BlockFaaS: Blockchain-enabled Serverless Computing Framework for AI-driven IoT Healthcare Applications

Muhammed Golec[1*], Sukhpal Singh Gill[2], Mustafa Golec[3], Minxian Xu[4], Soumya K. Ghosh[5], Salil S. Kanhere[6], Omer Rana[7], Steve Uhlig[2]

[1*] School of Electronic Engineering and Computer Science, Queen Mary University of London, United Kingdom and Abdullah Gul University, Kayseri, Turkey.
[2]School of Electronic Engineering and Computer Science, Queen Mary University of London, United Kingdom.
[3]Kütahya Dumlupınar University, Kütahya, Turkey.
[4]Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China.
[5]Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur, India.
[6]School of Computer Science and Engineering, The University of New South Wales (UNSW), Sydney, Australia.
[7]School of Computer Science and Informatics, Cardiff University, Cardiff, United Kingdom.

*Corresponding author(s). E-mail(s): m.golec@qmul.ac.uk;
Contributing authors: s.s.gill@qmul.ac.uk; mustafagolec36@gmail.com; mx.xu@siat.ac.cn; skg@cse.iitkgp.ac.in; salil.kanhere@unsw.edu.au; ranaof@cardiff.ac.uk; steve.uhlig@qmul.ac.uk;

Abstract

With the development of new sensor technologies, Internet of Things (IoT)-based healthcare applications have gained momentum in recent years. However, IoT devices have limited resources, making them incapable of executing large computational operations. To solve this problem, the serverless paradigm, with its

advantages such as dynamic scalability and infrastructure management, can be used to support the requirements of IoT-based applications. However, due to the heterogeneous structure of IoT, user trust must also be taken into account when providing this integration. This problem can be overcome by using a Blockchain that guarantees data immutability and ensures that any data generated by the IoT device is not modified. This paper proposes a BlockFaaS framework that supports dynamic scalability and guarantees security and privacy by integrating a serverless platform and Blockchain architecture into latency-sensitive Artificial Intelligence (AI)-based healthcare applications. To do this, we deployed the AIBLOCK framework, which guarantees data immutability in smart healthcare applications, into HealthFaaS, a serverless-based framework for heart disease risk detection. To expand this framework, we used high-performance AI models and a more efficient Blockchain module. We use the Transport Layer Security (TLS) protocol in all communication channels to ensure privacy within the framework. To validate the proposed framework, we compare its performance with the HealthFaaS and AIBLOCK frameworks. The results show that BlockFaaS outperforms HealthFaaS with an AUC of 4.79% and consumes 162.82 millijoules less energy on the Blockchain module than AIBLOCK. Additionally, the cold start latency value occurring in Google Cloud Platform, the serverless platform into which BlockFaaS is integrated, and the factors affecting this value are examined.

Keywords: Serverless Computing, Internet of Things, Healthcare, Privacy, Blockchain, AI.

1 Introduction

The Internet of Things (IoT) can provide the basis for a data exchange network that enables objects to work in a synchronous manner [1]. The data collected through sensors and actuators can be transferred to the cloud or other devices via a gateway. Moreover, the analysis results obtained using this data can help make decisions and support citizens within a smart city [2]. IoT use is spread over industrial, commercial, defence, and infrastructure scenarios [3]. One of these areas is healthcare, where applications can directly affect human lives. The general working mechanism of IoT-based healthcare applications is as follows: data is collected from patients using sensors and sent to a gateway via an IoT device. The gateway node is the middle layer that provides connectivity, manageability and security between the cloud and the IoT device [4]. In the final stage, the health data collected at the gateway is sent to the cloud for analysis and diagnostic studies [5].

1.1 Our Previous Works

In our previous IoT-based health studies, we developed three frameworks named iFaaSBus, AIBLOCK, and HealthFaaS, which detect COVID and Heart Disease Risk [6][7][8]. In these frameworks, health data is collected from users via IoT and sent to Machine Learning (ML) models deployed on serverless platforms. By detecting the condition, it is intended to avoid fatal instances and needless medical costs based on

the predictions generated by these ML models. ML models are applied to serverless platforms because, as the number of users using the frameworks increases, the amount of data that needs to be processed also increases. At the same time, system resources such as CPU and RAM are required for the training and prediction processes of ML models used in disease diagnosis. Considering that IoT devices generally consist of resource- and storage-limited devices, serverless computing is inspiring to meet these needs.

In all health applications, providing security and privacy to protect users' data, comply with legal regulations, and increase trust in applications is very important. JSON Web Token (JWT) and Open Authorization 2 (OAuth 2.0) are used to ensure user privacy and security within the framework of iFaaSBus. JWT has weaknesses such as key management, storage and verification of tokens [9]. However, OAuth 2 has weaknesses such as weak authentication and access token expiration [10]. Therefore, stronger security mechanisms are needed. Therefore, within the framework of AIBLOCK, Blockchain technology is used to ensure security, which guarantees the immutability of health data. On the other hand, in the HealthFaaS framework, no external method is used to ensure system security. Additionally, in both AIBLOCK and HealthFaaS frameworks, privacy is only provided in the IoT layer. Therefore, in this article, we propose a new framework called BlockFaaS to offer security and privacy by deploying the AIBLOCK framework to the HealthFaaS framework. We have used the XGB model in the BlockFaaS framework to get better ML prediction results than HealthFaaS and used the SHA3-224 hash function for higher-performance energy-sensitive blockchain transactions than AIBLOCK.

1.2 Motivation and Our Contributions

IoT devices are often insufficient in terms of resource and processing capacity (e.g., memory, processing, bandwidth and energy) [11]. Therefore, an architecture that can manage these limited resources is required to process the large amounts of data they generate [12]. With its high processing capacity and dynamic scalability feature, the serverless paradigm is an inspiring development for processing the massive amount of data produced in the IoT [6].In Serverless, there is no need to allocate resources that will be needed in advance. Instead, resources are automatically scaled according to the needs of the IoT, and with a pay-as-you-go pricing policy, the resources are charged based on their usage time.

User privacy is critical when integrating a serverless platform with the IoT. Due to the heterogeneous nature of IoT, it is difficult to ensure the integrity of data collected from patients, and this will cause a user privacy problem [13]. All communication channels between the serverless platform, the IoT device, and the database where patient data is stored are vulnerable to attack [14]. Therefore, in case of any attack on communication channels, patient data can be altered (immutability). Moreover, this may lead to severe consequences, such as misdiagnosis [15].Security and privacy concerns can be reduced by using Blockchain and Transport Layer Security (TLS). Blockchain technologies, which can be used everywhere, from healthcare to the financial sector and military technologies, guarantee data integrity and immutability with their unique architecture [16].

The main contributions of this paper are:

- We propose the BlockFaaS framework, which provides user privacy, security (data immutability), and dynamic scalability in a single framework for IoT-based healthcare applications,
- We implement a Blockchain-based and TLS-based architecture to guarantee security and privacy in IoT environments where data integrity is difficult to ensure due to their heterogeneous nature,
- We observe the effect of cold start delay caused by the serverless paradigm, which can cause problems in time-sensitive healthcare applications, and identify the most effective factors to reduce it,
- We compare the performance of BlockFaaS with AIBLOCK and HealthFaaS frameworks using the heart disease risk detection scenario,
- We evaluate the performance of BlockFaaS from three aspects: Blockchain energy cost and transaction time, ML prediction performance, and Quality of Service (QoS) parameters for the Serverless platform.

The remainder of the article is organized as follows: In section 2, some background information is explained to the reader. In section 3, related works are given and compared with the BlockFaaS framework. The section 4 describes the used data set and the architecture of the BlockFaaS framework. In section 5, a comprehensive performance evaluation of the framework is performed. Section 6 summarizes the work and highlights possible future directions.

2 BACKGROUND

The reader can better understand the BlockFaaS framework by reading through this section, which examines some concepts. First, serverless computing and its advantages are explained, followed by security and privacy concerns in the IoT, and in the last subsection, blockchain and TLS protocols are explained.

2.1 Serverless Computing

Modern cloud computing delivery models are terminologically examined under three main headings: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Function as a Service (FaaS) [17]. As it moves from IaaS to FaaS, infrastructure and server management are completely isolated from customers and undertaken by the cloud provider [18]. Serverless Computing, or in other words, Function as a Service (FaaS), does not point to a system where the server is not at all, contrary to popular belief [19]. Indicates that server management and other infrastructure operations are the service provider's responsibility [20].

Serverless computing attracts attention with its three advantages [21]:

- Customers only pay for the resources they use (Pay-as-you-go),
- Customers do not need to specify the resources they will use in advance. In case of a need for resources, the system automatically scales the resources,

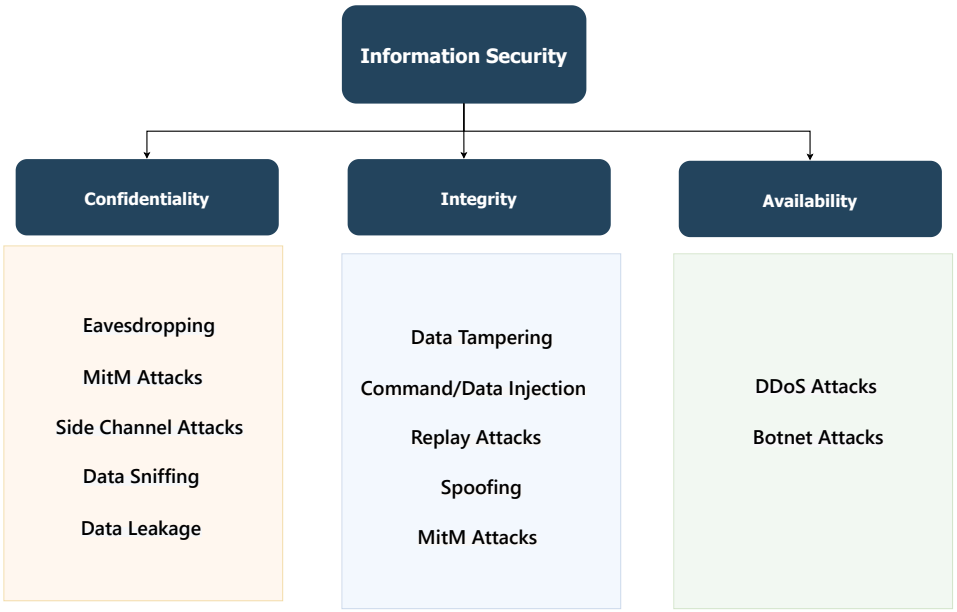

Fig. 1 CIA Triad in Information Security

- Server infrastructure management is the sole responsibility of the service provider. So, code developers only need to focus on business applications and coding.

2.2 Security & Privacy Concerns in IoT

IoTs usually consist of hardware-constrained devices, and it is difficult to apply security methods such as encryption to these devices, which require high processing power [22]. That's why it is vulnerable to many attacks [23]. The data transmitted in IoT may consist of various data such as weather, location, health data, and biometric data, depending on the type of application used. In case of unauthorized capture or modification of this data, a Confidentiality, Integrity, and Availability (CIA) violation in information security occurs [24]. CIA Triad is a general concept used to explain the core principles of information security [25]. Fig 1 shows these three concepts and attacks against the three concepts. These concepts can be explained as follows:

- **Confidentiality:** It refers to the fact that only authorized persons can access sensitive data held in communication channels and databases [26]. The most common attacks against Confidentiality are Eavesdropping, Man-in-the-Middle (MitM) Attacks, Side Channel Attacks, Data Sniffing, and Data Leakage. Any threat affecting the system's confidentiality will also cause a privacy problem [27]. Some measures

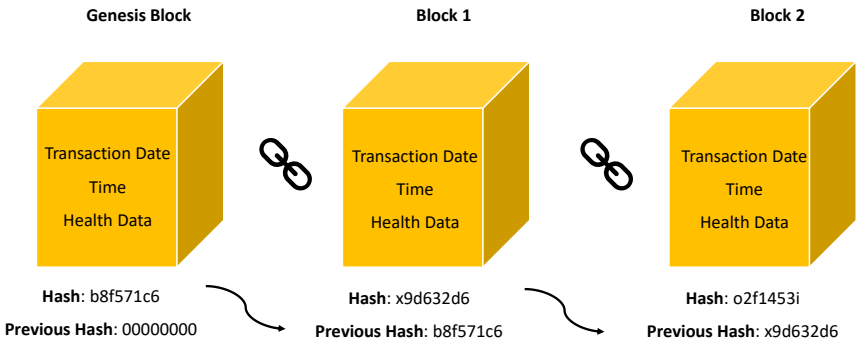

Fig. 2 Blockchain Mechanism

that can be taken to ensure confidentiality or privacy are encryption of communication channels and secure communication protocols (TLS Protocol, etc.) and user authentication (token, etc.) [28].

- **Integrity:** It refers to the modification or distortion of data held in communication channels and databases by unauthorized persons [24]. Data Tampering, Command/-Data Injection, Replay Attacks, and Spoofing are examples of the most common attacks against Integrity. Some measures that can be taken to ensure Integrity are Digital signatures and Blockchain [29].
- **Availability:** It refers to the constant availability of the information technology system [30]. Distributed Denial of Service (DDoS) and Botnet Attacks are examples of the most common attacks against Availability. Some measures that can be taken to ensure availability are the development of backup systems and infrastructure [31].

2.3 Blockchain & Transport Layer Security (TLS) Protocol

Blockchain emerged in the 21st century and is one of the most up-to-date technological innovations used in many fields, including IoT, Network Management, food security, and money transfers [32]. Blockchain working mechanism is given in Fig 2. The first block shown in the figure is called Genesis [33], and new blocks are added on top of Genesis. Blocks containing transaction date, time, and user data are digitally signed. After signing, the hash value is added to the next block. Each new block created is added to the end of the chain, combining itself and the hash of the block behind it. For this reason, a change in any of the blocks will affect the hash value of all blocks. Therefore, it is promising in providing data immutability of critical information such as health data in IoT-based health applications [34] .

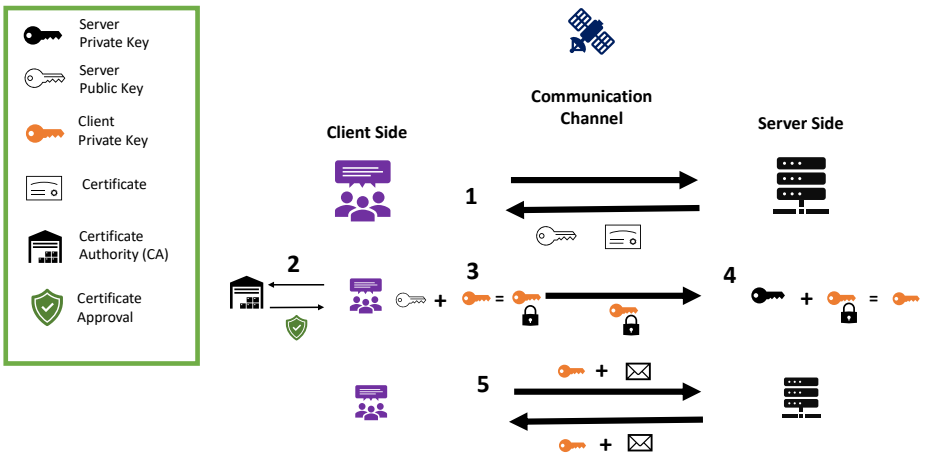

Fig. 3 TLS Protocol Mechanism

Blockchain technology can ensure the security of IoT-based healthcare systems against attacks such as MiTM by ensuring data integrity [7]. MiTM attacks are attacks in which data packets can be stolen and modified by eavesdropping on the communication channel between two targets [35]. This attack can cause many security problems in sensitive applications such as health applications and smart homes. Ahmed et al. [36] demonstrated that a MiTM attack could cause deaths in a healthcare scenario where automatic insulin dosage is adjusted for diabetics.

TLS protocol is a security protocol used to ensure the privacy of data transmitted in communication channels [37]. There are several versions still used today in web browsers and various applications. Fig 3 shows the working mechanism of the TLS protocol. (i) The client communicates with the server and receives the server certificate and the server public key. (ii) On the client side (internet browser), it checks the certificate received from the server in a Certificate Authority (CA). (iii) When this check is completed, the client creates a private key. The client's private key is encrypted with the server's public key and sent to the server. (iv) This data received by the server is decrypted with the server's private key. And it obtains the private key of the client. (v) The server and the client exchange data using the private key of the client in the communication between them (as explained in Step (iii)).

The TLS protocol can ensure the privacy of IoT-based healthcare systems against privacy attacks such as Eavesdropping [38]. Eavesdropping is an attack in which data is stolen by eavesdropping on communications between two devices. In healthcare applications where biometric data is used, this sensitive data can be stolen through Eavesdropping attacks. Biometric data are unique features of a person, such as fingerprints and facial recognition, and are used in various scenarios, from healthcare systems

to authentication methods [8] [39]. Since these data are personal and unchangeable, they will cause privacy problems if stolen [40].

Since the BlockFaaS framework provides encrypted protection with the TLS protocol on all communication channels, it is protected against external interventions. Thus, it guarantees to protect Internet of Things-based healthcare systems against privacy (confidentiality) attacks such as Eavesdropping, Side Channel Attacks, and Data Leakage. Additionally, the Blockchain module integrated into BlockFaaS ensures the integrity of patient data in transmission channels and databases. Thus, it guarantees to protect Internet of Things-based healthcare systems against security (Integrity) attacks such as MiTM, Data Tampering, Replay Attacks, and Spoofing.

3 Related Works

Taloba et al. [41] proposed a Blockchain-based architecture for multimedia data processing in IoT-Healthcare. In the study, they committed to protecting patients' security in real-time by integrating Blockchain and IoT. The proposed system against IoT attacks such as falsified assault and wormhole intrusion achieved a higher success rate of 86% compared to other studies. The authors of [42] presented an application that provides the security of healthcare documents using a Blockchain-based system. This application uses the Proof of Work (PoW) consensus algorithm to prevent fraudulent activity in the healthcare system. In their proposed study, Gupta et al. [43] detected patients' falls using the Convolutional Neural Network (CNN) model. Unlike other reviewed studies, they use Edge Computing, which has advantages such as lower latency and higher bandwidth than IoT. Balasundaram et al. [44] proposed a new Smart Healthcare System that detects health abnormalities using the U-Net and LSTM models. Multiple critical health data such as ECG and X-Ray are collected from patients via Multi-Model IoT (MMIoT) devices and transmitted to the server fastest with the 5G network. Golec et al. [6] proposed a new IoT-based healthcare system using K-Nearest Neighbor (KNN) to perform early COVID diagnosis. In this context, health data collected from users is sent to an ML model deployed on a serverless platform using JWT. According to the prediction results obtained in the ML model, it was aimed to reduce the spread of the disease and mortality rates by notifying the nearest health institution. The authors used the OAuth-2.0 authorization protocol to ensure user privacy. Cloud federations provide improved flexibility and performance by freeing users from dependence on a specific cloud provider. Doyle et al. [45] proposed a new framework that enables secure Virtual Machines (VM) in cloud federations to increase this flexibility and performance. The migration of VMs from one cloud provider to another is recorded using blockchain and HyperLedger solutions. In this way, it is aimed to prevent possible security breaches by guaranteeing data immutability. Additionally, in this study, the authors also ensure that SLA violations are recorded and charged. Golec et al. [7] proposed a new IoT-based framework that performs early COVID diagnosis in patients. They compared five ML models to identify the most successful ML model in diagnosing COVID. As a result of the comparison, they identified the ML model Decision Trees with the highest accuracy rate and deployed it on a serverless platform. The authors provided the processing

Table 1 Comparison of this paper with existing works. ×:= method does not support the property

Study	Diagnosis	Collection Type	Dynamic Scalability	Security	Privacy	AI
[41]	×	IoT	×	Blockchain	×	×
[42]	×	IoT	NO	Blockchain	×	×
[43]	Patients' Falls	IoT and Edge	×	×	×	CNN
[44]	Health Anomalies	MMIoT	×	×	×	U-Net LSTM
[45]	×	VM Logs	YES	Blockchain	×	×
[6]	COVID	IoT	YES	JSON Web Token	OAuth 2.0	KNN
[7]	COVID	IoT	YES	Blockchain	×	DTs
[46]	×	×	YES	×	×	×
[47]	×	IoMT	×	Blockchain	NuCypher Re-Encryption	×
[8]	Heart Disease Risk	IoT	YES	×	×	LightGBM
BlockFaaS (this paper)	Heart Disease Risk	IoT	YES	Blockchain	TLS	LightGBM

power and storage required for IoT thanks to the dynamic scalability feature of the serverless platform. Additionally, the authors used a Blockchain module to secure the framework. Apostolopoulos et al. [46] introduced a new resource-sharing paradigm for social cloud computing that offers both VM and serverless functions. In this paradigm, a high-yield, low-complexity algorithm is introduced that observes Pure Nash Equilibrium and considers user satisfaction. The authors demonstrated the performance of the framework they introduced using simulation. In [47], the authors proposed a hyperledger-based framework called BIoMT. This framework ensures effective resource consumption and security in Internet of Medical Things (IoMT) environments. At the same time, the NuCypher Re-Encryption method is used to ensure user privacy. Golec et al. [8] proposed a new IoT-based framework to reduce heart disease, one of the world's deadliest diseases, and the economic damage it causes. For this, they deployed the LightGBM model on Google Cloud Functions, a serverless platform. They used well-known feature selection methods in the literature to increase the prediction performance of the risk of developing heart disease. Additionally, they identified the cold start latency occurring in the serverless paradigm for latency-sensitive applications and the parameters affecting this time. They compared the performance of serverless and non-serverless platforms against the increasing number of users.

Table 1 clearly shows the differences between the existing literature and our proposed BlockFaaS framework. None of the existing works have considered dynamic scalability, security, and privacy in IoT-based healthcare applications simultaneously except iFaaSBus framework [6], which used JWT to provide security, however, iFaaSBus framework has several weaknesses, such as key management and storage and verification of tokens. To solve these issues, we propose a new framework called BlockFaaS to improve security using blockchain.

4 METHODOLOGY

This section explains the equations used when calculating energy cost and the cold start latency in subsection 4.1. Then, in subsections 4.2 & 4.3, BlockFaaS framework, including AI and Blockchain modules in the framework, are discussed in detail by showing them with pseudocodes. The flowchart describes different interactions in BlockFaaS and is given in subsection 4.4. In the section 4.5, the dataset used in the heart disease risk scenario to be integrated into BlockFaaS is explained.

4.1 Problem Formulation

In this subsection, we explain the formulas that we use for calculating the performance and energy cost of Blockchain's hash functions [48] and the cold start latency equation.

The following equation is used to calculate the energy cost (ϵ_{cost}) of the hash functions used in the Blockchain module.

$$\epsilon_{cost} = \rho \times t \tag{1}$$

Here ρ represents the power drawn by the processor. The actual power consumed by CPUs under load is not stable due to variables such as workload, processor operating mode, etc., and is therefore difficult to determine. Therefore, a suitable power target can be achieved using Thermal Design Power (TDP) [49]. TDP measures the amount of heat produced by components such as the CPU and GPU, expressed in Watts [50]. The TDP values drawn by the platforms can be accessed from the references [51][52]. Platforms to simulate serverless instance environments will be explained in detail later in subsection 5.1. The time t in the equation represents the time spent on the hashing process.

The following formula is used to calculate the cold start latency ($C_{Latency}$) occurring on the serverless platform. Here, ι_{first} is the latency calculated for the first request, and ι_{second} is the latency calculated for the second request. The initial request sent in sequence from the client to the server is referred to as the first request, while the subsequent request is denoted as the second.

$$C_{Latency} = \iota_{first} - \iota_{second} \tag{2}$$

4.2 BlockFaaS Framework

Figure 4 shows the general working mechanism of BlockFaaS. The steps are numbered sequentially to make it easier for the reader to follow the mechanism diagram. Using sensors and actuators, BlockFaaS sends 13 variables collected from patients via IoT to the gateway (1) and keeps copies of them in the gateway database (DB) (2). The gateway node is the layer that provides the connection between the cloud and the IoT device, and it is assumed to be a mobile device in this study. The health data collected at the gateway is sent to the AI model in the Serverless layer, and it is determined whether the patient has a risk of heart disease (3). A hash value is obtained by sending the prediction result from the AI and the patient health data to the Blockchain module (4). This hash value is stored in the serverless platform's DB (5). In the next step,

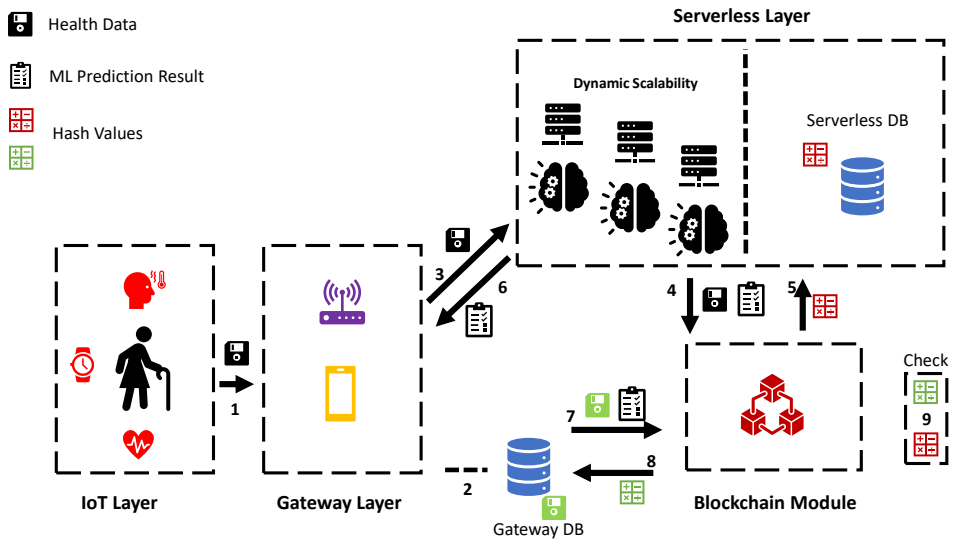

Fig. 4 BlockFaaS Framework.

the result from the AI model is sent to the gateway (6). The data previously stored in the gateway DB and the prediction result from the serverless module are sent to the blockchain module, and a new hash value is obtained (7, 8). This newly obtained hash value is compared with the hash value in the serverless module (9). If the two hashes are incompatible, the data has been tampered with. These steps continue for each patient. The newly created hash values are added to each other and stored in the DB of the serverless module. Further, every newly created hash value is subject to the same control mechanism, ensuring data immutability and user privacy. Because, if any health data is changed, all hash values will be affected following the working logic of the Blockchain. The Blockchain module used within the BlockFaaS framework is the SHA-3 Keccak version [53], which is accepted as the crypto standard in the USA and is also used in Ethereum. SHA-3 Keccak has advantages such as being resistant to length extension attacks and faster hashing (performance) than SHA-256 [54]. XGB ML model is used in the BlockFaaS framework AI module. To decide on the ML model to be used, four different models are compared by performance. More details are given in section 5.

4.3 Algorithms and Complexity Analysis

Algorithm 1 and Algorithm 2 show the pseudo-codes of the AI module and Blockchain modules deployed on the serverless platform, respectively. First (Algo 1), health data (H_D) to be sent to the AI module is collected via an IoT device and sensors. The collected H_D is transmitted to the gateway node that acts as the middle layer. The gateway node can be used as a user interface to display AI results to the user for new studies in the future. A copy of H_D transmitted to the gateway node is stored

in the gateway device's DB. H_D is sent to the AI module, that is, to the ML model deployed on the serverless platform, with the API created using Python Flask. The prediction result (P_R) and H_D created in the ML model are sent to the Blockchain module. In the second stage (Algo 2), P_R and H_D data are sent to the Blockchain module. These two data are combined (Data). First, the first block of the blockchain, Genesis, is created. Then, for each element of the combined Data, a new node (N) is obtained by using the time step (T_s), data, and the hash value of the previous block, respectively. The blockchain is created (C_f) by adding the Nvalues obtained for each data to the end of the block. The same process is repeated using H_D stored in the DB of the gateway device to obtain a new blockchain (C_s). In the control phase, these two blockchain values are checked by comparing them. In this way, integrity (security) is ensured for H_D. It should be noted that TLS protocol is used in all communication channels to ensure patients' privacy.

Algorithm 1 The BlockFaaS AI Module

Input: $\mathbf{H_{Di}}$
Output: P_R

Begin
    ⇀ IoT Layer ↽
        Send H_{Di}

    ⇀ Gateway Layer ↽
        $\sum_0^i H_{Di} \hookrightarrow H_D$
        Send H_D

    ⇀ AI Module ↽
        Return P_R
End

Time Complexity Analysis: Since there is no loop for Algo 1 and it only consists of commands such as send and return, the time complexity is O(1). In Algorithm 2, the time complexity value is O(n) because the for loop is used to obtain the hash value for each health data.

4.4 Flowchart

Fig 5 describes different interactions in BlockFaaS. First, preparation steps are applied for the dataset to be used in heart disease risk detection. In the second step, four ML algorithms are trained. In the third step, ML prediction results and ML performance results are compared. In the last step, performance evaluation is made by measuring the additional load that Blockchain brings to the system.

Algorithm 2 The BlockFaaS Blockchain Module

Input: $Data = H_{Di} + P_R$
Output: H

Begin
    #Blockchain Creation Phase#
    Create genesisBlock
    C_f = genesisBlock
    for data in Data:
        N = Block(T_s, data, C_f[-1].hash)
        C_f.append(N)
    #Control Phase#
    if $C_f == C_s$:
        Return Matched
End

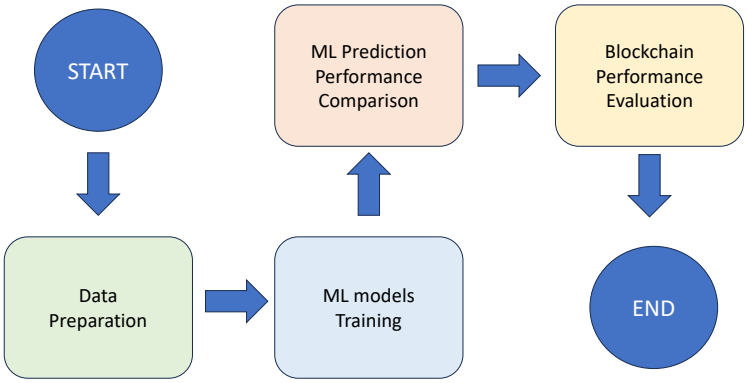

Fig. 5 Flowchart describes different interactions in BlockFaaS

4.5 Data Description

A dataset showing the risk of heart disease with 14 variables is used in this paper [55]. The dataset contains the following data for 303 patients: "age", "sex", "cp", "trestbps", "chol", "fbs", "restecg", "thalach", "exang", "oldpeak", "slope", "ca", "thal", and "target". These data represent, respectively, the patient's age, gender, chest pain symptom, blood pressure, blood glucose, cardiac electrocardiogram (ECG), heart rate, cardiac muscle ischemia, ST depression of the heart, slope ratio of the heart, number of blood vessels, thallium test result, and heart disease risk.

Table 2 System Configurations

Specifications	Platform 1	Platform 2
CPU	Intel Core i7-10750H	Intel Core i7-1165G7
Clock Speed	12M Cache, up to 5.00 GHz	12M Cache, up to 4.70 GHz
RAM	16 GB	32 GB
OS	Windows 10 Pro	Windows 11 Pro
TDP (W)	45	28

5 Performance Evaluation

In this section, we evaluate the performance of the BlockFaaS framework from three different aspects. In the first part, we measure the external load that the Blockchain module brings to the framework. Next, we benchmark the Blockchain modules used in the BlockFaaS and AIBLOCK frameworks in terms of performance and energy consumption. In the second part, we compare four different ML models to find the most effective ML model for the heart disease risk determination scenario. We compare HealthFaaS and BlockFaaS frameworks based on AUC in the heart disease risk determination scenario to extend the experiments and make more accurate performance comparisons. In the last part, we measure the QoS parameters of the serverless platform used in the BlockFaaS framework. We examine the factors that affect the cold start latency that occurs in serverless.

5.1 Experimental Setup

We use two different system configurations throughout the paper to simulate serverless instance environments and perform performance experiments. The system configurations are shown in Table 2. In BlockFaaS, Google Cloud Platform is used as a serverless platform [56] and the environment information for the GCP-Cloud Functions is as follows: Region: "europe-west1", Runtime: "Python 3.11" and RAM allocated: "512 MB".

5.2 Workloads

Platform 1 and Platform 2 are used to perform performance experiments of The Blockchain module (Hash functions). Our purpose in experimenting using two different platforms is to observe the load that Blockchain brings to the system under different system configurations. We use Platform 1 as only one system will be sufficient to evaluate the performance of The AI module. We create a workload using Apache-JMeter to measure the performance of the serverless platform on which The BlockFaaS framework is deployed [57]. To measure the response of the serverless platform under different workloads and to simulate an increasing number of users, we send different numbers of concurrent requests (100,200,..,1000) using JMeter.

5.3 Performance Metrics

While evaluating the performance of the BlockFaaS framework, the following metrics are used respectively.

- **Throughput**: It calculates the number of bits transmitted per second. It is expressed in units of kbit/second or bit/second.
- **Cold Start Latency**: In serverless computing, idle resources are terminated (Scale to Zero) to avoid unnecessary resource waste [14]. There is a certain delay in getting these resources ready for reuse in case of need. This delay is called cold start latency. Cold start is an undesirable delay in IoT-based healthcare systems such as patient monitoring.
- **Average Response Rate (ARR)**: It is known as the average response time between the Client-Server. It is one of the metrics used to measure cloud performance.
- **Blockchain Load**: Instant CPU-RAM usage percentage and processing times are used to measure the amount of additional load that the Blockchain module brings to the system.
- **Performance of AI Models**: Accuracy, Precision, Recall and F-Score are be used to measure the performance of AI models, respectively.

5.4 Experimental Results

In this section, all experiments and test results for the BlockFaaS framework are explained.

5.4.1 Measuring the impact of Blockchain

In this subsection, we compare the performance and energy consumption of Blockchain modules used in BlockFaaS and AIBLOCK frameworks. While the SHA3-224 hash function is used in the Blockchain module for the BlockFaaS framework, while the SHA256 hash function is used in the Blockchain module for the AIBLOCK framework. Fig 6 shows the performance results of two different SHA functions on two different platforms. Each experiment is performed 10 times, and the resulting average processing time is calculated using Eq1 1. As can be seen from the figure, the SHA3 (Keccak) function works much faster than SHA2.

Fig 7 shows the energy cost results of two different SHA functions on two different platforms. Each experiment is performed 10 times, and the amount of energy consumed per byte is shown using Eq2 2. It can be seen that the SHA3-256 function consumes less energy than SHA256. As a result, it is seen that the Blockchain Module used in the BlockFaaS framework works with higher performance and lower energy.

Three different parameters were used to calculate the overhead that the Blockchain module brings to the BlockFaaS framework. These are the CPU and RAM usage percentage of Platform 1, where the module is installed, and the processing time required for the blocks to be linked together. Only the Blockchain module is taken into account during our measurement, not the AI module. Table 3 shows the results. Block Size represents patients' health data, i.e. the number of patients using the framework. The Platform's CPU and RAM usage percentages while idle are 1% and 44.10% respectively. While CPU usage and processing time increase in direct proportion to the block size, the increase in RAM usage is negligible.

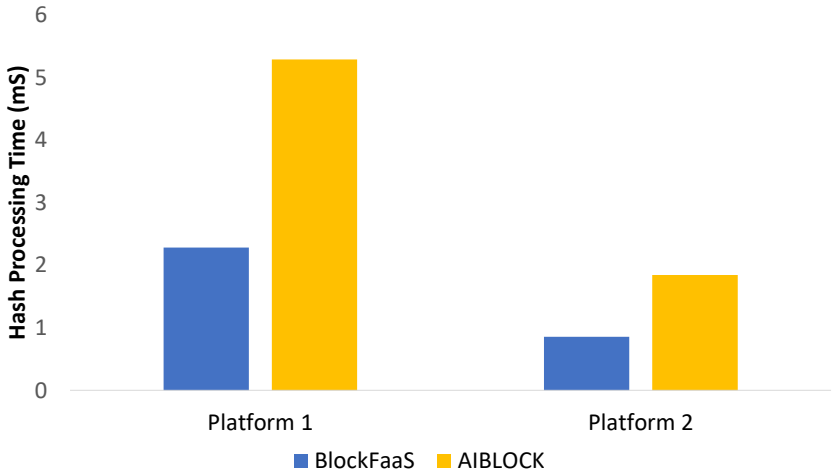

Fig. 6 Hash Performances of SHA Functions on Two Different Platforms

Table 3 Blockchain Module External Load for BlockFaaS

Block Size	CPU (%)	RAM (%)	Process Time (s)
100	4	44.00	0.056
300	6	44.40	0.212
500	7	44.40	0.385
1000	10	44.40	0.791
2000	14	44.50	1.44

5.4.2 Measuring the impact of AI-based prediction

In the BlockFaaS framework, we compare LR, KNN, XGB and CART using the units Accuracy, Precision, Recall, and F-Score, to find the most effective ML model in heart disease risk prediction. In the first step, we subjected the dataset to pre-processing for a better accuracy rate. First, variables whose correlation relationship with the target variable was less than 0.1 were removed from the dataset ("chol", "fbs"). Then, the dataset was normalized. This process is important for ML algorithms to achieve better convergence and higher performance. The first nine variables with the highest correlation were determined using the Inova feature selection method. Four different AI models, compared depending on Accuracy, Precision, Recall, and F-score parameters, are given in Table 4, respectively. The results showed that the most successful model was LR, with an Accuracy rate of 85.24%, and the most unsuccessful model was CART, with an accuracy rate of 72.13%.

In addition to using the above performance metrics when evaluating ML models, Area Under Curve (AUC) should also be taken into account. Because, while accuracy

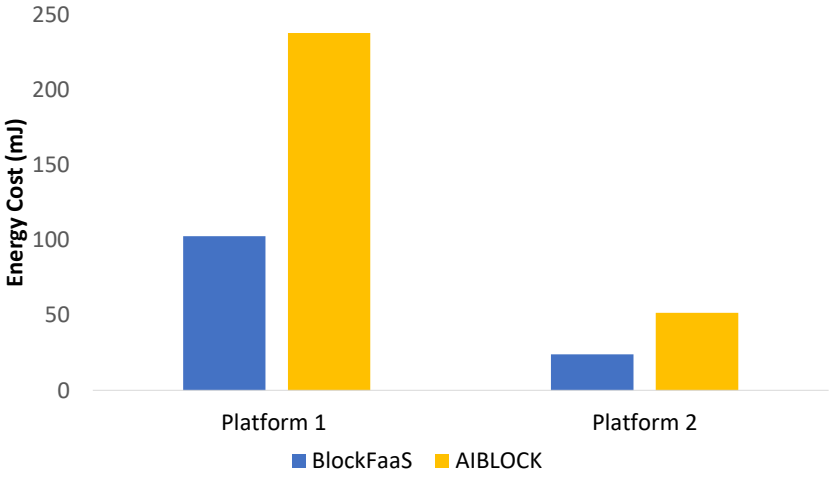

Fig. 7 Energy Cost of SHA Functions on Two Different Platforms

Table 4 Comparison of AI performances using BlockFaaS

Model	Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)
LR (BlockFaaS V.1)	85.24	83.8	91.2	87.3
KNN (BlockFaaS V.2)	83.6	81.6	91.2	86.1
XGB (BlockFaaS V.3)	80.32	82.4	82.4	82.4
CART (BlockFaaS V.4)	72.13	77.4	70.6	73.8

rates only explain the overall performance of a model, the AUC value explains the performance of the same model at different thresholds (positive and negative classes). Therefore, the AUC is resistant to imbalances between majority and minority classes. We use the AUC curve to determine the ML model used in BlockFaaS's AI module and to make a more accurate comparison with HealthFaaS. For BlockFaaS, we selected all the above ML models, while for HealthFaaS, we chose only the LightGBM model with the highest accuracy. Fig 8 shows the performance comparison for different ML models in terms of AUC. As can be seen, the model with the highest AUC is XGB with 86.71%. Therefore, the XGB model is used in the BlockFaaS model to obtain more accurate prediction results. AUC results show that BlockFaaS has better performance than HealthFaaS in terms of AUC.

5.4.3 Measuring the Scalability Performance

We used GCP Cloud Functions, a serverless platform within the BlockFaaS framework, and environment configuration in subsection 5.1. GCP Cloud Functions delivers the resources needed in IoT-based healthcare applications with dynamic scalability for

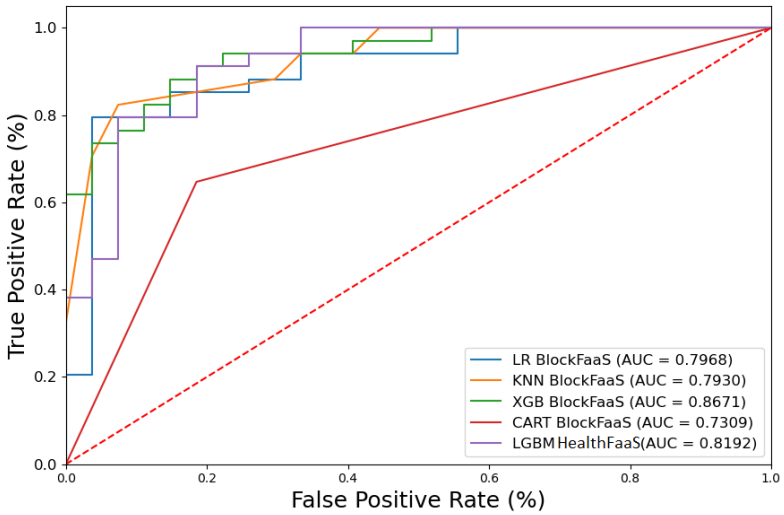

Fig. 8 Performance Comparison for different ML Models in terms of AUC

growing numbers of users and large computing operations. To test the scalability performance, we use QoS parameters such as Throughput, Average Response Rate (ARR), and Cold Start Latency in the heart disease risk scenario mentioned earlier. Then, we determine the cold start latency that occurs on the serverless platform and the ML model with the lowest cold start latency to minimize this latency. In the last experiment, we determined the factors that affect the cold start delay in Serverless computing.

Fig 9 shows the change in throughput (Serverless_Throughput) and average response rate (Serverless_ARR) of the serverless platform in response to the increasing number of users. An increasing number of concurrent requests (NCR) were sent to the ML model on the server via JMeter to represent the number of users using the system. Each HTTP request contains 11 health data used in the heart disease risk scenario. In direct proportion to the increasing amount of NCR, the bit/s rates in the transmission channels also increase. Therefore, the throughput value tends to increase continuously. When the NCR reaches 477, the Throughput value reaches its maximum and gradually decreases. This is because of resource contention that can occur when using the server's common resources (RAM, cache, etc.). When the ARR, another evaluation criterion, is examined, it is seen that the ARR increases with the increase in NCR. However, an exception is when 100 NCRs sent to the server have an ARR higher than 200 NCRs. The reason for this exception is the cold boot delay due to the serverless paradigm. This delay was measured as 30 milliseconds. Cold start latency is often an undesirable problem in time-sensitive IoT-based healthcare applications. To minimize this problem, it is important to identify the AI model with the lowest cold start latency. In our scenario, we deployed four different AI models to the server. And the cold start latency obtained for each is given in Figure 10. The results showed that the model with the highest cold start latency was XGB with 1324 mS, while the model with the least cold start latency was CART with 1197 mS.

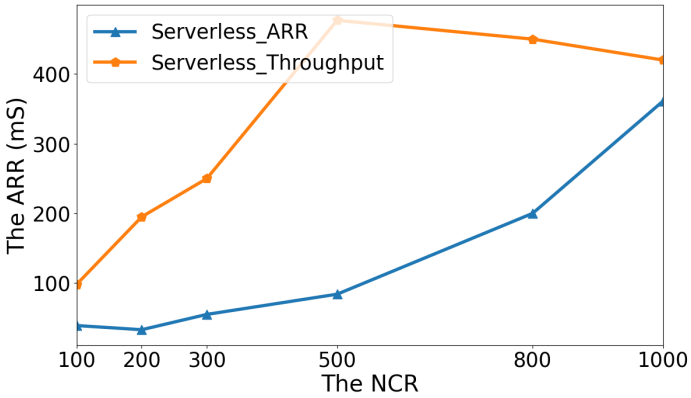

Fig. 9 The Scalability Performance of Serverless Platform.

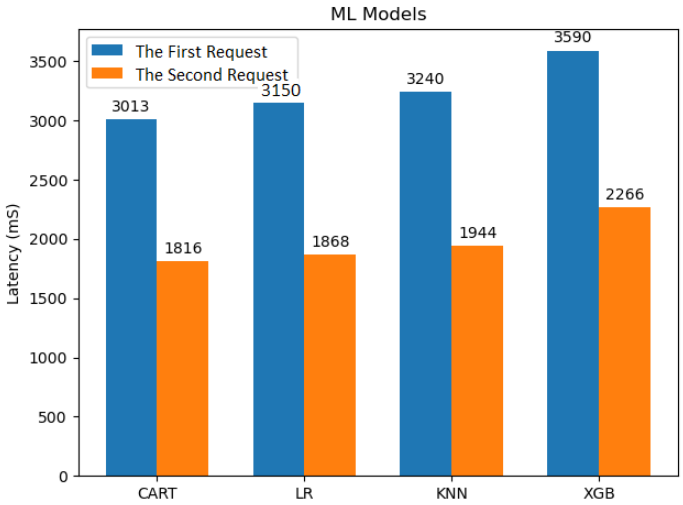

Fig. 10 Cold Start Latency for different ML Models.

5.4.4 Measuring the impact of Cold Start Latency

Three experiments were conducted to observe the factors affecting cold start latency in serverless computing. These experiments are the cold start latency rates corresponding to the RAM amount, software language, and package amount change. In the first experiment, we run a simple code script that returns "Hello World" to the screen using the Python 3.11 version. To observe the effect of the amount of RAM on a cold start, we use function instances with 256, 512, and 1024 RAM. As seen in Fig 11, the cold start latency rate also decreases as the amount of RAM increases. We repeat the same script in the second experiment using Python 3.11, Java 17 and Go language. It should be noted that the function instance in all three programming languages has 256 RAM. Fig 12 shows the cold start latency change according to the software languages.

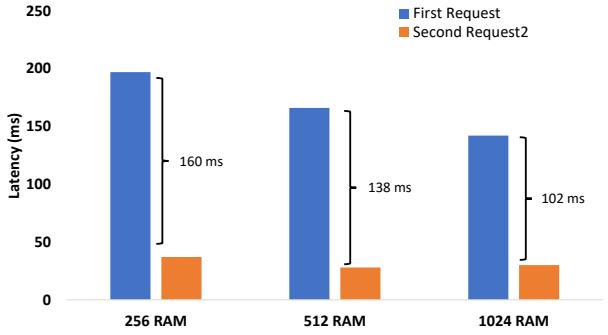

Fig. 11 The Impact of Different Main Memory Capacity (RAM) on Cold Start Latency.

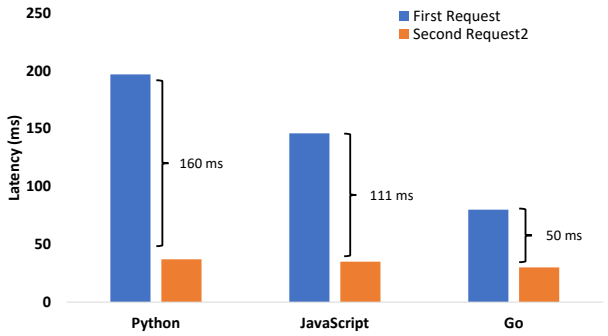

Fig. 12 The Impact of Different Software Languages on Cold Start Latency

Since the Go language is a compiled language, it is much faster than interpreted languages such as Python and Javascript and, therefore has a lower cold start latency (0.66 seconds). The reason why Javascript (1.18 seconds) is faster than Python (1.60) may be that it has highly optimized engines. As a result, we found that each software language has a different cold start latency. In the last experiment, we sent packets of varying sizes to a script written in Python. These package sizes are 1 kb, 15 Mb, and 40 Mb, respectively. Fig 13 shows the cold start latency change according to the varying sizes of packets. The cold start latency times obtained for each package are 1.70 s, 2.66, and 4.78 seconds, respectively. The results show that dependencies such as libraries and external files required for the execution of functions affect the cold start latency time.

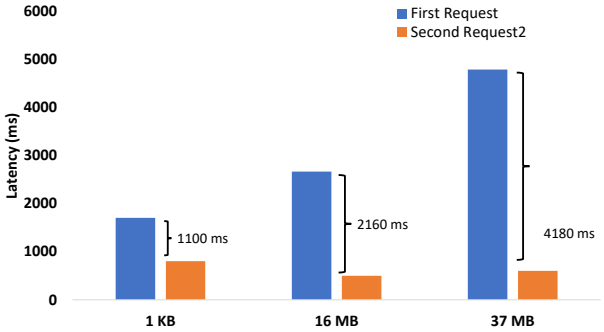

Fig. 13 The Impact of Uploading Different File Sizes (MB) on Cold Start Latency

6 Conclusions and Future Work

With the emerging use of IoT-based healthcare applications, resource-constrained IoT devices may be insufficient to process user-generated data. In addition, due to the heterogeneous nature of IoT devices, data integrity may limit a user's trust in decisions made using such data. This study proposes BlockFaaS, a framework designed with these vulnerabilities of IoT in mind. BlockFaaS automatically scales resources using a serverless platform and provides the external processing power required for IoT-based healthcare applications. In the BlockFaaS framework, a Blockchain module has been developed to address concerns such as user trust in IoT-based healthcare applications. This is ensured by guaranteeing data integration in transmission channels and databases. We have tested BlockFaaS from three aspects of a healthcare application scenario that detects heart disease risk. We tested BlockFaaS in three aspects of a healthcare application scenario detecting heart disease risk. First, the most successful model is determined by comparing the performances of 4 different artificial intelligence models in the heart disease risk detection scenario. Additionally, its prediction performance has been compared with the literature work HealthFaaS. Secondly, the performance and energy cost values of the Blockchain module used in BlockFaaS are compared with the literature study AIBLOCK. The last experiment measures QoS parameters such as throughput and cold start for the Google Cloud Functions serverless platform used in BlockFaaS. Additionally, factors affecting cold start latency are identified.

6.1 Promising Future Directions

In the BlockFaaS framework, it was assumed that health data was obtained through sensors and an IoT device. The study can be extended in the future by using a real IoT device, such as a wearable device [34]. Additionally, users can be informed by writing a mobile application for the BlockFaaS framework in the following studies [19]. Accuracy rates can be increased using advanced ML/DL models for the heart

disease risk detection scenario [23]. Eliminating cold start latency in the BlockFaaS framework is critical for time-sensitive IoT healthcare applications, an aspect that we aim to improve in future versions of BlockFaaS.

Acknowledgements

Muhammed Golec would express his thanks to the Ministry of Education of the Turkish Republic for their support and funding. This work is partially funded by Chinese Academy of Sciences President's International Fellowship Initiative (Grant No. 2023VTC0006). The authors would like to thank the Editor-in-Chief, area editor and anonymous reviewers for their valuable comments and helpful suggestions to improve the quality of the paper.

Abbreviations

Table 5 shows the list of acronyms.

Table 5 List of Acronyms

Abbreviation	Description
IoT	Internet of Things
TLS	Transport Layer Security
ML	Machine Learning
AI	Artificial Intelligence
PoW	Proof of Work
CNN	Convolutional Neural Network
MMIoT	Multi-Model IoT
IaaS	Infrastructure as a Service
PaaS	Platform as a Service
FaaS	Function as a Service
ARR	Average Response Rate
ECG	Cardiac Electrocardiogram
DB	Database
GCP	Google Cloud Platform

Declarations

- Funding
  This work is partially funded by Chinese Academy of Sciences President's International Fellowship Initiative (Grant No. 2023VTC0006)
- Conflict of interest/Competing interests
  On behalf of all authors, the corresponding author states that there is no conflict of interest.
- Ethics approval
  Not Available
- Consent to participate
  Not Available

- Consent for publication
  Not Available
- Availability of data and materials
  Not Available
- Code availability
  Not Available
- Authors' contributions
  Muhammed Golec (Conceptualization: Lead; Data curation: Lead; Formal analysis: Lead; Funding acquisition: Lead; Investigation: Lead; Methodology: Lead; Software: Lead; Validation: Lead; Writing – original draft: Lead) **Sukhpal Singh Gill** (Conceptualization: Lead; Data curation: Lead; Formal analysis: Lead; Funding acquisition: Lead; Investigation: Lead; Methodology: Lead; Software: Lead; Validation: Lead; Writing – original draft: Lead) **Mustafa Golec** (Conceptualization: Lead; Data curation: Lead; Formal analysis: Lead; Investigation: Lead; Methodology: Lead; Software: Lead; Validation: Lead; Writing – original draft: Lead) **Minxian Xu** (Conceptualization: Lead; Formal analysis: Lead; Funding acquisition: Lead; Investigation: Lead; Methodology: Lead; Writing – original draft: Lead) **Soumya K. Ghosh** (Supervision: Supporting; ; Writing – original draft: Lead; Writing – review & editing: Supporting) **Salil S. Kanhere** (Supervision: Supporting; ; Writing – original draft: Lead; Writing – review & editing: Supporting) **Omer Rana** (Supervision: Supporting; ; Writing – original draft: Lead; Writing – review & editing: Supporting) **Steve Uhlig** (Supervision: Supporting; ; Writing – original draft: Lead; Writing – review & editing: Supporting)

References

[1] Das, S.K., Benkhelifa, F., Sun, Y., Abumarshoud, H., Abbasi, Q.H., Imran, M.A., Mohjazi, L.: Comprehensive review on ml-based ris-enhanced iot systems: Basics, research progress and future challenges. Computer Networks, 109581 (2023)

[2] Dian, F.J., Vahidnia, R., Rahmati, A.: Wearables and the internet of things (iot), applications, opportunities, and challenges: A survey. IEEE access **8**, 69200–69211 (2020)

[3] Shanthamallu, U.S., Spanias, A., Tepedelenlioglu, C., Stanley, M.: A brief survey of machine learning methods and their sensor and iot applications. In: 2017 8th International Conference on Information, Intelligence, Systems & Applications (IISA), pp. 1–8 (2017). IEEE

[4] Kumar, M., *et al.*: Blockchain inspired secure and reliable data exchange architecture for cyber-physical healthcare system 4.0. Internet of Things and Cyber-Physical Systems **3**, 309–322 (2023)

[5] Gill, S.S., Tuli, S., Xu, M., Singh, I., Singh, K.V., Lindsay, D., Tuli, S., Smirnova, D., Singh, M., Jain, U., *et al.*: Transformative effects of iot, blockchain and artificial intelligence on cloud computing: Evolution, vision, trends and open

challenges. Internet of Things **8**, 100118 (2019)

[6] Golec, M., Ozturac, R., Pooranian, Z., Gill, S.S., Buyya, R.: Ifaasbus: A security- and privacy-based lightweight framework for serverless computing using iot and machine learning. IEEE Transactions on Industrial Informatics **18**(5), 3522–3529 (2021)

[7] Golec, M., Chowdhury, D., Jaglan, S., Gill, S.S., Uhlig, S.: Aiblock: Blockchain based lightweight framework for serverless computing using ai. In: 2022 22nd IEEE International Symposium on Cluster, Cloud and Internet Computing (CCGrid), pp. 886–892 (2022). IEEE

[8] Golec, M., Gill, S.S., Parlikad, A.K., Uhlig, S.: Healthfaas: Ai based smart healthcare system for heart patients using serverless computing. IEEE Internet of Things Journal (2023)

[9] Madden, N.: Misuse-resistant cryptography for jose/jwt. Position paper (2018)

[10] Kubovy, J., Huber, C., Jäger, M., Küng, J.: A secure token-based communication for authentication and authorization servers. In: Future Data and Security Engineering: Third International Conference, FDSE 2016, Can Tho City, Vietnam, November 23-25, 2016, Proceedings 3, pp. 237–250 (2016). Springer

[11] Zahoor, S., Mir, R.N.: Resource management in pervasive internet of things: A survey. Journal of King Saud University-Computer and Information Sciences **33**(8), 921–935 (2021)

[12] Samriya, J.K., et al.: Secured data offloading using reinforcement learning and markov decision process in mobile edge computing. International Journal of Network Management, 2243

[13] Liu, B., Yu, X.L., Chen, S., Xu, X., Zhu, L.: Blockchain based data integrity service framework for iot data. In: 2017 IEEE International Conference on Web Services (ICWS), pp. 468–475 (2017). IEEE

[14] Kumar, R., et al.: A robust and secure user authentication scheme based on multifactor and multi-gateway in iot enabled sensor networks. Security and Privacy, 335

[15] Liu, Y., Yu, J., Fan, J., Vijayakumar, P., Chang, V.: Achieving privacy-preserving dsse for intelligent iot healthcare system. IEEE Transactions on Industrial Informatics **18**(3), 2010–2020 (2021)

[16] Zhang, R., Xue, R., Liu, L.: Security and privacy on blockchain. ACM Computing Surveys (CSUR) **52**(3), 1–34 (2019)

[17] Singh, S., Chana, I., Singh, M.: The journey of qos-aware autonomic cloud

computing. It Professional **19**(2), 42–49 (2017)

[18] Singh, S., Chana, I.: A survey on resource scheduling in cloud computing: Issues and challenges. Journal of grid computing **14**, 217–264 (2016)

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

[20] Apostolopoulos, P.A., Tsiropoulou, E.E., Papavassiliou, S.: Risk-aware social cloud computing based on serverless computing model. In: 2019 IEEE Global Communications Conference (GLOBECOM), pp. 1–6 (2019). IEEE

[21] Cicconetti, C., Conti, M., Passarella, A., Sabella, D.: Toward distributed computing environments with serverless solutions in edge systems. IEEE Communications Magazine **58**(3), 40–46 (2020)

[22] Suo, H., Wan, J., Zou, C., Liu, J.: Security in the internet of things: a review. In: 2012 International Conference on Computer Science and Electronics Engineering, vol. 3, pp. 648–651 (2012). IEEE

[23] Iftikhar, S., *et al.*: Ai-based fog and edge computing: A systematic review, taxonomy and future directions. Internet of Things **21**, 100674 (2022)

[24] Samonas, S., Coss, D.: The cia strikes back: Redefining confidentiality, integrity and availability in security. Journal of Information System Security **10**(3) (2014)

[25] Yeboah-Boateng, E.O.: Cyber-security Challenges with Smes in Developing Economies: Issues of Confidentiality, Integrity & Availability (CIA). Institut for Elektroniske Systemer, Aalborg Universitet, ??? (2013)

[26] Kelly, G., McKenzie, B., *et al.*: Security, privacy, and confidentiality issues on the internet. Journal of Medical Internet Research **4**(2), 861 (2002)

[27] Waheed, N., He, X., Ikram, M., Usman, M., Hashmi, S.S., Usman, M.: Security and privacy in iot using machine learning and blockchain: Threats and countermeasures. ACM Computing Surveys (CSUR) **53**(6), 1–37 (2020)

[28] Ning, J., Huang, X., Poh, G.S., Xu, S., Loh, J.-C., Weng, J., Deng, R.H.: Pine: Enabling privacy-preserving deep packet inspection on tls with rule-hiding and fast connection establishment. In: Computer Security–ESORICS 2020: 25th European Symposium on Research in Computer Security, ESORICS 2020, Guildford, UK, September 14–18, 2020, Proceedings, Part I 25, pp. 3–22 (2020). Springer

[29] Zikratov, I., Kuzmin, A., Akimenko, V., Niculichev, V., Yalansky, L.: Ensuring data integrity using blockchain technology. In: 2017 20th Conference of Open

Innovations Association (FRUCT), pp. 534–539 (2017). IEEE

[30] Aminzade, M.: Confidentiality, integrity and availability–finding a balanced it framework. Network Security **2018**(5), 9–11 (2018)

[31] Kumar, P.R., Raj, P.H., Jelciana, P.: Exploring data security issues and solutions in cloud computing. Procedia Computer Science **125**, 691–697 (2018)

[32] Ye, T., Luo, M., Yang, Y., Choo, K.-K.R., He, D.: A survey on redactable blockchain: Challenges and opportunities. IEEE Transactions on Network Science and Engineering (2023)

[33] Zheng, Z., Xie, S., Dai, H.-N., Chen, X., Wang, H.: Blockchain challenges and opportunities: A survey. International journal of web and grid services **14**(4), 352–375 (2018)

[34] Singh, R., *et al.*: Edge ai: a survey. Internet of Things and Cyber-Physical Systems **3**, 71–92 (2023)

[35] Al-Shareeda, M.A., Anbar, M., Manickam, S., Hasbullah, I.H.: Review of prevention schemes for man-in-the-middle (mitm) attack in vehicular ad hoc networks. International Journal of Engineering and Management Research **10** (2020)

[36] Ahmad, U., Song, H., Bilal, A., Saleem, S., Ullah, A.: Securing insulin pump system using deep learning and gesture recognition. In: 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), pp. 1716–1719 (2018). IEEE

[37] Turner, S.: Transport layer security. IEEE Internet Computing **18**(6), 60–63 (2014)

[38] Chan, C.-l., Fontugne, R., Cho, K., Goto, S.: Monitoring tls adoption using backbone and edge traffic. In: IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 208–213 (2018). IEEE

[39] Golec, M., Gill, S.S., Bahsoon, R., Rana, O.: Biosec: A biometric authentication framework for secure and private communication among edge devices in iot and industry 4.0. IEEE Consumer Electronics Magazine **11**(2), 51–56 (2020)

[40] Bıçakcı, H.S., Santopietro, M., Boakes, M., Guest, R.: Evaluation of electrocardiogram biometric verification models based on short enrollment time on medical and wearable recorders. In: 2021 International Carnahan Conference on Security Technology (ICCST), pp. 1–6 (2021). IEEE

[41] Taloba, A.I., Elhadad, A., Rayan, A., Abd El-Aziz, R.M., Salem, M., Alzahrani,

A.A., Alharithi, F.S., Park, C.: A blockchain-based hybrid platform for multimedia data processing in iot-healthcare. Alexandria Engineering Journal **65**, 263–274 (2023)

[42] Sharma, P., Namasudra, S., Crespo, R.G., Parra-Fuente, J., Trivedi, M.C.: Ehdhe: Enhancing security of healthcare documents in iot-enabled digital healthcare ecosystems using blockchain. Information Sciences **629**, 703–718 (2023)

[43] Gupta, P., Chouhan, A.V., Wajeed, M.A., Tiwari, S., Bist, A.S., Puri, S.C.: Prediction of health monitoring with deep learning using edge computing. Measurement: Sensors **25**, 100604 (2023)

[44] Balasundaram, A., Routray, S., Prabu, A., Krishnan, P., Malla, P.P., Maiti, M.: Internet of things (iot) based smart healthcare system for efficient diagnostics of health parameters of patients in emergency care. IEEE Internet of Things Journal (2023)

[45] Doyle, J., Golec, M., Gill, S.S.: Blockchainbus: A lightweight framework for secure virtual machine migration in cloud federations using blockchain. Security and Privacy **5**(2), 197 (2022)

[46] Benedict, S.: Serverless blockchain-enabled architecture for iot societal applications. IEEE Transactions on Computational Social Systems **7**(5), 1146–1158 (2020) https://doi.org/10.1109/TCSS.2020.3008995

[47] Ayub Khan, A., Wagan, A.A., Laghari, A.A., Gilal, A.R., Aziz, I.A., Talpur, B.A.: Biomt: A state-of-the-art consortium serverless network architecture for healthcare system using blockchain smart contracts. IEEE Access **10**, 78887–78898 (2022) https://doi.org/10.1109/ACCESS.2022.3194195

[48] Winderickx, J., Braeken, A., Singelee, D., Mentens, N.: In-depth energy analysis of security algorithms and protocols for the internet of things. Journal of Cryptographic Engineering **12**(2), 137–149 (2022)

[49] Intel. https://www.intel.com/content/www/us/en/support/articles/000055611/processors.html

[50] Fan, Y., Winkel, C., Kulkarni, D., Tian, W.: Analytical design methodology for liquid based cooling solution for high tdp cpus. In: 2018 17th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm), pp. 582–586 (2018). IEEE

[51] Inteli7116. https://ark.intel.com/content/www/us/en/ark/products/208921/intel-core-i71165g7-processor-12m-cache-up-to-4-70-ghz-with-ipu.html

[52] Inteli710. https://www.intel.com/content/www/us/en/products/sku/201837/intel-core-i710750h-processor-12m-cache-up-to-5-00-ghz/specifications.html

[53] Dworkin, M.J.: Sha-3 standard: Permutation-based hash and extendable-output functions (2015)

[54] Gligoroski, D.: Length extension attack on narrow-pipe sha-3 candidates. In: International Conference on ICT Innovations, pp. 5–10 (2010). Springer

[55] Janosi: Heart Disease. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C52P4X (1988)

[56] Google. Google. https://cloud.google.com/functions

[57] apache. https://jmeter.apache.org/