Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot

HealthAIoT: AIoT-driven smart healthcare system for sustainable cloud computing environments

Han Wang [a], Kumar Ankur Anurag [a], Amira Rayane Benamer [b], Priyansh Arora [c], Gurleen Wander [d,e], Mark R. Johnson [e], Ranjit Mohan Anjana [f], Viswanathan Mohan [f], Sukhpal Singh Gill [a,*], Steve Uhlig [a], Rajkumar Buyya [g]

[a] School of Electronic Engineering and Computer Science, Queen Mary University of London, London, E1 4NS, UK
[b] LabInnov-Audensiel, Boulogne-Billancourt, France
[c] Microsoft, Schiphol, Netherlands
[d] Chelsea and Westminster Hospital, NHS Foundation Trust London, 369 Fulham Road, SW10 9NH, London, UK
[e] Department of Metabolism, Digestion and Reproduction - Faculty of Medicine, Imperial College London, South Kensington, SW7 2AZ, London, UK
[f] Madras Diabetes Research Foundation and Dr Mohan's Diabetes Specialities Centre, Gopalapuram, 600 086, Chennai, India
[g] Quantum Cloud Computing and Distributed Systems (qCLOUDS) Laboratory, School of Computing and Information Systems, University of Melbourne, Melbourne, VIC 3010, Australia

ARTICLE INFO

Keywords:
Cloud computing
Internet of Things
Artificial Intelligence
Artificial Intelligence of Things
Sustainable cloud computing
Energy consumption
Diabetes prediction

ABSTRACT

The evolution of Artificial Intelligence of Things (AIoT), coupled with advancements in edge and cloud computing, has enabled the development of real-time IoT-based applications. Integrating Internet of Things (IoT) and AI-driven edge–cloud services can address challenges such as early disease detection, system performance, data management and environmental sustainability in cloud-centric healthcare environments. To address these challenges, we propose **HealthAIoT**, a new architecture that utilises AIoT with cloud computing services to create a smart healthcare system. In our current implementation, HealthAIoT assesses the risk of developing Diabetes Mellitus in healthy individuals based on their personal health metrics and medical history; however, the proposed framework is fundamentally designed to be disease-agnostic and can be extended to incorporate detection and monitoring for other diseases. The HealthAIoT architecture consists of two main modules: a diabetes predictor and a cloud scheduler. The diabetes prediction module and cloud scheduler both utilise Multilayer Perceptron (MLP) models. The cloud scheduler manages health-related data and application requests from IoT devices, optimising resource utilisation and minimising the environmental impact of cloud services. The performance of the HealthAIoT framework is tested using realistic testbed CloudAIBus. Experimental results demonstrate that the MLP-based diabetes predictor achieves 78.30% accuracy and an F1-score of 0.7719 on unseen patient data while cloud scheduler achieves 93.6% accuracy. Further, system performance is evaluated using metrics including energy consumption, carbon-free energy usage, cost, execution time, and latency. By identifying individuals at the highest risk of developing diabetes, the framework enables targeted preventative interventions, optimise resource usage and maximises impact, while also serving as a foundational framework for broader healthcare applications.

* Corresponding author.
E-mail addresses: hanwang@qmul.ac.uk (H. Wang), k.anurag@se23.qmul.ac.uk (K.A. Anurag), a.benamer@audensiel.fr (A.R. Benamer), priyansh.arora@microsoft.com (P. Arora), g.wander@nhs.net (G. Wander), mark.johnson@imperial.ac.uk (M.R. Johnson), dranjana@drmohans.com (R.M. Anjana), drmohans@diabetes.ind.in (V. Mohan), s.s.gill@qmul.ac.uk (S.S. Gill), steve.uhlig@qmul.ac.uk (S. Uhlig), rbuyya@unimelb.edu.au (R. Buyya).

https://doi.org/10.1016/j.iot.2025.101555
Received 1 December 2024; Received in revised form 11 February 2025; Accepted 23 February 2025

1. Introduction

Internet of Things (IoT) introduces a wide range of sensors and devices that are capable of executing sophisticated computations and maintaining reliable communications, these technologies have been deployed across numerous sectors [1,2]. In healthcare, IoT-enabled Body Sensor Network (BSN) has moved beyond conventional applications to fundamentally transform patient care by enabling precise and proactive chronic disease monitoring and detection [3]. Advances in machine learning (ML) have substantially enhanced our ability to address complex health-related challenges [4]. The emerging paradigm of Artificial Intelligence of Things (AIoT) that integrates AI with IoT further improves the intelligence of sensing systems [5]. When these BSN/IoT sensors are integrated with AI and managed through public cloud services, they provide a robust framework for the development of environmentally sustainable and health-focused smart cities [6]. This integration facilitates the rapid and accurate processing of large volumes of real-time patient data, thereby supporting prompt diagnostics, timely updates to treatment strategies, improved patient care and overall streamlined healthcare innovation.

Diabetes Mellitus, a chronic disease responsible for 1.5 million deaths in 2019, with nearly 48% of these fatalities occurring in individuals under 70 years of age, is closely associated with severe complications affecting the cardiovascular system and kidneys [7,8]. Early risk assessment in non-diabetic individuals is therefore crucial to implementing timely preventative interventions. Delays in diagnosis have resulted in approximately 50% of individuals with Type 2 diabetes (T2D) presenting with complications at the time of diagnosis, which not only increase disease severity but also increases the likelihood of additional health issues [9,10]. Leveraging advanced computational methods can not only enable early intervention for T2D prevention but also facilitate continuous monitoring for the early identification of individuals at high risk.

1.1. Motivation

Modern approaches, such as the use of BSN sensor-based IoT devices [11], have addressed some challenges and brought significant convenience to daily life. However, further integration of AIoT technology with cloud computing in healthcare is needed to develop robust and comprehensive architectures for chronic disease risk assessment [12]. Traditional blood sugar monitoring devices, such as glucose metres are available for home use but can be costly. This can impose a financial burden on patients, particularly those managing other medical conditions with multiple monitoring devices. Relying on periodic blood sugar tests at clinical laboratories is both expensive and inconvenient, especially for individuals without a history of diabetes who want to minimise the risk of complications through early diagnosis. The advent of AIoT-enabled wearable devices, such as smartwatches and smart scales, has made it possible to continuously monitor general health-related parameters and track lifestyle changes. These smart devices generate comprehensive daily activity data that can be analysed using AI algorithms to provide accurate insights into individuals' risk of developing diabetes without additional cost. This can help users improve their lifestyle to prevent the disease or seek timely medical consultation if needed [13].

The traditional questionnaires for diabetes risk assessment often heavily rely on scoring systems that may not capture the complexity of disease risk factors. Furthermore, these methods often fail to account for individual differences due to variations in lifestyle and dietary habits resulting from cultural differences and environmental factors [14]. The advancement of AI models has enabled new insights into data that contain biases and discrepancies due to regional and cultural differences, which can identify these non-linear and complex relationships and interactions that traditional methods might overlook. AI techniques therefore can be utilised by both individuals and clinical practitioners to discover and capture unique lifestyle factors that affect diabetes risk, which traditional risk assessment methods may not detect, aiming to improve accuracy for various demographic groups [14]. The UK's National Health Service (NHS) has identified AI as a critical transformative technology, aiming to incorporate AI to promote healthcare innovation, support decision making and analyse data generated by digital health devices [15]. NHS is exploring various AI applications, such as the early identification of patients at high risk of hospital admission, continuous monitoring through IoT devices, and decision support during consultations [16]. For instance, AI can process patient data in real-time, offering possible diagnoses and suggesting follow-up investigations, thus improving diagnostic accuracy and efficiency. The effectiveness of AI algorithms in these prediction and decision-making tasks is particularly valuable in the current era due to increasing availability of large datasets [17]. AI algorithms are often more capable than traditional statistical analysis techniques in handling large datasets and capturing irregular patterns and complex relationships within the data [18]. Additionally, AI models can continuously improve as more data become available and more advanced algorithms are developed [19,20].

In addition to these deficiencies of traditional risk assessment methods, issues in the area of IoT and AI-based cloud integration in the healthcare sector remain, particularly in terms of performance, sustainability and Quality of Service (QoS) optimisation [21]. Global healthcare activities contribute an estimated 4%–5% of worldwide greenhouse gas emissions, with a substantial portion attributed to IT-related components [22,23]. This highlights a clear need to improve sustainability in the healthcare sector. Some major healthcare organisations have been increasingly emphasising targets for environmental sustainability and carbon neutrality. For instance, NHS has launched its 'Net Zero' targets in the UK, aiming to become the world's first carbon-neutral national health system by 2040 [24]. Similarly, the World Health Organisation (WHO) has been highlighting the need for the healthcare sector to combat climate change [25]. Additionally, to address the challenges of handling large data volumes on edge devices [26], there is an essential requirement for an AIoT–cloud based integrated solution. Such a solution should effortlessly handle user data, employ AI techniques for processing to overcome latency, optimise resource utilisation, and achieve energy-efficient operations while generating accurate prediction results for potential diabetes risk assessment [27]. Therefore, exploring an AI-driven edge–cloud solution for sustainable healthcare is essential to deal with the above challenges.

1.2. Our contributions

The primary purpose of HealthAIoT is to present an AIoT-driven cloud solution for efficient diabetes risk assessment, focusing on cloud QoS parameter optimisation while improving cloud sustainability. The proposed framework HealthAIoT addresses the challenges mentioned above. Its main contributions are as follows:

- **Design and Implementation of HealthAIoT architecture**: We developed a comprehensive architecture that integrates IoT devices with an AI-enabled cloud infrastructure for diabetes risk prediction. The framework includes a user-friendly frontend for diabetes assessment, its modular design enables that the same architecture can also be readily extended to monitor other chronic diseases, such as cardiovascular or respiratory conditions.
- **AI Cloud Resource Scheduler**: An innovative AI-based cloud resource scheduler was developed using a Multilayer Perceptron (MLP) model. The scheduler achieves 93.6% accuracy in predicting optimal computing resource allocation, ensuring efficient workload management. This approach enhances energy efficiency, reduces carbon footprint, and maintains application reliability and user data security, making it suitable for scalable and sustainable cloud environments.
- **Diabetes Risk Prediction Model**: A novel diabetes risk prediction model was proposed leveraging an MLP framework. The model achieves a peak accuracy of 78.30% and F1-score of 0.7719 on unseen data. These results indicate its potential for real-world deployment in healthcare scenarios, providing a reliable tool for early diabetes risk assessment.
- **Explainable AI (XAI) for Explaining Diabetes Prediction**: We incorporate XAI technique, specifically SHAP (SHapley Additive exPlanations), was incorporated. This approach quantifies and visualises the contribution of each dataset feature to the final decision-making process of the disease prediction model. This is particularly critical in healthcare applications where understanding the rationale behind predictions and improving transparency are essential for clinical adoption.
- **Comprehensive Performance Evaluation**: A thorough performance evaluation of the proposed architecture was conducted using the real testbed CloudAIBus [28]. Key QoS parameters including energy consumption, carbon-free energy usage, cost, execution time and latency were analysed. AI performance metrics such as accuracy and F1-score were evaluated to assess the effectiveness of the proposed solution. The results demonstrate the architecture's ability to balance performance, efficiency, and sustainability in real-world cloud environments.

1.3. Organisation

The rest of the paper is structured as follows: Section 2 presents details about recent related work in the healthcare sector that employs IoT devices and cloud services. Section 3 discusses the methodology used to develop the proposed architecture, including system design, communication topology, dataset processing and model training for the HealthAIoT framework. Section 4 explains the experimental environment setup and the performance evaluation metrics including AI performance metrics and QoS parameters. Section 5 discusses experimental outcomes and result analysis as well as the research limitations. Section 6 concludes the paper along with suggestions for future work.

2. Related work

The integration of cloud computing and AIoT in healthcare represents a transformative shift in the delivery of healthcare services. Recent surveys highlight that IoT devices, such as wearable sensors and medical equipment continuously collect health-related data, which is then transmitted via various communication technologies such as Wi-Fi and Bluetooth to cloud platforms [29,30]. These cloud systems provide scalable computing resources and advanced data processing capabilities, enabling remote patient monitoring and personalised treatment plans [31]. However, while many studies have leveraged these capabilities, much of the discussion around cloud performance remains largely theoretical. The existing research often does not incorporate cloud-side optimisation, such as resource allocation, energy consumption and operational cost, nor does it adequately address the need for XAI to improve user acceptance in healthcare settings.

2.1. Existing studies

Over the past decades, IoT applications have evolved significantly from basic data collection tools to systems capable of intelligent analytics and real-time processing [32–35]. Recent research has further emphasised the integration of cloud services with edge devices to support continuous healthcare monitoring and predictive analysis using advanced ML models and secure communication protocols [36].

For instance, Hennebelle et al. [37] present HealthEdge framework that utilises ML algorithms (Random Forest and Logistic Regression (LR)) for T2D prediction in an integrated edge–cloud environment. In the framework, health metrics from IoT devices are pre-processed at edge servers before being sent to the cloud for further analysis. Their framework are tuned and evaluated on public datasets including PIMA Indian and Sylhet. Although it is promising in managing real-world risk-factor heterogeneity, HealthEdge does not address cloud optimisation aspects, such as resource allocation, energy consumption and operational cost, which are critical for deploying scalable IoT–cloud systems. Similarly, Khan et al. [38] present an advanced federated ensemble learning approach for healthcare monitoring. Their system integrates various IoT devices to capture medical images, specifically Chest X-rays (CXR) and Magnetic Resonance Imaging (MRI), which constructs local models at the edge. These local models are then combined using an ensemble strategy that distinguishes between malicious and non-malicious data to mitigate poisoning attacks. Their framework achieves notable diagnostic performance on public CXR and MRI datasets, with accuracy rates of 99.24% and 99.0% respectively, outperforming traditional centralised models. Nonetheless, their study primarily focuses on disease prediction and does not consider cloud performance metrics or cloud-side optimisation strategies. Additional contributions such as, Desai et al. [12] proposed HealthCloud, an architecture designed for monitoring heart patient data for early detection and prediction of cardiac complications using various ML models including Artificial Neural Network (ANN), K-Nearest Neighbour (KNN), LR, Support Vector Machine (SVM) and Gradient Boosting (GB), which evaluate QoS parameters such as execution time and latency. Their study primarily considers accuracy and F1-score without a comprehensive investigation into cloud sustainability. Ramkumar et al. [39] present an IoT-based heart disease prediction model employing Long Short-Term Memory (LSTM) and Recurrent Neural Network (RNN) architectures alongside a combined Lion and Krill Head optimisation technique. Despite achieving high accuracy and F1-scores, their framework also lacks a detailed analysis of cloud optimisation.

In another study, Alsubai et al. [40] propose a digital twin initiative to enable real-time monitoring and proactive management of athlete health during intensive training sessions. Their framework integrates IoT devices with edge and cloud computing to continuously capture both physiological and environmental data, which is then transformed into a dynamic digital twin representation. They applied Convolutional Neural Network (CNN) and Multi-scaled Long Term Memory (MLSTM) to predict athlete's adverse health conditions early. Their experimental simulations show their framework outperforms existing techniques with high temporal efficacy and predictive accuracy, underscoring the potential of combining digital twin technology with IoT–cloud integration for enhancing real-time healthcare monitoring [40]. Sharma et al. [41] implemented an Extreme Learning Machine (ELM) model for monitoring and early diagnosis of diabetes. The system's performance was evaluated based on resource utilisation, latency and scalability in both standalone deployment and Amazon Web Services (AWS) deployment scenarios. The proposed architecture addresses system failure and security through scalability and cloud deployment. However, this framework does not consider sustainability aspects of both deployment strategies. Atoum et al. [42] proposed a K-Nearest Neighbour (KNN) model to develop a real-time, continuous health monitoring system for anomaly detection in patient health. It utilises edge devices (Raspberry Pi processor) as the main computational unit, which continuously receives patient data from IoT sensors and performs data pre-processing, then transfers the data to the AWS cloud platform. The model processes the data to detect anomalies, and the results are transmitted back to the Raspberry Pi module for display via web interface. The system operates on a private Wi-Fi setup, ensuring secure inter-device and intra-device communication. Patients can access their data stored on the cloud platform through a secure login authentication mechanism. The paper analyses the model's performance using accuracy and latency metrics.

The authors [43] propose a FedSDM (Federated Learning-based Smart Decision Making) module for anomaly detection in Electrocardiogram (ECG) data within an integrated edge–cloud environment. This framework employs an autoencoder that comprises an encoder and a decoder implemented as MLP, which is used for reconstructing ECG data to detect anomaly based on reconstruction error, which also leverages FedAvg for aggregating local models trained on ECG data from distributed IoT devices to detect anomaly. Importantly, the paper evaluates the deployment of the proposed framework across edge, fog and cloud layers by measuring QoS parameters such as energy consumption, network usage, cost, execution time and latency. The findings demonstrate their proposed framework can reduce latency and resource usage within edge environment, which offers a strong baseline for comparative analysis.

Tuli et al. [44] introduced HealthFog, a novel framework that uses ensemble ML algorithms deployed on edge devices to predict heart disease in patients. The system uses the FogBus architecture based on Blockchain technology, to handle communication and data flow between IoT sensors and edge devices. The framework also manages data privacy and security to ensure application safety. In their framework, Aneka software is utilised to integrate the IoT–edge system with the cloud platform. The system's performance is evaluated based on QoS parameters including latency, execution time, network bandwidth usage, jitter and power consumption, while the model performance is evaluated based on prediction accuracy.

2.2. Critical analysis

We reviewed the existing studies in the related area and found extensive research on integrating IoT devices with cloud or edge computing paradigms for various health-monitoring aspects. Although significant advancements have been made in leveraging IoT sensors and edge–cloud services for health monitoring and disease prediction, our review reveals persistent research gaps that motivate the need for the proposed HealthAIoT framework (see Table 1). Many studies emphasise that the backbone of their frameworks relies on cloud services for deploying ML models and real-time analytics; however, these discussions of cloud performance are largely theoretical with limited investigation into critical aspects such as QoS parameter optimisation and sustainability. For example, Desai et al. [12] and Ramkumar et al. [39] apply various ML models, including ANN, KNN, LSTM and RNN for heart disease prediction, while their frameworks are predominantly centred on cloud services without exploration of cloud-side optimisation. Similarly, Sharma et al. [41] present an ELM-based model for diabetes diagnosis, but their evaluation mainly focuses on accuracy and F1-score, neglecting broader system-level metrics. Furthermore, studies by Atoum et al. [42] and Tuli et al. [44] apply ML techniques for health anomaly detection and heart disease prediction; however, they do not adequately address the sustainability and efficiency with QoS parameters optimisation in IoT–cloud service integration, nor do they conduct deployment in the real testbed. Additionally, none of these works incorporate XAI methods to help to understand their models' decision-making processes, which is significant in interpretability of healthcare applications [45,46].

In summary, although these studies collectively demonstrate the potential of integrating IoT, edge and cloud computing for health monitoring and disease prediction, they frequently overlook critical aspects of cloud optimisation, framework integration and model explainability. This gap justifies the development of our HealthAIoT framework, which not only aims to achieve accessible disease prediction but also promote system sustainability, QoS optimisation and enhanced transparency through AI-driven scheduling and resource management.

Table 1
Comparison of proposed HealthAIoT with existing research.

Works	Computing Paradigm	Security	Algorithms Employed	AI Performance Metrics		Scheduler	Computing Parameters (QoS)	Real Testbed
				Accuracy	F1-score			
[37]	Cloud–Edge	✗	RF, LR	✓	✓	✗	–	✗
[38]	Cloud–Edge	✗	Federated Learning	✓	✓	✗	–	✗
[12]	Cloud	✓	KNN, LR, SVM, GB	✓	✓	✗	Latency, ET	✗
[39]	Cloud	✓	LSTM, RNN	✓	✓	✗	–	✗
[40]	Cloud–Edge	✓	MLSTM, CNN	✓	✓	✗	–	✗
[41]	Cloud	✓	ELM	✓	✓	✗	Latency	✗
[42]	Cloud–Edge	✓	KNN	✓	✗	✗	Latency	✗
[43]	Cloud–Edge	✗	Federated Learning	✓	✗	✗	EC, ET, Cost, Latency, Network	✗
[44]	Cloud–Edge	✓	EDL	✓	✗	FogBus	Latency, Execution Time, Energy Consumption	✗
HealthAIoT (Ours)	Cloud–AIoT	✓	MLP	✓	✓	MLP Scheduler	EC, ET, Carbon-free Energy Usage, Cost, Latency	CloudAIBus

Abbreviation used — KNN: K-Nearest Neighbour; LR: Logistic Regression; RF: Random Forest; SVM: Support Vector Machine; CNN: Convolution Neural Network; RNN: Recurrent Neural Network; GB: Gradient Boosting; LSTM: Long Short Term Memory; MLSTM: Multi-scaled Long Term Memory; ELM: Extreme Learning Machine; ML:Machine Learning; MLP: Multi-Layer Perceptron; EDL: Ensemble Deep Learning; ET: Execution Time; EC: Energy Consumption

3. Proposed methodology

This section introduces the architecture of our proposed framework and discusses its main components.

3.1. HealthAIoT framework

The HealthAIoT framework is built on an AIoT-Cloud platform that seamlessly integrates multiple hardware with software components. The framework utilises a RESTful API to facilitate inter-device communication. The proposed MLP scheduler serves as the core component of the architecture, managing resources utilisation, data acquisition, allocation and operations on user health data. The scheduler also manages the connection between the Gateway device and the system. To ensure security, the framework incorporates security measures that restrict access and communication to authorised Workers and users by operating within a private WLAN (Wireless Local Area Network). This approach protects sensitive data by mitigating risks of unauthorised internet access. Fig. 1 provides a detailed overview of the hardware and software components described below:

3.1.1. Hardware units

In this study, we have taken into account the following hardware components:

Body-sensor Network (BSN) IoT Devices: A smartwatch is used to collect health data such as heart rate and daily activity levels, which are essential for assessing general health conditions. Although ECG result and cholesterol levels are included in our research, the methods for obtaining these measurements are beyond the scope of the HealthAIoT architecture.

Gateway Device: A laptop, mobile phone or tablet can be used as a gateway device. The only requirements are the installation of either the Google Chrome or Firefox browser and access to the Internet.

3.1.2. Software units

In this work, we have considered the following software units:

Webpage Interface: A webpage interface (see Fig. 2) was developed in the form of a diabetes risk assessment questionnaire. This questionnaire allows users to input their health-related information, including lifestyle habits, health history and general personal information. The user can submit their responses to receive diabetes risk assessment results displayed on a result page.

Broker and Workers Modules: In our AI-based IoT–cloud framework, the system architecture comprises two main modules—the Broker and the Workers, which together function as computational units and a cloud task scheduler to manage data flow, resource management and task distribution across the network.

- Broker Module: The Broker module acts as the orchestrator of the entire system. On the front end, it handles connection requests and data received from the user gateway devices, providing responses to corresponding user queries. On the back end, it manages secure communication among devices and monitors the heartbeat signals and resource availability of the Worker modules, which allows it to determine the optimal allocation of user data processing tasks. This includes assigning tasks to the optimal Worker during normal operation, or reallocating tasks to another available Worker if necessary, or processing tasks locally in case of hardware failure. This dynamic allocation mechanism serves as a cloud task scheduler to optimise resource utilisation and ensure uninterrupted service.
- Workers Module: The Worker module performs two key functions. First, it sends continuous heartbeat messages along with QoS parameter statistics to the Broker, providing real-time updates on operational status and resource usage. Second, it executes the diabetes prediction tasks on the health data received from the Broker and sends the prediction results back to the Broker module. The Workers function as computational units, processing tasks assigned by the Broker and contributing to the overall performance and scalability of the HealthAIoT system.

Table 2
Sample VMs system metrics from BitBrains cloud dataset.

Timestamp [ms]	CPU cores	CPU capacity provisioned [MHZ]	CPU usage [MHZ]	CPU usage [%]	Memory capacity provisioned [KB]	Memory usage [KB]	Disk read throughput [KB/s]	Disk write throughput [KB/s]	Network received throughput [KB/s]	Network transmitted throughput [KB/s]
1376314846	1	2599.999	17.333	0.667	2 097 152	157 984.8	0	7.8	0	0.267
1376315146	1	2599.999	17.333	0.667	2 097 152	160 780.3	0	7.533	0	0.133
1376315446	1	2599.999	17.333	0.667	2 097 152	173 362.1	0	7.133	0	0.2
1376315746	1	2599.999	22.533	0.867	2 097 152	226 489.9	0	9	0	0.667
1376316046	1	2599.999	20.8	0.8	2 097 152	176 159.7	0	7.067	0	0.2

Table 3
Sample participant records from the diabetes health indicators dataset.

HighBP	HighChol	CholCheck	BMI	Smoker	Stroke	HeartDiseaseorAttack	PhysActivity	Fruits	Veggies	HvyAlcoholConsump	AnyHealthcare	NoDocbcCost	GenHlth	MentHlth	PhysHlth	DiffWalk	Sex	Age	Education	Income
1	1	1	40	1	0	0	0	0	1	0	1	0	5	18	15	1	0	9	4	3
0	0	0	25	1	0	0	1	0	0	0	0	1	3	0	0	0	0	7	6	1
1	1	1	28	0	0	0	0	1	0	0	1	1	5	30	30	1	0	9	4	8
1	0	1	27	0	0	0	1	1	1	0	1	0	2	0	0	0	0	11	3	6
1	1	1	24	0	0	0	1	1	1	0	1	0	2	3	0	0	0	11	5	4

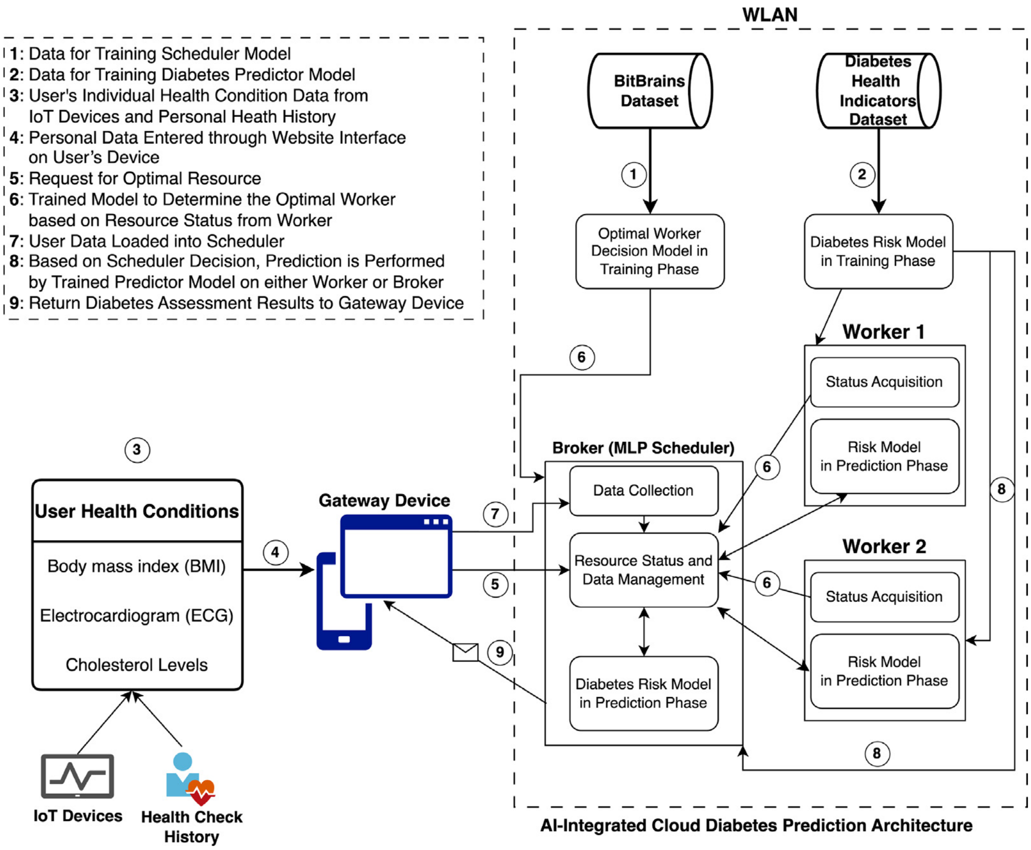


Fig. 1. HealthAIoT architecture.

3.2. HealthAIoT communication protocols

The HealthAIoT architecture handles a significant volume of network instructions and data exchanges among its components. To achieve this, robust and practical communication protocols are essential. Fig. 3 illustrates the topological representation of each network component in the sequence diagram. The AI Scheduler manages the information exchanges between devices within the HealthAIoT network. The HealthAIoT network follows a classic Leader–Follower model, with the Broker acting as the Leader and the Workers as Followers. All physical components (Broker, Workers and Gateway devices) operate within the same WLAN to ensure secure communication. Users access the diabetes assessment application webpage interface (see Fig. 2) through an HTTP request from the Gateway device. This interface is hosted on the Broker unit. The Broker routes the request based on two scenarios: processing the prediction locally, or assigning it to a Worker unit. Workers periodically send status updates and resource usage statistics to Broker, enabling it to determine the optimal Worker for processing in the second operational mode.

Each communication in the topology follows a standardised protocol. As shown in Fig. 3, there are two communication possibilities: communication via the Broker or via a Worker. The user accesses the HealthAIoT webpage interface from the Gateway device by sending a request to the Broker, which responds with a ‘200’ HTTP status code if the request is successful. The Broker then provides the Gateway device with the appropriate IP address (its own or that of the optimal Worker) depending upon the operation mode. The user then can send data for diabetes prediction to this IP address via the Gateway device. In the Broker communication mode, if the Workers are over-utilised or non-functional, the Gateway device immediately receives the Broker’s IP address without checking the status of the Workers. The Broker can handle the data processing depending on its resource availability. In the Worker communication mode, the Broker routes the IP address of the optimal Worker to the Gateway device. If a Worker failure occurs, the Broker provides the IP address of the next optimal Worker or itself. After this request–response cycle, the Gateway device sends the user data to the designated IP address (Broker or Worker) for processing and diabetes prediction. To enhance data security, user data accessed via the webpage interface is not stored within the HealthAIoT system.

3.3. AI-driven cloud scheduler model

The AI cloud scheduler is developed with an MLP network. It was then deployed on the Broker to serve two functions: it monitors system metrics of Workers and dynamically manages data-related requests to determine the Worker with the optimal resources for

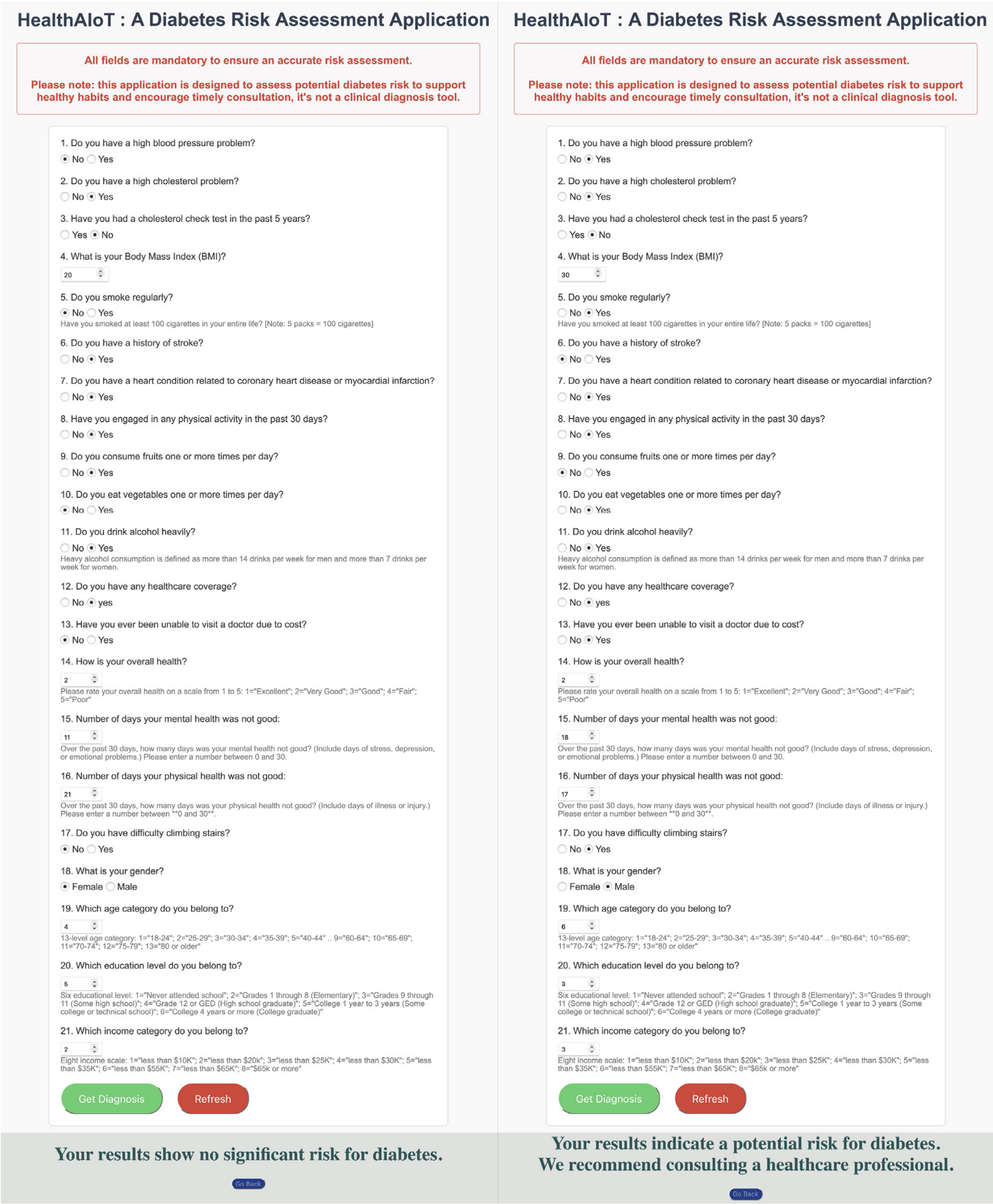


Fig. 2. HealthAIoT disease prediction user interface webpage.

processing tasks. It also oversees a fail-safe mechanism by migrating tasks to other available resources (either another Worker or itself) if the initially assigned Worker is unreachable. This ensures optimal throughput and prevents communication bottlenecks across devices by efficiently balancing task distribution. Fig. 4 illustrates the scheduling mechanism used to achieve load balancing, as well as the fail-safe mechanism implemented to prevent system disruption in case of Worker unit failure. The dataset and model structure and training processes are detailed below:

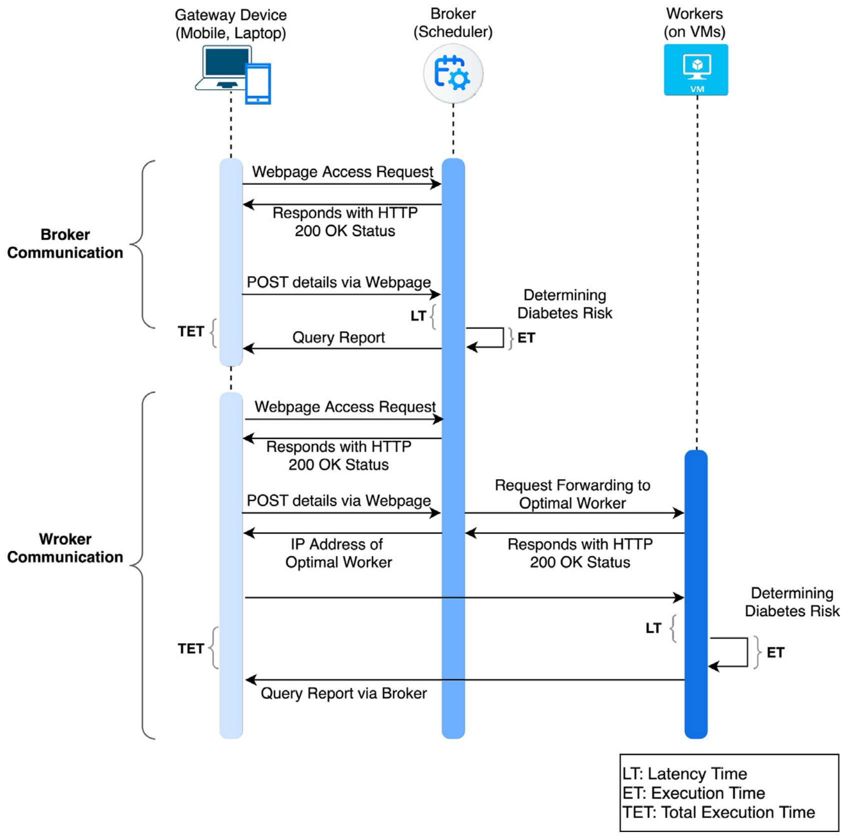


Fig. 3. HealthAIoT framework communication protocols.

3.3.1. BitBrains cloud dataset

In this research, a cloud workload dataset from BitBrains [47] was used to train the cloud scheduler model. The dataset contains 11 features (see Table 2) capturing various system metrics that provide comprehensive insights into the resource usage and performance of VMs in the real cloud environment: (a) Timestamp (time in ms), (b) CPU Cores (number of cores), (c) CPU Capacity Provisioned (requested CPU capacity in MHz), (d) CPU Usage (utilisation in MHz), (e) CPU Usage Percentage (utilisation in percentage), (f) Memory Capacity Provisioned (requested memory capacity in KB), (g) Memory Usage (memory utilisation in KB), (h) Disk Read Throughput (disk read speed in KB/s), (i) Disk Write Throughput (disk write speed in KB/s), (j) Network Received Throughput(network received speed in KB/s), (k) Network Transmitted Throughput(network transmitted speed in KB/s).

3.3.2. Cloud dataset preparation and pre-processing

Data processing starts with loading the Bitbrains dataset [47]. Dimensionality reduction is performed to retain only the columns related to CPU Usage, Memory Usage, Memory Capacity Provisioned, Network Received Throughput and Network Transmitted Throughput. Next, data cleaning is conducted to remove any rows containing only null values. On the refined dataset, Memory Usage (in percentage) is calculated from the values of Memory Usage and Memory Capacity Provisioned columns; this dataset is referred to as D. The data in D are then randomly shuffled and split in half. The two halves are placed side by side to create a new dataset with eight feature columns, four features for each worker (represented as W_1 and W_2), which enables the analysis of interactions between pairs of Workers based on these system metrics as shown Eq. (1):

$$W_1, W_2 = \left\lfloor \frac{D}{2} \right\rfloor \tag{1}$$

The scoring function combines all the features for each Worker on a row-wise basis to determine the optimal Worker. The features and target columns are then split in two separate DataFrames for further computation. The split dataset is divided into a training set and a test set with an 80:20 ratio. The data is normalised using the StandardScaler function from the sklearn module to ensure consistency across all features. This aids in better feature learning by the models. The normalised data is then converted into tensors. The scoring function is defined on a 4-dimensional input vector $(\text{cpu}, \text{mem}, \text{recv}, \text{send})$ as follows Eq. (2):

$$S(\text{cpu}, \text{mem}, \text{recv}, \text{send}) = \text{cpu} + \text{mem} + \text{recv} + \text{send} \tag{2}$$

where cpu represents CPU utilisation of the VM in percentage, mem represents memory utilisation of the VM in percentage, $recv$ represents VM network received bandwidth in Mbps, and $send$ represents VM network transmitted bandwidth in Mbps.

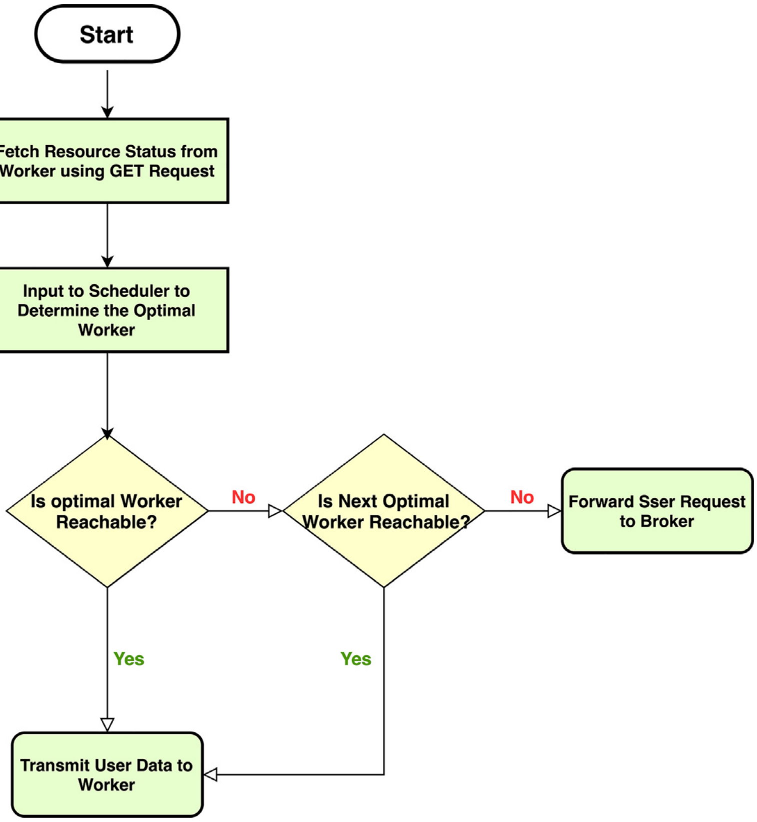


Fig. 4. HealthAIoT task scheduling mechanism.

3.3.3. Scheduler model structure, parameters, and tuning process

Table 4 shows the details of the scheduler structure and its training parameters setup used, the Algorithm 1 presents the corresponding pseudocode. In the pseudocode, $\mathcal{DL}$train and $\mathcal{DL}$test denote the training and test data loaders respectively, which provide batches of input features X_{batch} and corresponding labels y_{batch}. The model parameters θ are updated during training via backpropagation using the Adam optimiser (configured with a learning rate α and a weight decay λ). The total number of training epochs is denoted by E and the cross-entropy loss function $\text{Loss}(\cdot)$ measures the deviation between the predicted outputs $\hat{y}\text{batch} = \mathcal{M}(X\text{batch}; \theta)$ and the true labels y_{batch}. An evaluation function $\mathcal{EM}$ computes the ratio of correct predictions to the total number of samples, providing both training and test accuracies.

The scheduler incorporates a neural network-based supervised learning algorithm with one hidden layer. Glorot initialisation is utilised to ensure balanced weight distribution across layers [48]. The Adam optimiser (with a learning rate of 0.001 and weight decay of 0.00001) is used to update model parameters to balance convergence. To prevent overfitting and improve generalisation, a dropout rate of 0.3 is applied to the hidden layer, and a Leaky Rectified Linear Unit activation function is used. Model training involves batch-wise forward passes to compute predictions, followed by backward passes to calculate gradients and update parameters via backpropagation, which minimises the loss and enhances prediction accuracy [49]. After each epoch, the model evaluates its performance on both training and test data by averaging the batch-wise accuracies.

In terms of computational efficiency, each batch requires a forward pass (with a cost proportional to the batch size B times the feature size f) a backward pass (with a cost that scales with the number of parameters p). Therefore, over E epochs, the overall time complexity can be approximated as Eq. (3):

$$O(E \times (B \times f + p)) \tag{3}$$

Table 4
Structure and parameters of the cloud scheduler model.

Component	Configuration
Input Size	64 X 8 (Batch Size X Number of Features)
One Hidden Layer	8 input neurons, 32 output neurons
Activation Function	Leaky ReLU Applied to hidden layer
Dropout Layer	Applied to hidden layer, dropout rate = 0.3
Output Layer	32 input neurons, 2 output neurons (number of classes)
Output Size	64 X 2 (Batch Size X Number of Classes)
Optimizer	Adam with Learning rate (α) = 0.001 and weight decay (λ) = 0.00001
Parameter Initialisation $\mathcal{X}$	Glorot Initialisation

Algorithm 1 MLP-based Cloud Scheduler Training & Evaluation

Initialise model $\mathcal{M}$ with parameters θ using Glorot initialisation.
Initialise empty lists: `losses`, `trainAcc`, `testAcc`.
for epoch = 1 to E **do**
  Set $\mathcal{M}$ to **train** mode.
  for each batch $(X_{\text{batch}}, y_{\text{batch}})$ in $\mathcal{DL}_{\text{train}}$ **do**
    Compute $\hat{y}_{\text{batch}} \leftarrow \mathcal{M}(X_{\text{batch}}; \theta)$.
    Compute loss $l \leftarrow \text{Loss}(y_{\text{batch}}, \hat{y}_{\text{batch}})$.
    Reset gradients of θ.
    Backpropagate: update θ via Adam with learning rate α and weight decay λ.
    Append l to `losses`.
  end for
  Set $\mathcal{M}$ to **evaluation** mode.
  Compute training accuracy `trainAcc` $\leftarrow \mathcal{EM}(\mathcal{M}, \mathcal{DL}_{\text{train}})$.
  Compute test accuracy `testAcc` $\leftarrow \mathcal{EM}(\mathcal{M}, \mathcal{DL}_{\text{test}})$.
  Save model checkpoint.
end for
return Best model checkpoint.

3.4. Diabetes prediction model

The diabetes prediction model is trained to assess the individuals' risk of developing diabetes based on their response on the questionnaires interface. The trained model was deployed on Workers on Google Cloud Platform (GCP) to conduct the assessment. This section describes the dataset and prediction model development process.

3.4.1. Diabetes dataset

This research uses the Diabetes Health Indicators Dataset from Centres for Disease Control and Prevention (CDC), which is publicly available from the UCI Machine Learning Repository [50,51] and contains 253,680 participant records. Each record provides a snapshot of an individual's health profile, lifestyle habits and demographic characteristics relevant to diabetes risk. The dataset was collected and anonymised by the CDC, it does not include any personally identifiable information (PII) such as names or addresses. Instead, participants are represented by integer-coded attributes for demographic and health metrics.

The dataset was fetched via the ucimlrepo Python library from the UCI Machine Learning Repository. It is distributed as a CSV file with 21 features (columns) plus a single binary target column. The CDC aggregated these health indicators from a diverse adult population, including metrics for blood pressure, cholesterol, physical activity, and self-reported health ratings. Individuals are classified as either non-diabetic (0) or diabetic (1). In the dataset, 84.24% of the participants are reported as having no diabetes or diabetes only during pregnancy, 1.83% are identified as having pre-diabetes, and 13.93% are diabetic. Fig. 5 shows the correlation matrix for the 21 features of the diabetes dataset to explore how risk factors interrelate. The heat map of correlation coefficients represents the clusters of positively correlated risk indicators such as HighBP, Stroke, HeartDiseaseorAttack and possible negative associations such as PhysActivity and BMI, which help with the interpretation of health-behaviour linkages in the dataset. See Table 3 for the example of five records in the dataset, illustrating how features are coded, most of them are integer- or binary-coded with Age, Education, and Income represent ordinal levels. The dataset's features are summarised as follows:

Health Conditions: cholesterol level, Body Mass Index (BMI), history of stroke, heart disease or heart attack;

Health Behaviours: smoking status, physical activity in the last 30 days, regular consumption of fruits and vegetables, and alcohol consumption;

Medical Access: cholesterol check within five years, access to healthcare, and affordability of doctor visits;

Self-Reported Health: general health rating, number of days with mental or physical health issues, and mobility issues;

Demographics: sex, age, education level, and income range.

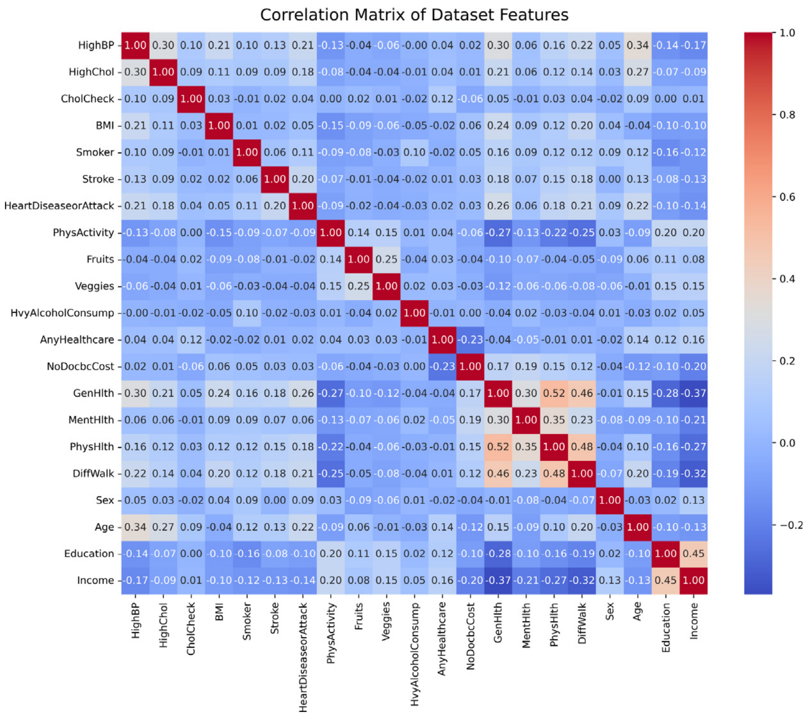


Fig. 5. Correlation matrix of diabetes health indicators dataset features.

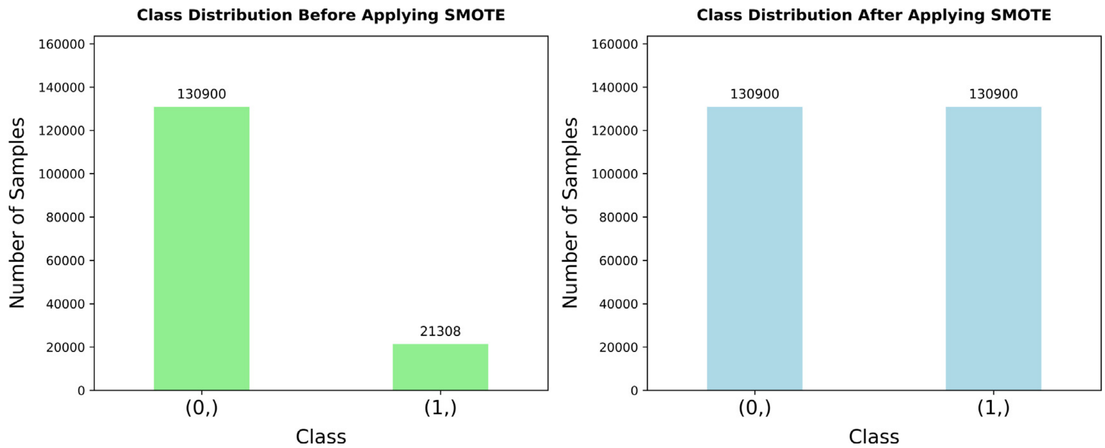


Fig. 6. Comparison of data distribution before and after applying SMOTE.

3.4.2. Diabetes dataset preparation and pre-processing

The CDC Diabetes Health Indicators dataset [50,51] is imported from the UCI Machine Learning Repository using its ucimlrepo Python package. The dataset is divided into training, validation and test sets with a 60:12:28 ratio and then normalised using StandardScaler function from sklearn module to ensure uniform scaling of features. There was a significant class imbalance in the dataset, which was addressed using the Synthetic Minority Oversampling Technique (SMOTE) to balance the class distributions and enhance model training [52]. SMOTE was applied only to the training set to ensure the model learned meaningful patterns and relationships from the data without introducing bias into the validation and test sets. The class distribution figures before and after applying SMOTE are shown in Fig. 6. Finally, the datasets were converted into tensors and batch-loaded for training using the TensorDataset and DataLoader functions from the Torch framework.

3.4.3. Diabetes prediction model structure, parameters, and tuning process

Table 5 shows the details of the diabetes prediction model structure and corresponding parameters used during the training process. The pseudocode is presented in Algorithm 2. $\mathcal{DL}$train and $\mathcal{DL}$test represent the training and test data loaders respectively,

Table 5
Structure and parameters of the diabetes prediction model.

Component	Configuration
Input Size	128 X 21 (Batch Size X Number of Features)
Two Hidden Layers	First Layer: 21 input neurons, 128 output neurons
	Second Layer: 128 input neurons, 64 output neurons
Activation Functions	Leaky ReLU applied to both hidden layers
Dropout Layer	Applied to second hidden layer, dropout rate = 0.24237
Output Layer	64 input neurons, 2 output neurons (number of classes)
Output Size	128 X 2 (Batch Size X number of Classes)
Two Batch Normalisation Layers	Applied on First Hidden Layer with 128 Features
	Applied on 2nd Hidden Layer with 64 Features
Optimiser	AdamW with learning rate (α) = 0.006771 and weight decay (λ) = 0.000031
Parameter Initialisation $\mathcal{X}$	Glorot initialisation

which provide mini-batches of input features X and corresponding labels y. The model parameters are represented by θ, which are updated during training via backpropagation using the AdamW optimizer $\mathcal{O}$, configured with a learning rate α and weight decay λ. The total number of training epochs is denoted by E. The loss function $\mathcal{L}$ is the cross-entropy loss that measures the deviation between the predicted outputs $\hat{y} = \mathcal{M}(X;\theta)$ and the true labels y. An evaluation function $\mathcal{EM}$ computes the ratio of correct predictions to the total number of samples, providing training, validation and test accuracies. Early stopping $\mathcal{ES}$ is employed to stop training when no improvement in validation accuracy is observed over a specified number of epochs.

Fig. 7 shows the overview for developing and deploying the diabetes prediction model. The diabetes prediction model employs a MLP architecture, which contains an input layer with 21 neurons corresponding to the 21 diabetes dataset features. The model builds upon the cross-entropy loss function and the AdamW optimiser (with a learning rate of 0.006771 and weight decay of about 0.000031 to minimise the prediction errors and balance convergence speed and regularisation. It also utilises Glorot initialisation to balance weight distribution across different layers. To reduce overfitting and improve the model's generalisation, a dropout rate of 0.24237 is applied to the second hidden layer to randomly deactivate neurons during training. The training process comprises batch-wise forward passes to compute predictions, loss calculations and subsequent parameter updates via backpropagation. After each epoch, the model's accuracy is evaluated on the validation set, and finally its performance is assessed on the test dataset. The overall time complexity of this training process is also given by Eq. (3), which accounts for both the forward and backward passes, thereby ensuring computational efficiency while performing the diabetes prediction task.

Algorithm 2 Diabetes Prediction Model Training and Evaluation

Initialise model $\mathcal{M}$ with parameters θ using Glorot initialisation; set up Early Stopping $\mathcal{ES}$, batch size $\mathcal{B}$, and data $\mathcal{X}$.
Configure Cross-Entropy Loss $\mathcal{L}$ and AdamW optimizer $\mathcal{O}(\alpha, \lambda)$.
for epoch = 1 **to** E **do**
    Set $\mathcal{M}$ to **train** mode.
    for each mini-batch $(X, y) \in \mathcal{DL}_{\text{train}}$ **do**
        Compute predictions $\hat{y} \leftarrow \mathcal{M}(X;\theta)$.
        Compute loss $l \leftarrow \mathcal{L}(y, \hat{y})$.
        Reset gradients and update θ via backpropagation using $\mathcal{O}$.
        Record l in `losses`.
    end for
    Set $\mathcal{M}$ to **evaluation** mode.
    Compute training accuracy $a_{\text{train}} \leftarrow \mathcal{EM}(\mathcal{M}, \mathcal{DL}_{\text{train}})$.
    Compute validation accuracy $a_{\text{val}} \leftarrow \mathcal{EM}(\mathcal{M}, \mathcal{DL}_{\text{val}})$.
    if Early stopping condition is met (via $\mathcal{ES}$) **then**
        break
    end if
    Save checkpoint of $\mathcal{M}$.
end for
Compute test accuracy $a_{\text{test}} \leftarrow \mathcal{EM}(\mathcal{M}, \mathcal{DL}_{\text{test}})$.
Return the best model $\mathcal{M}$.

4. Performance evaluation

To validate the proposed HealthAIoT framework in a real-time application, we deployed the AI cloud scheduler (Broker) and the Workers on GCP and employed a comprehensive set of evaluation metrics, including QoS parameters and AI performance metrics. The experiment setup and evaluation metrics are explained as follows.

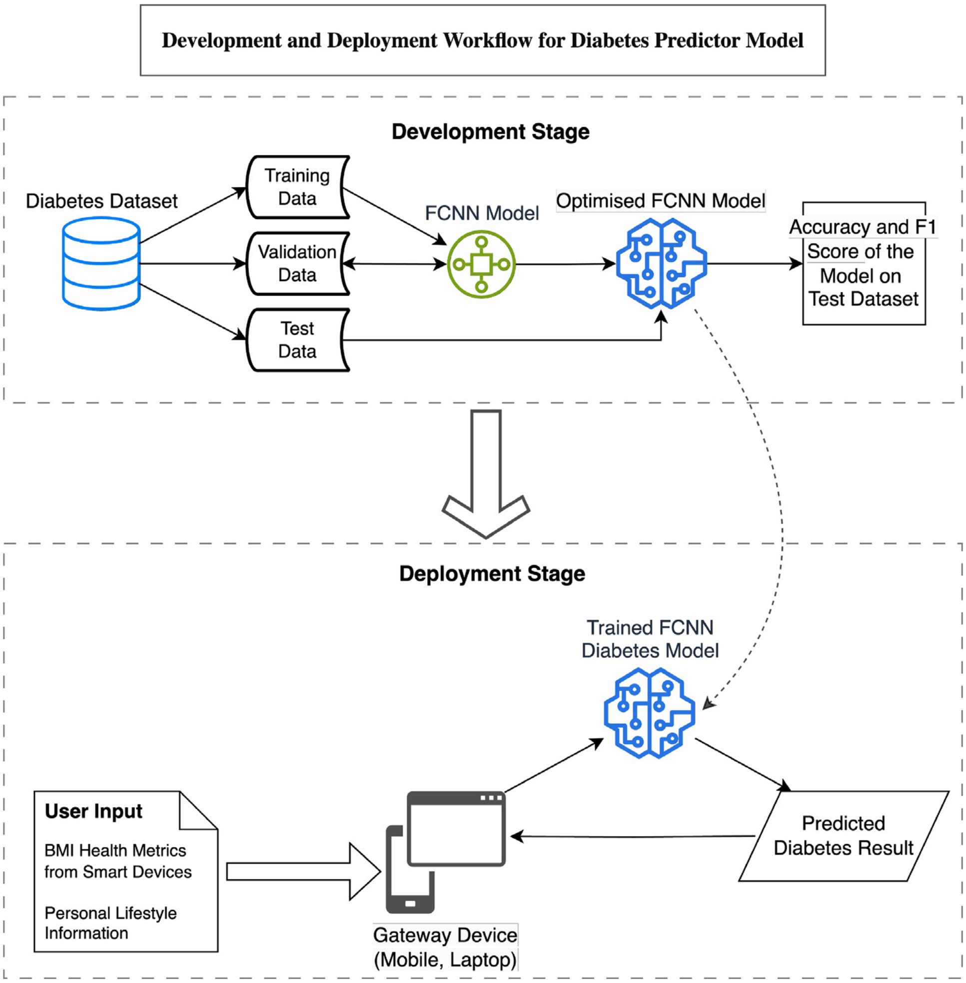


Fig. 7. Development and deployment workflow for diabetes prediction model.

4.1. Experimental setup

The experiment of our proposed framework HealthAIoT was performed and tested on CloudAIBus, CloudAIBus is an AI-based testbed designed for improving resource allocation in cloud environments [28], it can function as a flexible framework for integrating and testing predictive models. Our experimental configuration is explained in detail: The Broker was running on Apple MacBook Air with M2 64-Bit ARM processor and 8 GB RAM, running macOS Ventura 13.4.1; The Workers were hosted on Google Cloud Platform (GCP) e2-small VMs with 2vCPU (1 shared core), 2 GB RAM, 15 GB Balanced Persistent Disk, running Ubuntu 20.04 LTS. Gateway devices used in this experiment were Google Pixel 6a and an Apple MacBook Air. The majority of the HealthAIoT framework was developed in Python, with Python 3.11.3 for Broker and Python 3.8.10 for the Workers (on GCP VMs). The webpage interface was developed using HTML. The HealthAIoT framework was developed using the following Python modules: torch v1.13.1+cpu for Workers, torch v2.3.1 for Broker, flask v3.0.3, scikit-learn v1.3.2, joblib v1.4.2, numpy v1.24.4, pandas v2.2.2, matplotlib v3.9.1, requests v2.32.3, imbalanced-learn (imblearn) v0.0 and ucimlrepo v0.0.7.

Power consumption of the Broker was measured using the Activity Monitor macOS application [53], while power consumption for the Workers is discussed in Section 4.2. Processor performance metrics (CPU utilisation and latency) for both Broker and Worker units were generated using the Python Psutil (process and system utilities) library v6.0.0.

4.2. Quality of service (QoS) parameters

In cloud computing services, requirements for QoS parameters vary in different application scenarios [54–56]. The choice of QoS parameters depends on the focus of the research [57]. In this research, QoS parameters of energy consumption, carbon-free energy usage, cost, execution time and latency are considered. Evaluating energy consumption of cloud services is essential when developing cloud-based applications, as the recent advancements of AI have consumed a considerable amount of energy, resulting in substantial CO_2 emissions [58]. Incorporating energy consumption and carbon-free energy usage can illustrate how our proposed framework leverages renewable energy, which is particularly important for healthcare applications that increasingly prioritise energy sustainability. The cost parameter is considered to assess the operation cost of the services, which is crucial for both service providers and end-users. The parameters of latency and execution time are considered. This is particularly important in cloud-based healthcare applications, where real-time monitoring and instant feedback to users and healthcare practitioners are essential to ensure patient care and user experience [59].

4.2.1. Energy consumption

For our experimental setup, the configured Workers utilise GCP VMs, designed to imitate the functionality of low-powered computing machines. To simulate an IoT-AI integrated cloud environment, the configured VMs are placed in the nearest VM zones to the Broker. The total power consumption formula is given as Eq. (4) [60]:

$$P_{\text{total}} = P_{\text{idle}} + P_{\text{cpu}}(u) + P_{\text{eth,idle}} + P_{\text{WLAN,idle}} \tag{4}$$

where P_{idle} is the power consumption of the device in the idle state, $P_{\text{cpu}}(u)$ represents the power utilised by CPU, $P_{\text{eth,idle}}$ represents power consumption of Ethernet module in idle state, $P_{\text{WLAN,idle}}$ represents power consumption WLAN module in idle state.

The $P_{\text{cpu}}(U)$ is calculated as Eq. (5), where U is the CPU utilisation ratio ranging between 0 and 1 [60]:

$$P_{\text{cpu}}(U) = 0.6191 \times U \tag{5}$$

4.2.2. Carbon-free energy usage

To calculate the Carbon-Free Energy (CFE) usage, the total energy consumption is calculated using Eq. (6):

$$E_{\text{total}} = P_{\text{total}} \times t \tag{6}$$

where E_{total} is total energy usage in watt-hours (Wh), P_{total} is the total power consumption in watts, and t represents the system's up-time in hours.

In our experiment, we consider the system to be running for 24 h to calculate the total energy consumption. GCP provides region- and zone-specific Carbon-Free Energy (CFE) percentages, which indicate the proportion of energy consumed by VMs that are generated from the carbon-free resources [61]. Employing Workers on GCP VMs significantly reduces the carbon footprint of implementing the HealthAIoT system. The carbon-free energy consumption is calculated using the following Eq. (7):

$$E_{\text{carbon-free}} = E_{\text{total}} \times \frac{\text{CFE}}{100} \tag{7}$$

4.2.3. Cost

The cost of operating the VMs in various modes of operation is given by Eq. (8):

$$Cost_{\text{total}} = 0.02 \times t \times n \tag{8}$$

$0.02 is the cost of operating the GCP VMs per hour for the configuration mentioned in Section 4.1 [62], *'n'* represents number of Worker units and *'t'* represents the total time VMs were operational, the total operational cost of our system was calculated based on a continuous 24-hour runtime.

4.2.4. Latency and execution time

With reference to the communication protocols diagram, the temporal metrics were monitored and analysed. The latency time measures the delay between when a user POSTs (HTTP request) the data and when the diabetes estimation model runs on the Broker or the Worker. Total execution time accounts for the entire time lapse from when the user data reaches the destination node for execution and data processing to when the user receives their result. For our deployment setup, we calculated all three parameters for different operation modes: Broker only, 1-Worker and 2-Worker.

4.3. AI model performance

To assess the model performance, we examined the test accuracy of the model, which is defined as the percentage of patients for whom the model correctly classifies the presence or absence of the disease [44]. Additionally, we employed a confusion matrix and calculated the F1-score to gain a deeper insight into the model's classification performance. The Confusion Matrix is a square 2 × 2 matrix, with each row and column combination representing a class and it assesses the model's ability to classify between these classes [63]. For our diabetes model, the four classes include true negative (TN) - a person is predicted as non-diabetic and is actually non-diabetic, false positive (FP) - a person is predicted as having diabetes but is actually non-diabetic, false negative (FN) - a person is predicted non-diabetic but actually has diabetes, true positive (TP) - a person is predicted as having diabetes and is actually diabetic. The F1-score is calculated by Eq. (9) [64]:

$$F1 - score = 2 \times \frac{Recall \times Precision}{Recall + Precision} \tag{9}$$

Where recall is defined as Eq. (10):

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

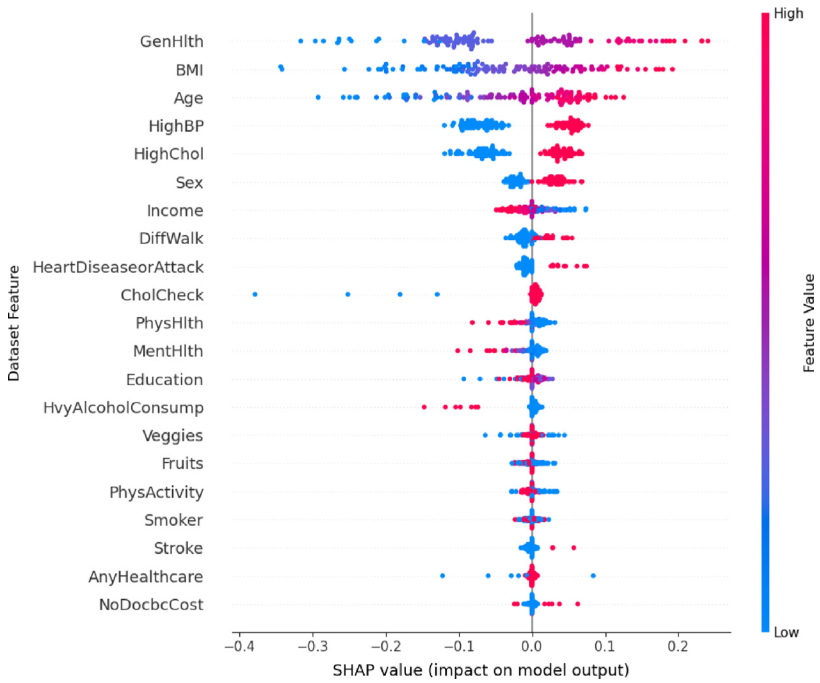


Fig. 8. SHAP swarm plot illustrating data feature impact on model predictions.

It represents the ratio of users diagnosed with diabetes to the actual total number of users with diabetes. Precision is defined as Eq. (11):

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

It represents the ratio of user diagnosed with diabetes to the predicted total number of users who are diabetic.

4.4. Explainable AI in healthcare application

In addition to achieving accessible disease prediction and sustainability system performance, our framework also emphasises the critical need for XAI in healthcare. The SHAP plot in Fig. 8 illustrates the contribution of each feature from the CDC Diabetes Health Indicators dataset to the diabetes prediction model's decision outcome. This visualisation enables healthcare professionals to validate whether these contributions align with their conventional approaches, leveraging their domain expertise to ensure model transparency and trustworthiness [65,66].

4.5. Experimental results

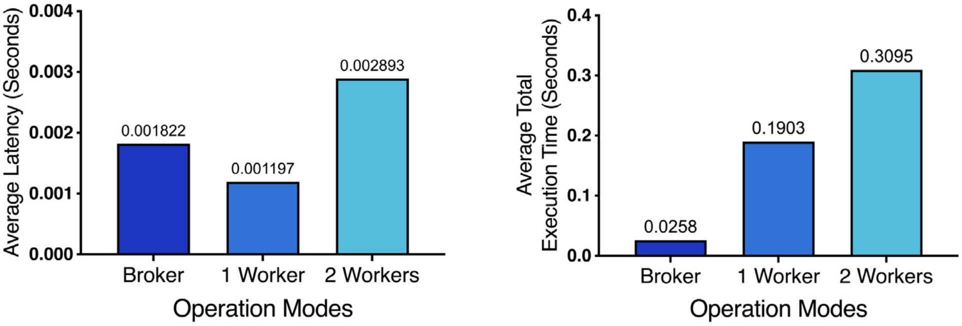
The HealthAIoT system's performance was evaluated across three operation modes: Broker only, 1-Worker, and 2-Worker. The experimental outcomes are discussed below:

4.5.1. System latency and execution time

Fig. 9(a) shows the latency variations across operation modes. The 1-Worker mode exhibits the lowest latency, outperforming both the Broker and 2-Worker modes. This is attributed to the efficient distribution of computational tasks between the Broker and a single Worker, minimising processing delays. Fig. 9(b) shows that total execution time increases with the number of Workers, the Broker mode completes tasks fastest, while the 2-Worker mode incurs the longest execution time (0.3095s on average).

4.5.2. Power and carbon-free energy usage

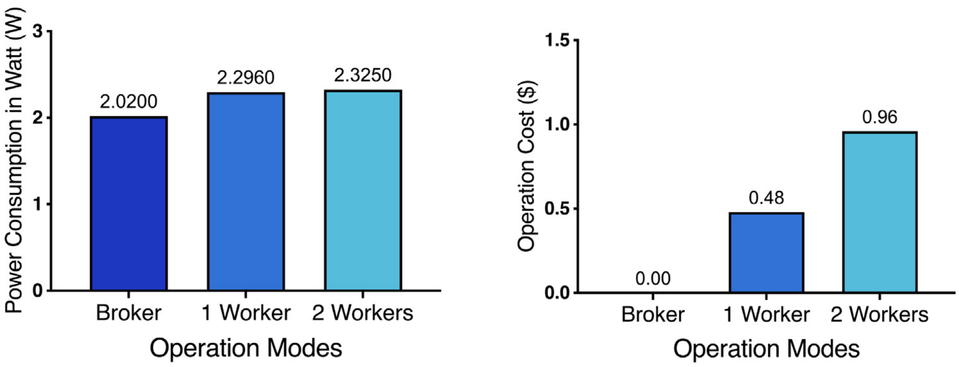
As discussed in Section 4.2, power consumption was calculated from the CPU utilisation ratio for each mode. The Broker mode consumes the least power (Fig. 10(a)), as it avoids cloud resource utilisation. Power consumption scales with Workers, with the 2-Worker mode consumes the most. Fig. 11 shows a critical sustainability advantage, both 1-Worker and 2-Worker modes leverage carbon-free energy, while Broker mode relies entirely on carbon-intensive sources. This highlights the environmental benefits of cloud-based processing despite higher total energy consumption.



(a) System Latency in Different Modes

(b) Total Execution Time in Different Modes

Fig. 9. Comparison of system latency and execution time in different modes.



(a) Power Consumption in Different Modes

(b) Total Operation Cost in Different Modes

Fig. 10. Comparison of power consumption and operation cost in different modes.

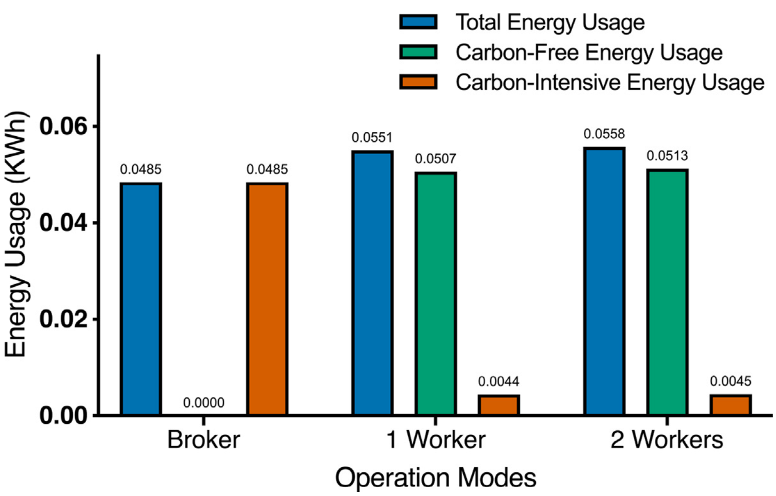


Fig. 11. Green energy usage in different modes.

4.5.3. Operation cost

Operational costs increase with cloud resource usage (Fig. 10(b)). Since the Broker runs locally on the end device, Broker mode incurs no cost. As the Workers are deployed on GCP, we calculated the total operation cost of the system based on GCP's pay-per-use pricing model. This demonstrates a cost-performance trade-off when scaling cloud resources.

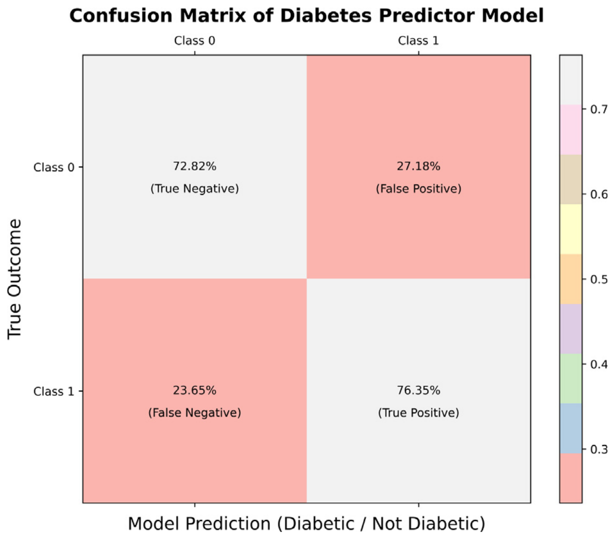


Fig. 12. Confusion matrix of diabetes prediction model.

Table 6
System latency and total execution time comparison (in seconds).

Comparison	Frameworks	Broker	1-Worker Mode	2-Workers Mode
System Latency	HealthFog [44]	0.0146	0.0200	0.0174
	FedSDM [43]	0.0333	0.2273	–
	HealthAIoT	0.0018	0.0011	0.0028
Total Execution Time	HealthFog [44]	0.2408	0.3741	0.3660
	FedSDM [43]	1.60071	2.3214	–
	HealthAIoT	0.0258	0.1903	0.3095

Table 7
Model average accuracy comparison in percentage.

Frameworks	HealthAIoT	FedSDM [43]	HealthFog [44]
Accuracy	78.3%	93.75%	72%

4.5.4. Accuracy, confusion matrix and F1-score

The diabetes prediction model achieves an overall accuracy of 78.30% and an F1-score of 0.7719 on unseen data. The confusion matrix shown in Fig. 12 indicates the model correctly identifies non-diabetic individuals with 72.82% precision and diabetic cases with 76.53% recall. However, in 23.65% of cases, it misclassifies diabetic individuals as non-diabetic (risk of delayed intervention) and 22.18% of non-diabetic cases as diabetic (potential overdiagnosis). These errors could be mitigated by incorporating more diverse training data to improve generalisability.

5. Discussion and performance comparison with existing works

The HealthAIoT framework balances performance, cost and sustainability through its AI-driven scheduler. Compared to the baselines HealthFog [44] and FedSDM [43] (as discussed in Section 2), our framework demonstrates superior efficiency in latency, execution time, and energy usage while maintaining competitive accuracy for diabetes risk prediction.

5.1. System latency and execution time

We discuss the performance results of the HealthAIoT system in contrast to the benchmarks of HealthFog [44] and FedSDM [43] as reviewed in Section 2. As shown in Table 6, HealthAIoT achieves significantly lower latency and execution time across all operation modes compared to both baselines. In Broker mode, HealthAIoT reduces latency by 87.7% (HealthFog's 0.0146s) and 94.6% (FedSDM's 0.0333s). Similarly, total execution time in 1-Worker mode (0.1903s) is 49.1% lower than HealthFog's 0.3741s and 91.8% lower than FedSDM's 2.3214s. These improvements benefit from the lightweight architecture that employs a streamlined MLP model optimised for edge–cloud environment, reducing computational overhead; also due to AI-driven scheduler that dynamically allocates tasks between the Broker and Workers, minimising coordination delays inherent in FedSDM's federated averaging and HealthFog's Blockchain-based FogBus.

Table 8
Power usage comparison (in watt)

Frameworks	Broker	1-Worker Mode	2-Workers Mode
HealthFog [44]	2.22	2.83	3.4422
HealthAIoT	2.02	2.296	2.325

5.2. Energy efficiency and sustainability

While direct power consumption comparisons are unavailable for FedSDM, HealthAIoT's Broker mode consumes 58% less power than HealthFog (Table 8), primarily due to avoiding resource-intensive Blockchain operations (used in HealthFog). Furthermore, HealthAIoT's use of carbon-free energy in 1-Worker and 2-Worker modes addresses a critical gap in both baselines, which do not prioritise environmental sustainability.

5.3. Accuracy trade-offs

As shown in Table 7, HealthAIoT achieves 78.3% accuracy, outperforming HealthFog (72%) but lagging behind FedSDM (93.75%). This discrepancy arises from fundamental differences in task complexity and architectural priorities. FedSDM's autoencoder-based anomaly detection for ECG data benefits from federated learning (FedAvg), which aggregates diverse ECG datasets across edge devices, improving generalisability. However, this comes with higher latency and energy consumption. While FedSDM prioritszes accuracy for ECG anomaly detection, HealthAIoT emphasises real-time diabetes risk prediction with XAI interpretability. The 0.7719 F1-score and SHAP-based feature explanations make it clinically actionable despite slightly lower accuracy.

5.4. Limitations

The following are the main limitations of this research work:

- Edge-Cloud Environment: The Workers are deployed on Google Cloud VMs to emulate low-powered edge devices, prioritising reproducibility and controlled benchmarking. While real-world testing on devices such as Raspberry Pi is needed, the framework's low latency advantage suggest strong potential for deployment on actual edge hardware.
- Dataset Scope and Framework Adaptability: The current diabetes model is trained on a U.S.-centric dataset, this may limit its generalisability to global populations. However, the HealthAIoT framework is designed as a modular platform, its architecture separates the prediction module from the cloud scheduler, enabling seamless integration of region-specific datasets or alternative predictors (such as for cardiovascular or other chronic diseases) without redesigning the core IoT–cloud infrastructure.
- Balancing Accuracy with Deployment Efficiency: The diabetes predictor achieves 78.3% accuracy and an F1-score of 0.7719, which might be relatively lower than state-of-the-art models for related tasks. This trade-off is because of our intentional use of a lightweight MLP architecture, which are optimised for fast inference and compatibility with resource-constrained IoT–edge environments. While deeper architectures such as transformer-based or multimodal frameworks might improve accuracy [67], they would compromise real-time responsiveness, which is a critical requirement for clinical deployment.

6. Conclusions and future work

We propose HealthAIoT framework as a scalable and sustainable architecture for integrating IoT devices with AI-enabled cloud services, prioritising real-time healthcare applications. We tested the framework in three operation modes, Broker only, 1-Worker and 2-Workers in terms of QoS parameters and predictive performance. In 1-Worker mode, HealthAIoT achieves a system latency of 0.0011s and total execution time of 0.1903s, outperforming HealthFog's 0.0200s and 0.3741 respectively, and FedSDM which reports 0.2273s latency in 1-Worker mode. This improvement stems from the lightweight MLP-based scheduler, which avoids computational overheads associated with HealthFog's ensemble models and FedSDM's federated learning framework. On average, our HealthAIoT framework consumed 20.11% less power compared to HealthFog model. The improvement on environmental sustainability addressed the lack presented in existing studies.

The overall performance of HeathAIoT demonstrate potential of clinical deployment, with its modular design and its AI scheduler that is separate from disease-specific prediction modules, which allows seamless adaptation to other healthcare application. By prioritising IoT–cloud orchestration over task-specific optimisation, the framework provides a foundational infrastructure for scalable and energy-efficient healthcare analytics.

6.1. Future research directions

To address limitations and enhance practical utility, future work will focus on:

- In future work, we aim to implement and validate the HealthAIoT framework in real-world healthcare environments. This validation will help assess the practicality of our framework in addressing real-world challenges such as data security, system scalability and integration with existing healthcare infrastructures.
- To enhance the predictive performance, future work could integrate hybrid architectures such as MLP model with attention mechanisms to improve accuracy without compromising latency, as well as expand training data to include diverse demographic cohorts such as UK Biobank [68] to reduce geographic bias.
- Our HealthAIoT system operates on an end device and two cloud VMs architecture that process data from IoT devices. The further work can incorporate low-powered edge devices (e.g., Raspberry Pi) or more number of Workers across various locations to assess the performance under various intermittent network connectivity.
- While HealthAIoT framework has demonstrated promising performance in the integration of IoT and cloud with diabetes risk prediction, it still can further explore the potential of our system for the management of diseases beyond diabetes, leveraging its adaptable and modular structure to support diverse healthcare applications. This can be achieved by incorporating additional IoT components into the HealthAIoT framework, such as Wireless Sensor Networks (WSNs), Bluetooth Low Energy (BLE) and ZigBee and some advanced sensor technologies that can harvest energy from the environment, these enhancements would enable both sustainability and expanded application scenarios [69].
- As quantum computing continues to evolve, the traditional cryptographic methods will likely become insufficient when it comes to handling sensitive data [70]. Some new cryptographic methods can be applied to enhance the protection of sensitive healthcare data in the future work, such as secure hash algorithm (SHA)-3 that can be utilised to provide robust encryption mechanism during health data transmission process [71–73].

Software availability

HealthAIoT has been released as open-source software. The code with experiment scripts and results can be accessed from the GitHub repository: https://github.com/HTXW/HealthAIoT

CRediT authorship contribution statement

Han Wang: Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Formal analysis, Data curation, Conceptualization. **Kumar Ankur Anurag:** Writing – original draft, Validation, Methodology, Investigation, Formal analysis. **Amira Rayane Benamer:** Writing – original draft, Validation, Methodology, Conceptualization. **Priyansh Arora:** Writing – original draft, Validation, Methodology, Conceptualization. **Gurleen Wander:** Writing – original draft, Validation, Methodology, Conceptualization. **Mark R. Johnson:** Writing – original draft, Validation, Supervision, Methodology, Conceptualization. **Ranjit Mohan Anjana:** Writing – original draft, Validation, Supervision, Methodology. **Viswanathan Mohan:** Writing – original draft, Validation, Supervision, Methodology, Conceptualization. **Sukhpal Singh Gill:** Writing – original draft, Validation, Supervision, Methodology, Conceptualization. **Steve Uhlig:** Writing – original draft, Validation, Supervision, Methodology, Conceptualization. **Rajkumar Buyya:** Writing – original draft, Validation, Supervision, Methodology, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] S.S. Gill, M. Xu, C. Ottaviani, P. Patros, R. Bahsoon, A. Shaghaghi, M. Golec, V. Stankovski, H. Wu, A. Abraham, et al., AI for next generation computing: Emerging trends and future directions, Internet Things 19 (2022) 100514.

[2] A. Al-Dulaimy, M. Jansen, B. Johansson, A. Trivedi, A. Iosup, M. Ashjaei, A. Galletta, D. Kimovski, R. Prodan, K. Tserpes, G. Kousiouris, C. Giannakos, I. Brandic, N. Ali, A.B. Bondi, A.V. Papadopoulos, The computing continuum: From IoT to the cloud, Internet Things 27 (2024) 101272, http://dx.doi.org/10.1016/j.iot.2024.101272.

[3] C. Li, J. Wang, S. Wang, Y. Zhang, A review of IoT applications in healthcare, Neurocomputing 565 (2024) 127017, http://dx.doi.org/10.1016/j.neucom.2023.127017.

[4] A. Alanazi, Using machine learning for healthcare challenges and opportunities, Informatics Med. Unlocked 30 (2022) 100924, http://dx.doi.org/10.1016/j.imu.2022.100924.

[5] P. Verma, A. Gupta, V. Jain, K. Shashvat, M. Kumar, S.S. Gill, An AIoT-driven smart healthcare framework for zoonoses detection in integrated fog-cloud computing environments, Software: Pr. Exp. 55 (1) (2025) 133–154, http://dx.doi.org/10.1002/spe.3366.

[6] S.M. Nagarajan, G.G. Deverajan, P. Chatterjee, W. Alnumay, U. Ghosh, Effective task scheduling algorithm with deep learning for Internet of Health Things (IoHT) in sustainable smart cities, Sustain. Cities Soc. 71 (2021) 102945, http://dx.doi.org/10.1016/j.scs.2021.102945.
[7] World Health Organization, Diabetes, 2024, URL https://www.who.int/news-room/fact-sheets/detail/diabetes. (Accessed: 2024-08-07).
[8] P.Z. Zimmet, D.J. Magliano, W.H. Herman, J.E. Shaw, Diabetes: a 21st century challenge, Lancet Diabetes Endocrinol. 2 (1) (2014) 56–64, http://dx.doi.org/10.1016/S2213-8587(13)70112-8.
[9] E.W. Gregg, N. Sattar, M.K. Ali, The changing face of diabetes complications, Lancet Diabetes Endocrinol. 4 (6) (2016) 537–547, http://dx.doi.org/10.1016/S2213-8587(16)30010-9.
[10] T. Yildiz, Diabetes risk assessment and awareness in a university academics and employees, SiSli Etfal Hast. Tip Bulteni / Med. Bull. Sisli Hosp. (2021) http://dx.doi.org/10.14744/SEMB.2021.84770.
[11] P. Gope, T. Hwang, BSN-care: A secure IoT-based modern healthcare system using body sensor network, IEEE Sensors J. 16 (5) (2016) 1368–1376, http://dx.doi.org/10.1109/JSEN.2015.2502401.
[12] F. Desai, D. Chowdhury, R. Kaur, M. Peeters, R.C. Arya, G.S. Wander, S.S. Gill, R. Buyya, HealthCloud: A system for monitoring health status of heart patients using machine learning and cloud computing, Internet Things 17 (2022) 100485, http://dx.doi.org/10.1016/j.iot.2021.100485.
[13] Z. Guan, H. Li, R. Liu, C. Cai, Y. Liu, J. Li, X. Wang, S. Huang, L. Wu, D. Liu, S. Yu, Z. Wang, J. Shu, X. Hou, X. Yang, W. Jia, B. Sheng, Artificial intelligence in diabetes management: Advancements, opportunities, and challenges, Cell Rep. Med. 4 (10) (2023) 101213, http://dx.doi.org/10.1016/j.xcrm.2023.101213.
[14] N. Razavian, S. Blecker, A.M. Schmidt, A. Smith-McLallen, S. Nigam, D. Sontag, Population-level prediction of type 2 diabetes from claims data and analysis of risk factors, Big Data 3 (4) (2015) 277–287, http://dx.doi.org/10.1089/big.2015.0020.
[15] NHS England, Artificial intelligence (AI) and machine learning, 2023, URL https://www.england.nhs.uk/long-read/artificial-intelligence-ai-and-machine-learning/.
[16] NHS England, The NHS AI Lab. URL https://transform.england.nhs.uk/ai-lab/.
[17] S. Dong, P. Wang, K. Abbas, A survey on deep learning and its applications, Comput. Sci. Rev. 40 (2021) 100379, http://dx.doi.org/10.1016/j.cosrev.2021.100379.
[18] B. Sanchez-Lengeling, E. Reif, A. Pearce, A. Wiltschko, A gentle introduction to graph neural networks, Distill 6 (8) (2021) http://dx.doi.org/10.23915/distill.00033.
[19] Y. Qin, J. Wu, W. Xiao, K. Wang, A. Huang, B. Liu, J. Yu, C. Li, F. Yu, Z. Ren, Machine learning models for data-driven prediction of diabetes by lifestyle type, Int. J. Environ. Res. Public Heal. 19 (22) (2022) 15027, http://dx.doi.org/10.3390/ijerph192215027.
[20] F. Krones, U. Marikkar, G. Parsons, A. Szmul, A. Mahdi, Review of multimodal machine learning approaches in healthcare, Inf. Fusion 114 (2025) 102690, http://dx.doi.org/10.1016/j.inffus.2024.102690.
[21] A.M. Nia, M. Mozaffari-Kermani, S. Sur-Kolay, A. Raghunathan, N.K. Jha, Energy-efficient long-term continuous personal health monitoring, IEEE Trans. Multi- Scale Comput. Syst. 1 (2) (2015) 85–98, http://dx.doi.org/10.1109/TMSCS.2015.2494021.
[22] L. Rodríguez-Jiménez, M. Romero-Martín, T. Spruell, Z. Steley, J. Gómez-Salgado, The carbon footprint of healthcare settings: A systematic review, J. Adv. Nurs. 79 (8) (2023) 2830–2844, http://dx.doi.org/10.1111/jan.15671.
[23] J. Malmodin, D. Lundén, The energy and carbon footprint of the global ICT and E&M sectors 2010–2015, Sustainability 10 (9) (2018) 3027, http://dx.doi.org/10.3390/su10093027.
[24] NHS England, Delivering a 'Net Zero' National Health Service. URL https://www.england.nhs.uk/greenernhs/a-net-zero-nhs/.
[25] W.H. Organization, WHO unveils framework for climate resilient and low carbon health systems, 2023, URL https://www.who.int/news/item/09-11-2023-who-unveils-framework-for-climate-resilient-and-low-carbon-health-systems.
[26] S. Nayak, R. Patgiri, L. Waikhom, A. Ahmed, A review on edge analytics: Issues, challenges, opportunities, promises, future directions, and applications, Digit. Commun. Networks 10 (3) (2024) 783–804, http://dx.doi.org/10.1016/j.dcan.2022.10.016.
[27] H. Wang, K.J. Mathews, et al., AmazonAICloud: proactive resource allocation using amazon chronos based time series model for sustainable cloud computing, Computing (2025) 1–23, http://dx.doi.org/10.1007/s00607-025-01435-w.
[28] S. Velu, S.S. Gill, S.S. Murugesan, H. Wu, X. Li, CloudAIBus: a testbed for AI based cloud computing environments, Clust. Comput. 27 (9) (2024) 11953–11981, http://dx.doi.org/10.1007/s10586-024-04562-9.
[29] C. Li, J. Wang, S. Wang, Y. Zhang, A review of IoT applications in healthcare, Neurocomputing 565 (2024) 127017, http://dx.doi.org/10.1016/j.neucom.2023.127017.
[30] A. Rejeb, K. Rejeb, H. Treiblmaier, A. Appolloni, S. Alghamdi, Y. Alhasawi, M. Iranmanesh, The Internet of Things (IoT) in healthcare: Taking stock and moving forward, Internet Things 22 (2023) 100721, http://dx.doi.org/10.1016/j.iot.2023.100721.
[31] H. Wang, V.S. Rajakumar, et al., StockAICloud: AI-based sustainable and scalable stock price prediction framework using serverless cloud computing, J. Supercomput. (2025) 1–25, http://dx.doi.org/10.1007/s11227-025-06984-7.
[32] N. Gershenfeld, R. Krikorian, D. Cohen, The internet of things, Sci. Am. 291 (4) (2004) 76–81, URL http://www.jstor.org/stable/26060727.
[33] L. Atzori, A. Iera, G. Morabito, The internet of things: A survey, Comput. Netw. 54 (15) (2010) 2787–2805, http://dx.doi.org/10.1016/j.comnet.2010.05.010.
[34] L.M. Dang, M.J. Piran, D. Han, K. Min, H. Moon, A survey on internet of things and cloud computing for healthcare, Electronics 8 (7) (2019) 768, http://dx.doi.org/10.3390/electronics8070768.
[35] V.P. Darcini S., D.P. Isravel, S. Silas, A comprehensive review on the emerging IoT-cloud based technologies for smart healthcare, in: 2020 6th International Conference on Advanced Computing and Communication Systems, ICACCS, 2020, pp. 606–611, http://dx.doi.org/10.1109/ICACCS48705.2020.9074457.
[36] M. Haghi Kashani, M. Madanipour, M. Nikravan, P. Asghari, E. Mahdipour, A systematic review of IoT in healthcare: Applications, techniques, and trends, J. Netw. Comput. Appl. 192 (2021) 103164, http://dx.doi.org/10.1016/j.jnca.2021.103164.
[37] A. Hennebelle, H. Materwala, L. Ismail, HealthEdge: A machine learning-based smart healthcare framework for prediction of type 2 diabetes in an integrated IoT, edge, and cloud computing system, Procedia Comput. Sci. 220 (2023) 331–338, http://dx.doi.org/10.1016/j.procs.2023.03.043.
[38] R. Khan, S. Taj, X. Ma, A. Noor, H. Zhu, J. Khan, Z.U. Khan, S.U. Khan, Advanced federated ensemble internet of learning approach for cloud based medical healthcare monitoring system, Sci. Rep. 14 (1) (2024) 26068, http://dx.doi.org/10.1038/s41598-024-77196-x.
[39] G. Ramkumar, J. Seetha, R. Priyadarshini, M. Gopila, G. Saranya, IoT-based patient monitoring system for predicting heart disease using deep learning, Measurement 218 (2023) 113235, http://dx.doi.org/10.1016/j.measurement.2023.113235.
[40] S. Alsubai, M. Sha, A. Alqahtani, M. Bhatia, Hybrid IoT-edge-cloud computing-based athlete healthcare framework: Digital twin initiative, Mob. Networks Appl. 28 (6) (2023) 2056–2075, http://dx.doi.org/10.1007/s11036-023-02200-z.
[41] S.K. Sharma, A.T. Zamani, A. Abdelsalam, D. Muduli, A.A. Alabrah, N. Parveen, S.M. Alanazi, A diabetes monitoring system and health-medical service composition model in cloud environment, IEEE Access 11 (2023) 32804–32819, http://dx.doi.org/10.1109/ACCESS.2023.3258549.
[42] M.S. Atoum, A. Pati, M. Parhi, B.K. Pattanayak, A.K. Habboush, M. Alnabhan, E. Qalaja, A fog-enabled framework for ensemble machine learning-based real-time heart patient diagnosis, Int. J. Eng. Trends Technol. 71 (8) (2023) 39–47, http://dx.doi.org/10.14445/22315381/IJETT-V71I8P204.
[43] S.M. Rajagopal, S. M., R. Buyya, FedSDM: Federated learning based smart decision making module for ECG data in IoT integrated Edge–Fog–Cloud computing environments, Internet Things 22 (2023) 100784, http://dx.doi.org/10.1016/j.iot.2023.100784.
[44] S. Tuli, N. Basumatary, S.S. Gill, M. Kahani, R.C. Arya, G.S. Wander, R. Buyya, 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. 104 (2020) 187–200, http://dx.doi.org/10.1016/j.future.2019.10.043.

[45] A. Sethi, S. Dharmavaram, S.K. Somasundaram, Explainable artificial intelligence (XAI) approach to heart disease prediction, in: 2024 3rd International Conference on Artificial Intelligence for Internet of Things (AIIoT), IEEE, 2024, pp. 1–6, http://dx.doi.org/10.1109/AIIoT58432.2024.10574635.

[46] R. Dwivedi, D. Dave, H. Naik, S. Singhal, R. Omer, P. Patel, B. Qian, Z. Wen, T. Shah, G. Morgan, R. Ranjan, Explainable AI (XAI): Core ideas, techniques, and solutions, ACM Comput. Surv. 55 (9) (2023) http://dx.doi.org/10.1145/3561048.

[47] S. Shen, V. van Beek, A. Iosup, GWA-T-12 Bitbrains Dataset, Delft University of Technology, 2015, URL http://gwa.ewi.tudelft.nl/datasets/gwa-t-12-bitbrains. Statistical Characterization of Business-Critical Workloads Hosted in Cloud Datacenters, 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), 2015, ShenZhen, China..

[48] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.

[49] H. Chen, J. Miao, R. Wang, H. Li, X. Zhang, A hierarchical deep reinforcement learning approach for throughput maximization in reconfigurable intelligent surface-aided unmanned aerial vehicle–Integrated sensing and communication network, Drones 8 (12) (2024) 717, http://dx.doi.org/10.3390/drones8120717.

[50] CDC, CDC Diabetes Health Indicators, 2023, http://dx.doi.org/10.24432/C53919, UCI Machine Learning Repository.

[51] Centers for Disease Control and Prevention, CDC - 2014 BRFSS Survey Data and Documentation. URL https://www.cdc.gov/brfss/annual_data/annual_2014.html.

[52] V. Rattan, R. Mittal, J. Singh, V. Malik, Analyzing the application of SMOTE on machine learning classifiers, in: 2021 International Conference on Emerging Smart Computing and Informatics, ESCI, IEEE, 2021, pp. 692–695, http://dx.doi.org/10.1109/ESCI50559.2021.9396962.

[53] A. Inc., Activity monitor user guide for mac, 2024, URL https://support.apple.com/en-gb/guide/activity-monitor/welcome/mac. (Accessed: 2024-08-01).

[54] S.S. Gill, R. Buyya, A taxonomy and future directions for sustainable cloud computing: 360 degree view, ACM Comput. Surv. 51 (5) (2018) http://dx.doi.org/10.1145/3241038.

[55] V. Hayyolalam, A.A. Pourhaji Kazem, A systematic literature review on QoS-aware service composition and selection in cloud environment, J. Netw. Comput. Appl. 110 (2018) 52–74, http://dx.doi.org/10.1016/j.jnca.2018.03.003.

[56] A.K. Bardsiri, S.M. Hashemi, QoS metrics for cloud computing services evaluation, Int. J. Intell. Syst. Appl. 6 (12) (2014) 27–33, http://dx.doi.org/10.5815/ijisa.2014.12.04.

[57] M. Jelassi, C. Ghazel, L.A. Saidane, A survey on quality of service in cloud computing, in: 2017 3rd International Conference on Frontiers of Signal Processing, ICFSP, IEEE, 2017, pp. 63–67, http://dx.doi.org/10.1109/ICFSP.2017.8097142.

[58] J. Dodge, T. Prewitt, R. Tachet des Combes, E. Odmark, R. Schwartz, E. Strubell, A.S. Luccioni, N.A. Smith, N. DeCario, W. Buchanan, Measuring the Carbon Intensity of AI in Cloud Instances, in: 2022 ACM Conference on Fairness, Accountability, and Transparency, ACM, New York, NY, USA, 2022, pp. 1877–1894, http://dx.doi.org/10.1145/3531146.3533234.

[59] A. Kumar, N. Gaur, A. Nanthaamornphong, Improving the latency for 5G/B5G based smart healthcare connectivity in rural area, Sci. Rep. 14 (1) (2024) 6976, http://dx.doi.org/10.1038/s41598-024-57641-7.

[60] A. Gougeon, B. Camus, A.-C. Orgerie, Optimizing green energy consumption of fog computing architectures, in: 2020 IEEE 32nd International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD), 2020, pp. 75–82, http://dx.doi.org/10.1109/SBAC-PAD49847.2020.00021.

[61] G. Cloud, Carbon free energy for Google Cloud Regions, 2024, URL https://cloud.google.com/sustainability/region-carbon. (Accessed: 2024-08-01).

[62] G. Cloud, VM instance pricing | compute engine: Virtual machines (VMs), 2024, URL https://cloud.google.com/compute/vm-instance-pricing. (Accessed: 2024-08-11).

[63] T. Hernández-Del-Toro, F. Martínez-Santiago, A. Montejo-Ráez, Assessing classifier's performance, in: Biosignal Processing and Classification using Computational Learning and Intelligence, Elsevier, 2022, pp. 131–149, http://dx.doi.org/10.1016/B978-0-12-820125-1.00018-X.

[64] B.D. Parameshachari, K. Hemalatha, J. Rahebi, M.R. Baker, B. Upendra Roy, Health-care monitoring of patient using CNN based model in internet of things, in: 2023 International Conference on Applied Intelligence and Sustainable Computing, ICAISC, IEEE, 2023, pp. 1–6, http://dx.doi.org/10.1109/ICAISC58445.2023.10200455.

[65] P.N. Srinivasu, N. Sandhya, R. Jhaveri, R. Raut, From blackbox to explainable AI in healthcare: Existing tools and case studies, Mob. Inf. Syst. 2022 (2022) 1–20, http://dx.doi.org/10.1155/2022/8167821.

[66] D. Chowdhury, S. Poddar, S. Banarjee, R. Pal, A. Gani, C. Ellis, R. Arya, S.S. Gill, S. Uhlig, Covidxai : explainable AI assisted web application for covid -19 vaccine prioritization, Internet Technol. Lett. 5 (2022) http://dx.doi.org/10.1002/itl2.381.

[67] Z. Yu, J. Wang, W. Luo, R. Tse, G. Pau, MPRE: Multi-perspective patient representation extractor for disease prediction, in: 2023 IEEE International Conference on Data Mining, ICDM, 2023, pp. 758–767, http://dx.doi.org/10.1109/ICDM58522.2023.00085.

[68] C. Sudlow, J. Gallacher, N. Allen, V. Beral, P. Burton, J. Danesh, P. Downey, P. Elliott, J. Green, M. Landray, B. Liu, P. Matthews, G. Ong, J. Pell, A. Silman, A. Young, T. Sprosen, T. Peakman, R. Collins, UK Biobank: An open access resource for identifying the causes of a wide range of complex diseases of middle and old age, PLOS Med. 12 (3) (2015) e1001779, http://dx.doi.org/10.1371/journal.pmed.1001779.

[69] Z.N. Aghdam, A.M. Rahmani, M. Hosseinzadeh, The role of the internet of things in healthcare: Future trends and challenges, Comput. Methods Programs Biomed. 199 (2021) 105903, http://dx.doi.org/10.1016/j.cmpb.2020.105903.

[70] M.M. Kermani, M. Zhang, A. Raghunathan, N.K. Jha, Emerging frontiers in embedded security, in: 2013 26th International Conference on VLSI Design and 2013 12th International Conference on Embedded Systems, IEEE, 2013, pp. 203–208, http://dx.doi.org/10.1109/VLSID.2013.222.

[71] S. Bayat-Sarmadi, M. Mozaffari-Kermani, A. Reyhani-Masoleh, Efficient and concurrent reliable realization of the secure cryptographic SHA-3 algorithm, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 33 (7) (2014) 1105–1109.

[72] B. Koziel, R. Azarderakhsh, M. Mozaffari Kermani, D. Jao, Post-quantum cryptography on FPGA based on isogenies on elliptic curves, IEEE Trans. Circuits Syst. I. Regul. Pap. 64 (1) (2017) 86–99, http://dx.doi.org/10.1109/TCSI.2016.2611561.

[73] M. Mozaffari-Kermani, S. Sur-Kolay, A. Raghunathan, N.K. Jha, Systematic poisoning attacks on and defenses for machine learning in healthcare, IEEE J. Biomed. Heal. Informatics 19 (6) (2015) 1893–1905, http://dx.doi.org/10.1109/JBHI.2014.2344095.