following the recurrent layers. Subsequently, two dense layers with 128 and 64 units are employed, ending in an output node activated by a sigmoid function to facilitate binary classification. Prior to training, the input data is reshaped to conform with the model's expected input format.

Hybrid CNN-LSTM

The model architecture begins with convolutional layers designed to capture meaningful patterns and feature interactions across the input variables. It includes two convolutional layers with 64 and 128 filters, each configured with a 3-by-3 receptive field. These are subsequently processed through batch normalization and max pooling operations to improve training stability and reduce data dimensionality. The processed feature maps are then passed to two LSTM layers containing 256 and 128 units, which help identify complex dependencies and temporal structures. The model concludes with two fully connected layers, where ReLU functions are applied to hidden units and a sigmoid function is used in the output layer to generate binary classification predictions for patient readmission.

Autoencoder as a feature extractor

An autoencoder is a specialized neural network architecture developed to perform dimensionality reduction by learning efficient, compressed representations of input data [18]. The model updates its weights via backpropagation, aiming to reduce the reconstruction error that measures how much the decoder's output deviates from the original input [30].

In this study, an autoencoder framework is utilized, consisting of two key components: an encoding unit and a decoding unit. The encoder maps input x onto a lower-dimensional latent representation z. Mathematically, the encoder can be described as follows:

1) It comprises a stack of fully connected layers. Each layer transforms its input through a linear projection, applies a ReLU nonlinearity, and incorporates dropout to help prevent overfitting. Specifically, for the *i*-th layer ($i = 1, 2, 3$).

$$h_i = \mathrm{Dropout}(\sigma(\mathrm{W_i} h_{i-1} + \mathrm{b_i})) \tag{7}$$

where h_{i-1} denotes the output vector from the preceding layer (with h_0 = x). $\mathrm{W_i}$ and $\mathrm{b_i}$ represent the trainable parameters – specifically, the weight matrix and bias term – associated with layer i. σ refers to the non-linear activation function applied in this context – specifically, the Rectified Linear Unit (ReLU), represented by the function $\sigma(z) = \max(0, z)$. Dropout operates by deactivating a random subset of neurons (rate = 0.3) at each training iteration, which helps prevent overfitting.

2) Encoding Layer: The final encoding layer produces the latent representation z:

$$z = \sigma(\mathrm{W_e} h_3 + \mathrm{b_e}) + \lambda \parallel z \parallel_1 \tag{8}$$

where $\mathrm{W_e}$ and $\mathrm{b_e}$ represent the learnable parameters – the weights and biases – associated with the encoder. λ is the regularization strength for the $\mathrm{L_1}$ activity regularizer, which encourages sparsity in z.

The decoder reconstructs the latent representation z back to the original input dimensions ˆx:

1) Fully Connected Layers: Each layer performs a linear transformation, followed by applying the ReLU function to introduce non-linearity.

$$h'_i = \sigma(\mathrm{W'_i} h'_{i-1} + \mathrm{b'_i}) \tag{9}$$

where h'_{i-1} denotes the activation output obtained from the previous layer in the network (with $h'_0 = z$). $\mathrm{W'_i}$ and $\mathrm{b'_i}$ are the weight matrix and bias term assigned to the *i*-th layer of the decoder.

2) Output Layer: The reconstruction is carried out by the final layer, which applies the tanh activation function to produce output values within the range of −1 to 1.

$$\hat{x} = \tanh(\mathrm{W_o} h'_3 + \mathrm{b_o}) \tag{10}$$

where $\mathrm{W_o}$ and $\mathrm{b_o}$ refer to the trainable parameters linked to the network's final output layer – namely, its weight matrix and corresponding bias component.

The training objective of the autoencoder is to reduce the discrepancy between the input and its reconstruction, which is commonly quantified by the Mean Squared Error (MSE) metric:

$$L_{AE}(x.\hat{x}) = \frac{1}{\mathrm{n}} \sum_1^n (x_i - \hat{x}_i)^2 \tag{11}$$

where n denotes the size of the input vector, referring to the complete set of input features.

In this work, the encoder includes three consecutive fully connected layers with 512, 256, and 128 neurons. Each layer incorporates dropout with a probability of 0.3, serving as a regularization mechanism to mitigate overfitting. In the decoder section, the encoded data is progressively reconstructed to its original dimensions through three fully connected layers with 128, 256 and 512 neurons.

The same data is also fed into a classification model using the original features (without the use of the autoencoder). The classification network is built with two intermediate layers, containing 64 and 32 units respectively. The model is designed to classify if a patient is likely to return for hospital admission, formulated as a binary prediction task. It consists of:

1) Fully connected layers, where each hidden layer performs a linear mapping followed by the ReLU non-linearity:

$$h_j = \sigma(\mathrm{W_j} h_{j-1} + \mathrm{b_j}) \tag{12}$$

where $\mathrm{h_{j-1}}$ corresponds to the output from the preceding layer (with $\mathrm{h_0}$ = x). $\mathrm{W_j}$ and $\mathrm{b_j}$ are trainable parameters defining the transformation applied in the *j*-th layer, specifically its weight matrix and associated bias term.

2) Output Layer: A sigmoid activation is applied in the output layer to produce a probability value p, which reflects the estimated chance of the patient being readmitted.

$$p = \sigma(\mathrm{W_o} h_2 + \mathrm{b_o}) \tag{13}$$

where $\mathrm{W_o}$ and $\mathrm{b_o}$ represent the parameters associated with the output layer, namely the weight matrix and bias term. The sigmoid activation maps the output into a probability score ranging from 0 to 1.

$$\sigma(z) = \frac{1}{1+\mathrm{e}^{-z}} \tag{14}$$

where z is the input to the sigmoid function.

The classification model is trained using binary cross-entropy loss as Equation (15):

$$L_{CLF}(y,p) = -\frac{1}{\mathrm{N}}\sum_{1}^{\mathrm{N}}(y_i \log(pi) + (1-yi)\log(1-pi)) \tag{15}$$

where y_i represents the binary label (0 or 1) assigned to the i-th sample. p_i indicates the model's estimated probability that the corresponding instance will result in readmission. N denotes the complete sample size used in the dataset.

Gated Recurrent Unit (GRU) model

This architecture is composed of three sequential layers containing 128, 256, and 128 units, respectively. This model includes dense layers consisting of 128 and 64 units, with normalization and dropout applied to improve generalization. Finally, the output layer applies a sigmoid function. Figure 4 illustrates the proposed architecture based on this model.

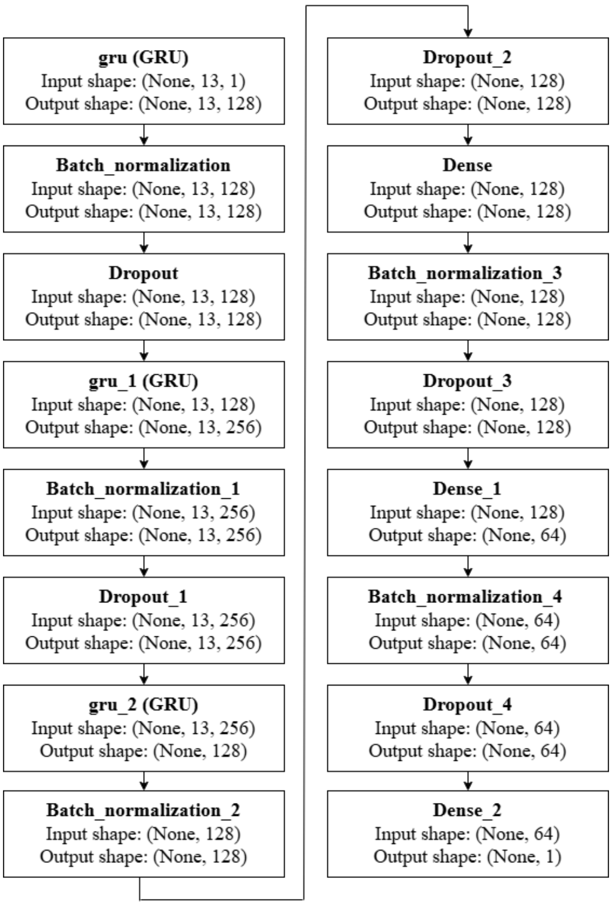


Fig. 4. The suggested model architecture utilizing GRU networks

2.4. Model training setup and hyperparameters

The experiments were conducted in a Google Colab environment with a Tesla T4 GPU (16 GB GPU memory) and approximately 14 GB of system RAM. The models were implemented in Python 3.12 using TensorFlow, PyTorch, Pandas, NumPy, scikit-learn, Matplotlib, and Seaborn, ensuring reproducibility and computational efficiency.

All neural models were optimized using the Adam algorithm, with a 0.0001 learning rate, a choice made due to its robustness and consistent results across various deep learning scenarios. Model performance was assessed using the binary cross-entropy criterion, which is well-suited for tasks involving two-class classification. The model employed a batch size of 32 during training, selected as a compromise between computational efficiency and training stability. Training was conducted over up to 100 training cycles with an early termination mechanism, which terminated the process if no performance gain was observed for 15 consecutive epochs. To further stabilize learning, the scheduler decreased the learning rate tenfold when validation performance plateaued for 7 epochs, with a minimum threshold of 10^{-6}. After training, evaluation was performed on the models based on a set of established performance measures.

3. Performance metrics

This study focuses on prediction of 30-day readmission of diabetic patients and use several metrics to measure models' performance. The evaluation utilized common classification metrics, including accuracy, precision, recall, F1-score, and the confusion matrix. Their definitions, along with their significance to the current research, are presented in the subsequent sections.

3.1. Accuracy

Accuracy, as defined in this work, represents the ratio of all model predictions that correctly classified cases of either 30-day readmission or no readmission. But, accuracy by itself is not enough to measure the model performance since the dataset could be imbalanced. For instance, if most patients have a "no readmission" then a model that predicts all patients are "no readmission" will get relatively high accuracy, but will not allow to identify patients that are actually going to be readmitted. The formula of accuracy is presented in Equation (16) [7, 26].

$$Accuracy = \frac{TP+TN}{Tp+TN+FP+FN} \tag{16}$$

- True Positives (TP): Cases where the model accurately identified patients who were actually readmitted within 30 days.
- False Positives (FP): Instances in which the model wrongly classified a patient as being readmitted within 30 days, although no readmission actually occurred.
- True Negatives (TN): Instances in which the model accurately determined that the patient was not readmitted.
- False Negatives (FN): Situations in which the model failed to predict readmission, classifying a patient as not readmitted while they were in fact readmitted.

3.2. Precision

Precision, in this study, refers to the proportion of correctly predicted 30-day readmissions relative to all cases the model classified as readmissions. This metric is especially relevant in contexts where false positive outcomes carry significant consequences or risks. For example, if patients are incorrectly classified as likely to be readmitted within 30 days, they may receive unnecessary additional treatments. The expression for computing precision is provided in Equation (17) [15].

$$Precision = \frac{TP}{Tp+FP} \tag{17}$$

3.3. Recall

Recall in this study indicates how well the model identifies patients who were actually readmitted within 30 days. This metric is particularly critical when the priority is to identify all potential readmission cases, even at the expense of occasionally mislabeling low-risk patients. Equation (18) presents the formula for calculating recall [26].

$$Recall = \frac{TP}{Tp+FN} \tag{18}$$

3.4. F1-score

The F1-score is obtained by combining precision and recall through their harmonic formulation, enabling a balanced assessment of both. In this work, where the main objective is to forecast 30-day hospital readmissions, this measure assists in identifying models that appropriately trade-off between sensitivity to true readmissions and control over false positives. Equation (19) outlines its mathematical formulation [7].

$$F1 - score = \frac{2*Precision*Recall}{Precision+Recall} \qquad (19)$$

3.5. Confusion matrix

The confusion matrix is a structured table that organizes prediction outcomes into four categories (TP, FP, TN and FN). It sheds light on how effectively the model can separate patients likely to be readmitted within 30 days from those who are not [23, 28].

In this study, evaluation metrics are selected according to the specific research objectives. As the primary aim is to predict hospital readmission of diabetic patients within 30 days, recall and precision serve as the core indicators. Recall is emphasized to ensure that high-risk individuals are not missed, while precision helps to reduce false positives and thereby lower healthcare costs. The F1-score, an aggregated metric, reflects the balance between correctly identifying relevant cases (recall) and achieving accuracy in positive classifications (precision). To offer a comprehensive evaluation of model performance, this study considers multiple criteria – namely classification accuracy, precision, recall, F1-score, and confusion matrix analysis. This multi-faceted assessment aids in identifying the most appropriate model aligned with the study's objectives and strengthens the prediction of hospital readmissions among diabetic patients.

4. Results and discussion

Among all tested architectures, the LSTM model yielded the highest performance metrics, with 0.74 for accuracy, 0.73 for precision, 0.74 for recall, and an F1-score of 0.73. Its capacity to model intricate relationships and represent non-linear dependencies – particularly in identifying positive cases and maintaining high recall – establishes it as the top-performing model. The results underscore how deep learning techniques can effectively handle complex data structures.

The Vanilla LSTM model achieved performance metrics closely aligned with those of the standard LSTM, reporting values of 0.73 for accuracy, 0.72 for precision, 0.73 for recall, and 0.72 for the F1-score. This demonstrates that strong performance can still be achieved using a simpler architecture.

The GRU architecture delivered solid performance, achieving 0.71 in accuracy, 0.70 in precision, 0.71 in recall, and an F1-score of 0.70. Nonetheless, its outcomes were marginally outperformed by the LSTM-based configurations. Given GRU's relatively simpler structure, this minor performance gap was anticipated. Nevertheless, it remains a competitive option among the evaluated models.

The hybrid CNN-LSTM model achieved 69% accuracy, 72% precision, 69% recall, and a 70% F1-score. These results indicate that combining CNN and LSTM did not significantly enhance overall accuracy or performance. While its accuracy is higher than some other models, the lower recall suggests a reduced ability to identify patients likely to be readmitted within the 30-day window.

The CNN model achieved 70% accuracy, 72% precision, 70% recall, and an F1 value of 71%, indicating performance on par with the hybrid CNN-LSTM architecture. However, given its simpler architecture, it provided relatively better efficiency.

The autoencoder model achieved 67% accuracy, 69% precision, 67% recall, and a 68% F1-score. This suggests that representation learning-based methods can also yield acceptable results. Although this model lagged behind other neural network-based methods, it still demonstrated superior performance relative to classical machine learning techniques.

Among traditional models, the decision tree achieved the most favourable results, with an accuracy of 67%, precision of 69%, recall of 67%, and an F1-score of 68%. Its ability to capture nonlinear relationships contributed to its relatively high accuracy.

Logistic regression, with 55% accuracy, 56% precision, 55% recall, and a 56% F1-score, exhibited weak performance, indicating its limitations in identifying complex patterns within medical data. Its lack of flexibility in handling intricate relationships reduced its effectiveness in correctly predicting readmissions.

The SVM model recorded the lowest performance among all tested methods, with respective values of 51% for accuracy, 57% for precision, 51% for recall, and 52% for the F1-score. Its low recall resulted in failing to identify a significant number of actual readmission cases, limiting its reliability for this task.

As shown in Figure 5, the confusion matrix illustrates how effectively the LSTM model detects patients likely to be readmitted within 30 days. The model correctly classified 18,443 patients as not readmitted, demonstrating its strong capability in identifying those who do not return after hospitalization. Additionally, it accurately detected 5,903 readmitted patients, highlighting its effectiveness in recognizing at-risk individuals. The model demonstrates strong discriminatory power between the two categories, indicating its robustness in forecasting patient readmission.

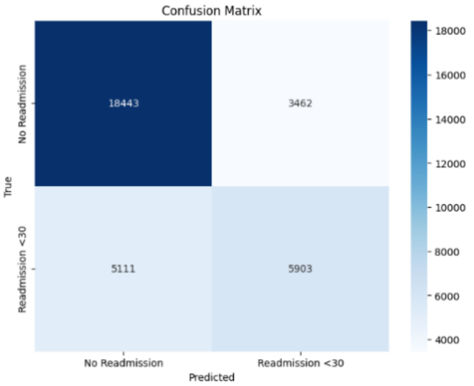


Fig. 5. LSTM-based confusion matrix for patient readmission prediction

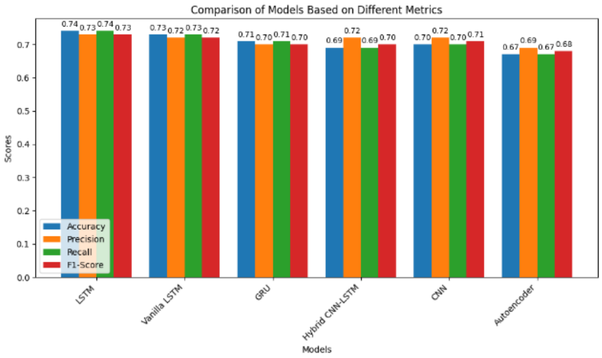


Fig. 6. Performance metrics visualization for various deep learning architectures

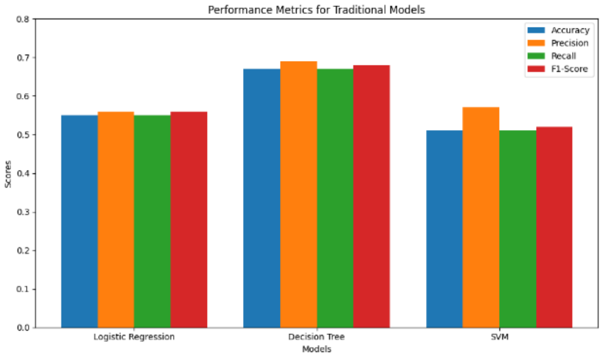


Fig. 7. Visual comparison of key performance measures in machine learning models

Figure 6 illustrates how various deep learning models perform across the defined evaluation metrics, whereas Figure 7 showcases a comparable analysis for classical machine learning approaches. The key findings from these analyses are outlined in Table 6.

The analysis indicates that deep neural networks and conventional models each exhibit unique operational traits, benefits, and constraints. Among various evaluation criteria, the F1-score – which captures the interplay between precision and recall – serves as a dependable indicator of model quality, particularly in the presence of class imbalance. However, for this specific problem, recall is considered the most critical evaluation metric. This is because minimizing the false negative rate – ensuring high-risk patients are correctly identified – is essential for both clinical and practical reasons. Missing even a small number of high-risk patients expected to be readmitted within the 30-day period could have severe and potentially life-threatening consequences. In contrast, while a higher false positive rate may result in unnecessary treatments, it does not carry the same immediate risk to the patient's life.

Table 6 presents the performance metrics for all evaluated models. Achieving a recall of 0.74, the LSTM model showed the highest effectiveness in detecting individuals who may require readmission within 30 days. These findings indicate that the model performs well in detecting a substantial number of clinically at-risk individuals, a factor that significantly contributes to evidence-based medical decisions and the refinement of therapeutic planning. Conversely, classical approaches such as logistic regression and SVM yielded weaker results – likely due to their limited capacity to model the complex structures present in healthcare data.

Given the nature of the problem, the most suitable model selection approach should prioritize improving recall while maintaining acceptable levels for other metrics. Recurrent models like LSTM are able to learn more intricate patterns by capturing long-term relationships in the data, which enables them to more accurately identify high-risk patients. Overall, the results emphasize the steady superiority of deep learning techniques compared to conventional methods.

Table 6. Comparative overview of model performance outcomes

Prediction Method	Accuracy	Precision	Recall	F1-score
LSTM	0.74	0.73	0.74	0.73
Vanilla LSTM	0.73	0.72	0.73	0.72
GRU	0.71	0.70	0.71	0.70
Hybrid CNN-LSTM	0.69	0.72	0.69	0.70
CNN	0.70	0.72	0.70	0.71
Autoencoder	0.67	0.69	0.67	0.68
Logistic Regression	0.55	0.56	0.55	0.56
Decision Tree	0.67	0.69	0.67	0.68
SVM	0.51	0.57	0.51	0.52

To further illustrate the stability and consistency of each model across different random seeds, Table 7 presents the performance metrics (accuracy and macro F1-score) for four runs for each of the four top models. This detailed analysis highlights the variability of model performance and provides a more robust assessment of each architecture's effectiveness in predicting patient readmission.

Table 7. Model performance stability across different seeds

Model	Test 1 (Acc/F1)	Test 2 (Acc/F1)	Test 3 (Acc/F1)	Test 4 (Acc/F1)
LSTM	0.7321/0.74	0.7292/0.73	0.7103/0.71	0.7250/0.73
Vanilla LSTM	0.7280/0.70	0.7300/0.71	0.7319/0.71	0.7294/0.70
GRU	0.7275/0.70	0.7392/0.71	0.7329/0.70	0.7342/0.71
Hybrid CNN-LSTM	0.6750/0.68	0.6987/0.70	0.6930/0.70	0.7159/0.71

From the results in Table 7, it can be observed that the LSTM-based models generally maintain higher accuracy and F1-scores across different runs, whereas the Hybrid CNN-LSTM exhibits lower metrics and greater variability. These findings further support the earlier conclusion that recurrent architectures, particularly LSTM, are the most effective for this task.

Table 8 provides a comparative summary of the top-performing models developed in this study alongside methodologies adopted in related research. The results underscore the improved performance of the proposed deep learning approaches relative to traditional methods in predicting readmissions among diabetic patients.

Table 8. Comparison of top results from this study and previous research

Prediction Method	Accuracy	Precision	Recall	F1-score	Source
LSTM	0.74	0.73	0.74	0.73	Proposed
Vanilla LSTM	0.73	0.72	0.73	0.72	Proposed
GRU	0.71	0.70	0.71	0.70	Proposed
Autoencoder	0.67	0.69	0.67	0.68	Proposed
Neural Network	0.65	-	-	0.28	[5]
SVM	63.97	64.90	60.90	-	[29]
SVM	63.38	-	-	-	[22]
Gradient Boosting	65.00	0.18	0.57	-	[13]
MLP	61.91	-	-	-	[1]
Logistic Regression	61.97	-	-	-	[1]

5. Conclusion

In recent years, deep learning techniques have had a profound impact on advancements in medical diagnostic systems. The capability of deep learning systems to interpret and derive insights from vast and intricate datasets has brought significant innovation to the medical domain, supporting more precise, streamlined, and individualized healthcare solutions [18, 19]. This research demonstrates how deep learning can transform clinical prediction tasks and offers a scalable framework for addressing similar challenges in the broader healthcare context. This represents a perennial challenge in healthcare and demonstrates the power of deep learning to forecast hospital readmissions within 30 days among diabetic patients. The authors leveraged the "Diabetes 130-US Hospitals" dataset available through the UCI Machine Learning Repository to analyze the effects of various patient features, demographics, medical history, and treatment details, on readmission rates, to uncover patterns driving readmissions. These results offer practical implications for reducing readmissions and improving health outcomes. By contrasting advanced neural network architectures – such as LSTM, GRU, CNN, and autoencoders – with classical machine learning techniques like Logistic Regression, Decision Trees, and SVM, the analysis demonstrates the relative advantages offered by deep learning methodologies. The LSTM model generated the highest accuracy (0.74), precision (0.73), F1-score (0.73), and recall (0.74), indicating a considerable improvement compared to traditional techniques. The GRU and Vanilla LSTM models achieved comparable performance, further underscoring the importance of recurrent neural networks in modelling intricate healthcare data.

6. Future research

Future studies could focus on enhancing hybrid architectures by incorporating advanced techniques such as transformers and exploring larger, more diverse datasets to improve both accuracy and generalizability. Additionally, enhancing model efficiency to minimize computational costs while integrating the strengths of both traditional and deep learning approaches could provide significant added value. Furthermore, future work could explore strategies to facilitate the practical adoption of these models in real-world clinical settings, such as their integration into clinical information systems, without compromising performance or efficiency.

Disclosure statement: The authors report there are no competing interests to declare.

Funding statement: There is no funding resource for this study.

Conflict of interest: There is no conflict of interest to declare.

Ethical Approval and Data Availability Statement: This study was conducted using a publicly available, fully de-identified dataset that ethical approval and informed consent were not required [8]. The dataset, titled Diabetes 130-US hospitals for years 1999–2008, is freely accessible from UCI Machine Learning Repository, 2014 [https://doi.org/10.24432/C5230J], for research and educational purposes.

References

[1] Ahmed M., Kabir M. J.: Dimensionality reduction in predicting hospital readmissions of diabetics. 5th International Conference on Advances in Electrical Engineering (ICAEE), 2019, 155–159 [https://doi.org/10.1109/ICAEE48663.2019.8975432].

[2] Alzubaidi L. et al.: Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. J Big Data 8(1), 2021, 53 [https://doi.org/10.1186/s40537-021-00444-8].

[3] Bhuvan M. S. et al.: Identifying diabetic patients with high risk of readmission. 2016, arXiv:1602.04257 [https://doi.org/10.48550/arXiv.1602.04257].

[4] Bojja G. R., El-Gayar O.: Predicting hospital readmissions of diabetic patients – A machine learning approach. Annual Research Symposium, Mar. 2019. [https://scholar.dsu.edu/research-symposium/24].

[5] Cynthia L. et al.: Boosting efficiency in a clinical literature surveillance system with LightGBM. PLOS Digital Health 3(9), 2024, e0000299 [https://doi.org/10.1371/journal.pdig.0000299].

[6] Chiu C. C. et al.: Predicting ICU readmission from electronic health records via BERTopic with long short term memory network approach. Journal of Clinical Medicine 13(18), 2024, 5503 [https://doi.org/10.3390/jcm13185503].

[7] Chicco D., Jurman G.: The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics 21(1), 2020, 6 [https://doi.org/10.1186/s12864-019-6413-7].

[8] Clore K. C. J.: Diabetes 130-US hospitals for years 1999–2008. UCI Machine Learning Repository, 2014 [https://doi.org/10.24432/C5230J].

[9] Cox M. et al.: Risk stratification with explainable machine learning for 30-day procedure-related mortality and 30-day unplanned readmission in patients with peripheral arterial disease. PLOS ONE 17(11), 2022, e0277507 [https://doi.org/10.1371/journal.pone.0277507].

[10] Drogkoula M., Kokkinos K., Samaras N.: A comprehensive survey of machine learning methodologies with emphasis in water resources management. Applied Sciences 13(22), 2023 [https://doi.org/10.3390/app132212147].

[11] Emi-Johnson O. G., Nkrumah K. J.: Predicting 30-day hospital readmission in patients with diabetes using machine learning on electronic health record data. Cureus 17(4), 2025, 82437 [https://doi.org/10.7759/cureus.82437].

[12] Forouhi N. G., Wareham N. J.: Epidemiology of diabetes 2019 [https://doi.org/10.17863/CAM.35630].

[13] Gawande R., Patil D. V.: Prediction of frequent hospitalisation of diabetic patients. Social Science Research Network, Rochester NY: 4043544, Feb. 25, 2022 [https://doi.org/10.2139/ssrn.4043544].

[14] Guido R. et al.: An overview on the advancements of support vector machine models in healthcare applications: A review. Information 15(4), 2024, 235 [https://doi.org/10.3390/info15040235].

[15] Hassan S. U., Ahamed J., Ahmad K.: Analytics of machine learning-based algorithms for text classification. Sustainable Operations and Computers 3, 2022, 238–248 [https://doi.org/10.1016/j.susoc.2022.03.001].

[16] Hochreiter S., Schmidhuber J.: Long Short-Term Memory. Neural Computation 9(8), 1997, 1735–1780 [https://doi.org/10.1162/neco.1997.9.8.1735].

[17] Hu T. L. et al.: Machine learning-based predictions of mortality and readmission in type 2 diabetes patients in the ICU. Applied Sciences 14(18), 2024, 8443 [https://doi.org/10.3390/app14188443].

[18] Jorge T. et al.: Early identification of out-of-roundness damage wheels in railway freight vehicles using a wayside system and a stacked sparse autoencoder. Vehicle System Dynamics 63(2), 2025, 232–257 [https://doi.org/10.1080/00423114.2024.2333771].

[19] Khomsi Z. et al.: Evaluating the feasibility of thermographic images for predicting breast tumor stage using DCNN. Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska 14(1), 2024, 99–104 [https://doi.org/10.35784/iapgos.5555].

[20] Levy J. J., O'Malley A. J.: Don't dismiss logistic regression: the case for sensible extraction of interactions in the era of machine learning. BMC Med Res Methodol 20(1), 2020, 171 [https://doi.org/10.1186/s12874-020-01046-3].

[21] Lyu Y. et al.: A heart failure classification model from radial artery pulse wave using LSTM neural networks. BMC Medical Informatics and Decision Making 25(1), 2025, 318 [https://doi.org/10.1186/s12911-025-02318-5].

[22] Munnangi H., Chakraborty G.: Predicting readmission of diabetic patients using the high performance support vector machine algorithm of SAS® Enterprise Miner™. SAS Global Forum, 2015.

[23] Mousavi S., Bahaghighat M., Ozen F.: Advancements in phishing website detection: A comprehensive analysis of machine learning and deep learning models. 32nd Signal Processing and Communications Applications Conference (SIU), 2024, 1–4 [https://doi.org/10.1109/SIU61531.2024.10601036].

[24] Park H. A.: An introduction to logistic regression: From basic concepts to interpretation with particular attention to nursing domain. J. Korean Acad. Nurs. 43(2), 2013, 154–164 [https://doi.org/10.4040/jkan.2013.43.2.154].

[25] Pinem J., Astuti W., Adiwijaya K.: Explainable ensemble learning framework with SMOTE, SHAP and LIME for predicting 30-day readmission in diabetic patients. Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi) 9(5), 2025, 784–791 [https://doi.org/10.29207/resti.v9i5.6977].

[26] Sadr H. et al.: Cardiovascular Disease Diagnosis: A Holistic Approach Using the Integration of Machine Learning and Deep Learning Models. European Journal of Medical Research 29(1), 2024, 455 [https://doi.org/10.1186/s40001-024-02044-7].

[27] Sarafraz I., Agahi H., Mahmoodzadeh A.: Convolutional neural network (CNN) configuration using a learning automaton model for neonatal brain image segmentation. PLoS ONE 20(1), 2025, e0315538 [https://doi.org/10.1371/journal.pone.0315538].

[28] Shamseen A. et al.: Developing a parallel classifier for mining in big data sets. IIUM Engineering Journal 22(2), 2021 [https://doi.org/10.31436/iiumej.v22i2.1541].

[29] Shih D. H. et al.: Thirty-day re-hospitalization rate prediction of diabetic patients. Journal of Internet Technology 21(7), 2020. [https://jit.ndhu.edu.tw/article/view/2444].

[30] Shrestha A., Mahmood A.: Review of deep learning algorithms and architectures. IEEE Access 7, 2019, 53040–53065 [https://doi.org/10.1109/ACCESS.2019.2912200].

[31] Somvanshi M. et al.: A review of machine learning techniques using decision tree and support vector machine. International Conference on Computing Communication Control and Automation (ICCUBEA), 2016, 1–7 [https://doi.org/10.1109/ICCUBEA.2016.7860040].

[32] CDC: Methods for the National Diabetes Statistics Report. Diabetes [https://www.cdc.gov/diabetes/php/data-research/methods.html] (accessed: 25.10.2024).

[33] WHO: Diabetes [https://www.who.int/health-topics/diabetes] (accessed: 25.08.2024).

[34] IABAC®: What is the difference between machine learning and deep learning. IABAC® [https://iabac.org/blog/what-is-the-difference-between-machine-learning-and-deep-learning] (accessed: 25.10.2024).

Shiva Saffari
e-mail: Shiva.saa21@gmail.com

She is an M.Sc. student in software engineering at Imam Khomeini International University, where she is deeply passionate about artificial intelligence (AI), machine learning (ML), and data science. As a research assistant at the Artificial Intelligence and Smart Systems Technology (AIST) Lab. She actively contributes to cutting-edge research projects that leverage AI and ML to solve complex problems.

https://orcid.org/0009-0001-5696-0823

Ph.D. Mahdi Bahaghighat
e-mail: Bahaghighat@eng.ikiu.ac.ir

He is an assistant professor at Imam Khomeini International University, where he is a driving force in the fields of artificial intelligence, machine learning, and data science. He is the visionary founder of the Artificial Intelligence and Data Armor (AIDA) research group, which focuses on pioneering research in AI-driven solutions and data protection technologies. Additionally, he serves as the Head of the Artificial Intelligence in Science and Technology (AIST) Lab, where he leads innovative projects at the intersection of AI and scientific applications. His work is dedicated to advancing intelligent systems, developing robust data-driven methodologies, and fostering collaboration in the AI and data science communities.

https://orcid.org/0000-0002-1813-8417