Privacy-Preserving Diabetes Prediction Using Federated Learning in Edge-Based Healthcare

Gabriel Gomes de Oliveira[1*†], Suja A. Alex[2†], J. Renees Jenisha[3†], Abdullah Ayub Khan[4†], Vania V. Estrela[5†], Shilpa Mahajan[6†], Asif A. Laghari[7†], Asiya Khan[8†]

[1*]Telecommunications Department, School of Electrical and Computer Engineering, State University of Campinas - FEEC/UNICAMP, Av. Albert Einstein, Nº 400 - Cidade Universitária, Campinas, 13083-852, São Paulo, Brasil.

[2]Department of Information Technology, St. Xavier's Catholic College of Engineering, hunkankadai, Nagercoil, 629003, State, India.

[3]Department of Information Technology, St. Xavier's Catholic College of Engineering, hunkankadai, Nagercoil, 629003, State, India.

[4]Departament Informatics and Computing, Fakulti Informatik Dan Komputeran, UniSZA, Zainal Abidin, Kampus Besut, Besut, 22200, Terengganu, Malaysia.

[5]Telecommunications Department, Fluminense Federal University (UFF), Street Passo da Pátria, 156, Bloco D - São Domingos, Niterói, 24210-240, Rio de Janeiro, Brazil.

[6]Department of CSE, The NorthCap University, HUDA Sector 23-A, Gurugram, 122017, Haryana, India.

[7]Software College, Shenyang Normal University, HUDA Sector 23-A, Shenyang, 110136, Liaoning, China.

[8]School of Engineering, Computing and Mathematics, University of Plymouth, Drake Circu, PL4 8AA, Plymouth, United Kingdom.

*Corresponding author(s). E-mail(s): oliveiragomesgabriel@ieee.org;
Contributing authors: suja@sxcce.edu.in; renees.502245@sxcce.edu.in;
abdullahayub.bukc@bahria.edu.pk ; vania.estrela.phd@ieee.org;
shilpa@ncuindia.edu; asiflaghari@synu.edu.cn;
asiya.khan@plymouth.ac.uk;
†These authors contributed equally to this work.

Abstract

The growing prevalence of diabetes highlights the need for scalable, accurate, and privacy-conscious testing technologies. To train models, traditional machine learning (ML) techniques often rely on centralised infrastructure that aggregates patient data from multiple sources onto a single server. Even though these methods frequently produce highly accurate predictions, they raise serious concerns about data privacy, legal compliance, and computational scalability. This article presents a Federated Learning (FL) framework for diabetes risk prediction, utilising the PIMA Indian Diabetes dataset to overcome these issues. Without sending raw patient data, the suggested approach utilizes a supervised Deep Neural Network (DNN) that has been cooperatively trained across several decentralised clients, including wearable health devices and institutional medical servers. Each client independently trains the model on its local dataset, sharing only model parameters with a central server. The global model is updated through Federated Averaging (FedAvg), thereby safeguarding data privacy while preserving diagnostic accuracy. Experimental results validate the effectiveness of the proposed approach in balancing performance and privacy, demonstrating its potential for real-world deployment in edge-based smart healthcare systems.

Keywords: Federated Learning, Edge Computing, Decentralized Intelligence, Deep Neural Network, Diabetes Prediction, Healthcare.

1 Introduction

Millions of people worldwide have diabetes, a chronic metabolic disease marked by persistent hyperglycemia brought on by the body's incapacity to either make enough insulin or use the insulin it does produce. Over 422 million people worldwide have diabetes, and by 2045, that figure is predicted to rise to 693 million, according to the World Health Organisation (WHO). The illness causes around 1.6 million deaths annually, making it a serious public health concern [1]. Diabetes mellitus is still difficult to diagnose early and accurately, especially in its early stages. However, artificial intelligence (AI) and machine learning (ML) techniques provide useful decision-support tools that can help clinicians detect the disease early, improve diagnostic accuracy, and lessen clinical workload [2]. While this approach can yield accurate predictions, it presents critical challenges, particularly in data privacy, security, and regulatory compliance (e.g., "General Data Protection Regulation" (GDPR), HIPAA) [3–5]. Additionally, the centralized paradigm imposes substantial computational and communication overhead, making it unsuitable for modern, decentralized healthcare ecosystems.

Electronic Medical Records (EMRs) contain private health information. Yet, exploiting the Internet of Things (IoT) [20, 21] entails defending information privacy. Expanding the IoT rationale to medicine gives rise to the Internet of Medical Things (IoMT) [3–9]. The IoMT also works with networked services [10–12], and health-tailored systems [13, 14], permitting EMR handovers [15, 16], communication exchanges, and knowledge acquisition [17, 18]. Privacy apprehensions for patients,

caregivers, and others augment medical care threats. Thus, smart healthcare's major hardships are privacy and data security. Patients may think their EMRs, such as glucose levels, blood pressure, and pulse rate, are only accessible to authorized healthcare personnel with their consent and monitoring. Electronic Medical Records (EMRs) contain private health information. Yet, exploiting the Internet of Things (IoT) [19, 20] entails defending information privacy. Expanding the IoT rationale to medicine gives rise to the Internet of Medical Things (IoMT) [3–9]. The IoMT also works with networked services [10–12], and health-tailored systems [13, 14], permitting EMR handovers [15, 16], communication exchanges, and knowledge acquisition [17, 18]. Privacy apprehensions for patients, caregivers, and others augment medical care threats. Thus, smart healthcare's major hardships are privacy and data security. Patients may think their EMRs, such as glucose levels, blood pressure, and pulse rate, are only accessible to authorized healthcare personnel with their consent and monitoring.

Federated Learning (FL) has emerged as a revolutionary paradigm for overcoming these constraints. Without sending raw data from its original source, FL enables training a global machine learning model across multiple dispersed clients, such as wearable devices, hospital servers, or mobile health applications. FL guarantees privacy-preserving collaborative learning while keeping data locally by exchanging only model parameters, which is particularly helpful in sensitive medical applications. Using the PIMA Indian Diabetes Dataset, we provide FedDiabCare, a Federated Deep Learning (FDL) framework for diabetes risk assessment. The framework leverages a supervised DNN classifier. It is designed to operate efficiently on edge computing platforms and smart healthcare devices, enabling decentralized intelligence in real-time health monitoring systems [6, 7].

Due to bandwidth limitations, data sensitivity, and regulatory restrictions, centralized training models are sometimes unfeasible or unsuitable in the healthcare industry, particularly for geographically dispersed hospitals or personal health devices. Edge-based health-monitoring devices have unrealized computational potential for localized training and inference, but current ML pipelines underutilise them. The following are the main contributions of this study:

- **Development of FedDiabCare:** The FedDiabCare is a novel privacy-preserving FL framework designed for diabetes risk assessment. The framework leverages a DNN architecture that enables collaborative model training across distributed edge devices without sharing raw patient data;
- **Evaluation in Edge-Based Smart Healthcare:** The FedDiabCare framework is extensively evaluated using the PIMA Indian Diabetes Dataset, demonstrating its effectiveness in predicting diabetes risk while preserving data privacy, supporting decentralized training, and maintaining model performance in edge-enabled healthcare environments;
- **Comprehensive Performance Evaluation:** A thorough evaluation of the proposed FedDiabCare framework is conducted using key FL metrics, including model accuracy, communication overhead, convergence rate, and data privacy effectiveness. The results validate the practical viability and robustness of the system in edge-based smart healthcare environments.

This paper is outlined as follows. Section 2 goes over related literature and experiments. The proposed FedDiabCare system's modeling and problem formulation are presented in Section 3. Section 4 sheds light on the overall framework's performance, explaining experiments and result breakdowns. Section 5 discusses some forthcoming improvements. Finally, conclusions are drawn in Section 6.

2 Related Works

To improve the predictive and classification accuracy for diabetes and other unbalanced datasets, deep learning models, including CNN and LSTM architectures, are used in conjunction with class-balancing strategies such as SMOTE and genetic algorithm-based oversampling. These studies made a substantial contribution to improving model performance in healthcare data analysis, especially when data were unbalanced [6, 7, 21]. Particularly in the healthcare industry, where data sensitivity and regulatory compliance are critical, FL has emerged as a revolutionary paradigm for privacy-preserving ML. FL ensures that raw data never leaves its source by permitting collaborative model training across decentralized devices or institutional servers, unlike standard centralized techniques. This enables intelligent, real-time prediction across distributed clients without compromising data privacy [22–24].

This decentralized framework mitigates the risks of data aggregation while enabling powerful predictive capabilities. To further reinforce privacy and security, various privacy-preserving aggregation protocols have been proposed within FL systems, including one-time pad encryption, homomorphic encryption, and secure multiparty computation - each allowing secure computations without exposing raw data [23, 24]. Differential privacy has also been widely adopted to add statistical noise to model updates, thereby obfuscating individual data points during training. In addition, blockchain technologies are increasingly integrated to provide immutable, decentralized logging, and trusted execution environments are leveraged to enhance hardware-based security. Together, these techniques elevate the trustworthiness and robustness of FL in sensitive settings such as healthcare.

Several studies have explored the application of FL for diabetes-related tasks. In [24], researchers utilized FL to investigate the association between Type 2 Diabetes Mellitus (T2DM) and healthcare expenditures using distributed datasets from Statistics Netherlands and the Maastricht Study. Their findings revealed a significantly higher financial burden for T2DM patients compared to those with prediabetes, underscoring the utility of FL in analyzing sensitive medical and economic data. Meanwhile, Tichler et al. [25] proposed a patient-centric decision aid for T2DM that emphasizes shared decision-making and personalized treatment pathways. Although they do not employ FL directly, their focus on user-centric care aligns with FL's goal of retaining patient data control while enhancing healthcare intelligence. FL has also shown promising results in real-time glucose level prediction, a crucial aspect of diabetes management. Falco et al. [26] introduced an FL-inspired evolutionary framework that addresses challenges related to data privacy and interpretability. Their system demonstrated improved prediction of future glucose levels by sharing knowledge across

decentralized datasets, validating the efficacy of FL in continuous monitoring applications. In another significant contribution, Astillo et al. [27] developed a DL-based anomaly detection framework for Diabetes Management Control Systems (DMCS) within implantable Internet of Things (IoT) for medical devices. The system integrated Convolutional Neural Networks (CNNs) and Multilayer Perceptrons (MLPs) within a federated architecture, ensuring data remained on the device while enabling collaborative learning. They reported Root Mean Square Error (RMSE) values ranging from 27.45 to 51.79 mg/dL for glucose prediction across varied intervals, demonstrating the practicality of FL-enabled DL in low-latency, embedded healthcare systems. Another research focused hybrid learning paradigm, integrating Split Learning (SL) and FL, to enhance privacy preservation during model training [28–30]. This approach enables collaborative training of DL models across disparate, privately maintained health datasets while preserving the original records and model parameters. This approach also uses the Dynamic Weight Correction Strategy (DWCS) to quantify the drift between models from adjacent communication rounds and adjusts the model weights.

While these approaches contribute valuable insights, the proposed FedDiabCare framework uniquely combines FL, DNN, and Edge Computing to facilitate real-time, privacy-preserving diabetes risk prediction across distributed healthcare infrastructures. Unlike prior work, FedDiabCare is explicitly tailored for edge-enabled smart healthcare systems, bridging the gap between privacy-aware model training and deployability in real-world, resource-constrained environments [30–32].

3 System Model and Problem Formulation

The proposed system, FedDiabCare, is an FDL framework designed to predict diabetes risk in a privacy-preserving, resource-constrained edge-computing environment. At its core, FedDiabCare employs a DNN classifier trained on the PIMA Indian Diabetes Dataset, with training distributed across multiple decentralized clients, such as wearable healthcare sensors, mobile devices, and institutional edge nodes. The overview of the proposed framework appears in Fig. 1.

3.1 Problem Formulation

In the FedDiabCare framework, the goal is to minimize the global empirical risk across all participating clients while preserving data privacy through FL. Consider the total number of clients to be K, and each client $k \, \epsilon \, \{1, 2, \ldots, K\}$ have a local dataset D_k containing n_k data samples. The total number of data samples across all clients is represented in Equation (1). Equation (2) defines the global objective function, which is formulated as a weighted sum of local objective functions.

$$n = \sum_{k=1}^{K} n_k \tag{1}$$

$$\min_{w} F(w) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(w) \tag{2}$$

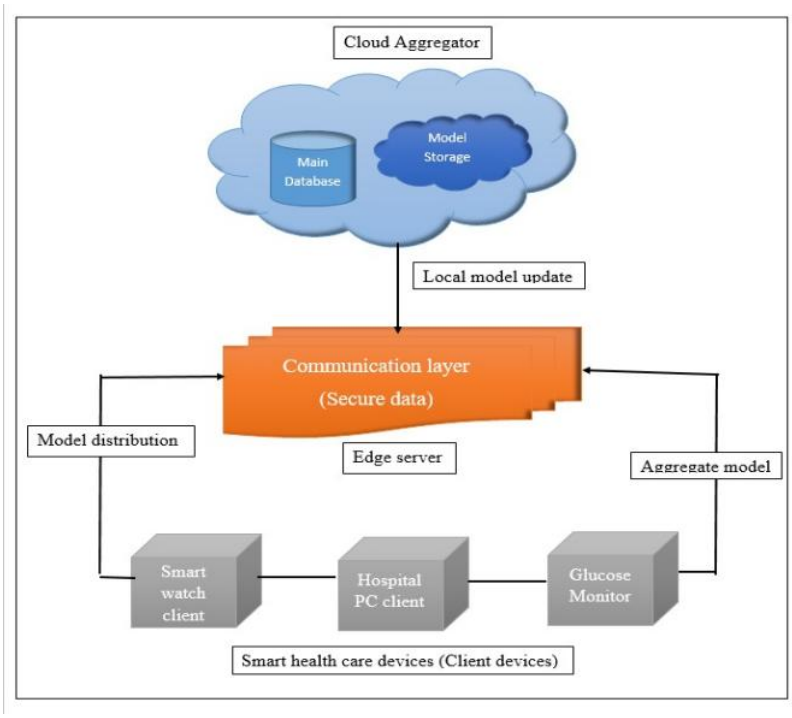

Fig. 1 Overview of FedDiabCare Framework.

Where w denotes the set of model parameters of the neural network, $F_k(w)$ denotes the local loss function at the client k, and is defined in Equation (3) as

$$F_k(w) = \frac{1}{n_k} \sum_{i=1}^{n_k} l(f_w(x_i^k), y_i^k) \tag{3}$$

Where x_i^k denotes i^{th} data sample and label from the client k, $f_w(x)$ denotes the output of the neural network with parameters w, $l(.)$ denotes the binary cross-entropy loss function used for diabetes classification and defined as

$$l(f_w(x), y) = -[y \log f_w(x) + (1-y) \log(1 - f_w(x))] \tag{4}$$

As per Equation (5), after local optimization on each client, the FedAvg algorithm is applied at the central server to update the global model:

$$w^{(t+1)} = \sum_{k=1}^{K} \frac{n_k}{n} w_k^{(t+1)} \tag{5}$$

Where $w_k^{(t+1)}$ denotes locally updated model at client k after training in round t, $w^{(t+1)}$ denotes the aggregated global model for the next round.

This ensures that learning is collaborative and privacy-preserving, as each client contributes to the global model only via gradients or weight updates, without sharing raw patient data.

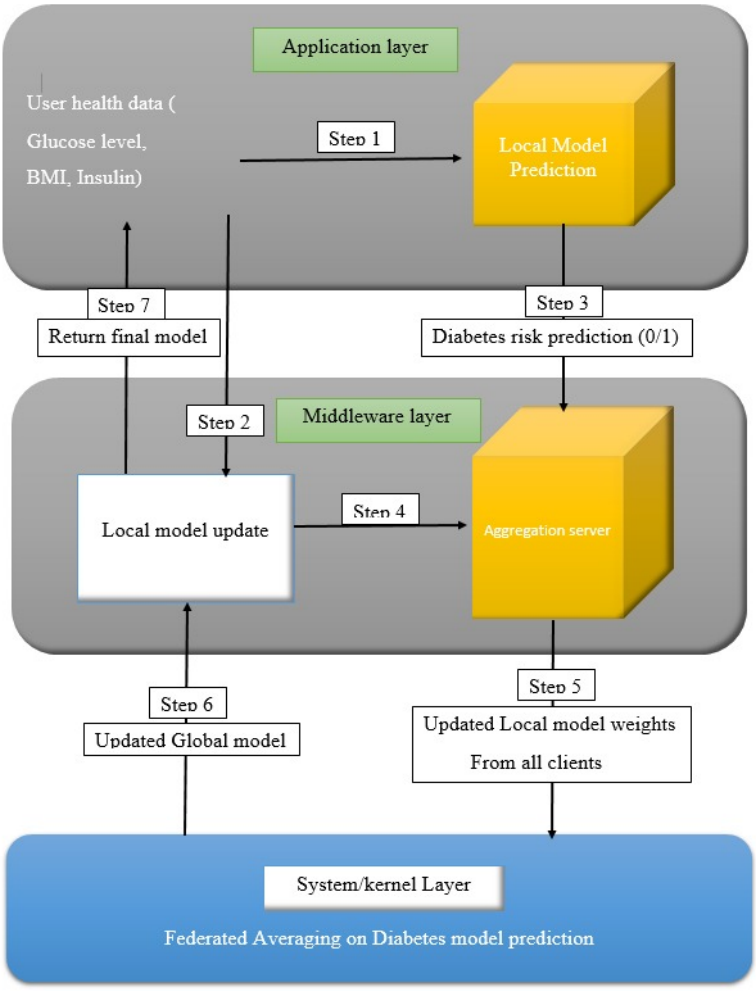

Fig. 2 Proposed FedDiabCare Framework.

3.2 System Architecture

Figure 2 shows the proposed FedDiabCare framework, composed of Local Clients, a Federated Server, and a Communication Protocol.

- **Local Clients (Edge Devices):** Each client represents a smart healthcare device (e.g., a wearable sensor or edge server) equipped with a local copy of the DNN model. These devices possess limited computational capacity and access only their own local patient data.
- **Federated Server (Aggregator):** A centralized coordinator (e.g., a hospital or cloud aggregator) collects only the model updates (weights or gradients) from participating clients without accessing their raw data. The server aggregates these updates using the FedAvg algorithm to refine the global model.

- **Communication Protocol:** Communication between clients and the server is secure and transmits only model parameters, thereby ensuring data locality and compliance with privacy requirements.

3.3 Execution Models in FedDiabCare

The FedDiabCare framework operates through two distinct but interdependent execution models: the Local Execution Model and the Cloud Execution Model. These components work collaboratively to enable privacy-preserving, distributed learning across multiple edge-enabled healthcare environments.

3.3.1 Local Execution Model

The Local Execution Model represents the decentralized stage of the FL workflow, in which training occurs directly on client-side devices such as smartphones and smartwatches, or on edge or hospital servers. Every participating client maintains a private dataset that may include real-time physiological sensor readings from wearable devices, Electronic Medical Records (EMRs), or prior patient records. Each device in this architecture runs a lightweight DNN instance optimized for binary diabetes classification. The model is trained locally for a predefined number of epochs on the client's data. Importantly, there is no raw data exchange. Instead, the device computes model-weight updates (i.e., gradients) and transmits them to a central server. To enhance bandwidth efficiency, only essential model parameters (rather than the entire model or raw data) are shared, thereby significantly reducing communication overhead. This localized learning approach ensures compliance with privacy regulations while leveraging the computational capabilities of modern edge devices.

3.3.2 Cloud Execution Model

The Cloud Execution Model operates on a centralized server or cloud platform that coordinates the collaborative training process without accessing raw patient data. Its primary responsibility is to aggregate model updates received from all participating clients using the FedAvg algorithm. Consider w_t^k represents the local model weights from the client k, and n_k denote the number of data samples at that client. The server computes the global model w_t as defined below.

$$w_t = \sum_{k=1}^{K} \frac{n_k}{n} w_t^k \tag{6}$$

After aggregation, the refined global model is redistributed back to all clients through their respective edge servers, where the next round of local training begins. This process is repeated iteratively until model convergence or a specified error threshold is reached.

3.3.3 Model Synchronization and Convergence

The continuous interplay between the local and cloud execution models enables the global training model to be synchronized across multiple clients, even when data

distributions are heterogeneous. Over successive training rounds, the global model incrementally improves its predictive performance, converging toward a solution that generalizes well across all participating clients while ensuring data privacy, scalability, and low latency.

3.4 Federated Learning Algorithm for FedDiabCare

This section presents the proposed FL algorithm for privacy-preserving diabetes prediction using a distributed DNN classifier. The system ensures data locality by allowing each client (e.g., an edge device or a hospital server) to train the model on its private data while sharing only model parameters with a central server, never the raw data. The global model is updated via FedAvg across multiple communication rounds.

The global dataset is denoted as $D = \{(x_i, y_i)\}_{i=1}^{n}$, where each input vector $x_i \epsilon R^d$ represents the health-related parameters of an individual (e.g., glucose level, BMI, age), and the corresponding binary class label $y_i \epsilon \{0, 1\}$ indicates whether the individual is diabetic (1) or non-diabetic (0). The system consists of K participating clients, such as edge devices or hospital nodes, each holding a private subset of the dataset. The learning process is carried out over R communication rounds, in which each round comprises local training followed by centralized aggregation. During local training, each client updates its model using E local epochs and a learning rate α. The model used is a DNN with parameters w, and the output layer employs the sigmoid activation function $\sigma(z) = \frac{1}{1+e^{-z}}$, which maps the model's linear output to a probability value between 0 and 1, suitable for binary classification. The local objective function on each client is $F_k(w)$, which is typically a binary cross-entropy loss computed over the client's data. Model updates from all participating clients are aggregated at the central server via a weighted average of local data sizes to obtain the global model parameters. The FL algorithm for FedDiabCare is stated subsequently.

Algorithm 1: Federated Diabetes Risk Prediction with FedAvg

Input: Number of clients K, communication rounds R, local epochs E, learning rate α, initial model weights w_0

Output: Final global model weights w_R

1. Initialize global model weights w_0
2. For each round r=1 to R:
   a. The server selects a subset $S_r \epsilon \{1, 2, \ldots, K\}$ of participating clients
   b. For each client $k \epsilon S_r$
      i. Initialize local model: $w_k^r = w^{(r-1)}$
      ii. For each local epoch $e = 1$ to E:

          1. Update local weights using mini-batch gradient descent:

$$w_k^r \leftarrow w_k^r - \alpha . \nabla F_k(w_k^r)$$

2. Send the updated local model w_k^r to server

c. Server aggregates updates using Federated Averaging:

$w^r = \sum_{k \in S_r} \frac{n_k}{n} \, w_k^r$.

3. Return global model weights w_R

Once training converges, the final model $f_x(x)$ is used for diabetes prediction $\widehat{y} = \sigma\left(w^T x\right)$, where $\widehat{y} \epsilon [0, 1]$ represents the predicted probability of diabetes for a given input x. This algorithm ensures that all training data remain on-device, and only parameter updates are shared, thereby preserving data privacy, enhancing security, and complying with regulatory frameworks (e.g., HIPAA, GDPR).

4 Experimental Analysis

4.1 Dataset

The experiments in this study utilize the PIMA Indian Diabetes Dataset (PIDD), a widely used benchmark dataset in biomedical and ML research. Collected by the National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK), the dataset is designed to support the prediction of Type 2 Diabetes Mellitus using clinically relevant diagnostic features. Owing to its realistic medical context and public accessibility, PIDD has become a standard in evaluating classification models in healthcare analytics. The dataset comprises 768 samples, each with 8 input features and a binary target label. The target column, referred to as Outcome, indicates whether a subject has diabetes (1) or not (0). The dataset is particularly suitable for evaluating classification algorithms, anomaly detection techniques, and risk assessment models due to its moderate size and structured feature set.

The PIMA Indian Diabetes Dataset comprises several clinically relevant features for predicting the likelihood of Type 2 diabetes. Each record contains eight input attributes and one output class. The features include Pregnancies, which indicates the number of times a patient has been pregnant, and Glucose, which represents the plasma glucose concentration measured 2 hours after an oral glucose tolerance test and serves as a primary diagnostic indicator. Blood Pressure refers to the diastolic blood pressure in mmHg, commonly monitored in metabolic assessments. Skin Thickness measures the triceps skinfold thickness, providing an estimate of subcutaneous fat. The Insulin feature records serum insulin levels, which are critical for assessing insulin resistance or deficiency; however, missing values may occur. The Body Mass Index (BMI) is calculated by dividing a patient's weight by the square of their height and is a key indicator of body fat composition. The Diabetes Pedigree Function (DPF) quantifies hereditary risk based on family history, while age records the patient's age in years. The final column, Outcome, is the binary target variable indicating whether diabetes is present (1) or absent (0).

For example, a sample record from the dataset, such as 6, 148, 72, 35, 0, 33.6, 0.627, 50, where 1, represents a 50-year-old female patient who has had six pregnancies, a glucose concentration of 148 mg/dL, a blood pressure of 72 mmHg, skinfold thickness of 35 mm, and a BMI of 33.6. No insulin level is recorded, and her diabetes pedigree function score is 0.627. The final value of 1 in the outcome column indicates a positive diabetes diagnosis. This combination of features provides valuable insights for training ML models to effectively detect and predict diabetic risk.

4.2 Experimental Setup

The experiments were conducted on a personal computing system sufficient for simulating FL scenarios involving moderate-sized datasets. The system used was a DESKTOP-K9GFCI device equipped with a 12th Gen Intel® Core™ i5-1235U processor operating at a base frequency of 1.30 GHz, along with 8.00 GB of RAM (7.68 GB usable). The machine runs Windows 11 Home Single Language 64-bit, with an x64-based processor, providing the necessary computational support and compatibility for all FL tools and frameworks used during the experimentation. All experimental trials were conducted in isolated virtual environments created with Python's built-in venv module to ensure reproducibility, maintain package dependencies, and avoid environment conflicts. The experiments were implemented using Python 3.11.7 and executed in Jupyter Notebook, which served as the integrated development environment (IDE). The TensorFlow DL framework (version 2.16.1) was utilized for model construction, local training at edge nodes, and coordination with the FL workflow. A variety of essential Python libraries were employed throughout the implementation to facilitate data preprocessing, model development, evaluation, and visualization. NumPy (v1.26.4) was used extensively for numerical operations and to efficiently handle multidimensional arrays. Pandas (v2.2.1) supported tasks related to data loading, cleaning, and transformation, making it easier to manage the structured tabular format of the PIMA dataset. For training and evaluating ML models, Scikit-learn (v1.4.2) provided vital tools for data splitting, model evaluation metrics, and preprocessing utilities. For visualization, Matplotlib (v8.8.2) was used to plot training curves, performance graphs, and loss metrics. At the same time, Seaborn (v0.13.2) enabled the creation of more refined statistical graphics such as heatmaps and distribution plots. Together, these libraries ensured a smooth, efficient development pipeline, enabling rigorous experimentation and clear analysis of results.

4.3 Evaluation Metrics

To assess the performance of the proposed FedDiabCare FL framework for diabetes risk prediction, a set of well-established evaluation metrics was employed. These metrics provide a comprehensive understanding of the model's classification capability, robustness, and clinical relevance in predicting diabetic outcomes. The following metrics and their corresponding mathematical formulations were used:

Accuracy: It measures the model's overall correctness by calculating the ratio of correct predictions (true positives and true negatives) to the total number of predictions. It is defined in Equation (7).

$$\text{Accuracy} = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \tag{7}$$

Precision: It amounts to the proportion of correctly predicted positive cases (True Positives) among all predicted positives, reflecting the model's ability to avoid false positives. It is defined in Equation (8).

$$\text{Precision} = \frac{T_P}{T_P + F_P} \tag{8}$$

Recall (also called sensitivity) measures the proportion of actual positive cases correctly identified by the model. This is crucial in medical applications where missing a diabetic patient can be critical. It is described in Equation (9).

$$\text{Recall} = \frac{T_P}{T_P + F_N} \tag{9}$$

F1-score: It is the harmonic mean of precision and recall. It balances both metrics, especially when class imbalance is present, as is often observed in healthcare datasets. It is expressed in Equation (10).

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{10}$$

Area Under the Curve (AUC): This score reflects the model's ability to discriminate between classes across all thresholds. A higher AUC denotes a better-performing classifier. It is defined in Equation (11).

$$\text{AUC} = \int_0^1 \text{TPR}(\text{FPR})\, d(\text{FPR}) \tag{11}$$

where IPR is the true positive rate, and FPR is the false positive rate. AUC is typically calculated from the ROC curve using trapezoidal approximation.

Communication Cost: In the context of FL, communication cost refers to the total amount of data exchanged between clients and the central server over R communication rounds. It is described in Equation (12).

$$\text{Communication Cost} = R \times K \times |w| \tag{12}$$

where R denotes the number of communication rounds, K denotes the number of clients, and $|w|$ denotes the size of the model.

Convergence Rate: The convergence rate reflects the number of communication rounds required to achieve a stable or near-optimal global model performance. It is defined in Equation (13).

$$\text{Convergence Rate} = \min\{r \in \{1, \ldots, R\} \mid \Delta L^r < \varepsilon\} \tag{13}$$

4.4 Hyperparameter Tuning

This section outlines the hyperparameters used in the development and optimization of the proposed FedDiabCare FL framework. Each hyperparameter was carefully selected and tuned to balance model accuracy, convergence stability, and computational efficiency across distributed edge devices. Tuned hyperparameters emerge in Table 1.

1) Learning Rate: The learning rate is a critical hyperparameter that directly influences the convergence behavior of neural network-based models. This study employed two distinct learning rates: one for local training on client devices and another for global model aggregation on the server.

Table 1 Caption text

Hyperparameter	Tuned Range	COptimal Value
Client Learning Rate	[0.001, 0.01, 0.1]	0.01
Server Learning Rate	[0.1, 0.5, 1.0, 1.5]	1.0
Batch Size	[10, 16, 32]	10
Local Epochs	[1, 2, 5]	2
Loss Function	Not Tuned	Binary Cross-Entropy

Author et. al. (2026)

- ***Client Optimizer:*** A learning rate of 0.01 was applied to the local optimizer on each client. This value brought about an effective trade-off between training speed and model stability. Larger values led to divergence or oscillations in local training, whereas smaller values significantly slowed convergence.
- ***Server Optimizer:*** Global aggregation stemmed from blending Stochastic Gradient Descent (SGD) with a learning rate of 1.0. This aggressive setting helped accommodate client update variances and ensured efficient parameter synchronization. Lower values led to slower convergence, whereas higher values compromised the model's stability.

2) Batch Size: Batch size determines the number of training samples processed in one forward and backward pass. It was tuned within the range of [10, 16, 32], with 10 identified as the optimal value. This smaller batch size is better suited to the limited memory and processing power of edge devices, enabling frequent updates while avoiding gradient instability.

3) Number of Local Epochs: Local epochs define the number of times a client iterates over its local dataset before sending model updates to the central server. This parameter was evaluated over [1, 2, 5], with 2 epochs yielding the best balance between local computation load and communication frequency. Fewer epochs led to slower global convergence, while more epochs increased client-side overhead and the risk of model drift.

4) Loss Function: The Binary Cross-Entropy (BCE) loss function was employed, consistent with the binary classification objective of predicting diabetes onset. Although this hyperparameter was not tuned, it was selected to align with the output layer's sigmoid activation function and to ensure proper gradient propagation during training.

4.5 Methods for Comparison

To validate the effectiveness of the proposed FedDiabCare framework for diabetes risk prediction, a comparative analysis was conducted against several well-established ML and DL models. The comparison includes eight baseline classifiers implemented in an FL environment using the same dataset and preprocessing pipeline to ensure consistency.

1) Deep Convolutional Neural Network (DCNN)

The Deep CNN (DCNN) is a class of DL models primarily known for its ability to learn spatial hierarchies in structured data, particularly images. In this study, CNN was adapted to handle tabular health records as input by reformatting input vectors to simulate spatially correlated features. In an FL setting, CNNs were deployed to

distributed clients (e.g., edge devices or institutional servers), where local training was performed on patient-specific data. Each CNN model comprises convolutional layers for feature extraction, pooling layers for downsampling, and fully connected layers for classification. The output layer relies on a sigmoid activation function for binary classification, predicting 1 for diabetic cases and 0 for non-diabetic cases. Model parameters were updated locally on each client and aggregated at the server using the FedAvg algorithm, consistent with the FedDiabCare pipeline.

2) Deep Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is an advanced variant of Recurrent Neural Networks (RNNs), designed to learn long-range temporal dependencies in sequential data. In this study, a Deep LSTM model was constructed using multiple stacked LSTM layers to enhance representational power and capture complex temporal patterns in patient health data. This architecture is particularly suitable for processing time-series medical records, such as patient monitoring logs, historical glucose measurements, or periodic diagnostic reports. Within the FL setup, each client processes its local sequential data using the Deep LSTM network, enabling the model to learn temporal health trends without transferring raw sequences to the server. The final LSTM layer is connected to a fully connected layer followed by a sigmoid activation function, which produces a binary classification output: 1 indicating a positive (diabetic) case, and 0 representing a negative (non-diabetic) case. Local model updates are periodically aggregated by the central server using FedAvg, thereby preserving privacy while maintaining predictive performance.

3) Binary Neural Network (BNN)

A BNN is a specialized class of neural networks in which both weights and activations are constrained to binary values, typically a {-1, +1} This binarization significantly reduces memory footprint, computational complexity, and inference latency, making BNNs particularly suitable for resource-constrained edge devices and FL environments, where efficient communication and energy consumption are critical concerns. In the proposed comparative setup, the BNN receives binary-encoded feature vectors derived from the healthcare dataset, structured explicitly for low-precision computation. The model architecture mimics a conventional feedforward neural network. Still, it employs binary operations for both forward and backward propagation, leveraging techniques such as XNOR-based computations and bit-packing to improve execution efficiency. The final output layer employs a sigmoid activation function for binary classification, yielding 1 for diabetic (positive) cases and 0 for non-diabetic (negative) cases. Within the FL paradigm, local training is executed on distributed clients, and only binarized model updates are transmitted to the server, further minimizing communication overhead and enhancing privacy preservation.

4) Shallow Neural Network (SNN)

The SNN is a basic form of neural architecture consisting of a single hidden layer between the input and output layers. Unlike deeper architectures such as DNNs or CNNs, SNNs are computationally lightweight and train faster, making them a suitable baseline for FL environments with limited edge-device resources. In this study, the SNN model was used to evaluate the trade-off between model complexity and training efficiency. Each client device receives preprocessed basic feature vectors extracted from

local patient datasets (e.g., glucose level, BMI, age), which serve as input to the SNN. The hidden layer applies a nonlinear activation function (e.g., ReLU or tanh), followed by a sigmoid-activated output layer that performs binary classification - outputting 1 for a diabetic case and 0 otherwise. Despite its simplicity, the SNN provides a valuable reference point for assessing the performance gains of deeper or more specialized architectures in federated healthcare applications. Model parameters are updated locally at the client level and then shared with the central server via the FedAvg algorithm, thereby preserving data privacy while enabling collaborative model refinement.

5) Support Vector Machine (SVM)

SVMs are classical supervised learning algorithms widely used for binary classification tasks. SVMs operate by identifying the optimal hyperplane that maximally separates data points of different classes in the feature space. It is particularly effective in high-dimensional settings, making it suitable for medical datasets with numerous diagnostic features.

In this study, the SVM model served as a baseline for classifying diabetes risk using feature vectors derived from distributed healthcare records, including glucose levels, BMI, and patient age. Each SVM model was trained on client-specific datasets within the FL framework. The model computes a decision boundary from the support vectors, the most informative data points for classification. For each local training phase, the SVM receives input features and outputs a binary class label: 0 for non-diabetic cases and 1 for diabetic cases. The federated learning implementation of SVM involves transmitting model parameters (e.g., weight vectors or support vectors) rather than raw patient data, thereby preserving data privacy. Although SVMs are not inherently distributed, they can be adapted to FL through federated optimization strategies or ensemble averaging of decision functions.

6) Logistic Regression (LR)

LR is a well-established statistical model for binary classification that estimates the probability that a given input belongs to a particular class. It applies a sigmoid (logistic) activation function to a linear combination of input features to map predictions to a probability range of [0,1]. Due to its simplicity, computational efficiency, and interpretability, LR is particularly well-suited to FL scenarios involving resource-constrained edge devices. In this study, LR serves as a lightweight baseline model for predicting diabetes risk using numerical health indicators, including glucose levels, BMI, insulin levels, and age. The model takes normalized feature vectors as input and outputs a binary classification label: 0 indicates non-diabetic status, and 1 indicates diabetic status.

Within the FL environment, each client performs local training on its private dataset, updating the model parameters (weights and bias) without exposing raw data. These updates are aggregated at the central server using the FedAvg strategy. Despite its linear nature, LR performs competitively across many healthcare applications and serves as a strong benchmark for evaluating more complex DL models.

7) Naive Bayes (NB)

NB is a probabilistic classification algorithm grounded in Bayes' theorem, which assumes strong independence among input features. Despite this simplifying assumption, NB often delivers robust performance on many large-scale classification tasks,

particularly when the feature-independence condition holds approximately. This study intended to implement NB as a lightweight baseline classifier within the FL framework for diabetes risk prediction. It accepts input feature vectors comprising both categorical and numerical attributes derived from patient health records. The algorithm computes the posterior probability of each class (diabetic or non-diabetic) and assigns the binary class label (0 or 1) corresponding to the higher probability. Its computational efficiency and rapid training make NB particularly suitable for resource-constrained edge devices participating in FL. Model updates based on local posterior distributions are shared with the central server, which aggregates them without access to raw data, preserving patient privacy.

4.6 Results Analysis

A comprehensive performance comparison of the proposed FedDiabCare framework with several baseline methods appears below.

4.6.1 AUC-Based Discriminative Analysis

The Area Under the ROC Curve (AUC) is a robust metric for evaluating the discriminative ability of classification models, particularly in medical diagnosis tasks where both sensitivity and specificity are critical. As shown in Fig. 3, the DNN achieves an AUC of 0.84, indicating superior performance in distinguishing between diabetic and non-diabetic cases and suggesting that the DNN is highly reliable at ranking positive instances higher than negative ones, making it an ideal choice for federated healthcare applications. Among traditional ML models, NB and Support Vector Machine (SVM) also exhibit strong performance with AUC values of 0.82 and 0.81, respectively. These results underscore their capability to function effectively even under simplified assumptions or linear boundaries, particularly in resource-constrained edge environments.

In contrast, models like a CNN and an SNN performed poorly, with AUC values of 0.33 and 0.58, respectively, alluding to a limited ability to generalize across the decision boundary in FL setups, likely due to overfitting or inadequate feature representation from distributed data. The LR model also performed competitively with an AUC of 0.77, validating its applicability in interpretable, privacy-preserving scenarios. Overall, the AUC-based analysis confirms that the FedDiabCare framework, powered by DNN, offers the best discriminatory performance, justifying its deployment in smart healthcare systems where classification accuracy and data confidentiality are equally critical.

4.6.2 Confusion Matrix Analysis

Table 2 evaluates the classification behavior of the Deep CNN (DCNN) model within the FL framework. The generated confusion matrix provides insight into the types of classification errors the model makes. The matrix comprises the following results: True Positives (TP): 0, True Negatives (TN): 44, False Positives (FP): 5, False Negatives (FN): 40. These values indicate that the CNN correctly identified 44 non-diabetic (negative) cases but failed to detect any diabetic (positive) cases. Furthermore, it

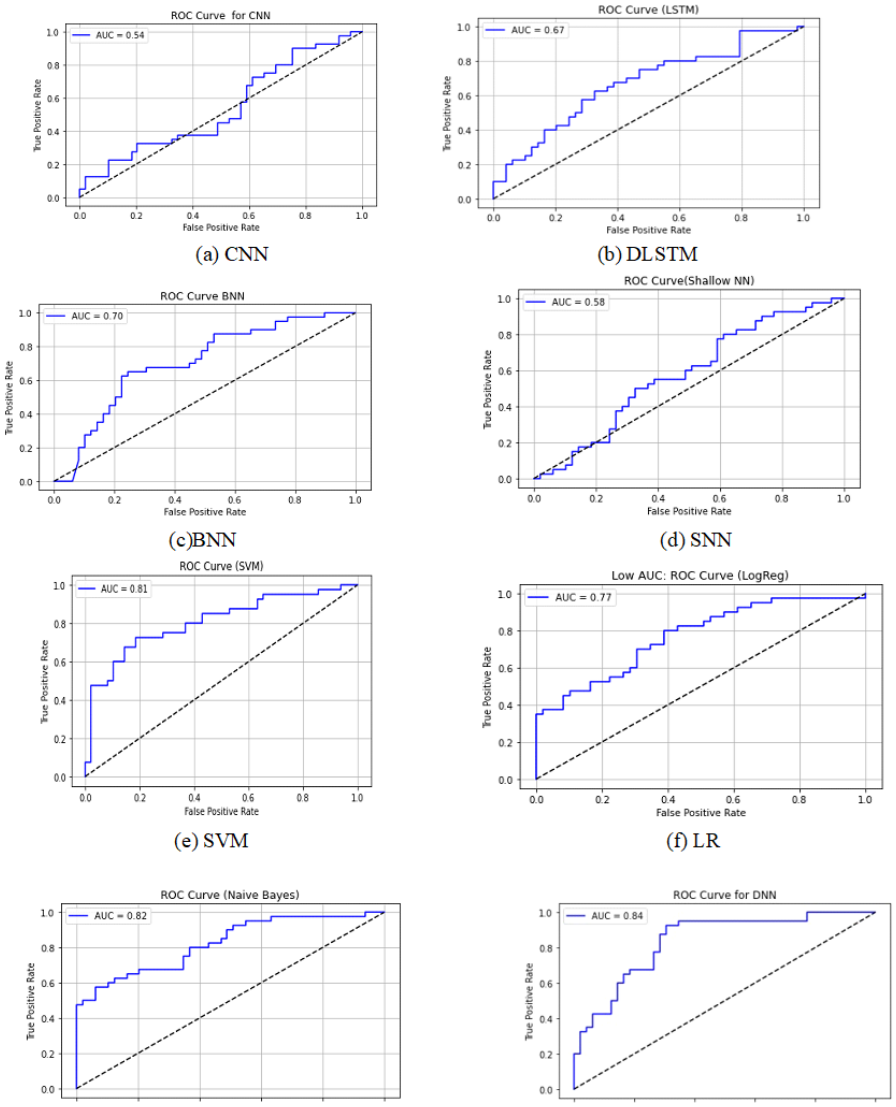

Fig. 3 AUC Analysis.

misclassified 5 non-diabetic samples as diabetic (false positives) and missed 40 actual diabetic cases (false negatives).

Table 3 presents the classification performance of the Deep LSTM (DLSTM) model, as quantified by a confusion matrix, which assesses the model's ability to distinguish between diabetic (positive) and non-diabetic (negative) cases. The confusion matrix for the DLSTM model is summarized as follows: True Positives (TP): 11, True Negatives (TN): 43, False Positives (FP): 6, False Negatives (FN): 29. These results

Table 2 Confusion Matrix for CNN.

Confusion matrix	Predicted 0	Predicted 1
Actual 0	44(TN)	5(FP)
Actual 1	40(FN)	0(TP)

Author et. al. (2026)

indicate that the DLSTM model correctly identified 11 diabetic cases and 43 non-diabetic cases. However, it missed 29 actual diabetic patients, leading to a relatively high false negative rate, and incorrectly classified 6 non-diabetic patients as diabetic (false positives). Compared to CNN, the DLSTM shows improved sensitivity (recall) by correctly identifying a portion of the true positives, although a considerable number of false negatives remain. The balance between false positives and true negatives also indicates a reasonable specificity. This performance demonstrates that the temporal learning capabilities of DLSTMs provide an advantage for capturing sequential dependencies in healthcare time-series data, thereby improving the detection of diabetic cases compared to purely spatial models.

Table 3 Confusion Matrix for DLSTM.

Confusion matrix	Predicted 0	Predicted 1
Actual 0	43(TN)	6(FP)
Actual 1	29(FN)	11(TP)

Author et. al. (2026)

Table 4 shows the performance of the BNN was evaluated using a confusion matrix to assess its effectiveness in predicting diabetes risk. The BNN model operates by taking binary-encoded feature vectors derived from the PIMA Indian Diabetes dataset as input and producing a binary output: 1 for diabetic and 0 for non-diabetic cases. The confusion matrix results are as follows: True Positives (TP): 34, True Negatives (TN): 45, False Positives (FP): 4, False Negatives (FN): 6. These results demonstrate that the BNN correctly classified a majority of both diabetic and non-diabetic cases, yielding a high true positive rate (recall) and low false negative rate, which is critical in medical diagnostics. Additionally, the model produced only 4 false positives, indicating strong precision, and successfully identified 45 non-diabetic cases. The balanced distribution of predictions across all four categories (TP, TN, FP, FN) reflects the BNN's robust capability to generalize from binary-encoded data while maintaining computational efficiency - an essential feature for deployment in resource-constrained FL environments. The model achieves a strong trade-off between sensitivity and specificity, suggesting its suitability for reliable, privacy-preserving diabetes risk assessment on edge devices.

Table 4 Confusion Matrix for BNN.

Confusion matrix	Predicted 0	Predicted 1
Actual 0	45 (TN)	4(FP)
Actual 1	34(FN)	6(TP)

Author et. al. (2026)

Table 5 shows that the performance of the SNN model, composed of a single hidden layer, was evaluated using a confusion matrix to assess its classification behavior on diabetes prediction tasks. The model receives basic input features from distributed client datasets and outputs a binary label: 0 for non-diabetic and 1 for diabetic. The confusion matrix results for the SNN model are as follows: True Negatives (TN): 49, False Positives (FP): 0, False Negatives (FN): 40, True Positives (TP): 0. These results highlight a significant prediction bias in the SNN model toward the negative class. While the model correctly identifies all non-diabetic cases (TN = 49) and avoids false positives (FP = 0), it fails to detect a single diabetic case, resulting in zero true positives and 40 false negatives.

Table 5 Confusion Matrix for SNN.

Confusion matrix	Predicted 0	Predicted 1
Actual 0	49 (TN)	0(FP)
Actual 1	40(FN)	0(TP)

Author et. al. (2026)

Table 6 shows the SVM classifier's performance, evaluated using a confusion matrix to assess its binary classification capability for diabetes risk prediction in an FL environment. The SVM model, trained on feature vectors from distributed healthcare data, outputs 1 for diabetic and 0 for non-diabetic predictions. The confusion matrix for the SVM model is as follows: True Positives (TP): 18, True Negatives (TN): 41, False Positives (FP): 8, False Negatives (FN): 22. The SVM model correctly identified 18 diabetic cases and 41 non-diabetic cases, demonstrating a balanced classification performance. The false-positive count of 8 indicates a moderate misclassification of non-diabetic cases as diabetic, while 22 false negatives suggest that several diabetic cases were missed. Despite these errors, the SVM still achieves a reasonable trade-off between precision and recall. These results illustrate that SVM effectively learns decision boundaries even in the presence of data partitioning and privacy constraints typical in FL.

Table 7 presents the confusion matrix for the LR model to assess its effectiveness in classifying diabetic and non-diabetic cases. As a widely used linear classifier in medical applications, LR outputs the probability of diabetes occurrence based on input health features, with the final prediction being binary: 0 for non-diabetic and 1 for

Table 6 Confusion Matrix for SVM.

Confusion matrix	Predicted 0	Predicted 1
Actual 0	42 (TN)	7(FP)
Actual 1	16(FN)	24(TP)

Author et. al. (2026)

diabetic. The confusion matrix for the model is as follows: True Positives (TP): 21, True Negatives (TN): 38, False Positives (FP): 11, False Negatives (FN): 19. These results indicate that Logistic Regression successfully detected over half of the diabetic cases, with 21 true positives, while also correctly classifying 38 non-diabetic samples.

Table 7 Confusion Matrix for Logistic Regression.

Confusion matrix	Predicted 0	Predicted 1
Actual 0	38(TN)	11(FP)
Actual 1	10(FN)	21(TP)

Author et. al. (2026)

Table 8 presents the confusion matrix to assess the performance of the Naive Bayes classifier for binary diabetes prediction in the FL context. Naive Bayes applies Bayes Theorem under the assumption of feature independence, making it computationally efficient and well-suited for edge devices. The model takes as input categorical or numerical health features and outputs a binary prediction: 0 for non-diabetic and 1 for diabetic. The confusion matrix is as follows: True Positives (TP): 27, True Negatives (TN): 37, False Positives (FP): 12, False Negatives (FN): 13. These results indicate that Naive Bayes achieved a strong TP rate, successfully identifying 27 diabetic cases, which contributes to a high recall value. Additionally, the classifier correctly predicted 37 non-diabetic samples. However, the 12 false positives slightly reduce precision, and the 13 false negatives indicate some missed diabetic cases.

Table 8 Confusion Matrix for Naive Bayes.

Confusion matrix	Predicted 0	Predicted 1
Actual 0	37(TN)	12(FP)
Actual 1	13(FN)	27(TP)

Author et. al. (2026)

The DNN model was evaluated using a confusion matrix to assess its classification performance on diabetes risk prediction. Designed with multiple fully connected layers, the DNN is capable of learning complex, nonlinear relationships from the PIMA dataset in an FL setup. The model outputs binary predictions: 0 for non-diabetic and 1 for diabetic, with Table 9 displaying the resulting confusion matrix. These values indicate that the DNN correctly classified 29 diabetic and 36 non-diabetic cases, reflecting a high true-positive rate and strong generalization across classes. The 11 false negatives imply some diabetic cases were missed, while the 13 false positives slightly reduced precision by incorrectly labeling non-diabetic individuals as diabetic.

Table 9 Confusion Matrix for DNN.

Confusion matrix	Predicted 0	Predicted 1
Actual 0	36(TN)	13(FP)
Actual 1	11(FN)	29(TP)

Author et. al. (2026)

4.6.3 Performance Comparison and Discussion

Table 10 presents a comparative summary of evaluation metrics across all models, including Accuracy, Precision, Recall, F1-Score, and Area Under the ROC Curve (AUC), to assess overall classification performance. The comparative evaluation highlights the superior performance of the DNN model for federated diabetes risk prediction. The DNN achieved the highest F1 Score (0.78), indicating the most effective balance between precision (0.793) and recall (0.767) among all evaluated models. It also achieved the highest Area Under the ROC Curve (AUC = 0.84), demonstrating strong discriminative power in distinguishing diabetic from non-diabetic cases. In contrast, models such as CNNs and Shallow Neural Networks (SNNs) underperformed significantly, with F1 Scores of 0.0 and AUC values of 0.54 and 0.58, respectively, highlighting their limitations in learning from distributed healthcare data. Classical ML models such as SVM and NB performed moderately well, achieving F1 Scores of 0.675 and 0.683, and AUCs of 0.81 and 0.82, respectively. However, they still fell short compared to the DNN. These results affirm the efficacy of the FedDiabCare framework, demonstrating that a well-structured DL model operating within an FL paradigm can deliver high accuracy and robustness while preserving data privacy. This validates the framework's suitability for smart healthcare environments involving decentralized and sensitive medical data.

Future Work

FL-edge AI (FLEAI) systems (FLEAISs) can alleviate limitations that call for decentralized schemes by combining cost-effective on-device computation together with privacy-preserving training . However, these combined praxes pose several obstacles. FL clashes with non-independent and identically distributed (i.e., non-IID) information, communication operating costs, and heterogeneousness among partaking devices. While edge AI (EAI) enables low-latency computation, it often faces resource

Table 10 Performance Analysis of Models.

Model	Accuracy	Precision	Recall	F1 Score	AUC
CNN	0.494	0.0	0.0	0	0.54
LSTM	0.607	0.647	0.356	0.456	0.67
BNN	0.688	0.895	0.85	0.72	0.70
SNN	0.550	0	0	0	0.58
SVM	0.741	0.774	0.60	0.675	0.81
LR	0.663	0.656	0.52	0.583	0.77
Naive Bayes	0.719	0.692	0.675	0.683	0.82
DNN	0.749	o.793	0.767	0.78	0.84

Author et. al. (2026)

constraints and scalability issues [a1]. Hence, a holistic integration of FLEAISs is paramount for addressing and alleviating these caveats, thereby enabling private, scalable, and effective AI deployments. Below, the authors address some challenges that hinder the progress of FLEAISs [33–36].

4.6.3 Heterogeneity

When compounding heterogeneous devices (e.g., ranging from legacy IoT sensors to up-to-the-minute GPUs and blossoming quantum hardware), an FLEAIS can significantly boost privacy while lessening broadcast overhead. Still, major bottlenecks arising from straggler effects, non-IID data, and hardware variance can hinder overall performance, reducing average model accuracy relative to homogeneous circumstances. Advanced, heterogeneity-aware frameworks and quantum optimization (QO) are now being adopted to boost efficiency. Concerning device heterogeneity, it leads to non-uniform capabilities, resulting in imbalanced training besides sluggish convergence rates [37–42]. The framework preserves raw data on neighboring devices, leverages differential privacy, and ensures robust privacy protection.

4.6.4 Communication Overhead

Communication demands may hinder frequent synchronization of model updates, increasing latency and bandwidth [43, 44]. Elevated communication overhead throughout the training stage poses a major challenge, notably when clients are wireless edge devices coupled to the central server. The first remediation technique is to reduce the total data exchanged amongst clients and the central server, thereby substantially reducing the model parameter size and leading to notable drops in both computational intricacy and communication overhead. If edge devices have very limited computing and communication power, an approach is to use methods that involve only the forward pass, with each client broadcasting only one scalar to the server per training iteration.

4.6.5 Blockchain Technology (BT) Scalability

The scalability of Blockchain Technology (BT) can be challenging even for traditional BT frameworks, creating hitches at scale. A scalable FLEAIS must accommodate heterogeneous devices and cut communication costs [45]. BT-reliant, secure aggregation structures to ensure trust, tampering-proof model updates, and decentralized accountability [46]. Intermediate aggregation on edge servers, together with BT-focused updates, significantly lessens communication costs.

4.6.6 Exploration of Quantum Computing (QC)

Quantum-enhanced optimization can accelerate global model convergence and improve accuracy, surpassing traditional FL approaches. Quantum AI (QAI) to boost computational adeptness, predominantly in circumstances requiring complex optimizations with high-dimensional model calculations [a7]. The reliance on quantum simulators limits the direct applicability of QO in real-world settings. Contemporary quantum hardware faces challenges (e.g., noise, decoherence, and restricted qubit availability) that may undermine the scalability and efficiency of potential upgrades to an enhanced QO layer. Furthermore, while BT appears secure and scalable in simulation, performance restraints may arise in networks with a massive number of nodes. Additional examination of hybrid architectures and lightweight consensus mechanisms may help address these challenges.

Notwithstanding its effectiveness, the QO layer incurs greater computational effort, particularly as it scales to real quantum hardware. The complexity of the implementation also increases, as integrating BT and QAI requires sophisticated expertise and infrastructure, posing barriers to adoption in resource-constrained settings.

5 Conclusion

In this research, we proposed FedDiabCare, a scalable, privacy-preserving FL framework that leverages a DNN trained on the PIMA Indian Diabetes Dataset to predict diabetes risk. FedDiabCare enables decentralized, on-device training with periodic parameter sharing, thereby greatly reducing privacy risks, communication overhead, and the risk of data leakage, in contrast to traditional centralized ML approaches that require aggregating sensitive patient data on a single server. With a final classification accuracy of 75%, an F1-score of 0.78, and a sensitivity of 0.76, the suggested system demonstrates strong predictive performance in a simulated FL environment, successfully identifying diabetes cases. Furthermore, the model achieved an AUC of 0.84, highlighting its robust discriminative power in binary classification. Beyond predictive performance, the framework also promotes user trust, regulatory compliance, and system transparency - key factors in real-world clinical adoption. The architecture is modular and upgradable, allowing for seamless integration of improved FL algorithms or DL models. In future work, the model can be extended to capture complex nonlinear patterns in patient health data through advanced neural architectures. Additionally, incorporating secure aggregation protocols, homomorphic encryption, or differential privacy mechanisms could enhance protection against inference and

reconstruction attacks during model updates.

Declarations:

Availability of data and material: All data supporting the conclusions of this study have been published and are cited accordingly.

Competing Interests: The authors declare no competing interests.

Funding: This research received no external funding.

Author Contributions: The authors contributed equally to this work. All authors were involved in the study design, data collection, analysis, and manuscript preparation. All authors have read and approved the final version of the manuscript.

Acknowledgements: This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Clinical Trial Number: Clinical trial number: not applicable.

Consent to Publish Declaration: Consent to Publish declaration not applicable.

Consent to Participate Declaration: Consent to Participate declaration not applicable.

Ethics Declaration: Ethics declaration: not applicable.

References

[1] Islam, M.M., Rahman, M.J., Menhazul Abedin, M., Ahammed, B., Ali, M., Ahmed, N.F., Maniruzzaman, M.: Identification of the risk factors of type 2 diabetes and its prediction using machine learning techniques. Health Systems **12**(2), 243–254 (2023)

[2] Khokhar, P.B., Gravino, C., Palomba, F.: Advances in artificial intelligence for diabetes prediction: insights from a systematic literature review. Artificial intelligence in medicine **164**, 103132 (2025)

[3] Renuka, O., RadhaKrishnan, N., Priya, B.S., Jhansy, A., Ezekiel, S.: Data privacy and protection: Legal and ethical challenges. Emerging threats and countermeasures in cybersecurity, 433–465 (2025)

[4] Estrela, V.V., Laghari, A.A., Lopes, R.T., Khan, A.A., Yin, S., Deshpande, A.: Introduction to intelligent healthcare in a post-pandemic world. In: Intelligent Healthcare Systems, pp. 3–23. CRC Press, ??? (2023)

[5] Estrela, V.V., Jumani, A.K., Laghari, A.A., Laghari, R.A., Khan, A.A., Jesus, M.A., Sroufer, R., Lopes, R.T.: Internet of medical things (iomt) layers for medical cyber-physical systems. In: Intelligent Healthcare Systems, pp. 45–65. CRC Press, ??? (2023)

[6] Estrela, V.V., Jesus, M.A., Intorne, A.C., Batista, K.K., Deshpande, A., Shi, F., Laghari, A.A., Khan, A.A., Oliveira, L.P.: Medical visual theragnostic systems using artificial intelligence (ai)–principles and perspectives. In: Intelligent

Healthcare Systems, pp. 301–321. CRC Press, ??? (2023)

[7] Estrela, V.V., Bukhsh, K., Rind, M.M., Shaikh, S., Khan, A.A., Laghari, A.A., Deshpande, A., Andreopoulos, N., Terziev, A.: Deep learning as a driving force for better drug development. In: Intelligent Healthcare Systems, pp. 217–235. CRC Press, ??? (2023)

[8] Ibrahim, M., Al-Wadi, A., Elhafiz, R.: Security analysis for smart healthcare systems. Sensors **24**(11), 3375 (2024)

[9] Sathya, M., Madhan, S., Jayanthi, K.: Internet of things (iot) based health monitoring system and challenges. International journal of engineering & technology **7**(1.7), 175–178 (2018)

[10] Johri, P., Balu, V., Jayaprakash, B., Jain, A., Thacker, C., Kumari, A.: Quality of service-based machine learning in fog computing networks for e-healthcare services with data storage system: P. johri et al. Soft Computing, 1–13 (2023)

[11] Elshafie, H., Mahmud, M., Jamil, H.A., Tayfour, O.E., Jarboui, S., Naveed, Q.N.: Ai as a service for 6g–satellite network architectures: Enabling next-generation e-healthcare care and beyond. In: 2025 IEEE International Conference on E-health Networking, Application & Services (Healthcom), pp. 1–6 (2025). IEEE

[12] Rajak, S., Summaq, A., Kumar, M.P., Ghosh, A., Elumalai, K., Chinnadurai, S.: Revolutionizing healthcare with 6g: A deep dive into smart, connected systems. IEEE Access **12**, 194150–194170 (2024)

[13] Mathkor, D.M., Mathkor, N., Bassfar, Z., Bantun, F., Slama, P., Ahmad, F., Haque, S.: Multirole of the internet of medical things (iomt) in biomedical systems for managing smart healthcare systems: An overview of current and future innovative trends. Journal of infection and public health **17**(4), 559–572 (2024)

[14] Vărzaru, A.A.: Assessing the relationships of expenditure and health outcomes in healthcare systems: a system design approach. In: Healthcare, vol. 13, p. 352 (2025). MDPI

[15] Ileana, M., Petrov, P., Milev, V.: Secure integration of sensor networks and distributed web systems for electronic health records and custom crm. Sensors **25**(16), 5102 (2025)

[16] Bellamkonda, S.: Enhancing network security in healthcare institutions: Addressing connectivity and data protection challenges (2019)

[17] Raceanu, A.R.: Mapping stakeholders and their networks for strategic communication in the romanian health sector–a public relations approach. ARS Medica Tomitana **29**(3), 153–165 (2024)

[18] Zhang, T., Zhou, X., Liu, J., Cheng, B., Xu, X., Qi, L., Tian, Q., Wan, Z.: Qoe-driven data communication framework for consumer electronics in tele-healthcare system. IEEE Transactions on Consumer Electronics **69**(4), 719–733 (2023)

[19] Kiran, K.V., Rao, T.S.: Routing mechanism ensuring congestion free communication in wireless sensor networks enabled by internet of things for applications in smart healthcare. International Journal of Electrical and Computer Engineering (IJECE) **15**(3), 2874–2887 (2025)

[20] Iqbal, A., Nauman, A., Qadri, Y.A., Kim, S.W.: Optimizing spectral utilization in healthcare internet of things. Sensors **25**(3), 615 (2025)

[21] Ahmed, A., Shahzad, A., Naseem, A., Ali, S., Ahmad, I.: Evaluating the effectiveness of data governance frameworks in ensuring security and privacy of healthcare data: A quantitative analysis of iso standards, gdpr, and hipaa in blockchain technology. PloS one **20**(5), 0324285 (2025)

[22] Alex, S.A., Jhanjhi, N., Humayun, M., Ibrahim, A.O., Abulfaraj, A.W.: Deep lstm model for diabetes prediction with class balancing by smote. Electronics **11**(17), 2737 (2022)

[23] Alex, S.A., Nayahi, J.J.V., Shine, H., Gopirekha, V.: Deep convolutional neural network for diabetes mellitus prediction. Neural Computing and Applications **34**(2), 1319–1327 (2022)

[24] Alex, S.A., Nayahi, J.J.V., Kaddoura, S.: Deep convolutional neural networks with genetic algorithm-based synthetic minority over-sampling technique for improved imbalanced data classification. Applied Soft Computing **156**, 111491 (2024)

[25] Alex, S.A.: Imbalanced data learning using smote and deep learning architecture with optimized features. Neural Computing and Applications **37**(2), 967–984 (2025)

[26] Alex, S.A., Singh, N., Adhikari, M.: Digital twin-based dynamic resource provisioning using deep q-network on 6g-enabled mobile edge networks. In: 2025 17th International Conference on COMmunication Systems and NETworks (COMSNETS), pp. 774–781 (2025). IEEE

[27] Liu, Z., Guo, J., Yang, W., Fan, J., Lam, K.-Y., Zhao, J.: Privacy-preserving aggregation in federated learning: A survey. IEEE Transactions on Big Data (2022)

[28] Sun, C., Soest, J., Koster, A., Eussen, S.J., Schram, M.T., Stehouwer, C.D., Dagnelie, P.C., Dumontier, M.: Studying the association of diabetes and healthcare cost on distributed data from the maastricht study and statistics netherlands using a privacy-preserving federated learning infrastructure. Journal of Biomedical Informatics **134**, 104194 (2022)

[29] Tichler, A., Hertroijs, D.F., Ruwaard, D., Brouwers, M.C., Elissen, A.M.: Development of a patient decision aid for type 2 diabetes mellitus: a patient-centered approach. BMC primary care **26**(1), 81 (2025)

[30] De Falco, I., Della Cioppa, A., Koutny, T., Ubl, M., Krcma, M., Scafuri, U., Tarantino, E.: A federated learning-inspired evolutionary algorithm: Application to glucose prediction. Sensors **23**(6), 2957 (2023)

[31] Astillo, P.V., Duguma, D.G., Park, H., Kim, J., Kim, B., You, I.: Federated intelligence of anomaly detection agent in iotmd-enabled diabetes management control system. Future Generation Computer Systems **128**, 395–405 (2022)

[32] Yang, Z., Chen, Y., Huangfu, H., Ran, M., Wang, H., Li, X., Zhang, Y.: Dynamic corrected split federated learning with homomorphic encryption for u-shaped medical image networks. IEEE Journal of Biomedical and Health Informatics **27**(12), 5946–5957 (2023)

[33] Brecko, A., Kajati, E., Koziorek, J., Zolotova, I.: Federated learning for edge computing: A survey. Applied Sciences **12**(18), 9124 (2022)

[34] Thomas, S.G., Myakala, P.K.: Beyond the cloud: Federated learning and edge ai for the next decade. Journal of Computer and Communications **13**(2), 37–50 (2025)

[35] Wu, L., Ruan, W., Hu, J., He, Y.: A survey on blockchain-based federated learning. Future Internet **15**(12), 400 (2023)

[36] Gill, S.S., Golec, M., Hu, J., Xu, M., Du, J., Wu, H., Walia, G.K., Murugesan, S.S., Ali, B., Kumar, M., *et al.*: Edge ai: A taxonomy, systematic review and future directions. Cluster Computing **28**(1), 18 (2025)

[37] Qiao, C., Li, M., Liu, Y., Tian, Z.: Transitioning from federated learning to quantum federated learning in internet of things: A comprehensive survey. IEEE Communications Surveys & Tutorials **27**(1), 509–545 (2024)

[38] Kalapaaking, A.P., Khalil, I., Rahman, M.S., Atiquzzaman, M., Yi, X., Almashor, M.: Blockchain-based federated learning with secure aggregation in trusted execution environment for internet-of-things. IEEE Transactions on Industrial Informatics **19**(2), 1703–1714 (2022)

[39] Prigent, C., Costan, A., Antoniu, G., Cudennec, L.: Enabling federated learning across the computing continuum: Systems, challenges and future directions. Future Generation Computer Systems **160**, 767–783 (2024)

[40] Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V.: Federated optimization in heterogeneous networks. Proceedings of Machine learning and systems **2**, 429–450 (2020)

[41] Karimireddy, S.P., Kale, S., Mohri, M., Reddi, S., Stich, S., Suresh, A.T.: Scaffold: Stochastic controlled averaging for federated learning. In: International Conference on Machine Learning, pp. 5132–5143 (2020). PMLR

[42] Fukami, T., Murata, T., Niwa, K., Tyou, I.: Dp-norm: Differential privacy primal-dual algorithm for decentralized federated learning. IEEE Transactions on Information Forensics and Security **19**, 5783–5797 (2024)

[43] Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H.B., Patel, S., Ramage, D., Segal, A., Seth, K.: Practical secure aggregation for privacy-preserving machine learning. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 1175–1191 (2017)

[44] Tan, A.Z., Yu, H., Cui, L., Yang, Q.: Towards personalized federated learning. IEEE transactions on neural networks and learning systems **34**(12), 9587–9603 (2022)

[45] Wang, Z., Wu, F., Yu, F., Zhou, Y., Hu, J., Min, G.: Federated continual learning for edge-ai: A comprehensive survey. arXiv preprint arXiv:2411.13740 (2024)

[46] Ning, W., Zhu, Y., Song, C., Li, H., Zhu, L., Xie, J., Chen, T., Xu, T., Xu, X., Gao, J.: Blockchain-based federated learning: A survey and new perspectives. Applied Sciences **14**(20), 9459 (2024)