Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 7268571, 20 pages
https://doi.org/10.1155/2022/7268571

*Research Article*

# Diagnose Diabetic Mellitus Illness Based on IoT Smart Architecture

**Abhilash Pati,[1] Manoranjan Parhi,[1] Binod Kumar Pattanayak,[1] Debabrata Singh,[1] Debabrata Samanta,[2] Amit Banerjee,[3,4] Sajal Biring,[4,5] and Goutam Kumar Dalapati[4,6]**

[1] *Siksha 'O' Anusandhan University, Bhubaneswar, Odisha, India*
[2] *Department of Computer Science, CHRIST University, Bangalore, India*
[3] *Physics Department, Bidhan Chandra College, Asansol 713 303, India*
[4] *Organic Electronics Research Center, Ming Chi University of Technology, New Taipei City, Taiwan 24301*
[5] *Department of Electronic Engineering, Ming Chi University of Technology, New Taipei City, Taiwan 24301*
[6] *Sunkonnect, 1 Cleantech Loop, Singapore 637141*

Correspondence should be addressed to Debabrata Samanta; debabrata.samanta369@gmail.com, Amit Banerjee; amitbanerjee.nus@gmail.com, and Sajal Biring; biring@mail.mcut.edu.tw

Received 11 May 2022; Revised 14 July 2022; Accepted 28 July 2022; Published 25 August 2022

Academic Editor: Kuruva Lakshmanna

Obtaining a quick remote diagnosis of heart disease has proven problematic in recent days. To overcome such issues in e-Healthcare systems, Internet of Things (IoT) applications have been deployed using cloud computing (CC) approaches. There are still a number of disadvantages to using CC, including latency, bandwidth, energy usage, and security and privacy concerns. Fog computing (FC), a CC development, may be able to overcome these obstacles. DiaFog enabling remote users for real-time diagnosis of diabetic mellitus disease (DMD) has been proposed in this study, which is based on the combined ideas of IoT, cloud, and fog computing, as well as an ensemble deep learning (EDL) technique. The proposed system is trained with EDL approaches on the integrated dataset of two diabetes mellitus disease datasets (DMDDs), namely, Pima Indians Diabetes Dataset (PIDD) and Hospital Frankfurt Germany Diabetes Dataset (HFGDD), obtained from the UCI-ML and Kaggle repository, respectively, and the integrated dataset of these two. The suggested system has been used to demonstrate accuracy, precision, recall, *F*-measure, latency, arbitration time, jitter, processing time, throughput, energy consumption, bandwidth utilization, network utilization, scalability, and more. In the remote instantaneous diagnosis of diabetic patients, the integration of IoT-fog-cloud is useful. The results of the trials show the value of employing FC principles and their applicability for speedy diabetic patient remote diagnosis. PACS-key is describing text of that key PACS-key describing text of that key.

## 1. Introduction

The first digital revolution, i.e., the connection of numerous networks known as the Internet, is regarded as an all-time brilliant invention. The evolving phase continues, and we are now in the second digital revolution, the Internet of Things (IoT), which is essential to long-distance communications. The Internet of Medical Things (IoMT) is a cutting-edge network that offers a global healthcare system that can cure any condition of any location [1, 2]. The globe is becoming more industrialized, and the deceased rate is rising. However, the number of lifestyle illnesses has been increased in the same period. Type 2 diabetes, heart attack, hypertension attack, and obesity are among these disorders. The kind of nutrition, degree of stress, lack of physical activity, and environmental variables are all critical contributing factors to various disorders. In some instances, the side effects of these disorders may result in life-threatening symptoms such as paralysis, shortness of breath, irregular heartbeat, cardiac arrest, and chest discomfort, all of which need immediate medical treatment. Wearables sensors and IoT applications are becoming more popular for inexpensive e-Healthcare systems [3, 4]. These IoT applications in e-Healthcare systems have enabled health professionals to monitor patients remotely while allowing patients to access e-Healthcare services easily.

*1.1. Diabetes Mellitus Disease (DMD).* Diabetes, also known as diabetes mellitus disease (DMD), is a common chronic metabolic illness characterized by high blood glucose levels that may lead to various health problems affecting the kidneys, heart, and eyesight. There are three basic forms of DMD, according to the World Health Organization (WHO) [5]:

(i) Type 1 DMD (T1DMD) is an autoimmune disease that causes insulin levels to drop dramatically

(ii) Type-2 DMD (T2DDMD) is caused by insulin-producing cells in the pancreas malfunctioning and insulin resistance in the peripheral organs

(iii) Gestational DMD (GDMD) is a kind of diabetes that affects pregnant women who have blood glucose levels that are higher than usual

Because of the health risks associated with DMD, it is critical to keep an eye on vulnerable populations, such as children, the elderly, and pregnant women [6, 7].

*1.2. IoT-Fog-Cloud Integration Approach.* Primarily, IoT applications are based only on CC. The CC creates a complete bundle for the individuals. e-Healthcare systems are aimed at making patients' lives simpler and more convenient. Current e-Healthcare systems rely heavily on IoT-enabled smart devices. Real-world applications of CC and its enlarged variations, such as edge computing (EC) and FC, have lately emerged [8, 9]. Traditional cloud networking infrastructures include restrictions like lesser data transmission speed, mobile traffic management, and privacy and security issues. So to circumvent the limits and operate as a bridge between various terminals and cloud servers, FC was created [10]. The IoT is built on the combined concepts of FC and CC ideas [11]. FC is a feature of CC that allows for reduced latency in cloud servers [12]. Fog-based designs efficiently cope with e-Healthcare system issues such as scalability, readability, flexibility, and energy awareness [13]. The FC tries to improve node-to-node communication while saving bandwidth [14]. The FC may be utilized to enhance disease diagnosis and prediction accuracy [15, 16]. The IoT-fog-cloud integration architecture generally comprises three layers, as depicted in Figure 1.

*1.3. Ensemble Learning (EL) in Disease Diagnosis.* Ensemble learning (EL) classification-based algorithms have recently been suggested to tackle classification error concerns in machine learning (ML) applications. The researchers, hence, proposed some ensemble classifiers for software fault prediction in [17]. For multiclass imbalanced data classification, ensemble classifiers are also used [18, 19]. The support vector machine (SVM), *k*-nearest neighbors (KNN), naive Bayes (NB), decision tree (DT), artificial neural network (ANN), fuzzy decision tree, and logistic regression- (LR-) based learning approaches have been used for diabetes prediction [20]. However, these methods suffer from low classification accuracy and computational complexity despite their popularity. Thus, a unique ensemble strategy is required to increase diabetes classification accuracy and deep learning (DL) in many datasets to justify that the proposed model is efficient.

*1.4. Research Gap and Motivation.* In recent years, the rising deceased rate from chronic diseases such as diabetes has posed a danger to people worldwide. Furthermore, bringing medical advantages closer to these patients in real-time is a societal challenge. Previously, diabetes patients used self-monitoring of blood glucose (SMBG) approaches such as pricking their fingers numerous times per day to test their blood glucose levels [8, 21]. There are several disadvantages to using such tactics. Then, emerged the idea of IoT health sensors, which replaced conventional sensors that lacked Bluetooth capabilities to automatically input users' detected data into smartphones through a specialized mobile application. A wide range of studies has been conducted employing the integration principles of IoT, CC, and FC, with most studies focusing on smart cities and smart homes. In e-Healthcare systems, the integration notion is also essential. It is worth noting that these studies are hardware-based yet have a real-time influence on society; nonetheless, they might be a one-time expenditure for a particular ailment. In recent days, fast remote diagnosis of any sickness has become a sought-after task.

*1.5. Research Questions.* The following research questions (RQs) have been considered in this study:

RQ1. What are the key outcomes of using a preprocessed integrated dataset for the diagnosis of DMDs?

RQ2. What are the major benefits of involving EL approaches with DL techniques in predicting a specific disease?

RQ3. What are the motives for using the integrated architecture in e-Healthcare systems, as well as the primary projected benefits?

RQ4. What is the primary objective of the IoT-fog-cloud integrated framework in processing e-Healthcare systems?

RQ5. Is it conceivable under the proposed work for the user to restrict third-party exposure to their clinical records?

*1.6. Objective and Key Contributions of the Research.* The rapid diagnosis of diabetes patients remotely is a need which is the main objective of this research. In this paper, DiaFog enabling remote users for real-time diagnosis of diabetes mellitus disease based on integrated concepts of IoT, cloud, and fog computing and ensemble deep learning (EDL) has been proposed. The proposed system is trained with EDL approaches on the integrated dataset of two diabetes mellitus disease datasets (DMDDs), namely, Pima Indians Diabetes Dataset (PIDD) and Hospital Frankfurt Germany Diabetes Dataset (HFGDD), obtained from the UCI-ML and Kaggle repository, respectively, and the integrated dataset of these two.

This paper's main goal and contributions may be described this way:

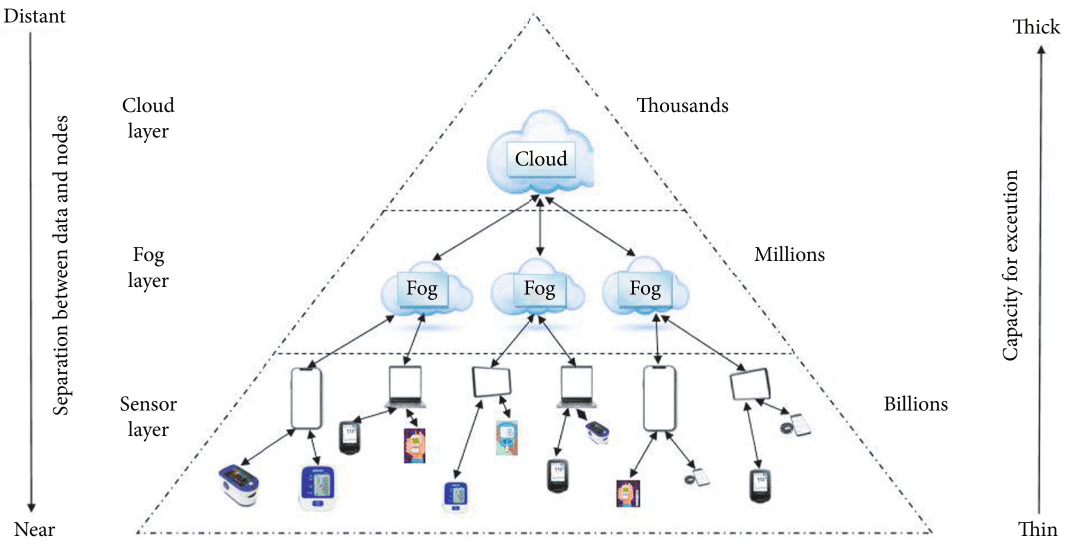


Figure 1: The architecture of IoT-fog-cloud integration.

(i) Building a portable automated diabetes patient diagnostic system based on EDL techniques

(ii) Using various frameworks and simulators previously recommended for IoT-fog-cloud integration for ultimate predictive analytics

(iii) Examining the work in terms of numerous evaluation metrics as well as network metrics on the diabetes disease integrated dataset

(iv) Addressing the findings and making comparisons of the findings with those of previous research investigations

(v) Highlighting the major areas where further IoT-fog-cloud computing integrated studies can foster the application of the methods

(vi) Identifying and analyzing prior work in diabetes disease diagnostics done by various authors in real time from afar

*1.7. The Organization of the Paper.* The following is the order in which the paper is organized: Section 2 discusses the spectrum of research work conducted in this field with a table containing the summary of these researches. Section 3 covers this work's architectural features, including the proposed work's design and the proposed model's working principle. Section 4 describes the efficient examination of the proposed work, comparing it with some related results considered in this research. Section 5 concludes with the study's pros and cons and the possible extensions to the proposed work.

## 2. Related Work

Kaur et al. have introduced a cloud IoT-based framework named CI-PDF for diabetes prediction considering accuracy, sensitivity, and specificity as evaluative parameters on the PIDD dataset and claimed to have achieved 94.5% of prediction accuracy by combining neural network (NN) and DT approaches [22]. Priyadarshini et al. have presented DeepFog, a fog computing-based deep neural architecture for predicting stress type, diabetes, and hypertension attacks using standard datasets and open-source software tools, and claimed to have achieved a superior and competitive method in comparison to others [15]. Fernández-Caramés and Fraga-Lamas have introduced an IoT continuous glucose monitor- (CGM-) based system that claims to offer a translucent and truthful blood sugar data source from a population in a quick, flexible, scalable, and low-cost manner by accessing the collected blood sugar samples and warning them in the case of a dangerous situation being detected [21]. Barik et al. have introduced FogLearn, a fog computing-based framework for the application of $K$-means clustering in Ganga River Basin Management and real-world feature data for detecting diabetes patients suffering from diabetes mellitus and found that fog computing holds a lot of promise for medical and geospatial big data analysis [23]. Fernández-Caramés et al. have created and implemented a system that improves commercial CGMs in terms of IoT capabilities, allowing them to monitor patients remotely and alert them about the severity of their conditions. And they claimed to have developed a better technique for diagnosing patients' illnesses remotely in real time [6]. Gia et al. have developed a fog-based structure for remote health monitoring and fall detection. The system provides numerous progressive amenities such as ECG feature extraction, security, and locally distributed storage. In addition, the system operates accurately, and the wearable sensor node is energy efficient [24]. Devarajan et al. proposed an energy-efficient fog-assisted healthcare system that manages glucose levels based on evaluative measures such as energy efficiency, prediction accuracy, computational complexity, and latency on two datasets from the UCI repository diabetes dataset and the Physical Activity Monitoring Dataset (PAMAP2). The experimental results show that fog over cloud computing has increased bandwidth efficiency, reduced latency, and enhanced accuracy [25]. Abdel-Basset et al. have suggested a novel framework based on computer-propped diagnosis and IoT to detect and observe type 2 diabetes patients and indicated the validity and robustness of the proposed algorithms considering accuracy and execution time as the performance evaluators [26]. Haq et al. have developed a filter method based on the DT-ID3 (Iterative Dichotomiser 3) model for essential feature selection in comparison to two ensemble learning algorithms, Ada Boost and RF, using prediction accuracy and computation time as evaluative measures, and found that the DT algorithm based on selected features improves the classifier's performance [27]. Kumari et al. have proposed an ensemble voting classifier that uses the ensemble of three ML algorithms, viz., LR, NB, and RF for the classification considering the evaluative measures like accuracy, precision, recall, and $F1$-score on PIDD and claimed to have achieved comparatively enhanced results on binary classifications [28]. Geetha and Prasad have built a hybrid model named T2DDP that doctors can effectively use to treat diabetic patients by employing supervised classification algorithms such as NB and ensemble algorithms like bagging with RF and AdaBoost for DT and found that the forecast will be submitted to the patient's cell phone at an early stage to make the immediate decisions about the health risk [29]. Shynu et al. have introduced efficient blockchain-based secure healthcare services for disease prediction in fog computing, considering purity, normalized mutual information (NMI), and accuracy as performance evaluators on PIDD and Cleveland heart disease dataset (CHDD) and thereby claimed that the proposed work efficiently clusters and predicts the disease compared to other methods [30]. Singh et al. have introduced an ensemble-based framework named eDiaPredict employing XGBoost, SVM, RF, NN, and DT to predict diabetes status among patients considering performance parameters like accuracy, sensitivity, specificity, Gini Index (GI), precision, the area under the curve (AUC), the area under the convex hull (AUCH), minimum error rate (MER), and minimum weighted coefficient (MWC) on PIDD and claimed that the proposed model could provide patients with a practical and precise prediction of diabetes based on glucose concentrations [31]. Rajput et al. have proposed a reference model for assisting rural people in India who have diabetes in characterizing two diabetes victims at an early stage using KNN, LR, SVM, RF, DT, and NB classifiers, considering accuracy, misclassification rate (MCR), recall, precision, prevalence, and $F1$-score as evaluative parameters on PIDD, and claimed to have achieved improved communication and interaction between patients [32]. Table 1 depicts an overview of the works conducted relating to this field.

## 3. Proposed Work: DiaFog

This section contains information on the various datasets, materials, and techniques employed in this study and the proposed work's architecture, design, and operation, designated as DiaFog.

### *3.1. Materials and Methods.*

This section is for the background study related to this research work. The simulation tool iFogSim, the simulating framework FogBus, one of the popular cloud service providers, Amazon Web Services (AWS), and the cloud computing platform Aneka are discussed briefly here, along with the datasets considered in training the model. In addition, a detailed discussion on the techniques considered in this research.

*3.1.1. Dataset Description.* DiaFog, the suggested model, is tested on three diabetic disease datasets: the Hospital Frankfurt Germany Diabetes Dataset (HFGDD), the Pima Indians Diabetes Dataset (PIDD), taken from the Kaggle and UCI-ML repository, respectively, and the Integrated Diabetes Dataset (IDD) of these two [33–35]. The HFGDD has 2000 persons, whereas the PIDD has 768 patients; both have nine columns. The binary result column contains two classes, each of which accepts the values "0" or "1," with "0" indicating the absence of diabetes and "1" indicating the existence of diabetes illness. Additionally, there are 1316 normal individuals and 684 diabetic individuals in HFGDD, while there are 500 normal individuals and 268 diabetic individuals in PIDD. An experiment's IDD was created by combining characteristics from both datasets. The suggested filtering and normalizing approaches handle all datasets with some missing values. Table 2 provides a summary of these datasets. There are 2768 cases in the IDD, each with a unique attribute. A deep machine learning technique cannot be used for the short dataset with nominal values. As a result, all nominal data is transformed into numeric values for the EDL model to work. Table 3 shows a summary of the dataset's characteristics.

*3.1.2. Deep Learning (DL) and Activation Function (AF).* Deep learning (DL), hierarchical learning (HL), or deep structured learning (DSL), a subset of ML, is gaining interest in the categorization of data points [36]. The primary types of DL include the recurrent neural network (RNN), deep neural network (DNN), convolution neural network (CNN), and artificial neural network (ANN). An ANN is a system that uses weighted inputs to learn. These inputs are then processed to generate an output. As the ANN learns, new routes emerge. Paths with greater weightings in the model are considered more significant (or create more desired outcomes). The bulk of DL structures and algorithms employ the ANN framework. An ANN has neurons (interconnected nodes). A multilayer perceptron (MLP) is a feedforward ANN that employs backpropagation to train the network. It is utilized for supervised learning, parallel distributed computing, and algorithmic neurobiology. The dataset was trained using MLP, a DL approach. The MLP function approach is presented as follows [37, 38]:

$$V_q = \theta_q + \sum_{p=1}^{n} W_{qp} X_p, \tag{1}$$

$$Y_q = f_q(V_q). \tag{2}$$

Table 1: A summary of some related works done.

| Author | Materials and methods | Dataset(s) used | Evaluative measures | Findings |
|---|---|---|---|---|
| Kaur et al. [22] | CI-PDF, a cloud IoT-based diabetes prediction platform, was introduced | PIDD | Accuracy, sensitivity, and specificity | Achieved 94.5% of prediction accuracy by combining DT and NN approaches |
| Priyadarshini et al. [15] | Introduced DeepFog, a deep neural architecture based on fog computing for forecasting stress, diabetes, and hypertension attacks | Stress, T2D, hypertension datasets | Accuracy, precision, recall, and $F1$-score | Achieved a superior and competitive method in comparison to others |
| Fernández-Caramés and Fraga-Lamas [21] | Introduced an IoT CGM-based system to remotely monitor patients by accessing blood sugar samples obtained and to notify them if a problematic scenario is discovered | Not mentioned | Prediction accuracy | This study claimed to be able to give a population with transparent and trustworthy blood sugar data quickly, easily, and affordably |
| Barik et al. [23] | Introduced the FogLearn framework for $K$-means clustering in Ganga River Basin Management using real-world feature data for diagnosing diabetes patients | PIDD | Accuracy | As a consequence, fog computing has great potential for extensive data analysis in medicine and geography |
| Fernández-Caramés et al. [6] | Designed and built a solution that allows continuous commercial CGMs to monitor patients remotely and warn them about their problems | Not mentioned | Prediction accuracy | The authors claimed to have developed a better technique for diagnosing patients' illnesses remotely in real time |
| Gia et al. [24] | A fog-based system for remote health monitoring and fall detection was introduced | Not mentioned | Energy efficiency | The suggested procedure is precise, and the worn sensor node saves energy |
| Devarajan et al. [25] | To manage blood glucose levels, propose an energy-efficient fog-assisted healthcare system | PAMAP2 and PIDD | Energy efficiency, accuracy, computational complexity, and latency | As a consequence, the fog over cloud computing has improved bandwidth efficiency, latency, and classification accuracy |
| Abdel-Basset et al. [26] | To diagnose and monitor type 2 diabetes patients, a new framework based on computer-assisted diagnostics and the IoTs was proposed | Personally collected data | Accuracy and execution time | The suggested algorithms were shown to be valid and resilient |
| Haq et al. [27] | Developed a filter method based on the ID3-DT model | Clinical data | Accuracy and computation time | Found that the decision tree algorithm based on selected features improves the classifier performance |
| Kumari et al. [28] | Proposed an ensemble voting classifier that employs the ensemble of three ML algorithms, viz., RF, LR, and NB | PIDD | Accuracy, precision, recall, and $F1$-score | Achieved comparatively enhanced results on binary classifications |
| Geetha and Prasad [29] | Proposed T2DDP is a hybrid model that uses supervised classification algorithms like NB and ensemble algorithms like bagging with RF and AdaBoost for DT to help physicians properly treat diabetic patients | PIDD | Accuracy, precision, recall, and $F1$-score | It was discovered that the predicted outcome would be sent to the patient's mobile phone at an early stage, allowing them to make quick judgments concerning the health risk |
| Shynu et al. [30] | Introduced efficient blockchain-based safe healthcare services for illness prediction in fog computing | PIDD and CHDD | Accuracy, purity, and NMI | In comparison to existing approaches, the suggested work efficiently clusters and predicts illness |
| Singh et al. [31] | eDiaPredict, an ensemble-based system that uses XGBoost, RF, SVM, NN, and DT to forecast diabetes status in patients, was introduced | PIDD | Accuracy, GI sensitivity, AUC, specificity, precision, AUCH, MER, and MWC | The proposed model can provide patients with a practical and precise prediction of diabetes based on glucose concentrations |
| Rajput et al. [32] | Proposed a reference model for aiding rural people in India who are suffering from diabetes | PIDD | Accuracy, MCR, recall, precision, prevalence, and $F1$-score | The authors stated that they were able to increase patient communication and engagement |

Table 2: A brief description of considered datasets.

| Dataset used | Number of characteristics | Number of instances |
|---|---|---|
| HFGDD | 9 (8 considered as inputs) | 768 (500 healthy and 268 diabetic) |
| PIDD | 9 (8 considered as inputs) | 200 (1316 healthy and 684 diabetic) |
| IDD | 9 (8 considered as inputs) | 2768 (1816 healthy and 952 diabetic) |

Table 3: A brief description of considered features of the datasets.

| Sl. no. | Features | Meaning | Values |
|---|---|---|---|
| 1 | Pregnancies | Number of times the women is pregnant | A numeric value (between 0 and 17) |
| 2 | Glucose | After 2 hours, plasma glucose concentration was determined in an oral glucose tolerance test (OGTT) | A numeric value (between 0 and 199) |
| 3 | Blood pressure | Diastolic blood pressure (in mmHg) | A numeric value (between 0 and 122) |
| 4 | Skin thickness | Thickness of triceps skin fold (in mm) | A numeric value (between 0 and 99) |
| 5 | Insulin | 2-hour serum insulin (in $\mu$U/ml) | A numeric value (between 0 and 846) |
| 6 | Body mass index (BMI) | Index mass of the body (weight in kg/height in m$^2$) | A numeric value (between 14 and 80.6) |
| 7 | Diabetes pedigree function (DPF) | Diabetes mellitus family history | A numeric value (between 0.078 and 2.42) |
| 8 | Age | Age in years | A numeric value (between 1 and 120) |
| 9 | Outcome | Diabetes disease diagnosis | 0: healthy and 1: diabetic |

Here, $V_q$ is the $q$ number of linear combinations of $p$ number of inputs, $X_p$ is the $p$ number of inputs, $\theta_q$ is the $q$ number of bias, $W_{qp}$ is the weighted connection between the neuron $j$ and the input $X_p$, $f_q(\quad)$ is the $q$ number of activation functions, and $Y_q$ is the $q$ number of outputs.

A DNN uses a layered NN with several layers of neurons. DNNs are made up of numerous linked perceptrons, each of which is a single neuron. In a DNN, dense layers are those where all inputs are densely connected to all outputs. DNNs may also have hidden layers. A hidden layer is a point between the NN's input and output where the activation function (AF) transforms the incoming data. It is called a hidden layer since it is not visible from the system's inputs or outputs. The deeper the NN, the more data recognition. The AF multiplies the input delivered to a node by weight and capacity. The function determines the signal's range. In DNN, each layer may be switched on or off, with the output of one layer feeding the input of the next layer ahead. A DNN has more hidden layers than other NNs. The dataset was trained using DNN, a DL algorithm. The DNN model is given as [38–40]

$$P_b^m(y) = W_t^m yx) + B_t^m, \tag{3}$$

$$P_b^m(\widehat{y}) = P_a^m(\widehat{y}), \quad m = 1. \tag{4}$$

Here, each preactivation function $P_b^m(y)$ is normally a linear operation involving the matrix $W_t^m(y)$ and the bias $B_t^m$, which can be integrated into a parameter $P_a^m$.

$$P_b^m\left(\widehat{H}^{m-1}\right) = P_a^m\left(\widehat{H}^{m-1}\right), \quad m > 1. \tag{5}$$

The $\widehat{y}$ denotes the addition of 1 to the vector $y$. The form of hidden-layer activation functions $H^m(y)$ is often the same at each level, but this is not always the case.

The activation function (AF) creates a weighted total and then adds bias to it to determine whether a neuron should be activated or not. The goal of these functions is to introduce nonlinearity into a neuron's output. ReLU (Rectified Linear Unit) became a prominent AF in DL and continues to give outstanding results today. It was built to solve the research's vanishing gradient issues. The sigmoid function has long been the most common AF in NNs. The sigmoid function's values are in the range $[0, 1]$, and because of its nature, tiny and big numbers sent through it will become values near zero and one, respectively. ReLU is the most often used AF, while the sigmoid function is the most commonly used AF for binary classification, which may be expressed as [37, 40, 41]

$$R(d) = \max\,(0, d), \tag{6}$$

$$S(d) = \frac{1}{1 + e^{-d}}. \tag{7}$$

Here, $R(d)$ and $S(d)$ are for ReLU and sigmoid AF, respectively, whereas the max () function finds the maximum value.

*3.1.3. Ensemble Learning (EL).* Ensemble learning (EL) may improve performance and accuracy in predictive analysis [42–44]. Ensemble approaches promise to reduce the bias and variance of the traditional learning algorithm's three flaws: the computational, symbolic, and statistical problems. Ensemble approaches include bagging, boosting, and stacking. Bagging is a strong, effective, and easy ensemble approach. This approach employs several copies of a training set utilizing the bootstrap with any classification or regression model. Bagging works well with unstable nonlinear models (little changes in the training set create large changes in the model). Boosting is a model averaging meta-algorithm. A popular ensemble approach is a potent learning principle [45, 46]. Multiple classifiers are built by various learning algorithms on the same dataset of feature vectors, and their classifications are stacked. Voting and averaging are also simple ensemble procedures. They are simple to comprehend and use. Both techniques begin by building several classification models on a training dataset. Create each base model using multiple splits of the same training dataset and algorithm or utilizing the same dataset with alternative methods. This method uses majority voting to choose the final output prediction that obtains more than half of the votes. The ensemble technique could not create a reliable prediction if no guess receives more than 50% of the votes. In weighted averaging, each model's forecast is multiplied by a weight, and then, the average is determined. A smoother model is frequently the result of this procedure. Assume we have $M$-bootstrap samples of size $N$ (approximations of $M$-independent subsets) indicated as $\{s_1^1, s_2^1, s_N^1\}, \{s_1^2, s_2^2, s_N^2\}, \cdots, \{s_1^M, s_2^M, s_{\mathrm{N}}^M\}$.

We might fit $M$ nearly independent weak learners (one from each subgroup) $w_1(s), w_2(s) \cdots, w_M(s)$ using data coefficients $d_1(s), d_2(s), \cdots, d_M(s)$ and then aggregate them using either majority voting or weighted averaging to achieve an EDL model with reduced variance as

$$\mathrm{MV}_M(s) = \mathrm{argmax}_p[\mathrm{card}\ (m|w_m(s) = P)], \tag{8}$$

$$\mathrm{WA}_M(s) = \frac{1}{M}\sum_{m=1}^{M} d_m(s) \times w_m(s). \tag{9}$$

*3.1.4. iFogSim and FogBus.* The iFogSim simulator proved helpful in evaluating alternate scheduling strategies for fog and clouds [47]. A range of situations may be considered with iFogSim, including latency, energy usage, network congestion, and operating expenses. Performance indicators are measured by emulating fog/edge devices and cloud data centers. This study used iFogSim for several simulations. This study also uses the FogBus paradigm [9], integrating IoT, CC, and FC concepts. It uses blockchain to guarantee communication security, privacy, and data integrity. It links Aneka's platform-based fog setup with the cloud via HTTP RESTful APIs. It enables platform-independent IoT application execution and interaction interfaces while computing instances. It lets developers construct apps and customers operate several apps simultaneously, and service providers manage their resources.

*3.1.5. AWS and Aneka.* Amazon Web Service (AWS), a CC service provider, is the most reliable cloud computing company, providing excellent web services and security [48]. This platform is a perfect example of actual cloud computing since it allows for data security, integrity, and availability. It provides on-demand services. The IT resources are cheap, and there is no upfront commitment. The CC platform Aneka enables developers to establish APIs [49]. Its primary design element is a service-oriented architectural (SOA) module. One of its primary features is its support for numerous programming models that describe the execution logic of programs using various abstractions. The framework's extensible SOA simplifies cloud administration and deployment while supporting various distributed application design patterns. Aneka uses cloud resources, whereas iFogSim and FogBus are used for different simulations and using fog resources.

*3.1.6. Platform and Languages Used.* The components of this work were written in a variety of programming languages. The Python programming language was used to create the preprocessing and EDL components. Jupyter python tool and SciKit Learn, Keras, and Tensorflow libraries were utilized in the EDL application. The approach, data filtering, and data processing in the intelligent gateway implementation are all made in Python to maintain compatibility with other services. We use the Pandas embedding library, a data structure library, to import the file into our Python environment. Numpy, Matplotlib, and additional libraries as needed are also loaded into the environment [50, 51]. In addition, the Android application is built using the App Inventor tool from MIT, and the web communications are carried out using the PHP programming language in this work.

*3.2. The Architecture.* DiaFog's architecture, as shown in Figure 2, incorporates a variety of techniques, hardware components, and software components required in this framework, as detailed below. The suggested study is based on previously specified frameworks and simulators for IoT-fog-cloud integration for ultimate predictive analytics.

*3.2.1. Hardware Components Used.* DiaFog consists of IoT health sensors, gateways, master PC node (MPN), fog worker node (FWN), and cloud data center node (CDCN), which are briefly covered here. The IoT health sensors collect data from diabetic patients and transfer it to gateway devices. For example, the "blood pressure sensor" measures systolic and diastolic pressures in mmHg. Gateways take patient data and share it with either MPN or FWNs. These gateways behave like fog. MPN assigns jobs to worker nodes using a resource manager or handles requests using a learned EDL model. When MPN and FWNs are overloaded, it forwards to CDCNs via cloud integrators, acting as a gateway device. FWN processes data using the learned EDL model and produces results as requested by gateway devices or MPNs. FWNs are Raspberry Pi devices in this work. CDCN is used to access cloud resources. MPN and FWNs are overloaded, and MPN forwards to CDCNs, acting as a gateway device.

*3.2.2. Software Components Used.* The proposed work comprises various software components, which are briefly discussed here. In the data preprocessing module, data from IoT health sensors such as blood pressure sensors is preprocessed and filtered before being sent to the cloud. Preprocessing of data improves model prediction accuracy. Preprocessed data from gateway devices are saved in a .csv file and utilized next. The data manager (DM) accepts preprocessed data from IoT health devices. Depending on the

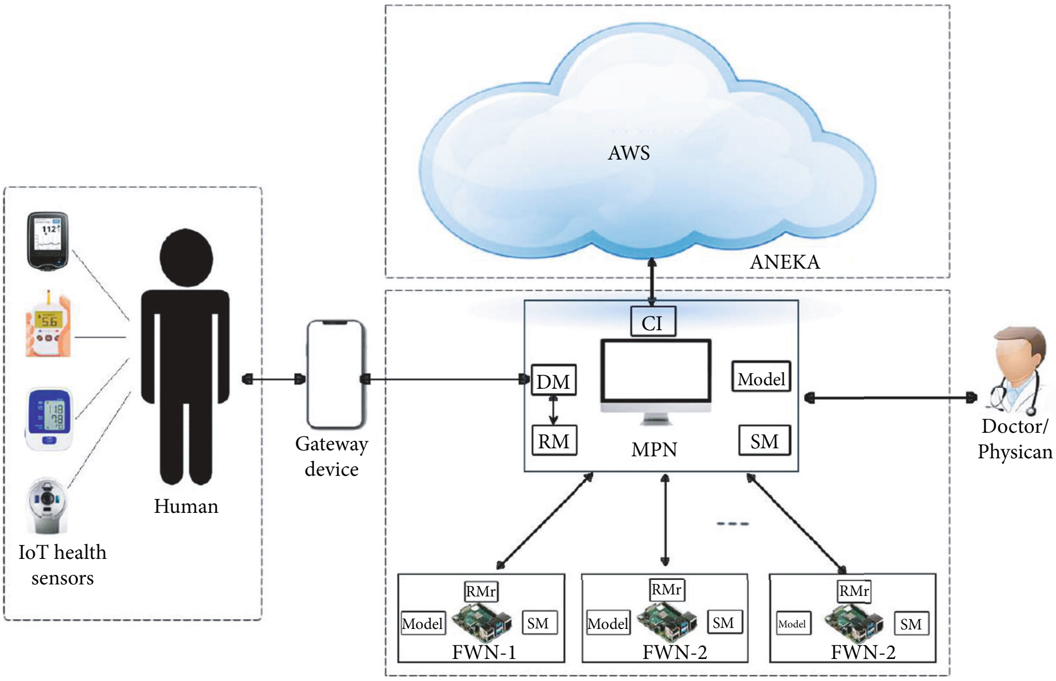


Figure 2: The architecture of DiaFog.

situation, it may combine data from many sources and change transmission frequency. The DM is in charge of deciding which FWNs to distribute the received data. The resource manager (RM) selects resources for programs to execute. The calculating server's RM determines each MPN and FWN's resource status. After collecting data, the RMs establish resources on FWNs and the cloud for applications. After receiving credentials from a gateway device, the MPN security manager (SM) validates them against the Credential Archive of Warehouse Service. In contrast, the FWN-SM oversees protected interactions of an FWN with others while conducting computations. The cloud integrator (CI) delivers storage and resource-providing instructions to the cloud. It gives context data for cloud-based instances like containers and virtual machines (VMs) to resource management. The resource monitor (RMr) allots resources to programs and tracks how successfully they are used in real time. A service provider-defined threshold is exceeded, an unexpected problem occurs, and the RM is notified. In this component, the dataset is used to train a DL approach to classify vector points (5vectors) generated by IoT health device preprocessing. On the other hand, it predicts and delivers results for the RM's tasks, a bagging classifier, and an EL approach for classification and averaging.

*3.3. The Design.* DiaFog's design begins with good dataset selection. The approach incorporates preprocessing of the raw patient dataset. The EDL model to be used at each design node is a step. An Android app is a vital tool for remote users. Experiment setup and implementation are the critical elements in the proposed study's design.

*3.3.1. Dataset Preprocessing.* The raw dataset is preprocessed and filtered here. The data were cleaned and normalized before training and testing to improve the model's prediction accuracy. In this study, the mobile application collects eight things every time, and the 9th data in the dataset is utilized as the anticipated outcome. Table 4 shows a sample of the considered integrated dataset.

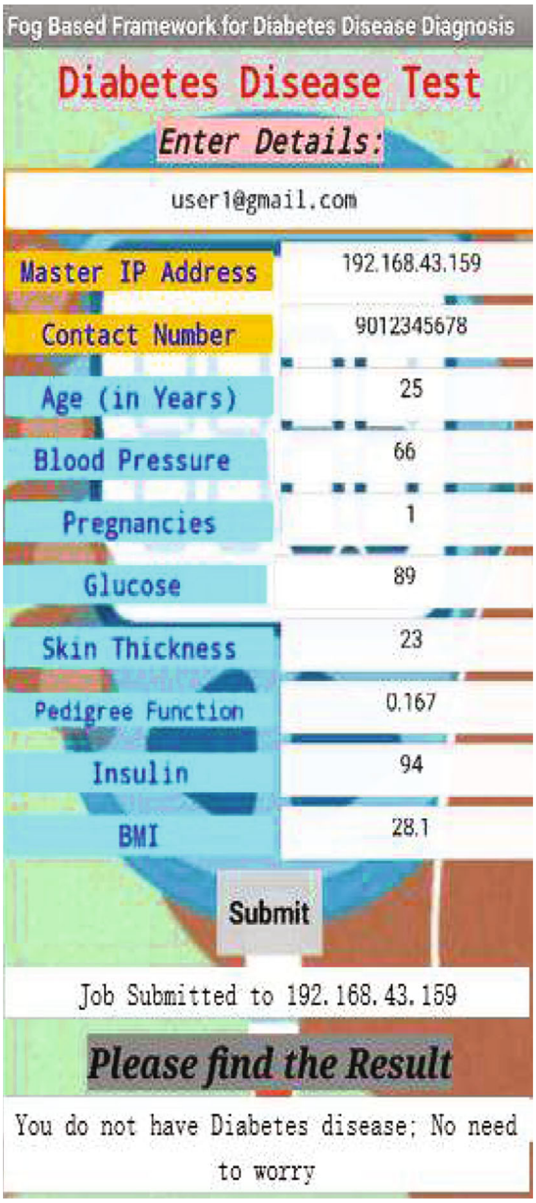
*3.3.2. Experimental Set-Up and Android App.* The set-up is implemented with some hardware configurations as evaluative hardware for the experiments in this work, including the primary gateway device (Xiaomi A2 with Android version 10), MPN (Dell with Core i3, Windows 10 64-bit OS, and 6 GB RAM), FWNs (five numbers of Raspberry Pi 4 with 4 GB SDRAM), and public cloud (AWS with Windows server and Aneka platform). In addition, 100 cell phones belonging to different people were utilized to test the scalability of the suggested concept.

The DiaFog.apk, an android interface built using MIT's App Inventor for this effort, will be utilized in a variety of Android-enabled gateway devices to gather data from remote users [52]. It serves as a connection point for IoT health devices and MPNs or FWNs [12]. As shown in Figures 3 and 4, the data input from the individuals is delivered to MPNs, and results are recorded.

*3.3.3. Implementation.* DiaFog's implementation section examines how the previously mentioned components are implemented. One of the most popular programming languages, Python, has been used to preprocess data and train the EDL model in recent years. For the predictive binary classification, the model uses ANN with a bagging classifier and majority voting classifier as EL techniques and DNN with a bagging classifier and weighted averaging as EL methods, as shown in Figure 5. Both EDL models were applied to the three datasets mentioned earlier: HFGDD,

Table 4: Samples from the preprocessed IDD dataset.

| Sl. no. | Pregnancies | Glucose | Blood pressure | Skin thickness | Insulin | BMI | DPF | Age | Outcome |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 6 | 148 | 72 | 35 | 0 | 33.6 | 0.627 | 50 | 1 |
| 2 | 1 | 85 | 66 | 29 | 0 | 26.6 | 0.351 | 31 | 0 |
| 3 | 8 | 183 | 64 | 0 | 0 | 23.3 | 0.672 | 32 | 1 |
| 4 | 1 | 89 | 66 | 23 | 94 | 28.1 | 0.167 | 21 | 0 |
| 5 | 0 | 137 | 40 | 35 | 168 | 43.1 | 2.288 | 33 | 1 |
| 6 | 5 | 116 | 74 | 0 | 0 | 25.6 | 0.201 | 30 | 0 |
| 7 | 3 | 78 | 50 | 32 | 88 | 31 | 0.248 | 26 | 1 |
| 8 | 10 | 115 | 0 | 0 | 0 | 35.3 | 0.134 | 29 | 0 |
| 9 | 2 | 197 | 70 | 45 | 543 | 30.5 | 0.158 | 53 | 1 |
| 10 | 8 | 125 | 96 | 0 | 0 | 0 | 0.232 | 54 | 1 |



Fog Based Framework for Diabetes Disease Diagnosis

Diabetes Disease Test

Enter Details:

user1@gmail.com

Master IP Address: 192.168.43.159

Contact Number: 9012345678

Age (in Years): 25

Blood Pressure: 66

Pregnancies: 1

Glucose: 89

Skin Thickness: 23

Pedigree Function: 0.167

Insulin: 94

BMI: 28.1

Job Submitted to 192.168.43.159

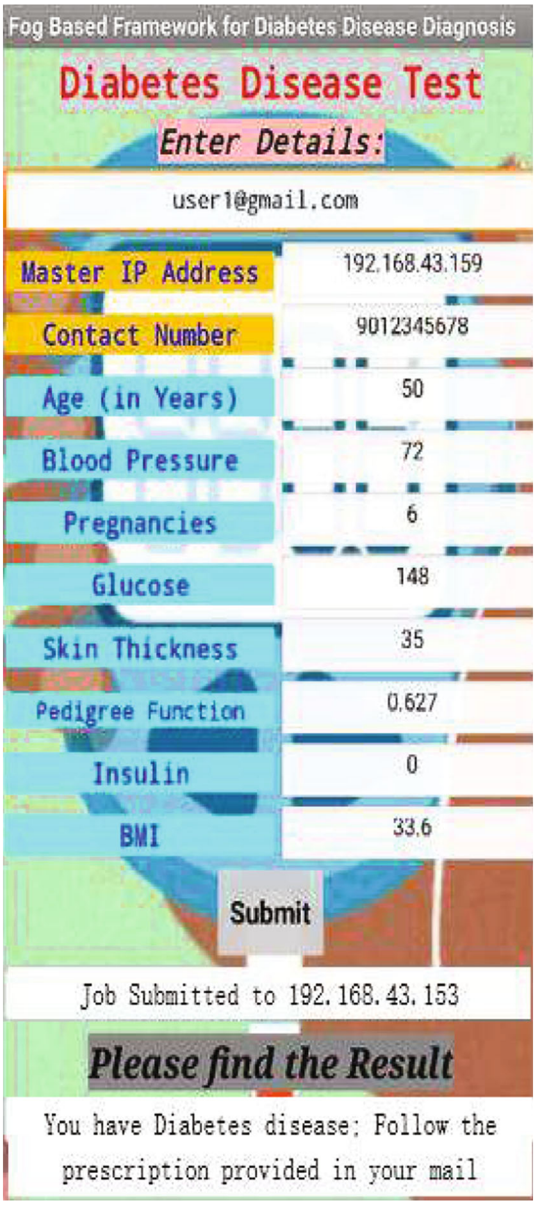
Please find the Result

You do not have Diabetes disease; No need to worry

Figure 3: Screenshot of the DiaFog app with diabetes response as no.

PIDD, and IDD. The results of the trials are then compared to determine which EDL model is the best. In the case of ANN, the ReLU function is used in all of the input, hidden, and output layers, but in the case of DNN, the ReLU function is used in both the input and hidden layers, and the sigmoid function is utilized at the output layer. The sizes of various layers in this study, including an input layer, a hidden layer, and an output layer, are 8, 3, and 2 in ANN and 8, 4, and 2 in DNN, respectively. The ninth feature is used to determine whether or not the patient has DMD. This proposed work's learning rate is 0.001 and 0.12 in ANN and DNN, respectively. In this study, the commonly used optimizer, Adam, is employed for modeling in both scenarios. Table 5 shows a summary of the DL approach setup. The Android app utilized in this project was created using MIT's App Inventor, and the online communications were done with PHP language. The data characteristics are stored in an excel file and then delivered to the MPN through HTTP post. The DM stationed within MPN is responsible for the subsequent conveyance of the data obtained. After any of the nodes have successfully processed the data supplied from the persons, the result is transmitted to the user's gateway device through the MPN.

*3.4. The Working Principle.* The working concept of this suggested work, DiaFog, is explained with several algorithm phases and a communication flow diagram. These networks are based on the master-slave idea, with MPN as the master and FWNs as slaves. The MPN, FWNs, and gateway devices are all on the same network. There are three ways to communicate: MPN alone, MPN with FWNs, or cloud only. In the first situation, MPN fulfills the task request and provides the result, whereas in the second case, FWN does so. When MPN detects insufficient resources, i.e., MPN and FWNs are overloaded, it forwards to CDCNs, acting as a gateway device. Algorithm 1 describes the primary function of the gateway device, whereas Algorithm 2 describes the primary role of the MPN. Besides, Algorithm 3 is for training the EDL model, and Algorithm 4 is for the test cases applied to the generated EDL model. This work's hardware components interact according to the prescribed framework. The communication chain shown in Figure 6 depicts a flow of work ordered by users remotely.



Fog Based Framework for Diabetes Disease Diagnosis

Diabetes Disease Test

Enter Details:

user1@gmail.com

| | |
|---|---|
| Master IP Address | 192.168.43.159 |
| Contact Number | 9012345678 |
| Age (in Years) | 50 |
| Blood Pressure | 72 |
| Pregnancies | 6 |
| Glucose | 148 |
| Skin Thickness | 35 |
| Pedigree Function | 0.627 |
| Insulin | 0 |
| BMI | 33.6 |

Job Submitted to 192.168.43.153

Please find the Result

You have Diabetes disease; Follow the prescription provided in your mail

Figure 4: Screenshot of the DiaFog app with diabetes response as yes.

### 3.5. *Algorithms Representing Detailed Working of DiaFog*

## 4. Empirical Analysis

A short discussion on different network characteristics and evaluative metrics is addressed, followed by a discussion on the outcomes acquired by this suggested effort dubbed DiaFog. This section also includes an overview of comparisons with similar works done. The performance of every planned task must be evaluated for research purposes. This study is built on previously suggested frameworks and simulators for IoT-fog-cloud integration for ultimate predictive analytics. These include performance parameters (accuracy, precision, recall, *F*-measure, etc.) and network parameters (latency, arbitration time, processing time, throughput, bandwidth consumption, jitter, network utilization, energy consumption, scalability, etc.).

*4.1. Performance Parameters.* The primary purpose of performance parameters is to find the confusion matrix, a real-to-anticipated-class matrix on which numerous evaluation metrics have been applied. $T_1$, $T_0$, $F_1$, and $F_0$ are abbreviations for the confusion matrix's true positive, true negative, false positive, and false negative. Some of the performance measures for classification purposes explored in this study include accuracy (Acc), precision (Pre), recall (Rec), and *F*-measure (F-M). The "Acc" is defined as the number of correct predictions divided by the total number of input samples. The "Pre" is defined as the ratio of properly predicted positive observations to the total number of correctly predicted positive observations. The "Rec" is defined as the proportion of successfully expected positive observations to the total number of properly predicted positive observations. The "F-M" is the weighted average of "Pre" and "Rec." The detailed evaluation formulas are as follows:

$$\mathrm{Acc} = \frac{(T_1 + T_0)}{(T_1 + T_0 + F_1 + F_0)}, \tag{10}$$

$$\mathrm{Pre} = \frac{T_1}{(T_1 + F_1)}, \tag{11}$$

$$\mathrm{Rec} = \frac{T_1}{(T_1 + F_0)}, \tag{12}$$

$$\text{F-M} = \frac{(2 \times \mathrm{Pre} \times \mathrm{Rec})}{(\mathrm{Pre} + \mathrm{Rec})}. \tag{13}$$

In this work, in order to validate the proposed EDL model, we evaluated it through six models. Here, model 1, model 2, and model 3 are the ANN along with the bagging Classifier and majority voting EL models applied on PIDD, HFGDD, and IDD datasets, respectively, whereas, model 4, model 5, and model 6 are the DNN along with the bagging classifier and weighted averaging EL models applied on PIDD, HFGDD, and IDD datasets, respectively. The observed results are then compared, as depicted in Table 6 and Figures 7–10. From experiments, it is revealed that model 6 outperforms other models in terms of the performance parameters like "Acc," "Pre," "Rec," and "F-M."

*4.2. Network Parameters.* Network characteristics are heavily influenced by the computing approach or level at which the fog-enabled IoT application is coordinated. Various network characteristics such as latency, arbitration time, total processing time, throughput, energy consumption, bandwidth, jitter, network utilization, and scalability are used to verify this suggested work. Layout 1 for MPN alone, layout 2 for MPN with 1 FWN, layout 3 for MPN with 2 FWNs, layout 4 for MPN with 3 FWNs, layout 5 for MPN with 4 FWNs, and layout 6 for CDCN only are the various configurations employed in this study for the assessment of various network metrics.

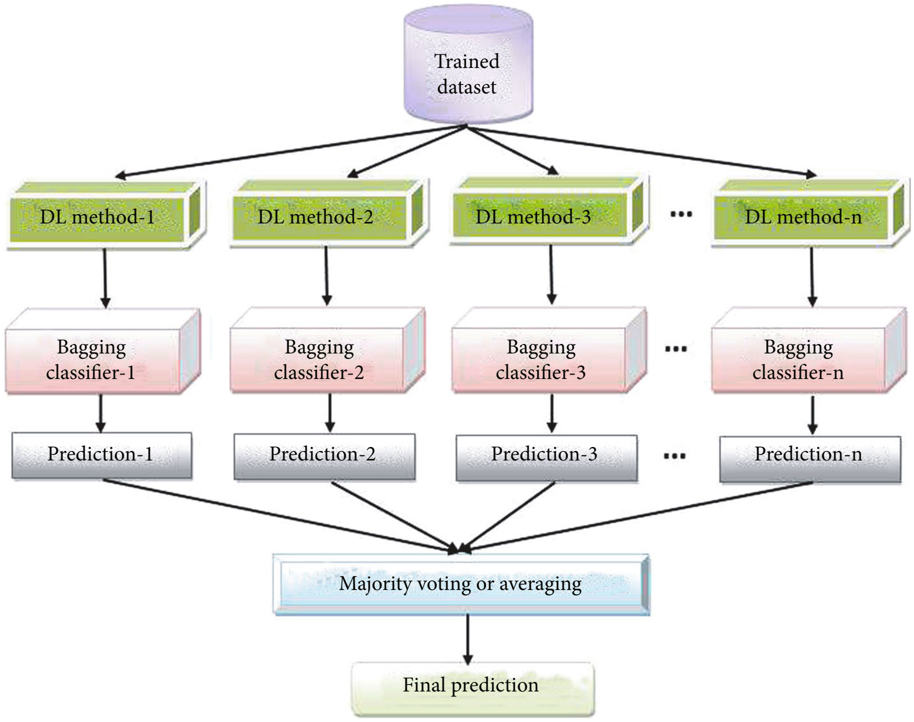


Figure 5: Basic diagram of the EDL model.

Table 5: Configuration of DL methods considered in DiaFog.

| DL methods | No. of input layers | No. of hidden layers | No. of output layers | Optimizer used | Learning rate | AF at input layers | AF at hidden layers | AF at output layers |
|---|---|---|---|---|---|---|---|---|
| ANN | 8 | 3 | 2 | Adam | 0.001 | ReLU | ReLU | ReLU |
| DNN | 8 | 4 | 2 | Adam | 0.120 | ReLU | ReLU | Sigmoid |

```
Inputs : $USR_i$
  Output : $RES_i$
  True: Gateway in active mode
    True Obtain $ATB_i$ using $IHD_i$ Submit $USR_i$ to $GTW_i GTW_i$ connected to $MPN$ Send $USR_i$ to $MPN$ using $GTW_i$ Obtain $RES_i$ Reset to obtain $ATB_i$ and submit $USR_i$ again.
```

Algorithm 1: Gateway primary function.

```
Inputs : $USR_i$
  Output : $RES_i$
  True : MPN in active mode
    True Obtain $USR_i MPN$(Available) outcome = =0 return $RES_H$ return $RES_D FWN_i$(Available) outcome = =0 return $RES_H$ return $RES_D CDCN_i$(Available) outcome = =0 return $RES_H$ return $RES_D$ Return $RES_i$ to $GTW_i$ using $MPN$
```

Algorithm 2: MPN primary function.

Latency is the time it takes for data to flow across a network. It also refers to the time it takes for a data packet to be recorded, transferred, processed by several devices, and finally received and decoded. The variation in latencies is estimated by adding transmission time and queuing delay, as shown in Figure 11. Because all contact is done through single-hop data transfers, the latency is nearly the same whether the work is submitted to MPNs or FWNs. The latency in a cloud arrangement is relatively significant due to multihop data transfer outside the network, which is the primary function of the FC. Arbitration time refers to the time limit for the MPN to respond to the gateway devices,

```
Inputs : $SDT_i$
   Output : $RES_i$
   True : Samples in training mode
      True i =1 to $M$ Obtain $SDT_i$ Training $DLA_i$ on $SDT_i$ Bootstrap $DLA_i$ using $BCF_i$ Apply Majority Voting or Weighted Averaging Calculate $PPR_i$ Generate an EDL Model
```

Algorithm 3: Training EDL model(s).

```
Inputs : $TDT_i$
   Output : $RES_i$
   True : Data in test mode
      True j =1 to $N$ Apply $TDT_i$ to generated EDL Model Assign $RES_i$ to the nearest outcome values {0, 1} Calculate predictive class ($RES_i$) Return $RES_i$
```

Algorithm 4: EDL model test cases.

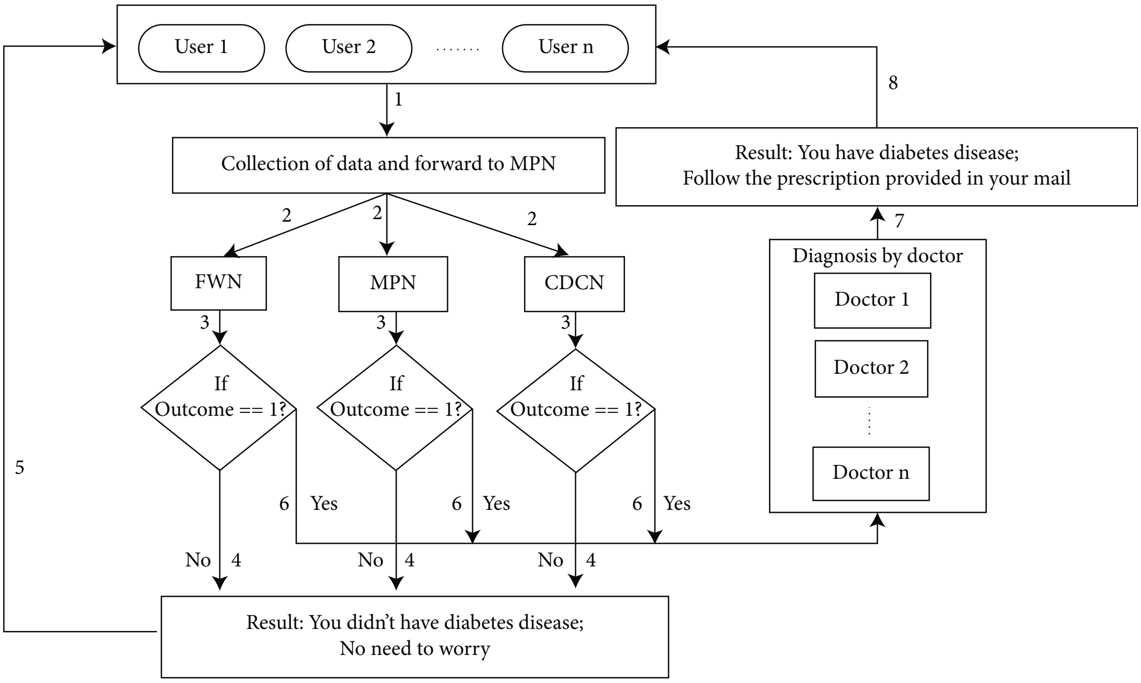


Figure 6: The flowchart of data communications in DiaFog.

Table 6: Observed results of various performance parameters.

| Performance parameters | EDL models | | | | | |
|---|---|---|---|---|---|---|
| | Model 1 | Model 2 | Model 3 | Model 4 | Model 5 | Model 6 |
| Accuracy (in %) | 81.6 | 90.2 | 94.5 | 83.5 | 93.4 | 98.6 |
| Precision | 0.72 | 0.91 | 0.96 | 0.81 | 0.95 | 0.99 |
| Recall | 0.69 | 0.88 | 0.94 | 0.76 | 0.93 | 0.96 |
| $F$-measure | 0.70 | 0.89 | 0.95 | 0.78 | 0.94 | 0.97 |

which might vary depending on the network setup. The arbitration time under different fog situations is shown in Figure 12. The incidence of arbitration is lower when assignments are routed directly to MPN or CDCNs. In other cases, time is spent balancing load between nodes, which reduces the arbitration rate. On the other hand, cloud processing is exceptionally fast due to its superior capability. The processing time is longer because the nodes in FWNs have less processing power and a lower clock frequency. Processing time is the time it takes for a task to be started, processed, and returned to the users. It also changes with setups. The processing characteristics under varied fog conditions are

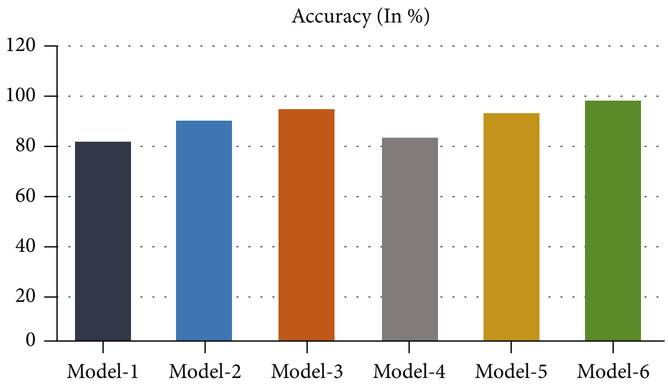


Figure 7: The accuracies of various models.

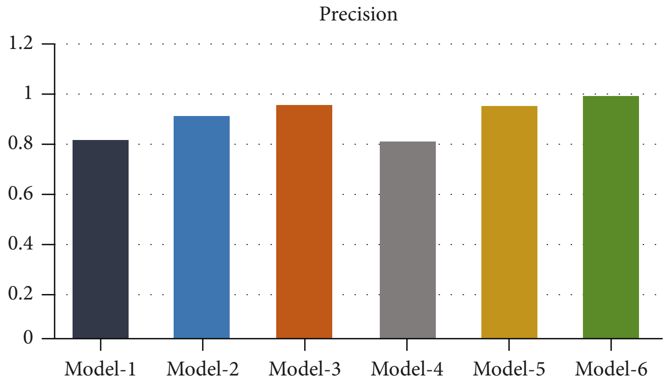


Figure 8: The precisions of various models.

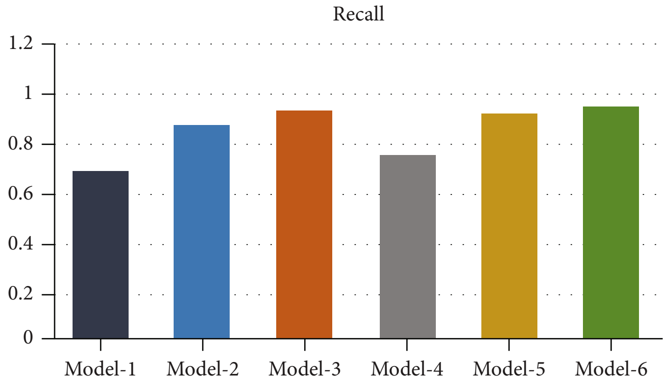


Figure 9: The recalls of various models.

shown in Figure 13 as well. One thing that may be seen is that the total processing time is considerably shorter in the case of cloud communications. Throughput is measured in bits per second, HTTP transactions per day, or millions of instructions per second. It is determined by the successful data packet delivery rate from any node to end users. Figure 14 shows a representation of the variation in throughputs, measured in megabits per second (Mbps), discovered for different configurations. Compared to the CDCN, the throughput numbers for MPN with FWNs are much greater than those for the CDCN. In order to compute the throughput, it is necessary to determine the rate at which data packets are successfully sent from fog nodes to end users. Essentially, jitter is the temporal delay of changes in time. Jitter is the variation in response time between task requests. It is vital for many real-world applications, such as e-Healthcare data analysis. Figure 15 depicts the variation of jitter with different configurations. Because MPN additionally conducts other responsibilities such as arbitration, security checking, and resource management, jitter is more significant in the MPN-alone scenario than when tasks are transmitted to FWNs; nevertheless, jitter is significantly

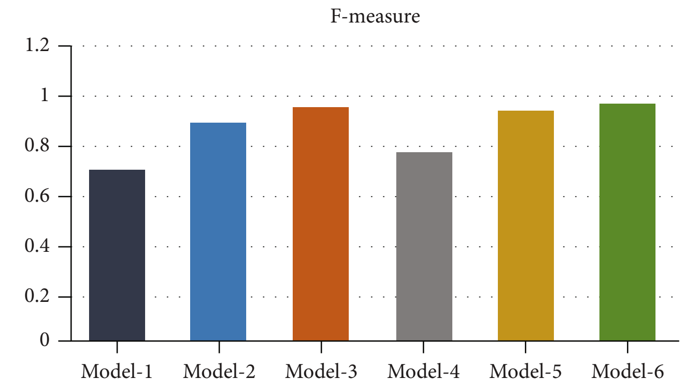


Figure 10: The *F*-measures of various models.

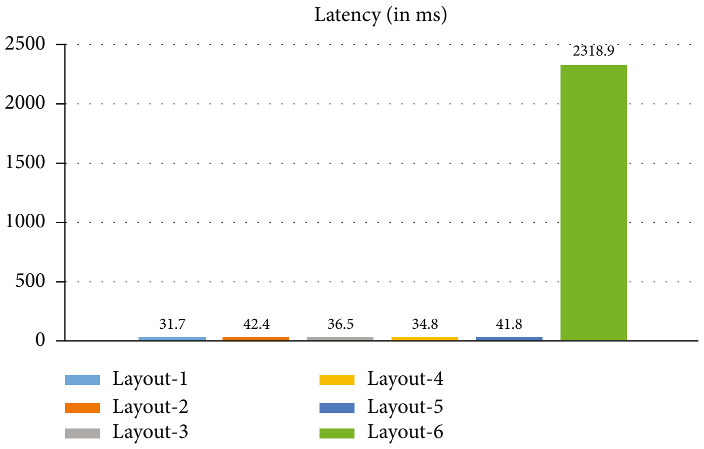


Figure 11: The latency of various configurations.

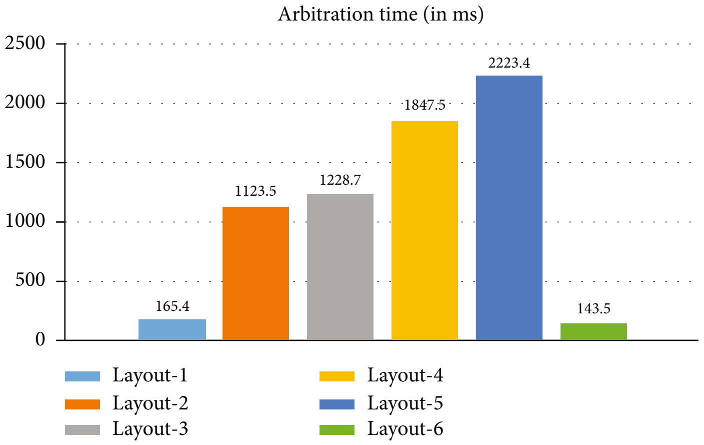


Figure 12: The arbitration time of various configurations.

higher when jobs are supplied to CDCN. Data transferred over an Internet connection is measured in bandwidth in a particular time period. It refers to the amount of data sent across a link in a specific time period (typically measured in kilobits per second (Kbps)). The circumstance, such as MPN alone, FWNs, or cloud, and the number of FWNs impact bandwidth consumption. Figure 16 demonstrates the variation in bandwidth utilization across all FWNs in different configurations. Because of the high number of heartbeat packets, it is required to check for security vulnerabilities and transmit data (through the cloud). As the number of FWNs increases, the amount of bandwidth used increases. Network utilization is the average rate of successful data transmission across a communication link. A fog

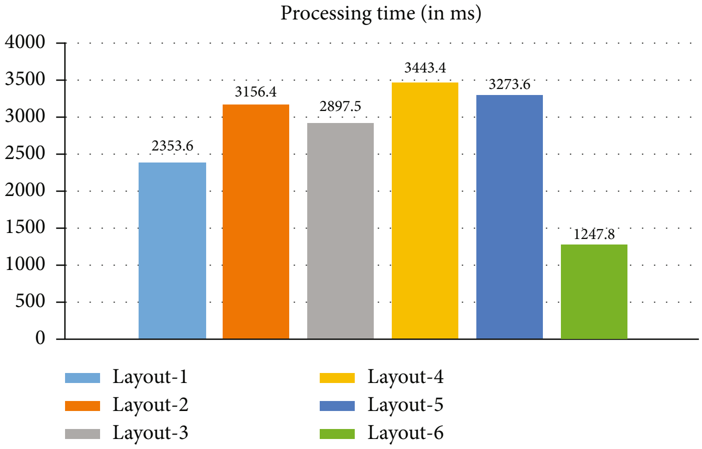


Figure 13: The total processing time of various configurations.

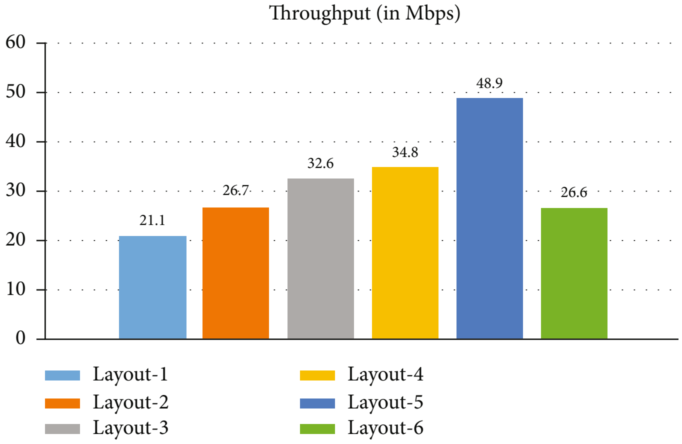


Figure 14: The throughput of various configurations.

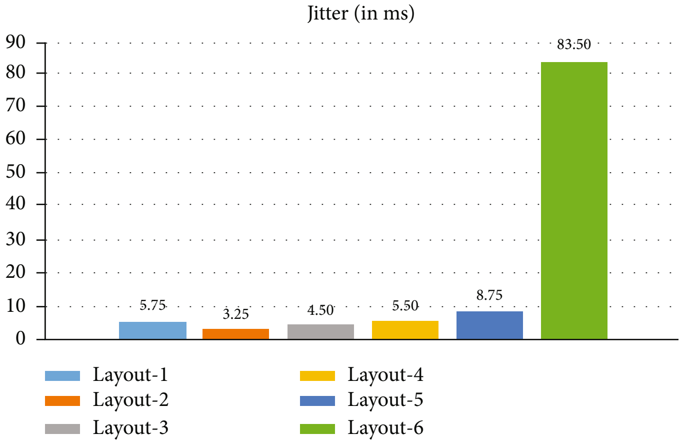


Figure 15: The jitter of various configurations.

computing design uses less network than a cloud computing system. The issue impacts the network utilization, including MPN alone, FWNs, or cloud, and the number of FWNs. Because the fog environment reduces the number of user requests routed to the cloud, as seen in Figure 17, network utilization time in the case of MPN and/or FWNs is much smaller than that of CDCNs.

Energy consumption is the total energy utilized by the system. Sensors and other system components need energy. The physical theorem uses the following formula to compute it [53]:

$$E = T \times p(X). \tag{14}$$

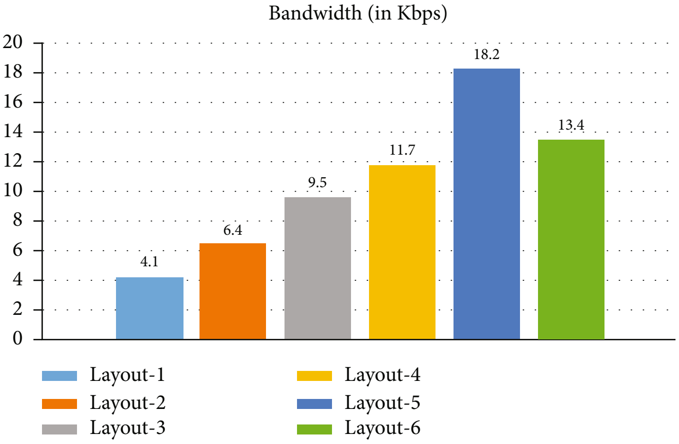


Figure 16: The bandwidth of various configurations.

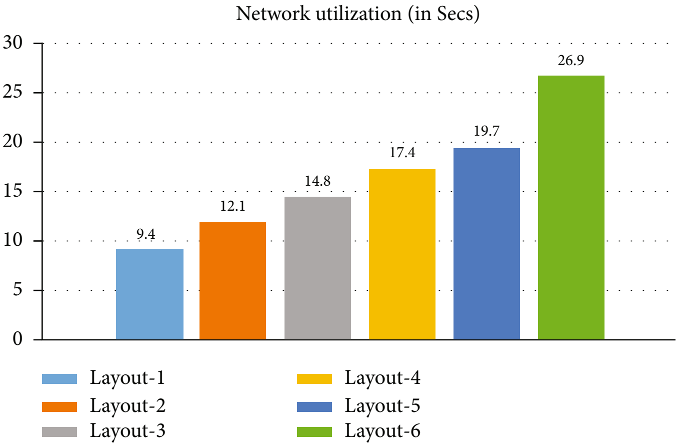


Figure 17: The total network utilization of various configurations.

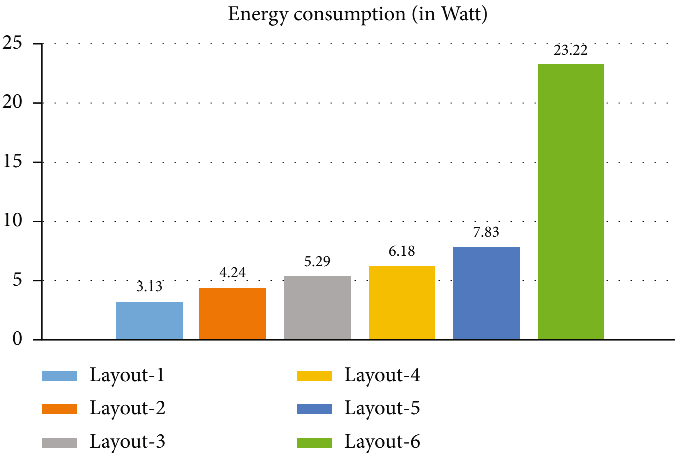


Figure 18: The energy consumption of various configurations.

Here, $E$ is for the energy, $p()$ is the power function, $X$ is the parameter set that impacts power, and $T$ is the task processing time. CDCN uses a significant amount of energy as compared to MPN or FWNs, as seen in Figure 18. CDCNs use a considerable amount of energy compared to FWNs due to this. As the number of FWNs increases, the amount of energy used by the proposed task increases. Table 7 depicts the averages of observed outcomes of various network parameters corresponding to multiple configurations based on the data collected.

Table 7: Observed results of various network parameters.

| Network parameters | Layouts | | | | | |
|---|---|---|---|---|---|---|
| | Layout 1 | Layout 2 | Layout 3 | Layout 4 | Layout 5 | Layout 6 |
| Latency (in ms) | 32.3 | 41.3 | 36.8 | 35.7 | 43.2 | 2459.9 |
| Arbitration time (in ms) | 165.4 | 1123.5 | 1228.7 | 1847.5 | 2223.4 | 143.5 |
| Processing time (in ms) | 2353.6 | 3156.4 | 2897.5 | 3443.4 | 3273.6 | 1247.8 |
| Throughput (in Mbps) | 21.1 | 26.7 | 32.6 | 34.8 | 48.9 | 26.6 |
| Jitter (in ms) | 5.75 | 3.25 | 4.50 | 5.50 | 8.75 | 83.50 |
| Bandwidth (in Kbps) | 4.1 | 6.4 | 9.5 | 11.7 | 18.2 | 13.4 |
| Network utilization (in sec) | 9.4 | 12.1 | 14.8 | 17.4 | 19.7 | 26.9 |
| Energy consumption (in watt) | 3.13 | 4.24 | 5.29 | 6.18 | 7.83 | 23.22 |

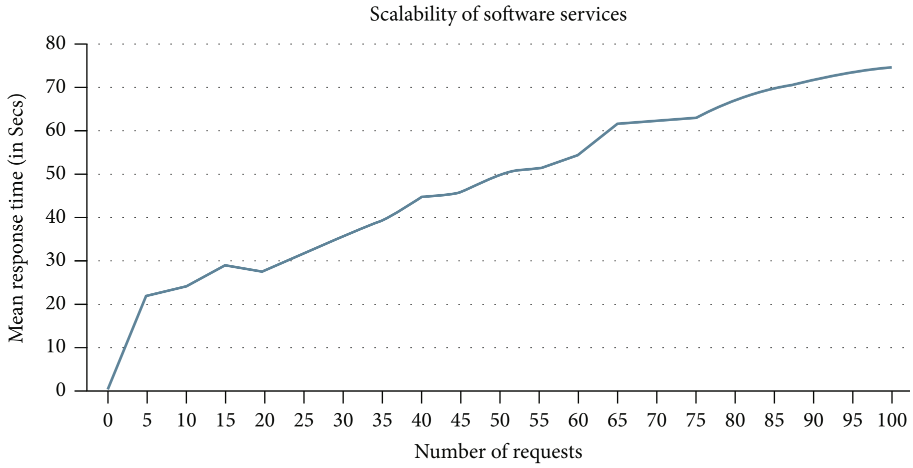


Figure 19: The mean response time concerning the number of requests.

Table 8: Comparison of the proposed work DiaFog with the considered existing works.

| Work | IoT | CC | FC | ML | DL | EL | Network parameters | | | | | | | | | Evaluative parameters | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | AT | LT | PT | TP | EC | BW | JT | NU | SC | Acc | Pre | Rec | F-M |
| [22] | Y | Y | N | Y | Y | N | N | N | N | N | N | N | N | N | N | Y | N | Y | N |
| [15] | Y | Y | Y | NN | Y | N | N | N | N | N | N | N | N | N | N | Y | Y | Y | Y |
| [21] | Y | Y | Y | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N |
| [23] | Y | Y | Y | Y | Y | N | N | N | N | N | N | N | N | N | N | Y | N | N | N |
| [6] | Y | Y | Y | N | N | N | N | N | Y | Y | Y | N | N | Y | Y | N | N | N | N |
| [24] | Y | Y | Y | Y | N | N | N | Y | N | N | Y | N | N | N | N | N | N | N | N |
| [25] | Y | Y | Y | Y | Y | N | N | N | N | N | N | Y | N | N | N | Y | Y | Y | Y |
| [26] | Y | Y | Y | N | N | N | N | N | Y | N | N | N | N | N | N | Y | N | N | N |
| [27] | N | N | N | Y | N | Y | N | N | Y | N | N | N | N | N | N | Y | N | N | N |
| [28] | N | N | N | Y | N | Y | N | N | N | N | N | N | N | N | N | Y | Y | Y | Y |
| [29] | N | N | N | Y | N | Y | N | N | N | N | N | N | N | N | N | Y | Y | Y | Y |
| [30] | Y | Y | Y | Y | N | N | N | N | Y | N | N | N | N | N | N | Y | N | N | N |
| [31] | N | N | N | Y | N | Y | N | N | N | N | N | N | N | N | N | Y | Y | Y | N |
| [32] | Y | Y | N | Y | N | N | N | N | N | N | N | N | N | N | N | Y | Y | Y | Y |
| DiaFog (proposed) | Y | Y | Y | N | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |

Scalability refers to the capacity of the IoT-fog-cloud-based system to raise the resources of software service delivery when higher revenue for the service is needed over time (i.e., a demand scenario) [54, 55]. A scalable infrastructure may increase resources to meet changing application demands while maintaining the infrastructure's constraints. As shown in Figure 19, our main worry is whether the system can scale up in quantity as consumers need it over time.

As the number of requests grows, the mean response time grows as well. The increase in mean reaction time is not exponential but relatively moderate. It is also noted that response times do not fluctuate with increased queries, indicating the research's scalability.

*4.3. Comparison.* The proposed framework is compared with some considered existing works involving various performance and network parameters. We included other aspects such as throughput, network usage, and scalability that were not evaluated before, demonstrating the work's uniqueness. Table 8 depicts a comparison of the proposed work, DiaFog, with several existing results employed in this research relevant to this suggested work. The following abbreviations are used in Table 8: yes (Y), no (N), latency (LT), arbitration time (AT), processing time (PT), throughput (TP), energy consumption (EC), bandwidth (BW), jitter (JT), network utilization (NU), accuracy (Acc), precision (Pre), recall (Rec), and $F$-measure (F-M) are some of the terms used.

## 5. Conclusion and Future Scope

To make an individual's life easier and more consistent, the FC idea with IoT implementations has played an important part in recent days. Because DMDs have a high mortality rate, it is beneficial if the patient may self-diagnose remotely by employing IoT applications. However, formal IoT implications only use CC for real-time data storage, analysis, etc., with several drawbacks like latency and network usage. To solve this problem, FC should be combined with IoT and CC. This research proposes DiaFog, a fog-enabled system for real-time diagnosis of diabetes patients utilizing EDL methods. The model is trained on the combined HFGDD and PIDD diabetes datasets from Kaggle and UCI-ML warehouses. The IDD is both cost-effective and sensitive to real-time diabetes patient diagnosis. Aspects of this study, DiaFog, are examined including accuracy, precision, recall, $F$-measure, latency, arbitration time, jitter, throughput, energy consumption, bandwidth use, network utilization, and scalability. This framework integrates IoT, CC, and FC ideas to guarantee low latency and high accuracy applications in patients' DMD diagnostics and remote prediction. The trials show that it is a viable and user-friendly platform for instantaneous remote DMD diagnosis.

DiaFog may be used to diagnose diabetes patients remotely, anywhere around the globe. This study work's combined IoT, FC, and CC strategy establishes low latency, high precision, etc. The research concluded that it is a robust and user-friendly framework for immediate remote DMD diagnostics. In terms of latency, network usage, energy usage, security, and privacy, the results support FC's appropriateness for real-time remote diabetic patient diagnostics. A person with a few sensors and a smartphone app can be diagnosed anywhere. Every study has its benefits and drawbacks. This study has certain disadvantages; for example, the general execution of the suggested work is difficult and expensive. The dataset we used in this study comprises 2768 instances, which looks little from a DL experiment perspective since more examples mean more accurate and exact conclusions. This modular design relies on a single network platform, a work restriction.

Moreover, the suggested work may be improved by incorporating additional well-known DL principles. This work's expansions may be used to treat various chronic disorders. Also, alternative CC platforms such as edge computing, mist computing, and surge computing should be used to extend this suggested architecture. Another area where we should work in the future is the problem of a single network platform. Individuals must be made aware of the importance of IoT, cloud, fog, and edge computing and other related technologies and their worldwide consequences in recent days.

## Notations

$\mathrm{USR}_i$: Users' data $\{\mathrm{USR}_1, \mathrm{USR}_2, \mathrm{USR}_3, \cdots\}$

$\mathrm{GTW}_i$: Gateways $\{\mathrm{GTW}_1, \mathrm{GTW}_2, \mathrm{GTW}_3, \cdots\}$

$\mathrm{IHD}_i$: IoT health devices $\{\mathrm{IHD}_1, \mathrm{IHD}_2, \mathrm{IHD}_3, \cdots\}$

$\mathrm{ATB}_i$: Attributes/features/characteristics ($\mathrm{ATB}_1$ for age, $\mathrm{ATB}_2$ for blood pressure, $\mathrm{ATB}_3$ for pregnancies, $\mathrm{ATB}_4$ for glucose, $\mathrm{ATB}_5$ for skin thickness, $\mathrm{ATB}_6$ for DPF, $\mathrm{ATB}_7$ for insulin, $\mathrm{ATB}_8$ for BMI, and $\mathrm{ATB}_9$ for outcome)

MPN: Master PC node

$\mathrm{FWN}_i$: Fog worker nodes $\{\mathrm{FWN}_1, \mathrm{FWN}_2, \mathrm{FWN}_3, \cdots\}$

$\mathrm{CDCN}_i$: Cloud data center nodes $\{\mathrm{CDCN}_1, \mathrm{CDCN}_2, \mathrm{CDCN}_3, \cdots\}$

$\mathrm{SDT}_i$: Training samples $\{\mathrm{SDT}_1, \mathrm{SDT}_2, \mathrm{SDT}_3, \cdots\}$

$\mathrm{TDT}_i$: Test data $\{\mathrm{TDT}_1, \mathrm{TDT}_2, \mathrm{TDT}_3, \cdots\}$

$\mathrm{BCF}_i$: Bagging classifier algorithms $\{\mathrm{BCF}_1, \mathrm{BCF}_2, \mathrm{BCF}_3 \cdots\}$

$\mathrm{DLA}_i$: DL algorithms $\{\mathrm{DLA}_1, \mathrm{DLA}_2, \mathrm{DLA}_3, \cdots\}$

$M$: Maximum number of iterations in training EDL model(s)

$N$: Number of test data

$\mathrm{PPR}_i$: Performance parameters ($\mathrm{PPR}_1$ for accuracy, $\mathrm{PPR}_2$ for blood precision, $\mathrm{PPR}_3$ for recall, and $\mathrm{PPR}_4$ for $F$-measure)

$\mathrm{RES}_i$: Results ($\mathrm{RES}_D$, i.e., diabetic when the value of outcome is 1, $\mathrm{RES}_H$, i.e., healthy when the value of outcome is 0).

## Data Availability

The IoT data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors of this manuscript declared that they do not have any conflict of interest.

## Acknowledgments

This work is partially supported by DST/TDT/DDP-38/2021, Device Development Programme (DDP), by the Department of Science Technology (DST), Ministry of Science and Technology, Government of India. Sajal Biring would like to thank Ministry of Science and Technology, Taiwan, under grant no. MOST 110-2221-E-131-019 for financial support.

## References

[1] S. Dubravac and C. Ratti, *The Internet of Things: Evolution or Revolution?*, vol. 1, Wiley, Hoboken, NJ, USA, 2015.

[2] A. Lakhan, M. A. Mohammed, S. Kozlov, and J. J. P. C. Rodrigues, "Mobile-fog-cloud assisted deep reinforcement learning and blockchain-enable IoMT system for healthcare workflows," *Transactions on Emerging Telecommunications Technologies*, vol. 1, 2021.

[3] N. Matar and M. Alnabhan, "Evaluating E-health services and patients requirements in Jordanian hospitals," *International Arab Journal of e-Technology*, vol. 3, no. 4, pp. 250–257, 2014.

[4] A. Haboush, M. N. Mohanty, B. K. Pattanayak, and M. Al-Tarazi, "A framework for wireless sensor network fault rectification," *International journal of multimedia and ubiquitous Engineering*, vol. 9, no. 1, pp. 133–142, 2014.

[5] World Health Organization, *Global Status Report on Non-Communicable Diseases 2014*, WHO, 2014, March 2020, https://apps.who.int/iris/handle/10665/148114.

[6] T. M. Fernández-Caramés, I. Froiz-Míguez, O. Blanco-Novoa, and P. Fraga-Lamas, "Enabling the Internet of mobile crowdsourcing health things: a mobile fog computing, blockchain and iot based continuous glucose monitoring system for diabetes mellitus research and care," *Sensors*, vol. 19, no. 15, p. 3319, 2019.

[7] A. Pati, M. Parhi, and B. K. Pattanayak, "COVID-19 pandemic analysis and prediction using machine learning approaches in India," in *Advances in Intelligent Computing and Communication*, pp. 307–316, Springer, Singapore, 2021.

[8] S. M. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K. S. Kwak, "The Internet of things for health care: a comprehensive survey," *IEEE Access*, vol. 3, pp. 678–708, 2015.

[9] S. Tuli, R. Mahmud, S. Tuli, and R. Buyya, "FogBus: a blockchain-based lightweight framework for edge and fog computing," *Journal of Systems and Software*, vol. 154, pp. 22–36, 2019.

[10] A. M. Rahmani, T. N. Gia, B. Negash et al., "Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: a fog computing approach," *Future Generation Computer Systems*, vol. 78, pp. 641–658, 2018.

[11] A. A. Mutlag, M. K. Abd Ghani, N. Arunkumar, M. A. Mohammed, and O. Mohd, "Enabling technologies for fog computing in healthcare IoT systems," *Future Generation Computer Systems*, vol. 90, pp. 62–78, 2019.

[12] S. Tuli, N. Basumatary, and R. Buyya, "EdgeLens: deep learning based object detection in integrated IoT, fog and cloud computing environments," in *2019 4th International Conference on Information Systems and Computer Networks, ISCON 2019*, pp. 496–502, Mathura, India, 2019.

[13] S. S. Gill, R. C. Arya, G. S. Wander, and R. Buyya, "Fog-based smart healthcare as a big data and cloud service for heart patients using IoT," in *Lecture Notes on Data Engineering and Communications Technologies*, vol. 26, pp. 1376–1383, Springer Science and Business Media Deutschland GmbH, 2019.

[14] D. C. Klonoff, "Fog computing and edge computing architectures for processing data from diabetes devices connected to the medical Internet of things," *Journal of Diabetes Science and Technology*, vol. 11, no. 4, pp. 647–652, 2017.

[15] R. Priyadarshini, R. K. Barik, and H. Dubey, "DeepFog: fog computing-based deep neural architecture for prediction of stress types, diabetes and hypertension attacks," *Computation*, vol. 6, no. 4, p. 62, 2018.

[16] R. O. Aburukba, M. AliKarrar, T. Landolsi, and K. El-Fakih, "Scheduling Internet of things requests to minimize latency in hybrid fog-cloud computing," *Future Generation Computer Systems*, vol. 111, pp. 539–551, 2020.

[17] I. H. Laradji, M. Alshayeb, and L. Ghouti, "Software defect prediction using ensemble learning on selected features," *Information and Software Technology*, vol. 58, pp. 388–402, 2015.

[18] G. Haixiang, L. Yijing, L. Yanan, L. Xiao, and L. Jinling, "BPSO-Adaboost-KNN ensemble learning algorithm for multi-class imbalanced data classification," *Engineering Applications of Artificial Intelligence*, vol. 49, pp. 176–193, 2016.

[19] L. Tang, Y. Wu, and L. Yu, "A non-iterative decomposition-ensemble learning paradigm using RVFL network for crude oil price forecasting," *Applied Soft Computing Journal*, vol. 70, pp. 1097–1108, 2018.

[20] S. El-Sappagh, M. Elmogy, F. Ali, T. Abuhmed, S. M. R. Islam, and S. K. Kwak, "A comprehensive medical decision–support framework based on a heterogeneous ensemble classifier for diabetes prediction," *Electronics*, vol. 8, no. 6, p. 635, 2019.

[21] T. M. Fernández-Caramés and P. Fraga-Lamas, "Design of a fog computing, blockchain and IoT-based continuous glucose monitoring system for crowdsourcing mHealth," *Multidisciplinary Digital Publishing Institute Proceedings*, vol. 4, no. 1, p. 37, 2018.

[22] P. Kaur, N. Sharma, A. Singh, and B. Gill, "CI-DPF: a cloud IoT based framework for diabetes prediction," in *2018 IEEE 9th annual information technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 654–660, Vancouver, BC, Canada, 2018.

[23] R. K. Barik, R. Priyadarshini, H. Dubey, V. Kumar, and K. Mankodiya, "FogLearn," *International Journal of Fog Computing (IJFC)*, vol. 1, no. 1, pp. 15–34, 2018.

[24] T. N. Gia, I. B. Dhaou, M. Ali et al., "Energy efficient fog-assisted IoT system for monitoring diabetic patients with cardiovascular disease," *Future Generation Computer Systems*, vol. 93, pp. 198–211, 2019.

[25] M. Devarajan, V. Subramaniyaswamy, V. Vijayakumar, and L. Ravi, "Fog-assisted personalized healthcare-support system for remote patients with diabetes," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 10, pp. 3747–3760, 2019.

[26] M. Abdel-Basset, G. Manogaran, A. Gamal, and V. Chang, "A novel intelligent medical decision support model based on soft computing and IoT," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4160–4170, 2019.

[27] A. U. Haq, J. P. Li, J. Khan et al., "Intelligent machine learning approach for effective recognition of diabetes in E-healthcare using clinical data," *Sensors*, vol. 20, no. 9, p. 2649, 2020.

[28] S. Kumari, D. Kumar, and M. Mittal, "An ensemble approach for classification and prediction of diabetes mellitus using soft voting classifier," *International Journal of Cognitive Computing in Engineering*, vol. 2, pp. 40–46, 2021.

[29] G. Geetha and K. M. Prasad, "An hybrid ensemble machine learning approach to predict type 2 diabetes mellitus," *Webology*, vol. 18, no. Special Issue 02, pp. 311–331, 2021.

[30] P. G. Shynu, V. G. Menon, R. L. Kumar, S. Kadry, and Y. Nam, “Blockchain-based secure healthcare application for diabetic-cardio disease prediction in fog computing,” *IEEE Access*, vol. 9, pp. 45706–45720, 2021.

[31] A. Singh, A. Dhillon, N. Kumar, M. S. Hossain, G. Muhammad, and M. Kumar, “eDiaPredict: an ensemble-based framework for diabetes prediction,” *ACM Transactions on Multimedia Computing Communications and Applications*, vol. 17, no. 2s, pp. 1–26, 2021.

[32] D. S. Rajput, S. M. Basha, Q. Xin et al., “Providing diagnosis on diabetes using cloud computing environment to the people living in rural areas of India,” *Journal of ambient intelligence and humanized Computing*, vol. 13, no. 5, pp. 2829–2840, 2022.

[33] H. F. Germany, *Diabetes Dataset*, Kaggle, 2020, October 2020, https://www.kaggle.com/johndasilva/diabetes.

[34] Pima Indian Diabetes Dataset, “UCI-ML repository,” 2020, March 2020, https://archive.ics.uci.edu/ml/datasets/pima+indians+diabetes.

[35] B. Ihnaini, M. A. Khan, T. A. Khan et al., “A smart healthcare recommendation system for multidisciplinary diabetes patients with data fusion based on deep ensemble learning,” *Computational Intelligence and Neuroscience*, vol. 2021, 11 pages, 2021.

[36] B. Shickel, P. J. Tighe, A. Bihorac, and P. Rashidi, “Deep EHR: a survey of recent advances in deep learning techniques for electronic health record (EHR) analysis,” *IEEE Journal of Biomedical and Health Informatics*, vol. 22, no. 5, pp. 1589–1604, 2018.

[37] H. Yan, Y. Jiang, J. Zheng, C. Peng, and Q. Li, “A multilayer perceptron-based medical decision support system for heart disease diagnosis,” *Expert Systems with Applications*, vol. 30, no. 2, pp. 272–281, 2006.

[38] T. T. Hasan, M. H. Jasim, and I. A. Hashim, “Heart disease diagnosis system based on multi-layer perceptron neural network and support vector machine,” *International Journal of Current Engineering and Technology*, vol. 77, no. 55, pp. 2277–4106, 2017.

[39] V. Sharma, A. Rasool, and G. Hajela, “Prediction of heart disease using DNN,” in *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, pp. 554–562, Coimbatore, India, 2020.

[40] P. Ramprakash, R. Sarumathi, R. Mowriya, and S. Nithyavishnupriya, “Heart disease prediction using deep neural network,” in *2020 International Conference on Inventive Computation Technologies (ICICT)*, pp. 666–670, Coimbatore, India, 2020.

[41] S. Sharma and M. Parmar, “Heart diseases prediction using deep learning neural network model,” *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 9, no. 3, pp. 2244–2248, 2020.

[42] S. Tuli, N. Basumatary, S. S. Gill et al., “HealthFog: an ensemble deep learning based smart healthcare system for automatic diagnosis of heart diseases in integrated IoT and fog computing environments,” *Future Generation Computer Systems*, vol. 104, pp. 187–200, 2020.

[43] I. E. Livieris, E. Pintelas, S. Stavroyiannis, and P. Pintelas, “Ensemble deep learning models for forecasting cryptocurrency time-series,” *Algorithms*, vol. 13, no. 5, p. 121, 2020.

[44] X. Y. Gao, A. A. Ali, H. Shaban Hassan, and E. M. Anwar, “Improving the accuracy for analyzing heart diseases prediction based on the ensemble method,” *Complexity*, vol. 2021, 10 pages, 2021.

[45] F. Ali, S. El-Sappagh, S. R. Islam et al., “A smart healthcare monitoring system for heart disease prediction based on ensemble deep learning and feature fusion,” *Information Fusion*, vol. 63, pp. 208–222, 2020.

[46] M. N. Uddin and R. K. Halder, “An ensemble method based multilayer dynamic system to predict cardiovascular disease using machine learning approach,” *Informatics in Medicine Unlocked*, vol. 24, p. 100584, 2021.

[47] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, and R. Buyya, “iFogSim: a toolkit for modeling and simulation of resource management techniques in the Internet of things, edge and fog computing environments,” *Software: Practice and Experience*, vol. 47, no. 9, pp. 1275–1296, 2017.

[48] S. Narula and A. Jain, “Cloud computing security: Amazon web service,” in *2015 Fifth International Conference on Advanced Computing & Communication Technologies*, pp. 501–505, Haryana, India, 2015.

[49] C. Vecchiola, X. Chu, and R. Buyya, “Aneka: a software platform for .NET-based cloud computing,” *High speed and large scale scientific computing*, vol. 18, pp. 267–295, 2009.

[50] A. Pati, M. Parhi, and B. K. Pattanayak, “IDMS: an integrated decision making system for heart disease prediction,” in *2021 1st Odisha International Conference on Electrical Power Engineering, Communication and Computing Technology (ODICON)*, pp. 1–6, Bhubaneswar, India, 2021.

[51] A. Pati, M. Parhi, and B. K. Pattanayak, “IADP: an integrated approach for diabetes prediction using classification techniques,” in *Advances in Distributed Computing and Machine Learning*, pp. 287–298, Springer, Singapore, 2022.

[52] Explore MIT App Inventor, *MIT App Inventor*July 2020, https://appinventor.mit.edu/.

[53] H. Liu, F. Yan, S. Zhang, T. Xiao, and J. Song, “Source-level energy consumption estimation for cloud computing tasks,” *IEEE Access*, vol. 6, pp. 1321–1330, 2017.

[54] S. Lehrig, H. Eikerling, and S. Becker, “Scalability, elasticity, and efficiency in cloud computing: a systematic literature review of definitions and metrics,” in *Proceedings of the 11th International ACM SIGSOFT Conference on Quality of Software Architectures*, pp. 83–92, Montreal , Canada, 2015.

[55] A. A. S. Ahmad and P. Andras, “Scalability analysis comparisons of cloud-based software services,” *Journal of Cloud Computing*, vol. 8, no. 1, pp. 1–17, 2019.