



Article

An IoT-Fog-Cloud Integrated Framework for Real-Time Remote Cardiovascular Disease Diagnosis

Abhilash Pati ¹, Manoranjan Parhi ², Mohammad Alnabhan ^{3,*}, Binod Kumar Pattanayak ¹, Ahmad Khader Habboush ⁴ and Mohammad K. Al Nawayseh ⁵

¹ Department of Computer Science and Engineering, Siksha 'O' Anusandhan (Deemed to be University), Bhubaneswar 751030, Odisha, India

² Centre for Data Science, Siksha 'O' Anusandhan (Deemed to be University), Bhubaneswar 751030, Odisha, India

³ Department of Computer Science, Princess Sumaya University for Technology, Amman 11941, Jordan

⁴ Department of Computer Seines and Information Technology, Jerash University, Jerash 26150, Jordan

⁵ Department of Management Information Systems, Business School, The University of Jordan, Amman 11942, Jordan

* Correspondence: m.alnabhan@psut.edu.jo

Abstract: Recently, it has proven difficult to make an immediate remote diagnosis of any coronary illness, including heart disease, diabetes, etc. The drawbacks of cloud computing infrastructures, such as excessive latency, bandwidth, energy consumption, security, and privacy concerns, have lately been addressed by Fog computing with IoT applications. In this study, an IoT-Fog-Cloud integrated system, called a Fog-empowered framework for real-time analysis in heart patients using ENsemble Deep learning (FRIEND), has been introduced that can instantaneously facilitate remote diagnosis of heart patients. The proposed system was trained on the combined dataset of Long-Beach, Cleveland, Switzerland, and Hungarian heart disease datasets. We first tested the model with eight basic ML approaches, including the decision tree, logistic regression, random forest, naive Bayes, k-nearest neighbors, support vector machine, AdaBoost, and XGBoost approaches, and then applied ensemble methods including bagging classifiers, weighted averaging, and soft and hard voting to achieve enhanced outcomes and a deep neural network, a deep learning approach, with the ensemble methods. These models were validated using 16 performance and 9 network parameters to justify this work. The accuracy, PPV, TPR, TNR, and F1 scores of the experiments reached 94.27%, 97.59%, 96.09%, 75.44%, and 96.83%, respectively, which were comparatively higher when the deep neural network was assembled with bagging and hard-voting classifiers. The user-friendliness and the inclusion of Fog computing principles, instantaneous remote cardiac patient diagnosis, low latency, and low energy consumption, etc., are advantages confirmed according to the achieved experimental results.

Keywords: internet of things (IoT); fog computing (FC); cloud computing (CC); ensemble machine learning (EML); ensemble deep learning (EDL); heart disease diagnosis



Citation: Pati, A.; Parhi, M.; Alnabhan, M.; Pattanayak, B.K.; Habboush, A.K.; Al Nawayseh, M.K. An IoT-Fog-Cloud Integrated Framework for Real-Time Remote Cardiovascular Disease Diagnosis. *Informatics* **2023**, *10*, 21. <https://doi.org/10.3390/informatics10010021>

Academic Editor: Antony Bryant

Received: 22 November 2022

Revised: 16 January 2023

Accepted: 18 January 2023

Published: 6 February 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The IoT is indispensable for long-distance connectivity, whereas the Internet is the most brilliant invention ever. The IoT in smart healthcare schemes, that is, the Internet of Medical Things (IoMT), allows health specialists to remotely diagnose patients, while patients can easily use medical services [1]. Fog computing (FC) and its base, cloud computing (CC), are the key healthcare technologies. Fog-cloud computing nodes provide paid e-healthcare services for all application classes and types [2]. The most prevalent and complicated health issue today on Earth is heart disease, which is the second most common cause of death and results from arterial infections and blood clots [3]. Arrhythmias, coronary artery disease, and cardiac valve dysfunction increase mortality [4]. To Obesity, aging, depression, smoking, sugar intake, inactivity, and high blood pressure are risk factors for developing

cardiovascular diseases [5,6] and must be avoided to live healthily and avoid a heart attack, heart failure, stroke, and blindness. Machine learning (ML) and its advancement, that is, deep learning (DL), along with ensemble learning (EL), improve predictive analytical accuracy and performance [7].

Currently, IoT-enabled smart devices play a crucial role in e-healthcare systems. IoT engagements have ushered in the notion of CC. CC and its expanded variants, such as FC and edge computing (EC), have recently played an important role in real-world applications [8]. The significant delay of CC makes it unsuitable for real-time applications such as e-healthcare. The FC at the network edge can run applications near data sources and act as a connection between endpoints and cloud servers [9]. Both FC and CC platforms have similar intakes, resulting in the advantages of FC with the use of latency reduction in cloud servers [10]. Concerns regarding readability, scalability, and other aspects of e-healthcare are addressed by Fog-based systems [11]. The IoT-Fog-Cloud integrated approach aims to improve communication between nodes, thereby conserving the bandwidth that can be used in areas such as disease diagnosis and improved prediction accuracies. Figure 1 depicts the IoT-Fog-Cloud integration.

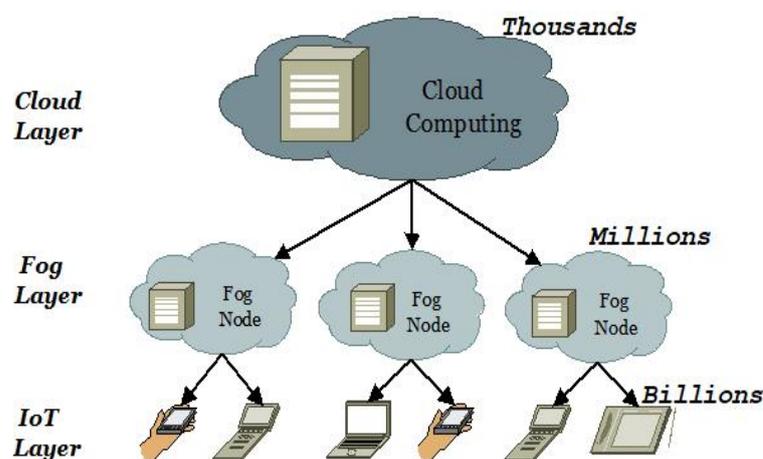


Figure 1. The IoT-Fog-Cloud integration.

1.1. Research Gap and Research Questions

A variety of research has been conducted using the combined concepts of IoT, CC, and FC, especially in smart homes and cities. The integration concept plays a significant role in e-healthcare systems. It should be noted that these studies are based on hardware but have an instantaneous impact on the world; however, they may be a single-time investment for a specific disease. Instantaneous remote diagnosis of any disease has been investigated in recent years.

The following research questions (RQs) were taken into account for this investigation:

RQ1. What is the main e-healthcare goal of the IoT-Fog-Cloud combined approach?

RQ2. Can the user utilize the suggested work to prevent other parties from accessing their medical records?

RQ3. What are the benefits of employing this combined approach in e-healthcare systems?

RQ4. What are the major paybacks of ensemble learning procedures for predicting a particular disease?

RQ5. What are the main results of employing an integrated preprocessed dataset to diagnose cardiac diseases?

1.2. Motivations and Objective

The increasing mortality rate due to cardiovascular disease has been threatening individuals globally in recent years. Another issue in society is getting medical care close to these individuals in real time. There is a demand for remote and immediate diagnosis of heart patients. The primary goal of this study was the remote instantaneous diagnosis of patients with heart disease. In this paper, a Fog-enabled framework for real-time diagnosis in heart patients using ensemble deep learning (FRIEND) is presented to improve the accuracy of diagnosis with unexplained data instantaneously concerning the integrated dataset of the four HDDs taken from the UCIML warehouse. We involved 16 performance parameters on various ensemble machine learning (EML) and ensemble deep learning (EDL) approaches. In addition, the suggested work was validated with nine network parameters.

1.3. Contributions

The basic contributions of this study are summarized as follows:

1. Identifying and assessing earlier work in heart disease diagnoses in real time from abroad;
2. Examining the work using many measures and an integrated heart disease dataset;
3. Constructing a simple, automated, EML- and EDL-based heart patient diagnosis system;
4. Using a variety of IoT-Fog-Cloud frameworks and simulators to perform the most accurate predictive analytics;
5. Addressing the results and comparing them with the results of earlier studies,
6. Finding crucial areas where this can be combined with Fog computing studies to promote the application of these methods.

1.4. Paper Organization

The remainder of this paper is organized as follows: Section 2 summarizes the field's research. Section 3 presents the datasets and methods used in this study. Section 4 discusses the architectural aspects of the work, including its design and working principles. Section 5 analyzes the efficiency of the proposed method by comparing it with related efforts, and Section 6 summarizes the pros and downsides and suggests expansions.

2. Existing Works

Caliskan and Yuksel [12] created a deep neural network (DNN) approach for coronary artery disease using Cleveland, Hungarian, Switzerland, and LongBeach HDDs, claiming that the model is cheaper than standard diagnostic methods. Choi et al. [13] introduced a recurrent neural network (RNN)-based model called GRAM, that is, a graph-based attention model, based on accuracy, data needs, and interpretability as evaluative parameters on two datasets from the Sutter Palo Alto Medical Foundation (PAMF) and MIMIC-III and claimed to have improved performance on low frequency and small datasets. Ali and Ghazal [14] invented RHAMDS using gesture and voice control for instantaneous data. Their approach minimized and eliminated automobile collisions by instantaneously recognizing the heart attacks of drivers. Gupta [15] used KNN and GA on a Hungarian HDD and discovered that the hybrid strategy is more precise and effective than the alternatives. Mustafa et al. [16] suggested an ensemble technique by merging five classifiers: SVM, ANN, NB, RF, and RA, for the diagnosis and prediction of heart disease recurrence. Zhenya and Zhang [17] presented a cost-effective ensemble technique for the diagnosis of cardiac diseases. On Cleveland, Statlog, and Hungarian HDDs, recall, precision, specificity, E, G-mean, MC, and AUC were enhanced. Ali et al. [18] predicted cardiovascular disease using Cleveland and Hungarian HDDs, considering precision, accuracy, F-measure, recall, RMSE, and MAE, which achieved enhanced cardiac disease prediction.

Moghadas et al. [19] presented a method for distantly nursing an individual's health using an Arduino board and an AD8232 sensor module and found KNN to be the best cardiac arrhythmia technique among NB, RF, KNN, and SVM linear. Baccouche et al. [20]

built an EL approach based on a CNN with BiGRU or BiLSTM to classify heart illness data from Mexico’s Medica Norte Hospital. They stated that the framework avoids imbalanced HDD classification and offers more accurate findings. Sun et al. [21] introduced FogMed, a Fog-based system for predicting heart sickness on an ECG dataset, claiming superior performance compared to earlier CC approaches. Tuli et al. [22] created an EDL-based smart healthcare system named HealthFog for autonomous cardiac patient analysis in FC and IoT-integrated environments, considering latency, power consumption, network bandwidth, accuracy, execution time, and jitter on Cleveland HDD, and found that it diagnoses cardiac patients remotely in real-time. Sharma and Parmar [23] created a neural network (NN)-based DL approach, DNN, for predicting cardiovascular disease using accuracy as the assessing parameter on the Cleveland HDD, providing the best accuracy compared to previous systems. Uddin and Halder [24] introduced MLDS, an EM-based multilayer dynamic technique, with feature selection approaches, including IGAE, GAIN, CAE, lasso, and ETs, and found that this procedure may predict cardiovascular disease effectively. The studies carried out in this field are summarized in Table 1, which provides an overview of these studies.

Table 1. A summary of some state-of-the-art works that are deemed to be connected.

Work	Materials and Methods	Dataset Used	Evaluative Measures	Findings
[12]	Proposed DNN classifier for categorizing coronary artery disease	Switzerland, LongBeach, Hungarian, and Cleveland HDDs	Accuracy, sensitivity, and specificity	The suggested classifier is affordable and readily available
[13]	Introduced RNN-based model termed GRAM, i.e., graph-based attention model	Datasets from PAMF and MIMIC-III	Accuracy, data needs, and interpretability	The proposed GRAM approach significantly improves performance on low-frequency diseases and small datasets
[14]	Introduced RHAMDS, i.e., real-time heart attack mobile detection service	Real-time dataset	Voice and gesture control	The planned RHAMDS reduces and prevents automobile crashes by identifying drivers who are having heart attacks
[15]	Introduced a hybrid technique combining KNN and GA approaches	Hungarian HDD	Accuracy	The suggested paradigm yields outcomes that are more precise and effective
[16]	Proposed an ensemble model with SVM, ANN, NB, RA, and RF to investigate and forecast the recurrence of cardiovascular illness	Hungarian and Cleveland HDDs	Accuracy, precision, F-M, ROC, RMSE, and kappa statistics	This ensemble technique achieves both high readability and predictive accuracy
[17]	Proposed a 5-classifier-based affordable ensemble technique to enhance the detection of heart disease	Cleveland, Statlog, and Hungarian HDDs	Precision, AUC specificity, E, recall, MC, and G-mean	This ensemble method outperforms others and offers a possible alternative
[18]	Feature fusion and EDL are used to predict cardiovascular disease	Cleveland and Hungarian HDDs	Accuracy, recall, MAE, precision, F-measure, and RMSE	This ensemble strategy leads to more accurate heart disease predictions
[19]	Presented a method for remote monitoring of a patient utilizing an Arduino board and an AD8232 sensor module	Tele ECG	Accuracy and macro-F1	The KNN classifier-based system is ideal for both classification and validation
[20]	Proposed CNN-based EL framework for heart disease data classification	Mexico’s Medica Dataset	Accuracy and F1-score	The suggested approach classifies imbalanced HDDs more accurately
[21]	Proposed FogMed framework for heart disease prediction	ECG dataset	Accuracy and time efficiency	The proposed framework outperforms existing CC methods.
[22]	Introduced EDL-based HealthFog for automatic cardiac patient analysis in FC and IoT	Cleveland HDD	Accuracy and network parameters	HealthFog detects cardiac patients remotely in real-time
[23]	Introduced NN-based DL for cardiovascular disease prediction	Cleveland HDD	Accuracy	This DNN technology outperforms previous methods
[24]	IGAE, GAIN, CAE, lasso, and ETs were used to introduce MLDS, an EM-based multilayer dynamic system	Realistic dataset from Kaggle	Accuracy, precision, ROC, and AUC	Cardiovascular disease can be accurately predicted by the suggested MLDS

3. Preliminaries

3.1. Dataset Description, Preparation, and Preprocessing

There are various heart disease datasets (HDDs) accessible worldwide, including Cleveland, Hungarian, Switzerland, and Long Beach HDDs. All these HDDs' attributes are numeric and taken from the UCI-ML warehouse [25]. In this study, an integrated heart disease dataset (IHDD) was formed by combining these HDDs, as all datasets had the same instance format, as shown in Table 2.

Table 2. A brief summary of the IHDD Dataset.

Dataset Used	Quantity of Attributes	Quantity of Instances
IHDD	76 (11 considered in this Work)	303 (Cleveland HDD), 123 (Switzerland HDD), 294 (Hungarian HDD), and 200 (Long Beach VA HDD)

Although there were 76 raw attributes, only 11 were considered for the investigation in this study, which are discussed in Table 3. In the raw datasets, 0–4 indicate the disease severity. Attributes 1, 2, 3, and 4 are considered as 1 for binary classification, indicating people with heart problems, and 0 as those without. Here, 10 items, including age, sex, rest ECG, etc., are collected via a mobile app, and the 11th datum, that is, the target, is used as a prediction. The IHDD is separated into training, test, and validation data in a ratio of 70:15:15. For better accuracy of the prediction results from the model, the data were preprocessed and filtered out with data cleaning, and data normalization was performed before subjecting the dataset to training and testing. Table 4 presents the representation of a subset of the dataset.

Table 3. Short description of attributes of IHDD dataset.

Sl.	Attributes	Meaning	Values
1	Age	Patient's age in years	A number value (between 29 and 79)
2	Sex	Patient's gender	0: female; 1: male
3	Chest pain	Type of chest pains	1: typical angina; 2: atypical angina; 3: non-anginal pain; 4: asymptomatic
4	Blood pressure	Resting blood pressure level (in mm Hg on admission to the hospital)	A numeric value (between 94 and 200)
5	Smoking years	Years that a person has smoked	A numeric value (between 5 and 45)
6	Fasting blood sugar	Level of fasting blood sugar	0: false, if FBS is less than or equal to 120 mg/dL; 1: true, if FBS is greater than 120 mg/dL
7	Diabetes history	Prior diabetic history	0: no history of diabetes; 1: history of diabetes
8	Family coronary history	Coronary disease in the family	0: no; 1: yes
9	Rest ECG	Electrocardiographic readings while at rest	0: normal; 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of >0.05 mV); 2: showing probable or definite left ventricular hypertrophy by Estes' criteria
10	Pulse rate	Reached maximum heart rate	A numeric value (between 71 and 202)
11	Target	Heart disease diagnosis	0: for <50% diameter narrowing; 1: >50% diameter narrowing

Table 4. Samples from pre-processed IHDD Dataset.

Sl.	Age	Sex	Chest Pain	Blood Pressure	Smoking Years	Fasting Blood Sugar	Diabetes History	Family Coronary History	Rest ECG	Pulse Rate	Target
0	63	1	1	145.0	20.0	1.0	1.0	1.0	2.0	150.0	0
1	67	1	4	160.0	40.0	0.0	1.0	1.0	2.0	108.0	0
2	67	1	4	120.0	35.0	0.0	1.0	1.0	2.0	129.0	0
3	37	1	3	130.0	0.0	0.0	1.0	1.0	0.0	187.0	0
4	41	0	2	130.0	0.0	0.0	1.0	1.0	2.0	172.0	0
5	56	1	2	120.0	20.0	0.0	1.0	1.0	0.0	178.0	0
6	62	0	4	140.0	0.0	0.0	1.0	1.0	2.0	160.0	1
7	57	0	4	120.0	0.0	0.0	1.0	1.0	0.0	163.0	0
8	63	1	4	130.0	0.0	0.0	1.0	0.0	2.0	147.0	0
9	53	1	4	140.0	25.0	1.0	1.0	1.0	2.0	155.0	0

3.2. Materials and Methods

This study used AWS to produce virtual machines (VMs) and Aneka for cloud resources, while iFogSim and FogBus simulated Fog resources. iFogSim explored fog and cloud scheduling and tested resource management and scheduling principles by simulating FC situations [26]. FogBus includes blockchain concepts to protect data, privacy, and communication and combines IoT, CC, and FC [27]. It allows developers to build apps, customers to execute apps, and service providers to manage resources. AWS delivers excellent cloud security and web services, that is, demand-based services [28]. Aneka is a cloud-based API creation platform [29] that is designed using a service-oriented architecture and can create programming models that abstract program execution logic.

Owing to the difficulty in diagnosing and predicting illnesses, artificial intelligence (AI) and its extensions, such as machine learning (ML) and deep learning (DL), are essential tools for empowering diabetic patients and doctors in their daily lives [30]. Logistic regression (LR) is a statistical method for predicting the link between a binary dependent variable and at least one of the independent variables, such as probabilities and sigmoid function. A decision tree (DT) creates a set of rules that can be used to classify data using the idea of decision making. Random forest (RF) is applied to reduce overfitting and improve prediction accuracy. Naive Bayes (NB) performs well under many actual circumstances, including document classification and spam filtering. K-Nearest Neighbors (KNN) is a technique for finding solutions to classification and regression problems; however, in the business sector, it is most typically used for classification challenges. In multidimensional space, a support vector machine (SVM) is used to classify data points using the proper hyperplane. AdaBoost, or adaptive boosting, is a statistical data classification algorithm. Extreme gradient boosting (XGBoost) is an ensemble machine-learning technique based on gradient boosting that uses decision trees. XGBoost performed better than AdaBoost because of system upgrades.

Deep learning (DL) is gaining popularity in data classification; its major types include deep neural networks (DNN), recurrent neural networks (RNN), convolutional neural networks (CNN), and artificial neural networks (ANN). DNN uses a layered neural network with many layers, where each layer can be turned on or off, and its output is the input for the following layer. Unlike other NNs, DNNs have fewer hidden layers and are used to train datasets [31]. Mathematically, a DNN is, as depicted in Equations (1) and (2), where each pre-activation function $P_a^l(x)$ is typically a linear procedure involving the matrix $W_s^l(x)$ and the bias B_s^l , which can be cohesive into a parameter P_r^l . In Equation (3), the \hat{x} denotes

the addition of 1 to the vector x . The form of the hidden-layer activation function $H^l(x)$ is often identical at each level; however, this is not always the case.

$$P_a^l(x) = W_s^l(x) + B_s^l \tag{1}$$

$$P_a^l(\hat{x}) = P_r^l(\hat{x}), \quad l = 1 \tag{2}$$

$$P_a^l(\hat{H}^{l-1}) = P_r^l(\hat{H}^{l-1}), \quad l > 1 \tag{3}$$

To determine neural activation and nonlinearize a neuron’s output, the activation function (AF) adds bias to a weighted sum. A rectified linear unit (ReLU) is a typical AF in DL that yields decent results; it was developed to address vanishing gradient problems. The most frequent AF in the DL group was the sigmoid. The sigmoid function transforms small and large values into values close to zero and one, respectively. The most widely used AF is ReLU, whereas binary classification uses the sigmoid, which is defined in Equations (4) and (5), respectively [32], where $Rel(x)$ is for the ReLU activation function and $Sigm(x)$ is the sigmoid activation function, whereas the $Max()$ function finds the maximum value.

$$Rel(x) = Max(0, x) \tag{4}$$

$$Sigm(x) = \frac{1}{1 + e^{-x}} \tag{5}$$

Ensemble learning (EL) improves the predictive analysis performance and accuracy [33]. The ML and DNN use ensemble approaches, such as bagging, voting, and averaging, to distribute data to worker nodes and obtain insights, as shown in Figure 2. The bootstrap aggregator, commonly known as the bagging meta-estimator, is an EL approach that may be used for classification and regression (BaggingClassifier and BaggingRegressor), whereas averaging averages, as in Equation (6), where $WA()$ is a function representing weighted averaging on p number of initial predictions x_p and weights assigned w_p , all predictions for a final predictive result and voting use an ensemble of various models to train and predict an output based on the class that has the highest likelihood of becoming the output.

$$WA_p = \frac{\sum_1^m x_p w_p}{w_p} \tag{6}$$

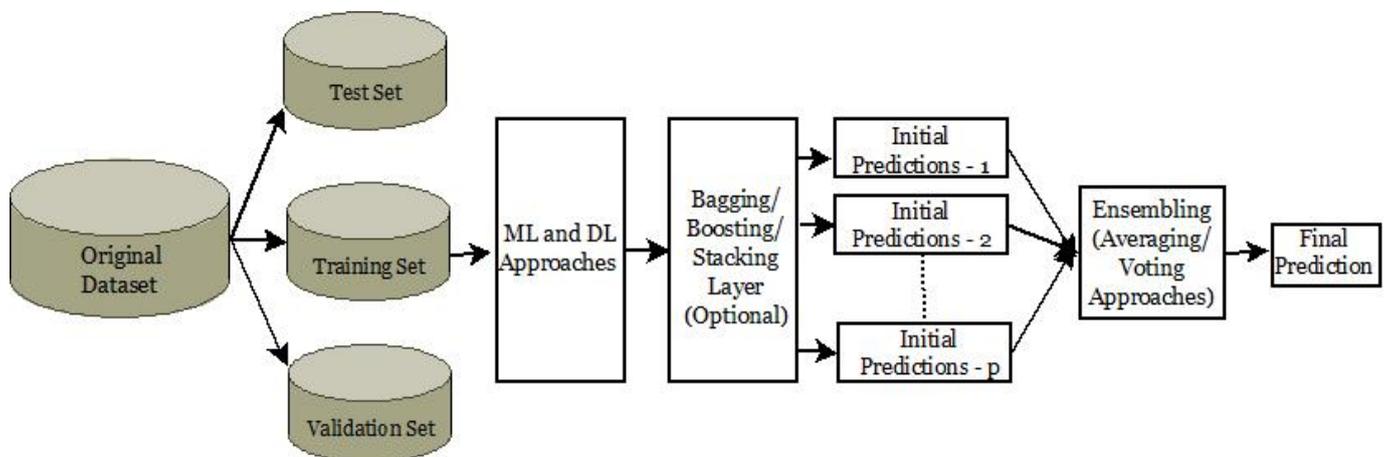


Figure 2. Ensembling of ML and/or DL Approaches.

4. Proposed Work: FRIEND

The architecture, design, and operation of the proposed method, called FRIEND, are described in this section.

4.1. Architecture of FRIEND

The architecture of FRIEND, shown in Figure 3, incorporates numerous hardware and software elements, as shown below. The hardware parts that are part of the suggested work include IoMT Health Sensors (IHSs), Gateway devices (GTDs), Master/Broker PC (MBP), Fog Worker Nodes (FWNs), and Cloud Data Center Nodes (CCNs), to name just a few. Through IHSs, data from patients with heart disease are sensed and transmitted to GTDs. Through GTDs, such as smartphones, tablets, and computers, patient data are accepted and transferred to either MBP or FWNs. These GTDs work in a manner akin to that of Fog gadgets. The MBP receives job requests from GTDs and either assigns FWNs to process them using an information manager (IM) or processes them using a learned DNN-based model (DBM) to produce results.

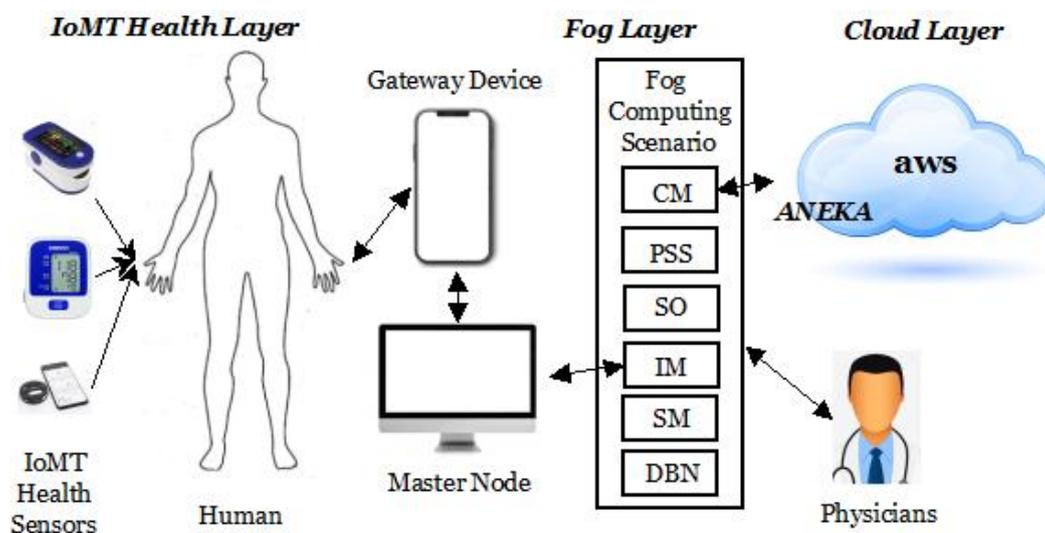


Figure 3. The architecture of FRIEND.

The MBP transfers the traffic to CCNs that employ a cloud manager (CM) when it detects that there are inadequate resources available, that is, when the MBP and FWNs are burdened and the MBP transforms into a GTD. In response to queries from GTDs or MBPs, FWN processes the data and produces results using the trained DBM, and then delivers the results back. In this study, Raspberry Pi devices were used as FWNs. If needed, the CCN is used to access cloud resources. The MBP functions as a GTD and forwards it to CCNs when it detects a lack of resources, that is when the MBP and FWNs are overloaded. Some of the software elements included in the suggested work are the information manager (IM), service manager (SM), privacy and security supervisor (PPS), cloud manager (CM), service observer (SO), and DNN-based model (DBM). The IM takes data from the IHSs that have been found and analyzed. Additionally, reliant on the circumstance, it can mix data from several sources and modify the occurrence of data transfers. The data's subsequent communications, or which FWNs will communicate with the next, are managed by the IM. The selection of sufficient resources for program implementation is under the purview of SM. Each MBP and FWN resource condition was determined by the SO of the computing server. To determine the needs of various applications, it leverages the directory of the warehouse service applications. After gathering the required data, the SM creates resources on the FWNs and the cloud for implementation. The MBP-PPS verifies user authentication credentials after obtaining them from a GTD, and the FWN-PPS manages seamlessly protected contacts of an FWN with others while performing computing operations. The CM warns the framework of cloud-based occurrences, such as containers and virtual devices, by sending storage and resource-providing requests to the cloud, that is, to the virtual devices. The SO allocates resources to various programs and monitors how well they can meet their implementation requirements in real time. It notifies the SM when resource

usage exceeds a provision-provider-defined threshold, or when something surprisingly tricky occurs. The dataset is used by the DL component to train the DBM to classify feature vectors, that is, vector points produced by data preprocessing from IHSs, which are feature vectors. Additionally, based on the tasks assigned to the SM, it anticipates and provides results for the data acquired via GTDs.

4.2. Design of FRIEND

FRIEND’s design discusses the Android application and experimental setup along with the implementation of this research.

4.2.1. Android Application and Experimental Setup

The FRIEND.apk, an Android interface created for this effort using MIT’s AppInventor, will be employed in a diversity of Android-compatible GTDs to gather sensed data from individuals [34]. It acts as an interface for the MBP, FWNs, and IHSs. The data input from the patients was sent to either the MBP, FWNs or CCNs, respectively, as depicted in Figure 4. A setup with various hardware configurations as evaluation hardware was implemented for the trials in this study. These configurations included an MBP (Dell with Windows 10 64-bit OS, Core i3, and 6 GB RAM) and a gateway device (Android Xiaomi A2 with Ver. 10), FWNs (four records of 4 GB SDRAM-based Raspberry Pi 4), and a public cloud (Aneka platform and AWS with Windows server). Additionally, the scalability of the proposed methodology was evaluated using 50 smartphones from various individuals in various configurations.

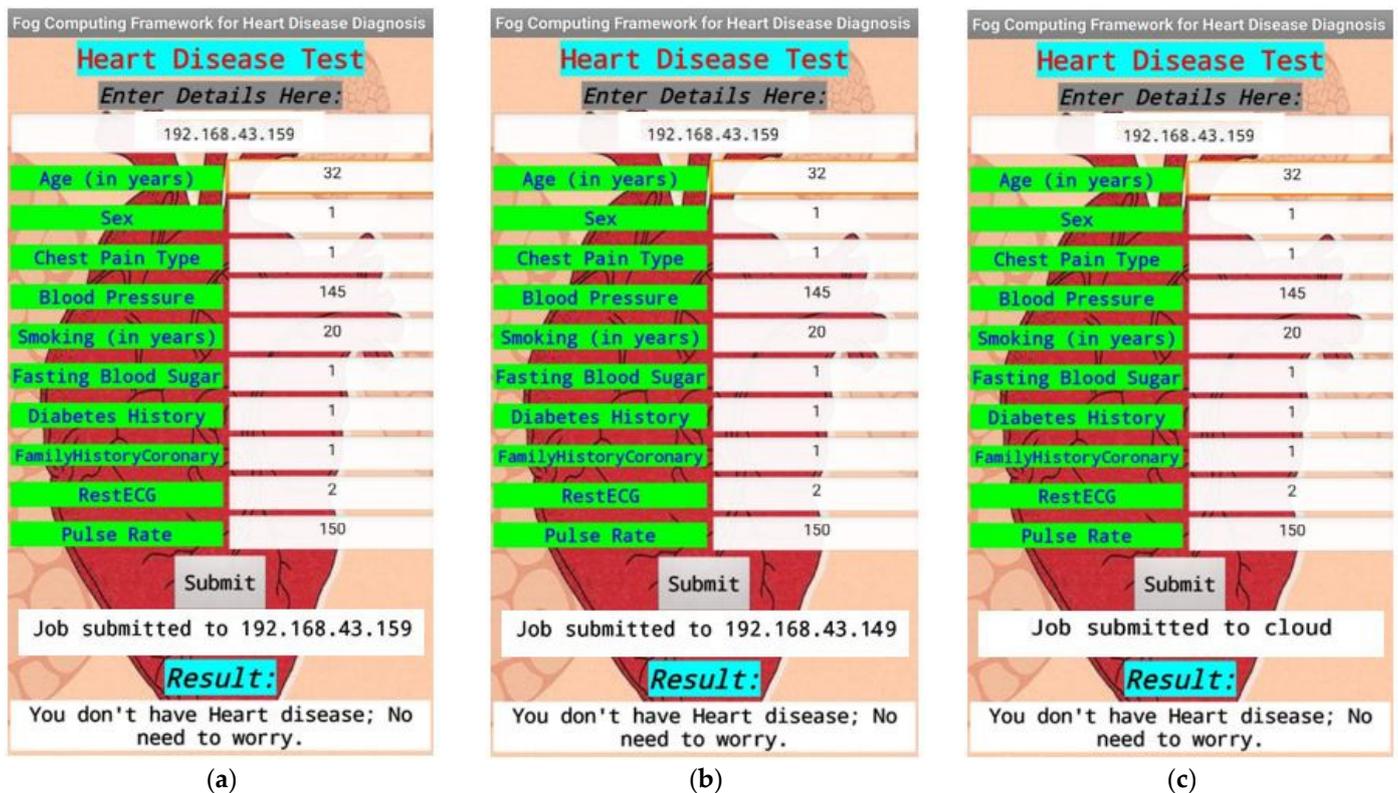


Figure 4. Android app communication with: (a) MBP; (b) FWN; (c) CCN.

4.2.2. Implementation

This section describes the execution of these components. In this study, we first considered eight basic ML approaches, namely, LR, DT, RF, NB, KNN, SVM, AdaBoost, and XGBoost, and initial predictions were made. Then, ensemble techniques, namely, weighted averaging and soft and hard voting, were applied to the initial predictions received. It

seems to be good to implement a DNN when the number of instances is higher along with the bagging classifier with these ensemble approaches, which were then subjected to various experiments to implement the proposed DBM model. Python was used for data pre-processing and training of the DBM model. The DNN and BaggingClassifier were used during model training for predictive binary classification and improved prediction accuracy. This model uses averaging and voting ensemble approaches to forecast important results by averaging all predictions from the assembled models. The ReLU is utilized in both the input and hidden layers, whereas the sigmoid is employed at the output layer. In this study, the input layer, hidden layer, and output layer are 10, 3, and 2, respectively, as 10 traits are inputs, and the result is binary as to whether the patient has heart disease. The learning rate in this study was 0.12. This study was modeled using the Adam optimizer. Table 5 summarizes the proposed DBM setup. The Android app in this work was made with MIT’s AppInventor, and web communications were performed using PHP. The data attributes were supplied to the MBP via HTTP Post in an Excel file. The MBP Data Manager communicates the data received. After any node processes patient data, a response is sent to the patient’s GTD through the MBP.

Table 5. Samples from pre-processed IHDD dataset.

Dataset	Model	Input Layers Numbers	Hidden Layers Numbers	Output Layers Numbers	Optimizer Used	Learning Rate	Input Layers AF	Hidden Layers AF	Output Layers AF
IHDD	DNN	10	3	2	Adam	0.12	ReLU	ReLU	Sigmoid

4.3. Working of FRIEND

The operating principle of the proposed FRIEND was described using algorithmic steps and a flowchart. In the proposed work, the MBP is the master and FWNs are the slaves. The MBP, FWNs, and GTDs were on the same network. The three types of communication include: (1) MBP, (2) MBP with FWNs, and (3) Cloud. The MBP attempts to perform the job request or requests any of the FWNs and returns the result. When the MBP and FWNs are overwhelmed, they forward to CCNs and become a GTD. Algorithm 1 explains how to build the suggested model, whereas Algorithm 2 explains how to train it utilizing various ML, DL, and EL ideas. The communication sequence of the proposed method is shown in Figure 5.

Algorithm 1: Detailed working of FRIEND

Notations:
PAD: Patients’ data
GTD: Gateway device
ITD: IoT device
ATR: Attribute
FWN: Fog computational node
CCN: Cloud Data center node
MBP: Master/broker node
RES: Result
Input:
 Patients’ data, $PAD_i = \{PAD_1, PAD_2, PAD_3, \dots\}$
 Gateway devices, $GTD_i = \{GTD_1, GTD_2, GTD_3, \dots\}$
 IoT devices, $ITD_i = \{ITD_1, ITD_2, ITD_3, \dots\}$
 Attributes, $ATR_i = \{ATR_1$ for age, ATR_2 for sex, ATR_3 for chest pain type, ATR_4 for blood pressure, ATR_5 for smoking years, ATR_6 for fasting blood sugar level, ATR_7 for diabetes history, ATR_8 for family history coronary, ATR_9 for ECG, ATR_{10} for pulse rate, ATR_{11} for target}
 Fog computational nodes, $FWN_i = \{FWN_1, FWN_2, FWN_3, \dots\}$
Output:
 Results, $RES_i = \{RES_0$ when value of target is 0, RES_1 when value of target is 1}
Steps of Algorithm:
 Obtain $ATR_3, ATR_4, ATR_6, ATR_9$ and ATR_{10} using ITD_i .
 Submit PAD_i obtained using ITD_i adding $ATR_1, ATR_2, ATR_5, ATR_7$, and ATR_8 to MBP using GTD_i .
if MBP = available (), **then**
 if target = 1, **then**
 return RES_1
 else
 return RES_0
 end if
end if

```

if  $FWN_i = \text{available}()$ , then
  if target == 1, then
    return  $RES_1$ 
  else
    return  $RES_0$ 
  end if
end if
if  $CCN = \text{available}()$  then
  if target == 1 then
    return  $RES_1$ 
  else
    return  $RES_0$ 
  end if
end if
Return  $RES_i$  to MBP.
Transfer  $RES_i$  to the corresponding  $GTD_i$ .
    
```

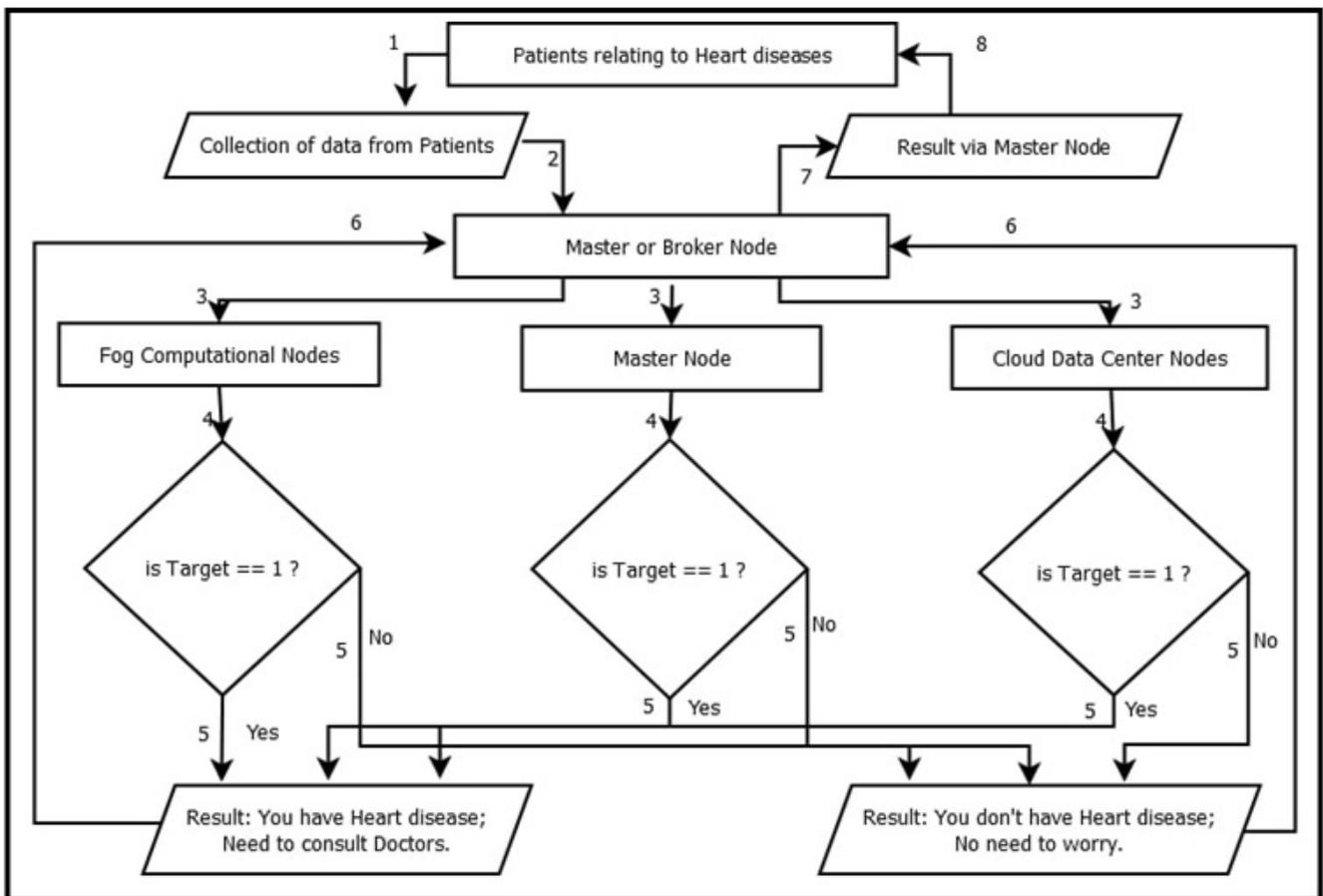


Figure 5. The workflow of FRIEND.

Algorithm 2: Pseudocode of EDL algorithm employed in FRIEND

Notations:

- S: Samples
- B: Bagging algorithm
- M: ML algorithm
- D: DL algorithm
- E: EL algorithm
- P: Total number of iterations
- R: Result

Input:

- Training samples, $S_i = \{S_1, S_2, S_3, \dots\}$
- Bagging algorithms, $B_i = \{B_1, B_2, B_3, \dots\}$
- ML algorithms, $M_i = \{M_1, M_2, M_3, \dots\}$
- DL algorithms, $D_i = \{D_1, D_2, D_3, \dots\}$
- EL algorithms, $E_i = \{E_1 \text{ for weighted averaging, } E_2 \text{ for soft voting, } E_3 \text{ for hard voting}\}$

Output:

Results, R

```

Steps of Algorithm:
Begin
for i = 1 to P do
  Obtain  $S_i$ 
  if Algorithm ==  $D_i$  then
    Training  $D_i$  on  $S_i$ 
    Bootstrap  $D_i$  using  $B_i$ 
  else
    Training  $M_i$  on  $S_i$ 
  end if
end for
Get  $R$ , Applying  $E_i$ 
Assign  $R$  to the nearest {0,1}
End.

```

5. Experimental Results and Discussion

First, a brief discussion of various network parameters and evaluative measures is provided. Next, the results obtained by experiments on these parameters are discussed, which leads to a discussion of the results obtained by the proposed method named FRIEND. A summary of comparisons with related studies is included in this section.

5.1. Analysis of Results Based on Performance Parameters

Empirical analysis of the results obtained is a major part of any proposed work. Performance measures aim to establish a real-to-anticipated class confusion matrix with a set of evaluation standards. For true and false positives, the confusion matrix is abbreviated as TR_{Pos} and FL_{Pos} , whereas for true and false negatives, it is abbreviated as TR_{Neg} and FL_{Neg} , respectively. In this study, classification can be performed using a variety of performance metrics, including precision or positive predictive value (PPV), accuracy (ACC), specificity or selectivity or true negative rate (TNR), recall or sensitivity or true positive rate (TPR), f1-score ($F1S$), misclassification rate (MCR), miss rate or false-negative rate (FNR), fall-out or false-positive rate (FPR), negative predictive value (NPV), false discovery rate (FDR), false omission rate (FOR), prevalence threshold (PT), balanced accuracy (BA), prevalence (PRV), threat score (TS), critical success index (CSI), and phi coefficient (ϕ_c) or Mathew's correlation coefficient (MCC), which can be formulated as in Equations (7)–(22).

$$ACC = \frac{TR_{Pos} + TR_{Neg}}{TR_{Pos} + TR_{Neg} + FL_{Pos} + FL_{Neg}} \quad (7)$$

$$PPV = \frac{TR_{Pos}}{TR_{Pos} + FL_{Pos}} = 1 - FDR \quad (8)$$

$$TPR = \frac{TR_{Pos}}{TR_{Pos} + FL_{Neg}} = 1 - FNR \quad (9)$$

$$TNR = \frac{TR_{Neg}}{TR_{Neg} + FL_{Pos}} = 1 - FPR \quad (10)$$

$$F1S = \frac{2 \times PPV \times TPR}{PPV + TPR} = \frac{2 \times TR_{Pos}}{2 \times TR_{Pos} + FL_{Pos} + FL_{Neg}} \quad (11)$$

$$MCR = \frac{FL_{Pos} + FL_{Neg}}{TR_{Pos} + TR_{Neg} + FL_{Pos} + FL_{Neg}} \quad (12)$$

$$FNR = \frac{FL_{Neg}}{TR_{Pos} + FL_{Neg}} = 1 - TPR \quad (13)$$

$$FPR = \frac{FL_{Pos}}{TR_{Neg} + FL_{Pos}} = 1 - TNR \quad (14)$$

$$NPV = \frac{TR_{Neg}}{TR_{Neg} + FL_{Neg}} = 1 - FOR \quad (15)$$

$$FDR = \frac{FL_{Pos}}{TR_{Pos} + FL_{Pos}} = 1 - PPV \tag{16}$$

$$FOR = \frac{FL_{Neg}}{TR_{Neg} + FL_{Neg}} = 1 - NPV \tag{17}$$

$$PT = \frac{\sqrt{FPR}}{\sqrt{TPR} + \sqrt{FPR}} \tag{18}$$

$$CSI = \frac{TR_{Pos}}{TR_{Pos} + FL_{Pos} + FL_{Neg}} \tag{19}$$

$$PRV = \frac{TR_{Pos} + FL_{Neg}}{TR_{Pos} + TR_{Neg} + FL_{Pos} + FL_{Neg}} \tag{20}$$

$$BA = \frac{TPR + TNR}{2} \tag{21}$$

$$MCC = \frac{(TR_{Pos} + TR_{Neg}) - (FL_{Pos} + FL_{Neg})}{\sqrt{(TR_{Pos} + FL_{Pos})(TR_{Pos} + FL_{Neg})(TR_{Neg} + FL_{Pos})(TR_{Neg} + FL_{Neg})}} \tag{22}$$

In this study, we first considered eight fundamental ML techniques: LR, DT, RF, NB, KNN, SVM, AdaBoost, and XGBoost. Initial predictions were created, and they are listed in Table 6 and displayed in Figure 6. However, the highest accuracy achieved among the ML techniques (i.e., with RF) was only 90.53%, which can be improved by combining these results using weighted averaging, soft voting, and hard voting, which are models known as M1, M2, and M3, respectively. When there are more instances in the datasets, it seems wise to use the DL approach, DNN, and bagging classifiers with weighted averaging and hard voting ensemble techniques, whose models, known as M4 and M5, respectively, are used in numerous experiments. According to reports based on performance measurements, model M5, that is, the DNN with bagging and hard voting classifiers, as shown in Table 7 and Figure 7, surpasses all other recommended models and is deemed to be the proposed DBM approach.

Table 6. Recorded results of various performance parameters based on various ML approaches.

ML Approaches	Recorded Performance Measures (in %)															
	ACC	PPV	TPR	TNR	F1S	MCR	FNR	FPR	NPV	FDR	FOR	PT	CSI	PRV	BA	MCC
LR	84.01	91.59	89.03	58.49	90.29	15.99	10.97	41.51	51.24	8.41	48.76	0.41	82.31	83.54	73.76	45.11
DT	89.91	94.88	93.35	68.18	94.11	10.09	6.65	31.82	61.86	5.12	38.14	0.37	88.87	86.34	80.76	59.08
RF	90.53	95.41	93.53	71.59	94.46	9.47	6.48	28.41	63.64	4.59	36.36	0.36	89.49	86.34	82.56	62.01
NB	88.04	94.56	91.31	68.48	92.91	11.96	8.69	31.52	56.76	5.44	43.24	0.37	86.75	85.71	79.89	55.39
KNN	87.42	93.90	91.37	62.49	92.62	12.58	8.63	37.51	53.51	6.09	46.59	0.39	86.25	86.34	76.93	50.48
SVM	87.73	94.11	91.58	62.79	92.82	12.27	8.42	37.21	53.47	5.89	46.53	0.39	86.61	86.65	77.18	50.86
AdaBoost	86.49	93.37	90.86	58.14	92.11	13.51	9.14	41.86	49.51	6.63	50.51	0.41	85.35	86.65	74.49	45.84
XGBoost	88.35	94.45	91.91	65.91	93.16	11.65	8.09	39.09	56.31	5.55	43.69	0.38	87.21	86.34	78.91	54.18

Table 7. Recorded results of performance parameters based on various proposed approaches.

Proposed Approaches	Recorded Performance Measures (in %)															
	ACC	PPV	TPR	TNR	F1S	MCR	FNR	FPR	NPV	FDR	FOR	PT	CSI	PRV	BA	MCC
M1	90.84	96.04	93.05	78.35	94.52	9.16	6.95	21.65	66.67	3.96	33.33	0.33	89.61	84.94	87.69	66.91
M2	91.46	96.45	93.48	79.35	94.94	8.54	6.52	20.65	66.97	3.55	33.03	0.32	90.37	85.71	86.41	67.96
M3	92.08	96.67	94.05	79.78	95.34	7.92	5.95	20.22	68.27	3.33	31.73	0.32	91.09	86.18	86.91	69.24
M4	92.39	97.14	94.28	76.12	95.59	7.61	5.72	23.88	60.71	2.86	39.29	0.33	91.74	89.61	85.21	63.82
M5	94.27	97.59	96.09	75.44	96.83	5.73	3.91	24.56	65.15	2.41	34.85	0.34	93.86	81.18	85.77	66.99

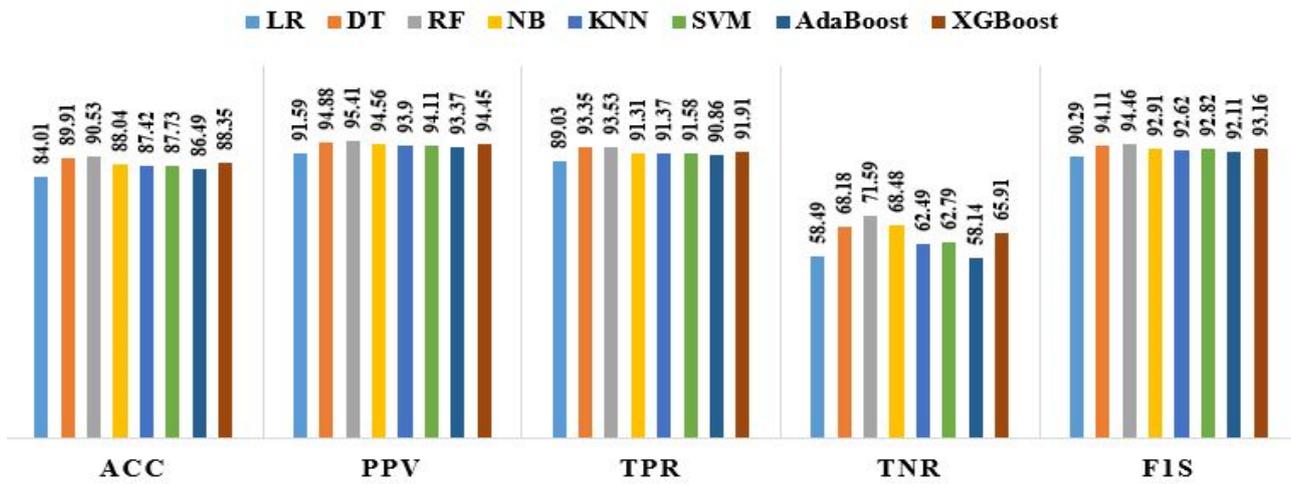


Figure 6. Results of several performance parameters based on different ML approaches were recorded.

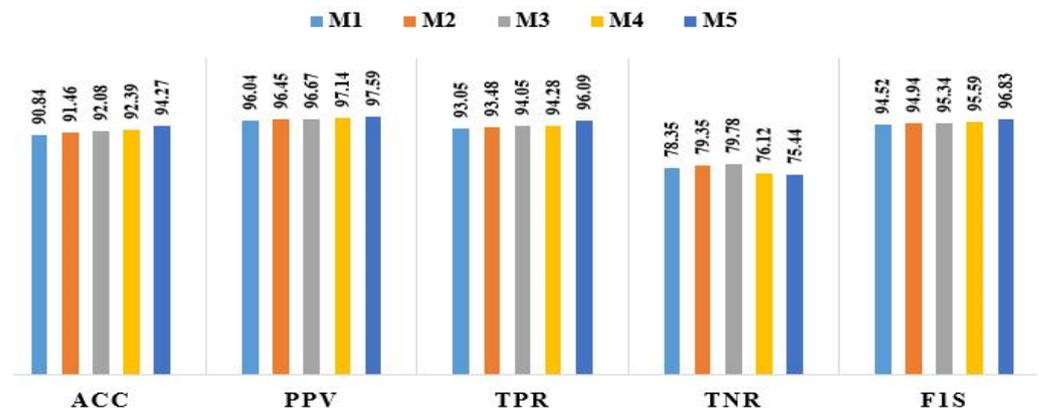


Figure 7. The results of several performance parameters were recorded using different ensemble techniques.

5.2. Analysis of Results Based on Network Parameters

The network parameters latency, arbitration time, total processing time, throughput, energy consumption, bandwidth, jitter, network utilization, and scalability were used to validate the proposed task. The various configurations considered for measuring network characteristics are Config-1 for the MBP only, Config-2 for the MBP with one FWN, Config-3 for the MBP with two FWNs, Config-4 for the MBP with three FWNs, Config-5 for the MBP with four FWNs, and Config-6 for the CCN only. Equation (23) can be used to express “latency (LT),” which is the length of time it consumes for data to move through a network and for a packet to be received, transmitted, and decoded at its destination. Arbitration time (AT) is the window of time during which the MBP must respond to GTDs, which may change according to the setup. The processing time (PT), which varies depending on the setup, is the period of time between starting a job, processing it, and returning user responses. The temporal time delay of a signal’s time change is known as the jitter (JT). Task response times may vary, and this is known as jitter. It is a crucial measurement that may be expressed in Equation (24) for many real-world applications, such as the analysis of data from e-healthcare. The quantity processed in a given time, such as the number of HTTP communications per day or bits transported per second, is known as throughput (TP). It can be calculated using Equation (25) to regulate the rate at which data packets are effectively transferred from any node to end nodes. The average data transfer rate over a communication path is known as network consumption or NC. Compared to cloud computing, Fog computing requires fewer networks. The system’s overall energy use was

measured as energy consumption (EC). Energy is used by the sensors, Fog nodes, and other components. Equation (26) and the physical theorem both measure and compute this in terms of Watt (W). Table 8 presents the observed findings for various network parameters corresponding to various setups.

$$LT_a = Time_{Res_a} - Time_{Req_a} \tag{23}$$

$$JT_a = \frac{\sum_1^m |Diff_a|}{m} \tag{24}$$

$$TP_a = \frac{m}{ET_{Aggr_a}} = Time_{Res_a} - Time_{Req_a} \tag{25}$$

$$EC_a = pow(X_a) \times Time_{Proc_a} \tag{26}$$

Here, $Time_{Res_a}$ is the response time, $Time_{Req_a}$ is the request time, $Diff_a$ is the difference in latencies, ET_{Aggr_a} is the aggregated execution time of m occurrences, $pow()$ acts as the function between power and characteristic parameters, and $Time_{Proc_a}$ is the processing time of a job, where X is the parameter set that affects the power.

Table 8. Recorded results of network parameters based on various proposed approaches.

Network Parameters	Recorded Network Measures					
	Config-1	Config-2	Config-3	Config-4	Config-5	Config-6
Latency (in ms)	29.8	38.9	37.4	36.5	44.6	2156.3
Arbitration time (in ms)	145.4	987.3	1388.9	1911.5	2334.7	134.1
Processing time (in ms)	2065.5	3124.3	2785.9	3454.9	3572.4	922.8
Jitter (in ms)	4.25	2.75	3.25	6.50	7.75	62.25
Throughput (in Mbps)	18.9	25.3	28.8	32.7	43.8	21.4
Bandwidth (in Kbps)	3.4	5.3	8.1	12.8	17.5	14.6
Network consumption (in secs)	8.2	11.8	13.7	18.5	19.1	28.6
Energy consumption (in Watt)	1.89	2.24	3.16	5.61	6.72	14.39

Figure 8 displays the observed latencies versus arbitration time over the configurations, where the latency is determined by adding the transmission and queue times. Every job submitted to the MBP or FWNs uses single-hop data transmissions; hence, the latency is the same. Owing to multi-hop data transmission outside the network, cloud latency is considerable; hence, FC is required. However, arbitration time is shorter when jobs are routed to the MBP or CCNs. In other instances, the load balancing affects the arbitration rate. Owing to enhanced processing, cloud processing is fast. Figure 9 shows the observed processing time versus jitter for the configurations. The amount of time it takes to begin, complete, and return a job to the users is known as the processing time. It is possible to observe that the overall processing time is significantly reduced when using cloud communications since cloud processing is extraordinarily quick owing to its higher capabilities. Because the nodes in FWNs have less processing power and a lower clock frequency, processing takes longer in cases when Fog computing infrastructures are used, whereas cloud communication has a shorter processing time. It shows a jitter with different settings, as shown in Figure 9. Jitter is stronger in the MBP-alone condition than when tasks are transferred to FWNs because the MBP handles arbitration, security checking, and resource management, whereas jitter is higher when jobs are supplied to the CCN.

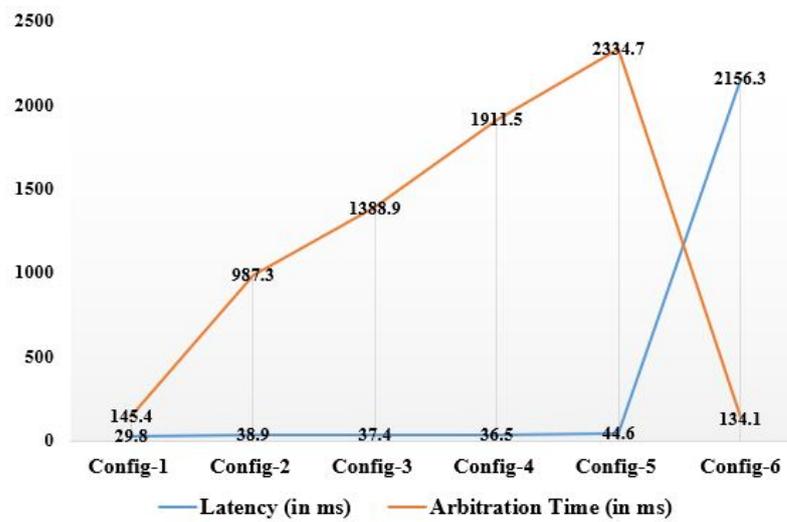


Figure 8. Latency vs. arbitration time over various configurations.

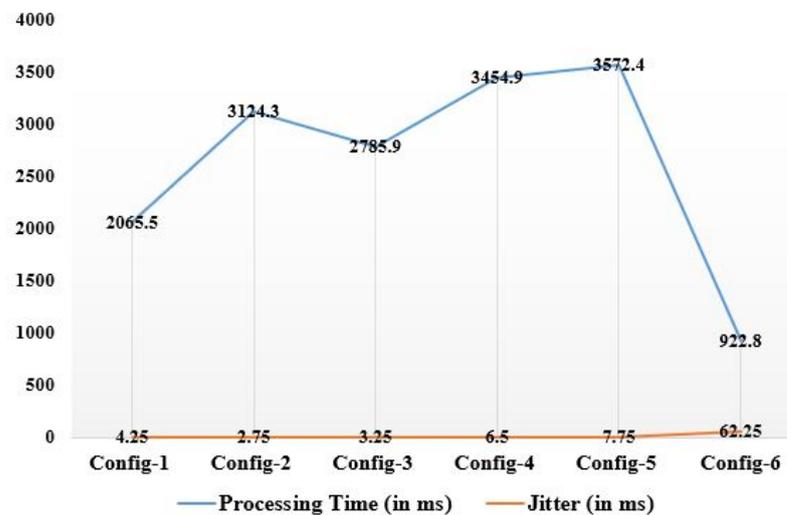


Figure 9. Processing time vs. jitter over various configurations.

Figure 10 shows the throughput versus bandwidth usage for various configurations. As shown in the figure, the throughput variance was Mbps for various settings. The MBP with FWNs has a better throughput than the CCN. As the number of FWNs and bandwidth utilization increases, security risks and data transmission (with clouds) must be checked. The figure depicts how bandwidth usage varies across all FCNs in various setups. Because of the maximum packets of heartbeats, checking of security issues and transmission of data (with cloud) are necessary as the number of FCNs grows, and bandwidth usage grows as well. Figure 11 shows the network usage versus energy consumption for various configurations. The figure shows that the MBP and/or FWNs have less network consumption time than CCNs because the fog environment reduces cloud user queries. In comparison with the MBP or FWNs, the CCN consumes a large amount of energy, as shown in Figure 11. As the number of FWNs increased, the energy consumption of the suggested work also increased.

Scalability is the IoT-Fog-Cloud system’s ability to increase software service delivery by augmenting software service resources when demand varies [35,36]. As shown in Figure 12, our key concern is whether the system can grow in quantity (scalability) over a long period when customers require it. The average response time increased in proportion

to the number of queries sent, although this increase was not exponential. The research is scalable because response times do not fluctuate with an increase in queries.

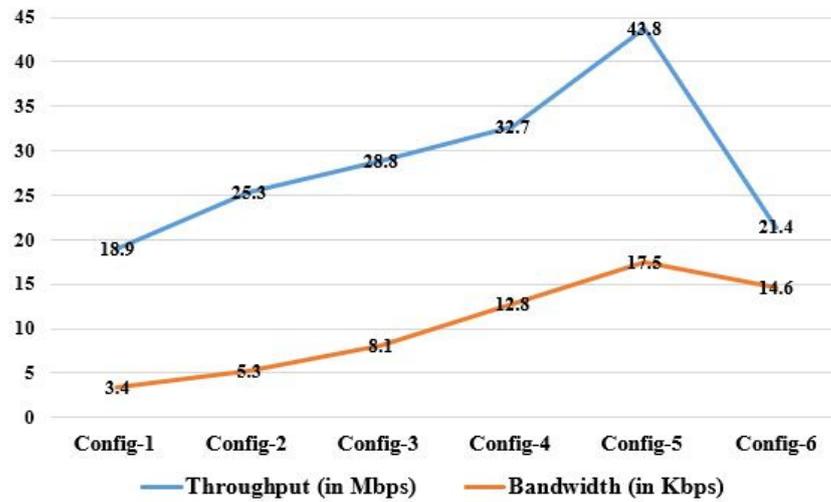


Figure 10. Throughput vs. bandwidth usage over various configurations.

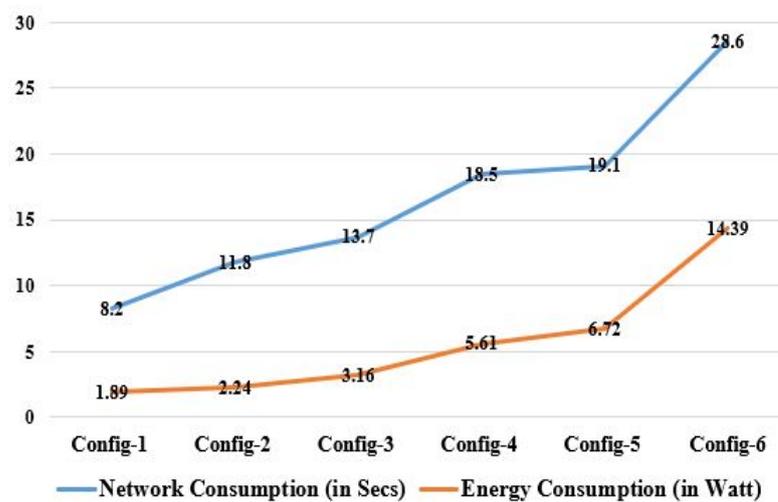


Figure 11. Network vs. energy consumption over various configurations.

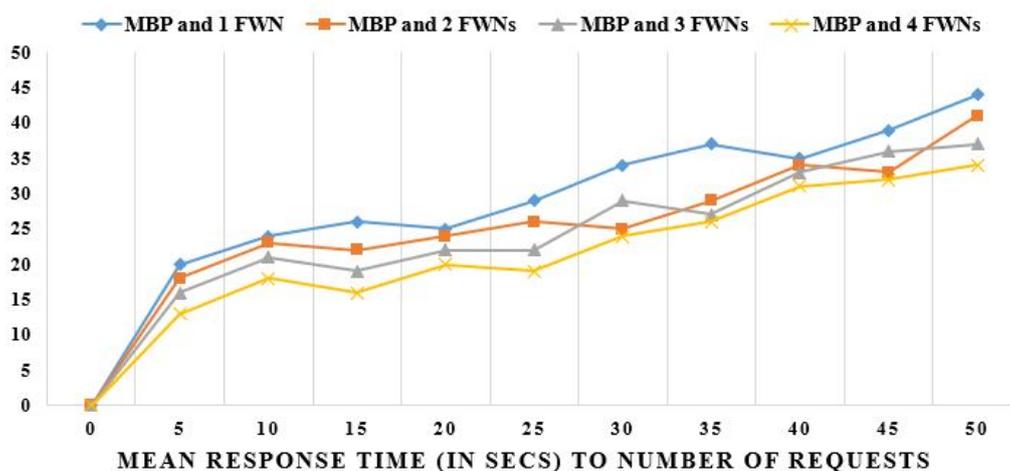


Figure 12. The mean response time (in secs) with respect to the number of requests by users.

6. Conclusions

E-healthcare systems are simple and consistent for individuals when the FC is used with IoT installations. Remote self-diagnosis is successful because heart disease-related mortality is high. Formal IoT implications only use CC methods, which have several drawbacks that can be resolved by combining FC, IoT, and CC in this suggested work. In this instance, we first tested the model using eight fundamental machine learning models: logistic regression, decision tree, random forest, naive Bayes, support vector machine, AdaBoost, and XGBoost. We then applied ensemble methods such as bagging classifiers, weighted averaging, and soft and hard voting to achieve enhanced results and, finally, used the DNN with the ensemble methods. To support this effort, these models were validated using 9 network parameters and 16 performance parameters. When the deep neural network was combined with bagging and hard voting classifiers, the accuracy, PPV, TPR, TNR, and F1 scores of the trials reached 94.27%, 97.59%, 96.09%, 75.44%, and 96.83%, which are significantly higher results. User-friendliness, FC-based design, instantaneous remote cardiac patient diagnosis, low latency, minimal energy consumption, etc., all of which are supported by trials, validate its name. FRIEND uses EDL techniques on the IHDD dataset to provide real-time, affordable, and enhanced cardiac diagnostics swiftly. Studies have revealed that it is a quick, user-friendly platform for cardiac remote diagnostics based on many considered performance and network criteria.

The results support the benefits of FC over CC In terms of latency, network, energy use, security, and privacy, as well as real-time remote cardiac diagnostics. For instantaneous distant cardiac patient diagnosis employing smartphones with sensors and apps that can diagnose heart patients whenever and wherever possible, integrating IoT, FC, and CC decreases latency and improves accuracy. The limitations of this study include its high cost, the use of a dataset with only 920 instances, which is not encouraging for DL trials, and the use of a single-platform-based approach. The aforementioned restrictions may be addressed by further studies using more DL approaches for various chronic conditions. This paradigm can be further enhanced by utilizing the concepts of edge, mist, and surge computing.

Author Contributions: Writing—original draft, A.P. and M.P.; writing—review and editing, B.K.P., M.A., A.K.H. and M.K.A.N.; supervision, B.K.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. DuBravac, S.; Ratti, C. The internet of things: Evolution or revolution? *AIG White Pap.* **2015**, *1*, 1–22.
2. Lakhan, A.; Mohammed, M.A.; Kozlov, S.; Rodrigues, J.J. Mobile-fog-cloud assisted deep reinforcement learning and blockchain-enabled IoMT system for healthcare workflows. *Trans. Emerg. Telecommun. Technol.* **2021**, e4363. [[CrossRef](#)]
3. Pati, A.; Parhi, M.; Pattanayak, B.K. IDMS: An Integrated Decision Making System for Heart Disease Prediction. In Proceedings of the 2021 1st Odisha International Conference on Electrical Power Engineering, Communication and Computing. Technology (ODICON), Bhubaneswar, India, 8–9 January 2021; pp. 1–6.
4. Gupta, R.; Mohan, I.; Narula, J. Trends in coronary heart disease epidemiology in India. *Ann. Glob. Health* **2016**, *82*, 307–315. [[PubMed](#)]
5. Shukla, S.; Thakur, S.; Hussain, S.; Breslin, J.G.; Jameel, S.M. Identification and Authentication in Healthcare Internet-of-Things Using Integrated Fog Computing Based Blockchain Model. *Internet Things* **2021**, *15*, 100422. [[CrossRef](#)]
6. Pati, A.; Parhi, M.; Pattanayak, B.K. IADP: An integrated approach for diabetes prediction using classification techniques. In Proceedings of the Advances in Distributed Computing and Machine Learning 2021, Singapore, 15–16 January 2021; pp. 287–298. [[CrossRef](#)]
7. Islam, S.R.; Kwak, D.; Kabir, M.H.; Hossain, M.; Kwak, K.S. The internet of things for health care: A comprehensive survey. *IEEE Access* **2015**, *3*, 678–708. [[CrossRef](#)]

8. Rahmani, A.M.; Gia, T.N.; Negash, B.; Anzanpour, A.; Azimi, I.; Jiang, M.; Liljeberg, P. Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: A fog computing approach. *Futur. Gener. Comput. Syst.* **2018**, *78*, 641–658. [CrossRef]
9. Mutlag, A.A.; Ghani, M.K.A.; Mohammed, M.A.; Lakhan, A.; Mohd, O.; Abdulkareem, K.H.; Garcia-Zapirain, B. Multi-Agent Systems in Fog–Cloud Computing for Critical Healthcare Task Management Model (CHTM) Used for ECG Monitoring. *Sensors* **2021**, *21*, 6923. [CrossRef]
10. Gill, S.S.; Arya, R.C.; Wander, G.S.; Buyya, R. August Fog-based smart healthcare as a big data cloud service for heart patients using IoT. In Proceedings of the International Conference on Intelligent Data Communication Technologies and Internet of Things, Coimbatore, India, 7–8 August 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 1376–1383.
11. Priyadarshini, R.; Barik, R.K.; Dubey, H. Deepfog: Fog computing-based deep neural architecture for prediction of stress types, diabetes and hypertension attacks. *Computation* **2018**, *6*, 62. [CrossRef]
12. Caliskan, A.; Yuksel, M.E. Classification of coronary artery disease data sets by using a deep neural network. *EuroBiotech J.* **2017**, *1*, 271–277. [CrossRef]
13. Choi, E.; Bahadori, M.T.; Song, L.; Stewart, W.F.; Sun, J. August GRAM: Graph-based attention model for healthcare representation learning. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017; pp. 787–795.
14. Ali, S.; Ghazal, M. April Real-time heart attack mobile detection service (RHAMDS): An IoT use case for software defined networks. In Proceedings of the 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), Windsor, ON, Canada, 30 April–3 May 2017; pp. 1–6.
15. Gupta, S. Classification of Heart Disease Hungarian Data Using Entropy, Knnga Based Classifier and Optimizer. *Int. J. Eng. Technol.* **2018**, *7*, 292–296. [CrossRef]
16. Mustafa, J.; Awan, A.A.; Khalid, M.S.; Nisar, S. Ensemble approach for developing a smart heart disease prediction system using classification algorithms. *Res. Rep. Clin. Cardiol.* **2018**, *9*, 33.
17. Zhenya, Q.; Zhang, Z. A hybrid cost sensitive ensemble for heart disease prediction. *BMC Med. Inform. Decis. Mak.* **2021**, *21*, 1–18. [CrossRef] [PubMed]
18. Ali, F.; El-Sappagh, S.; Islam, S.R.; Kwak, D.; Ali, A.; Imran, M.; Kwak, K.-S. A smart healthcare monitoring system for heart disease prediction based on ensemble deep learning and feature fusion. *Inf. Fusion* **2020**, *63*, 208–222. [CrossRef]
19. Moghadas, E.; Rezazadeh, J.; Farahbakhsh, R. An IoT patient monitoring based on fog computing and data mining: Cardiac arrhythmia use case. *Internet Things* **2020**, *11*, 100251.
20. Baccouche, A.; Garcia-Zapirain, B.; Olea, C.C.; Elmaghraby, A. Ensemble Deep Learning Models for Heart Disease Classification: A Case Study from Mexico. *Information* **2020**, *11*, 207. [CrossRef]
21. Sun, L.; Yu, Q.; Peng, D.; Subramani, S.; Wang, X. FogMed: A Fog-based Framework for Disease Prognosis Based Medical Sensor Data Streams. *Comput. Mater. Contin.* **2020**, *66*, 603–619. [CrossRef]
22. Tuli, S.; Basumatary, N.; Gill, S.S.; Kahani, M.; Arya, R.C.; Wander, G.S.; Buyya, R. HealthFog: An ensemble deep learning based Smart Healthcare System for Automatic Diagnosis of Heart Diseases in integrated IoT and fog computing environments. *Future Gener. Comput. Syst.* **2020**, *104*, 187–200.
23. Sharma, S.; Parmar, M. Heart diseases prediction using deep learning neural network model. *Int. J. Innov. Technol. Explor. Eng.* **2020**, *9*, 124–137. [CrossRef]
24. Uddin, M.N.; Halder, R.K. An ensemble method based multilayer dynamic system to predict cardiovascular disease using machine learning approach. *Inform. Med. Unlocked* **2021**, *24*, 100584. [CrossRef]
25. Heart Disease Data Set. UCI Machine Learning Repository. Available online: <https://archive.ics.uci.edu/ml/datasets/heart+disease> (accessed on 4 December 2020).
26. Gupta, H.; Vahid Dastjerdi, A.; Ghosh, S.K.; Buyya, R. iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments. *Softw. Pract. Exp.* **2017**, *47*, 1275–1296. [CrossRef]
27. Tuli, S.; Mahmud, R.; Tuli, S.; Buyya, R. Fogbus: A blockchain-based lightweight framework for edge and fog computing. *J. Syst. Softw.* **2019**, *154*, 22–36.
28. Narula, S.; Jain, A. February Cloud computing security: Amazon web service. In Proceedings of the 2015 Fifth International Conference on Advanced Computing & Communication Technologies, Rohtak, India, 21–22 February 2015; pp. 501–505.
29. Vecchiola, C.; Chu, X.; Buyya, R. Aneka: A software platform for NET-based cloud computing. *High Speed Large Scale Sci. Comput.* **2009**, *18*, 267–295.
30. Qalaja, E.K.; Abu Al-Haija, Q.; Tareef, A.; Al-Nabhan, M.M. Inclusive Study of Fake News Detection for COVID-19 with New Dataset using Supervised Learning Algorithms. *Int. J. Adv. Comput. Sci. Appl.* **2022**, *13*, 1–22. [CrossRef]
31. Hasan, T.T.; Jasim, M.H.; Hashim, I.A. Heart disease diagnosis system based on multi-layer perceptron neural network and support vector machine. *Int. J. Curr. Eng. Technol.* **2017**, *77*, 2277–4106.
32. Yan, H.; Jiang, Y.; Zheng, J.; Peng, C.; Li, Q. A multilayer perceptron-based medical decision support system for heart disease diagnosis. *Expert Syst. Appl.* **2006**, *30*, 272–281. [CrossRef]
33. Livieris, I.E.; Pintelas, E.; Stavroyiannis, S.; Pintelas, P. Ensemble deep learning models for forecasting cryptocurrency time-series. *Algorithms* **2020**, *13*, 121. [CrossRef]
34. Explore MIT App Inventor. MIT App Inventor. Available online: <https://appinventor.mit.edu/> (accessed on 24 July 2020).

35. Ahmad, A.A.-S.; Andras, P. Scalability analysis comparisons of cloud-based software services. *J. Cloud Comput.* **2019**, *8*, 10. [[CrossRef](#)]
36. Alnabhan, M.; Habboush, A.K.; Abu Al-Haija, Q.; Mohanty, A.K.; Pattnaik, S.; Pattanayak, B.K. Hyper-Tuned CNN Using EVO Technique for Efficient Biomedical Image Classification. *Mob. Inf. Syst.* **2022**, *2022*, 2123662. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.