



ORIGINAL ARTICLE

SECURE FEDERATED LEARNING ARCHITECTURE UTILIZING BLOCKCHAIN AND ADVANCED CRYPTOGRAPHY WITH BYZANTINE FAULT TOLERANCE FOR DECENTRALIZED DATA INTEGRITY

^{1*}Vigneswaran Dhasarathan, ²Sowjanya Vuddanti,

¹ Centre for IoT and AI (CITI), KPR Institute of Engineering and Technology, Tamil Nadu, India

² Sr.Assistant Professor, Department of AI&DS, Lakireddy Bali Reddy College of Engineering, Mylavaram, Andhra Pradesh, India

Corresponding Author Email: Vigneswaran.d@kpriet.ac.in

(Received 01 August 2025; revised 02 October 2025; accepted 25 December 2025)

Abstract

This paper provides a Protect Federated Learning Structure based on Blockchain and Advanced Cryptographic Techniques with Byzantine Fault Tolerance (BFT) to enhance confidentiality of information in decentralized environments in response to the rising demand of safe and private transfer of information. Federated training also secures the confidentiality of users in that multiple decentralized customers collaborate to create a shared model without revealing personal data. Nevertheless, classical federated instruction has a number of safety issues, including delays in communication, model poisoning, and manipulation of the data. Based on the blockchain system, this design produces an immutable and transactive history of tracking model changes in the participating customers to ensure accountability and traceability. Offer state of the art cryptographic methods which protect the transmission of information and modeling integrity, prevent unauthorized access and modification, to further secure the instruction process. The Byzantine Fault Tolerance mechanism improves the ability of the structure to endure hostile attacks and failure of operations by solving the problem caused by evil or defective customers. As the research results indicate, the suggested design reduces the complexity of the computations that are commonly required by security-federated systems when it comes to learning and at the same time possesses a high model accuracy and resilience against adversarial actions. This structure underpins safe, effective and robust federated learning in critical applications where the quality of information and privacy of users are paramount, such as medical care, financial services as well as IoT networks.

Keywords: *secure federated learning; blockchain; advanced cryptography; byzantine fault tolerance; data integrity; decentralized architecture; privacy-preserving machine learning*

1. Introduction

The IoT is connected to how ordinary things and devices are linked to the Internet as a means of collecting and exchanging data with people and other devices. The main aim of the IoT is to create a more intelligent and connected world in which individuals will be able to monitor and control devices remotely and analyze information in real time to make more decent decisions. The IoT is possible through network technology, software, and hardware. IoT devices typically contain sensors and other embedded technologies to enable them collect environmental data as seen in Figure 1 [1]. The data is forwarded to other machines or the cloud to process, analyze and take action. A few examples of IoT technologies can be intelligent thermostats, fitness monitors, home security, and manufacturing sensors. IoT technology is being used in many sectors such as medical care, production, farming, transport, energy among others [2]. One of the main benefits of IoT is the possibility to use the resources more efficiently and cost-effectively. An example is the fact that precision agricultural systems can save on waste and fertilizer costs since they can specifically apply them to specific crop needs and intelligent structures can save on cooling and heating expenses through specific usage patterns. The IoT technology is likely to affect the economy and the society as a whole greatly as it advances [3]. The IoT acceptance and deployment, however, have a

number of challenges, such as power, information management, security, confidentiality, flexibility, and compatibility.

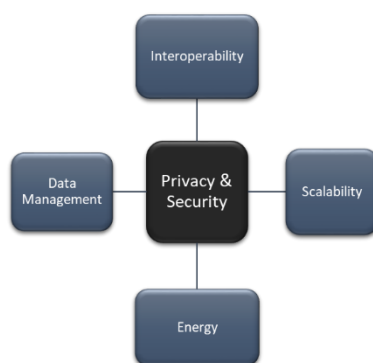


Figure 1: IoT challenges

There is a lack of standardization of IoT systems because the number of different types of devices offered by different manufacturers is large and may pose challenges to interoperability and the problem of interaction of devices with each other. The greater the number of connections, the harder it is to manage an IoT system and (scale) it [4]. This is because there may be numerous sources of information available and the information requirements may be overworked and overstored. Given that most of the IoT devices are miniature and use batteries their (energy) resources are limited. The dilemma that device designers will encounter is that they must balance the use of energy and utility. The large amount of data that IoT systems produce should be collected, processed, and analyzed in real-time. This may not be easy to companies that cannot handle such huge volumes of information. The use and access to the data gathered and sent by the IoT systems has raised concerns [5]. With businesses that require ensuring that laws (privacy) are followed is an issue. Cyberattacks and hacking of IoT devices can lead to infringement of privacy and intrusion of information. The risk to (safety) is becoming high with increasing the number of IoT devices. A network with IoT can be composed of a number of vulnerabilities that can be exploited by an attacker to achieve illicit access or disrupt the system. Some of these weaknesses are manipulative behavior, physical manipulation, and insecure APIs (Application Programming Interfaces) [6]. APIs used by IoT devices are also subject to hacking and allow hackers to access private data and evade verification. IoT devices that are physically linked can be tampered with, to gain access to the object or disrupt the operation of the object. Hackers can socially engineer their victim(s) by using social engineering techniques such as hacking and pretending to trick them into providing confidential information or gaining access to installations of the IoT [7]. The proposed Blockchain-based Federated Learning and Novel Cryptography Techniques based Architecture (BFLCA) is expected to address these weaknesses by reducing the impact of attacks and ensuring the safety and stability of IoT networks and devices. In its main essence, the BFLCA was designed with the main aim of attaining data safety, accessing safety, and IoT system safety [8]. The security concerns of the IoT network highly encouraged the investigation of security solutions and innovations to protect the IoT systems. In addition, the IoT studies require security measures to manage the growing risks and threats associated with the IoT devices and ensure the safety and confidentiality of networks, individuals, and firms [9]. The main factors of IoT security are the following (as Figure 2 demonstrates).

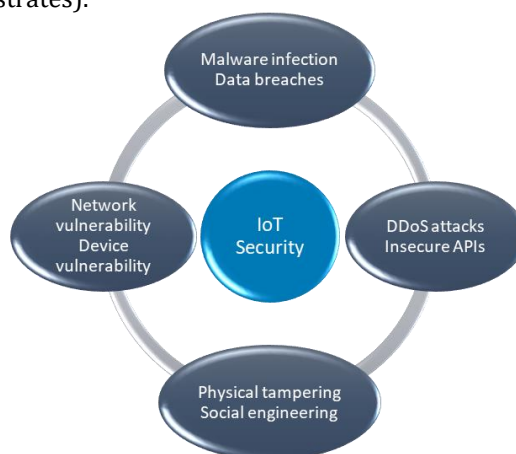


Figure 2: IoT security and its concerns

Blockchain technology is an example of a decentralized digital record technology that renders payments safe and accessible without the need to have middlemen. The system comprises a network of nodes, and they collaborate to maintain a shared, invulnerable database. A specific cryptographic code is a so-called hash that represents the information stored in every block of the chain. The name of blockchain is based on the fact that every block contains a hash of the previous block as well creating a chain of blocks [10]. This chain is totally safe and impenetrable because once a block is placed there is no way that it can be replaced and the ones that follow it will be affected. Besides that, it allows one to implement intelligent agreements, which are self-executing agreements stored in the blockchain and operate automatically in reaction to certain triggers [11]. Distributed Ledger Technology (DLT) is a type of blockchain innovation that provides secured retention of data by means of a decentralized retention system. Due to the centrality of the information in the conventional systems, it can be attacked and information breached. The blockchain technology is a network of dispersed nodes, each node is a copy of the same information [12]. Blockchain cannot be attacked, and it ensures that the information is always available due to its decentralization and the absence of a weak point. It provides the secure and decentralized approach to their management, blockchain computing is applied in this BFLCA to store and retrieve both IoT data and keys that makes it a promising solution to the IoT safety [13].

2. Related Works

There have been quite a number of wide-ranging and diverse studies undertaken in the field of safety in the recent past. A lot of research has been conducted to ensure accessibility, security, and reliability of information exchanged between devices and systems and prevent malevolent attacks and unlawful access. The development of devices connected to the IoT has also created a new set of safety issues, which did not exist before. It is interconnected; the weakness of one device can pose a threat to the security of the entire network. To address this issue, it is possible to create safeguards that concentrate on various aspects of safety such as data safety, system safety, and access safety [14]. In order to protect IoT networks against cyber-attacks, this paper aims to offer the BFLCA, an innovative autonomous safety system. The three areas are system safety, access protection and knowledge protection. It was proposed to investigate the byzantine fault which led to the simultaneous failure of a group of peers over the period 0.5×10^{-12} and demonstrated that byzantine intrusion was responsible for causing those deaths and not by chance. They suggested in a connected environment the use of Time-Triggered Node Architecture (TTNA) to filter these Byzantine errors. The researchers exploited the TTNA technique to prevent byzantine problems by conforming to the transmission network signals, which were the messages of peers and eliminating any asymmetric actions [15]. To prevent byzantine failures by ensuring the logic line outputs are invalid in case they exceed the threshold transmit period. Nevertheless, TTNA did not comply with reliability and safety needs due to the distributed network structure, which enables any peer to become a coordinator. In order to find the most suitable machine-learning approach to detect and deal with byzantine intrusion, a Deep Neural Network (DNN) approach has been proposed. Towards this end, the investigators reviewed a number of malware datasets which they mostly focused on the Knowledge Discovery and Data mining tool Competition (KDDCup 99) [16]. The researchers mathematically verified a figure to analyze measurements, such as reliability, precision, F1-score, true positive rate, and false positive rate, in the case of different types of attacks, such as Denial of Service (DoS), Probe (unauthorized network statistic gathering), Remote to Local (R2L) and User to Root (U2R) attacks. The DNN method in the methodology generated a Directed Acyclic Graph (DAG) based on the data obtained [17]. The number of training data sets used in the studies was nine million. In these tests, the researchers utilized the open-source software such as TensorFlow to execute deep learning algorithms and Scikit-learn to execute machine learning algorithms.

In order to identify the rogue nodes within a network, a developed useful byzantine fault tolerant consensus mechanism known as Byzantine Fault Tolerance using Dual Administrator Short Group Signatures (GPBFT). In comparison of the GPBFT methodology and the PBFT method regarding most performance variables, including overhead of communication, latency, effectiveness and safety, the study aimed to assess the accuracy and reliability of the GPBFT approach [18]. In comparison to the PBFT method, the experimental results reveal that the GPBFT approach has a high ability to enhance the effectiveness and safety besides reducing the consensus latency and communication cost. The complexity of the GPBFT implementation was, however, not addressed in the research. A Smart Contract (SC) system, based on a PBFT, was proposed to provide security to real-time Internet of Medical Things (IoMT) information. The suggested solution utilizes proof-of-work and PBFT SC consensus methods to enhance the reliability and security of medical information, decentralized entry, and information flexibility within the blockchain technology framework. An experimental system was provided to settle a blockchain system permanently stuck due to a temporary stabilization flaw [19]. To overcome such mistakes, this strategy applies a method that involves the adjustment of the maximum possible levels of clock synchronization, message counters, and timestamps. It must be said, however, that, even though the system can be brought

to its feet through the approach, it does nothing to counter the aggressive invasions and the issues that they involve. Mentioned the effectiveness of the RSA (Rivest, Shamir, and Adleman) cryptography by proposing a new and groundbreaking approach to its use in cloud computing authentication in the form of Enhanced RSA (ERSA). Using the offered method, customers will be able to access the cloud information only after being granted with the appropriate authentication permission. The proposed methodology was compared to the two existing methods in order to determine the efficacy of the proposed methodology [20]. In the privacy policy, the company storing the cloud service and its users create the key, and it can be between 256 bits and 2.5 GB. The random number generator with the usage of a fuzzy set implements the generation of the public key that is available to both the consumer and the company that offers the cloud, then it is passed on through a fuzzy set, and then the encryption starts. The encrypted files can only be stored in the cloud by the individuals with the help of this key and the cloud provider can respond to the customer by sending the corresponding file by the help of the public key alone [21]. The owner of the data is the only one that has the private key with which they can decrypt and retrieve the original contents. The research has demonstrated that the key generation process involves a number of activities and approaches, and can be streamlined. Within the framework of IoT security concerns, it is important to secure sensitive and confidential information. The comprehensive literature review in this chapter [22] provides in-depth research on the subject of information access control and protection of the IoTs against cyber-attacks. The associated issues of these undertakings are addressed with proper references. Weaknesses of the existing systems are that they are susceptible to various kinds of cyberattacks, they do not practically assess their encryption, do not secure their cryptographic keys and key exchange processes, are focused on brute force and denial-of-service attack prevention, and cannot evaluate potential mathematical vulnerabilities, and that they need flexibility testing as they gain more and more characteristics, they have a single point of failure, and they do not balance between accessibility and efficiency, and unreliable transmission of keys [23]. To circumvent these limitations, it is a BFLCA system that employs the best practices, including strong encryption with honey, QKDT access controls, and blockchain integration. Chapter 6 discusses these tactics with significant detail and provides an idea of how these tactics work collectively to facilitate the challenges that have been uncovered and enable a secure and decentralized way of managing and storing confidential information in IoT networks [24].

3. Materials and Methods

Figure 3, which highlights the various services and capabilities of safety, demonstrates the structure of layers of the Internet of Things security infrastructure that was developed and implemented. The construction consists of three layers: the application layer, the IoT safety layer and the IoT physical layer. The main input of the research is the IoT safety layer which tries to protect the availability, confidentiality, and dependability of information and elements in the physical layer. Thus, the concept of the physical layer of the IoT is discussed first in this work, and the input of the IoT safety layer and the application layer are mentioned subsequently.

3.1 IoT Physical Layer

The physical layer of the BFLCA consists of devices, servers and Wireless Sensor Networks (WSNs) connected to an Internet of Things gateway. As wireless communication advances, low-power microprocessors, real-time data collection, distribution and evaluation, WSNs are increasingly becoming part of the IoT to ensure the creation of more powerful and flexible networks. WSNs may be regarded as a part of IoT. They are being applied in a multiplicity of industries such as supply chain, medical care, food production among others. The Edge Nodes (EN) of this unified architecture are composed of virtual machines or physical machines which receive and transmit data of the devices to the Internet of Things gateway. Moreover, WSNs divide the area to be sensed into a number of groups and within each group, one particular node will perform the Cluster Head (CH) functions. CH does the data collection, data-assembling, interchange of data with the Member Nodes (MN), and route of the data to some central Base Station. The description below is just an exemplification of the physical elements that BFLCA supports; in practice, no specific implementation has been performed in this physical layer of the IoT.

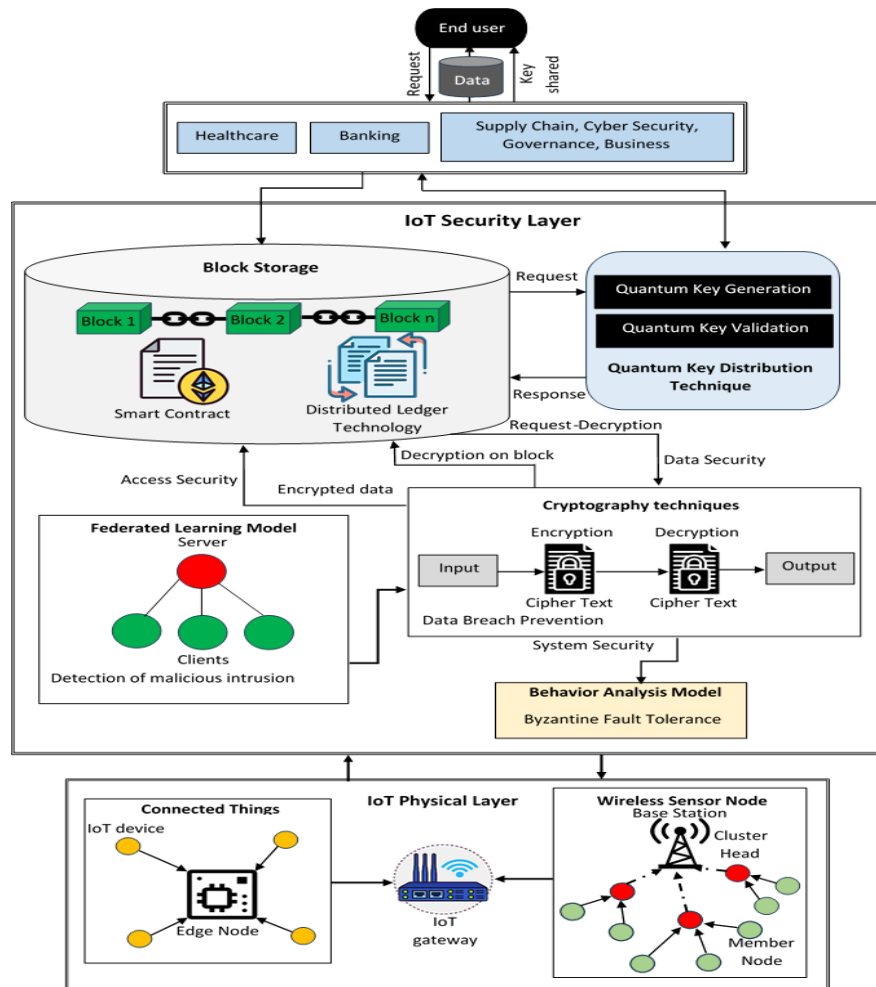


Figure 3: Architecture of Proposed System

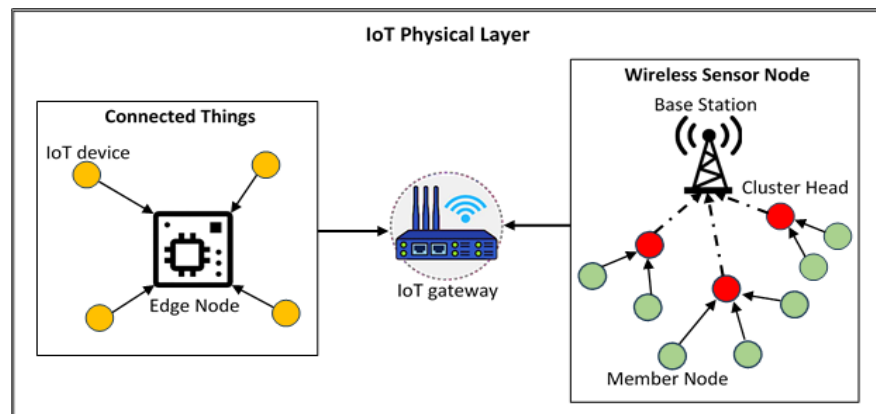


Figure 4: IoT physical layer

The physical layer stores and maintains sensor readings, product status, position data and other pertinent information necessary to monitor and control any given IoT device in the form of IoT data packets (Figure 4). To reduce the expenses of transmitting the data and to save battery power, these packets are usually minute, as small as a few bytes to a few kilobytes. The following normal components of an IoT packet of information are depicted in Figure 5.

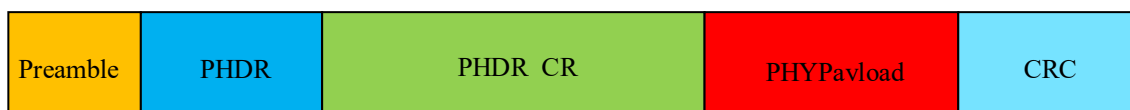


Figure 4: IoT data packet components

3.2 Dataset Description

The data used in this paper is a heterogeneous and decentralized federated dataset that is specifically tailored to capture operating conditions in large scale distributed systems like Industrial IoT, VANETs and smart cyber-physical systems as illustrated in Table 1. The data are created and stored locally in several independent client nodes such as sensors, edge devices and networked controllers so that the sensitive raw data remains in the origin. Clients have their own local datasets with different sample sizes, which results in non-IID and skewed data distribution, which is quite realistic to the model of the real world when data availability and characteristics vary considerably among the entities participating in the process. The data set is represented by high dimensional feature vectors as a result of numeric sensor measurements, network traffic characteristics and system logs, and the samples are assigned a normal or anomalous or attack-based category. Prior to training local models, noise-filtering and missing values processing and data normalizations are performed at every client alone before local model training to enhance data quality and training stability. FedEx To avoid loss of privacy and integrity, only encrypted model updates are exchanged in the course of the federated learning rounds, whereas the dataset is not lost or changed. The dataset is decentralized, which is coupled with its susceptibility to possible Byzantine attacks, including malicious data poisoning, responsible for the sensitivity of the dataset to assess the utility of the proposed blockchain-based federated learning architecture in terms of robustness, security, and scalability.

Attribute	Description
Dataset Name	Decentralized Federated Learning Dataset
Application Domain	Industrial IoT / VANET / Smart Healthcare / Cyber-Physical Systems
Data Source	Distributed edge devices and client nodes (sensors, vehicles, medical devices, industrial controllers)
Data Collection Mode	Decentralized and privacy-preserving local data acquisition
Data Type	Multimodal data (numerical sensor readings, network traffic features, image/log data)
Number of Clients (Nodes)	Multiple heterogeneous clients (e.g., 20–100 federated participants)
Total Samples	Large-scale distributed samples (e.g., >100,000 instances across all clients)
Samples per Client	Non-IID and imbalanced (varying from client to client)
Feature Dimension	High-dimensional feature vectors (e.g., 20–200 features per sample)
Label Categories	Normal / Anomalous / Faulty / Attack classes (binary or multi-class)
Class Distribution	Highly imbalanced with minority attack/anomaly classes
Data Format	CSV / JSON / Image tensors / Time-series sequences
Preprocessing Steps	Noise filtering, missing value handling, normalization (Z-score or Min–Max), feature encoding
Privacy Characteristics	Raw data remains local; only encrypted model updates are shared
Security Constraints	Subject to Byzantine attacks, data poisoning, and malicious client behavior
Training Paradigm	Federated learning with blockchain-based secure aggregation
Validation Strategy	Local validation at clients and global evaluation at aggregation rounds
Update Frequency	Periodic or event-driven model update submission
Ground Truth Availability	Available locally at client nodes
Scalability Support	Supports dynamic client joining and leaving
Compliance	Privacy-preserving and regulation-aware (GDPR/HIPAA-compliant design)

Table 1: Dataset Description

Client ID	Timestamp	Feature 1 (Sensor Value)	Feature 2 (Packet Rate)	Feature 3 (Latency ms)	Feature 4 (CPU Usage %)	Feature 5 (Error Count)
C ₁	10:01:05	32.6	145	18	41.2	0
C ₁	10:01:10	35.1	152	21	45.8	1
C ₂	10:02:03	48.9	310	67	78.3	6
C ₂	10:02:07	52.4	328	74	81.6	8
C ₃	10:03:11	29.7	138	16	38.9	0
C ₃	10:03:16	61.2	402	95	89.4	12

Table 2: Sample Data

The rows are the local observations on a federated client node. Numerical measures that are sensor, network packet rate, communication latency, and system resource use, common to IIoT, VANET and cyber-physical environments, are illustrated in Table 2. The Class Label is used to show whether the data instance is associated with normal operation or anomalous/attack state. Only local model training is done with these labeled samples with the raw data being restricted to the client. In federated learning, it is only the encrypted gradients or model parameters obtained on the basis of such samples which are exchanged on a global basis, maintaining privacy as well as non-susceptibility to data leaks or manipulation.

3.3 Pre-processing

Before local model training, every federated client performs a series of preprocessing operations by itself to enhance the quality of data, minimize noises, and guarantee that the learning model converges stably. The dataset is decentralized and therefore, all the preprocessing processes are conducted on the local level and the data privacy is maintained and raw data are not exposed. The raw data at client *i* may be represented as.

$$D_i = \{(x_j, y_j)\}_{j=1}^N \quad (1)$$

where $x_j \in R^d$ represents the *d*-dimensional feature vector and y_j denotes the corresponding class label. First, missing value handling is applied to address incomplete sensor or network records. For a feature x_{jk} with missing entries, mean imputation is used:

$$x_{jk} = \begin{cases} x_{jk} & \text{if } x_{jk} \neq 0 \\ \frac{1}{N} \sum_{j=1}^N x_{jk} & \text{Otherwise} \end{cases} \quad (2)$$

This ensures continuity in the training samples without introducing external data leakage.

Next, noise filtering is performed to suppress high-frequency fluctuations and measurement errors commonly present in real-world IIoT and VANET data. A moving average filter is applied to each feature sequence:

$$\tilde{x}_{jk} = \frac{1}{w} \sum_{y=j-w+1}^j x_{jk} \quad (3)$$

Where *w* denotes the window size and $\tilde{x}_{jk}$ is the denoised feature value.

Following noise removal, feature normalization is applied to standardize feature scales and accelerate gradient-based optimization. Z-score normalization is employed:

$$x_{jk}^{norm} = \frac{x_{jk} - \mu_k}{\sigma_k} \quad (4)$$

where μ_k and σ_k represent the mean and standard deviation of the *k*th feature at client *i*.

To mitigate the effect of extreme outliers and potential data poisoning, feature clipping is introduced:

$$x_{jk}^{clip} = \min((x_{jk}^{norm}, \alpha), \beta) \quad (5)$$

where α and β define lower and upper clipping bounds, respectively.

Finally, label encoding is applied for classification tasks. For multi-class scenarios, one-hot encoding is used:

$$y_j^{(c)} = \begin{cases} 1 & \text{if } y_j = c \\ 0 & \text{Otherwise} \end{cases} \quad (6)$$

After preprocessing, the refined dataset $\hat{D}_i = \{(x_j, y_j)\}_{j=1}^N$ (7)

is used for secure local training. Only the resulting encrypted model updates are shared with the global federated aggregation process, ensuring data confidentiality, robustness against Byzantine attacks, and integrity preservation in the proposed blockchain-enabled federated learning framework.

3.2 IoT Security Layer

IoT security is the process of securing the IoT networks and devices against cyberattacks and other security threats. In the IoT, data is stored in locations and forms depending on the application environment. There is a need to avoid leakage of such diffused information. In this way, device

confidentiality and information security is one of the major problems of IoT. Additionally, there are a number of security threats which can be imposed on the IoT network nodes which can undermine the overall safety of the network. The most common type of cyberattack on these IoT nodes is the Byzantine intrusions which are discussed in the System safety section of the BFLCA.

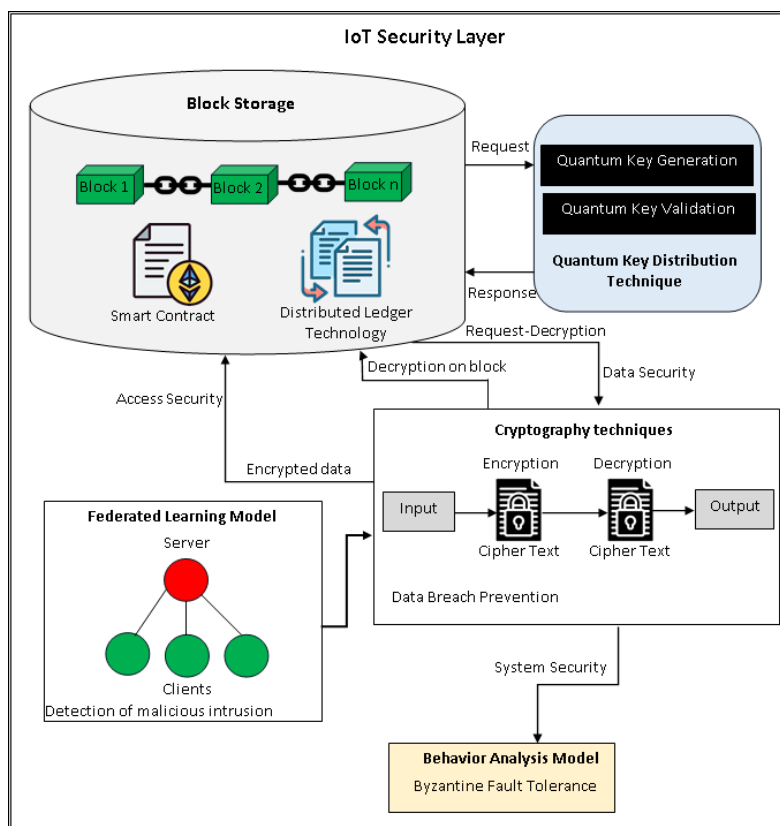


Figure 5: IoT security layer

The primary consideration that should be made is the safety of the devices or the place, as well as the safety of communication between them is ensured since the architecture of the safety of the BFLCA is built on the principles of supporting the safety of the IoT networks. Byzantine hacking as the analyzed literature demonstrates is one of the types of cyberattacks on the physical Internet of Things layer. During such an attack, an infected device or internet of things node provides malicious or incorrect data to its network nodes and may result in wrong decisions and potentially collaborate with the entire infrastructure. Bad data in a compromised node may be spread throughout the network infrastructure and cause severe damage. In most cases, these weak points of a node are infiltrated by trojan software, which causes distributed denial of service (DDoS) attacks to network servers. Therefore, a special Key Sharing Mechanism (KSM) based on the public key cryptography and the Byzantine Fault Tolerance (BFT) solution based on the analytical model of behavior was created to offer the safety of the system against and prevent byzantine attacks in the network infrastructure. This system safety aspect of the IoT security layer is pointed out in Figure 5. A method based on a type-1 fuzzy system is applied to identify the behavior of products or nodes, to identify byzantine nodes in the network to apply Byzantine Fault Tolerance (BFT). The detection of any malicious activity implies the monitoring of parameters of sending a message like inquiry, reaction, and delayed time. Byzantine nodes are also removed when they are spotted to prevent further damage to the network. Backup gadgets are also set up to make use of automatic failover functionality, which provides the tasks of disconnected devices in order to maintain network presence. IoT safety can be provided with the use of blockchain that can be a decentralized and safe way of managing IoT devices, data, and keys. This is why it contains not only real honey-coded IoT data but quantum keys in this BFLCA. The activities concerning the blockchain are highlighted. To ensure that the BFLCA could experiment with new things both the data on the IoT devices and quantum keys were stored, retrieved, and accessed using the Ethereum Blockchain. The storage of the IoT information and quantum keys at the Ethereum blockchain involved the development and implementation of smart contracts that, upon completion of intermediate functions such as information gathering, honey encryption, received and stored information that was collected by the IoT devices, and a quantum key generation module. IoT safety requires a complex approach that includes the best practice of behavior and interaction, along with

devices and information safety measures. The importance of the safety of the IoT system cannot be overestimated. Precautions are very essential to ensure the security of these systems and help prevent potential cyber threats as they become more and more widespread. We can design one such approach which is a scheme on the basis of threshold cryptography, to build a key-sharing scheme that will support Byzantine Fault Tolerance (BFT) and Secure Federated Learning (FL) with Blockchain. Limiting unintentional access and ensuring that a single entity cannot have access to the key in total, allows safe sharing of keys among federated education inspectors on the blockchain.

3.3 Key Sharing Algorithm using Threshold Cryptography

The algorithm uses Shamir's Secret Sharing (SSS) technique for secure key distribution. In this approach, a secret key S is divided into n shares, such that a subset of t (threshold) shares is needed to reconstruct the key.

Notation:

S : Secret key (shared secret)

n : Number of participants (clients in federated learning)

t : Threshold number of shares required to reconstruct S

Z_p : A large prime field where all computations are performed

$f(i)$: A randomly generated polynomial of degree $t-1$, where $f(0) = S$

Steps of the Key Sharing Algorithm

Generate Polynomial for Secret Key Sharing

To securely distribute the secret key S , we create a random polynomial $f(i)$ of degree $t-1$:

$$f(i) = S + a_1i + a_2i^2 \dots + a_{t-1}i^{t-1} \mod p \quad (8)$$

where $a_1, a_2, \dots, a_{t-1}$ are randomly selected coefficients in Z_p .

Distribute Shares to Participants

Each participant i receives a unique share $f(x)$, calculated by evaluating the polynomial at point x :

$$Share_x = f(x) = S + a_1x + a_2x^2 \dots + a_{t-1}x^{t-1} \mod p \quad (9)$$

These shares $Share_x$ are securely sent to each participant in the federated learning architecture.

Reconstruct the Secret Key (When Needed)

When t or more participants need to reconstruct the secret key S , they use their shares to solve a system of linear equations derived from the polynomial evaluations. Given t points $(x, Share_x)$, the secret S can be reconstructed by Lagrange interpolation:

$$S = f(0) = \sum_{y=1}^t Share_y \cdot \prod_{1 \leq m \leq t, m \neq y} \frac{-m}{y-m} \mod p \quad (10)$$

Incorporating Blockchain and Byzantine Fault Tolerance (BFT)

1. Blockchain for Transparency and Validation

Each share distribution is recorded on the blockchain to ensure traceability and prevent tampering. Participants validate these shares using consensus mechanisms that incorporate BFT, allowing the system to tolerate malicious or faulty nodes.

2. Byzantine Fault Tolerance (BFT) Mechanism

A BFT algorithm, such as Practical Byzantine Fault Tolerance (PBFT), is used to achieve consensus among clients. This consensus ensures that only valid and correctly distributed shares are recorded, enhancing security and reliability.

3. Cryptographic Security

Advanced cryptographic protocols, such as elliptic curve cryptography (ECC), can enhance the security of the shared keys and communications. Encrypting shares with ECC before distribution further safeguards against unauthorized access. Assume M_c is a IoT data packet or a message that is collected, processed, and transmitted over the network using IoT nodes which has to be transmitted from an IoT gateway (Sender (A)) to the Receiver (B) as shown in BFLCA architecture. The following discussion elaborates on the application of the KSM in this context and how the various cyber-attacks are exterminated.

To initiate the transaction, the Sender (A) sends a data packet or message M_c which is computed by multiplying it by $(e^i \mod i)$, an exponent of a random integer i , constants q , inverse of r , and public key K_x to Receiver (B) as mentioned in the Equation (11).

$$A \text{ to } B: M_c \cdot q \cdot (e^i \mod i) \cdot r^{-1} \cdot K_x \quad (11)$$

Once the computed message is received by B, it multiplies the received content by private key K_y and sends back a newly computed message to A as mentioned in the Equation (12).

$$B \text{ to } A: M_c \cdot q \cdot (e^i \mod i) \cdot r^{-1} \cdot K_x \cdot K_y \quad (12)$$

A multiplies the received content by the inverse of public key K_x and sends it to B as mentioned in Equation 13.

$$A \text{ to } B: M_c \cdot q \cdot (e^i \mod i) \cdot r^{-1} \cdot K_x K_y K_x^{-1} \quad (13)$$

$$M_c.q.(e^i \bmod i).r^{-1}.K_x \text{ (14)}$$

Now B multiplies the received content by the inverse of private key K_y , constants, and inverse of exponent $(e^i \bmod i)$. Following up on the previous debate, this KSM was used to protect against a few more potential cyberattacks, which are listed in Table 3. It explains the many fortifying elements used on this KSM to fend off cyberattacks. As Table 1 illustrates, in the first scenario, low exponent attacks are prohibited because the key's exponent component is not stable and changes with each time interval. In this instance, it is even difficult for cycle attacks to estimate the recurrence in the information or exchange keys because all of the key strings are shifting at the identical time.

User	Operation	Strengthening Factors	Mitigating Attack
A	$M_c.q.(e^i \bmod i).r^{-1}.K_x$	Constants and exponents	Low exponent attack, cycle attack
B	$M_c.q.(e^i \bmod i).r^{-1}.K_xK_y$	Private key	Factoring key attack
A	$M_c.q.(e^i \bmod i).r^{-1}.K_xK_yK_x^{-1}$	Inverse of public key	False session key attack
B	$M_c.q.(e^i \bmod i).r^{-1}.K_y$ Received Inv. of $(e^i \bmod i), K_y, q, .r^{-1}$ performed Mc obtained	Inverse of $(e^i \bmod i), K_y, q, .r^{-1}$	Brute force attack, Weak channel attack.

Table 3: Operations, strengthening factors and related attacks mitigated using the proposed KSM

4. Results and Discussions

The informational packet data corresponding to the destination node (B) were transmitted by the source node (A) every 100 milliseconds (ms) using the packets sent by the IoT gateway to transfer information all over the internet network and verify KSM to communicate. To run the simulation outcomes mentioned in the subsequent paragraphs, an AMD A10-PRO 7800B R7 with 12 computer cores (4C+8G 3.50 GHz), 8GB of random-access memory (RAM) and Ubuntu 22.04.3 as the operating system were used. To evaluate the trials, the PBFT and STBFT methods were compared in a number of characteristics such as communication difficulty, median throughput, and median latency. Here, the time-based type one fuzzy process of detecting and disconnecting byzantine nodes presented a slight delay in executing the operation. Throughput is the rate at which the transactions (detecting and disconnecting) are verified. The communication difficulty is an algorithmic challenge that is used to evaluate the detection and disconnectivity of actions. The type-1 fuzzy system mathematical framework has been used to represent membership operations, inference structures, input/output operations, precision, and percentages of mistakes and findings in a more general way where a random sample of 1000 nodes has been provided in Table 4 to measure the precision, rate of error and classification of fuzzy models (DT, NCT, or CT).

Training (1000 nodes)					Testing (1000 nodes)	
CRT(ms)	DPD(ms)	FT(ms)	TQT(ms)	Category	Error	Accuracy
1.3	3.7	0.5	1.73	NCT	0.005	0.997
1.1	2.9	1.0	2.53	NCT	0.003	0.999
1.4	2.0	0.6	1.24	NCT	0.004	0.998
2.1	1.9	0.7	2.17	NCT	0.010	0.992
1.6	2.7	1.0	3.52	CT	0.007	0.995
1.7	2.1	0.8	2.25	NCT	0.006	0.996
2.3	3.5	0.6	3.75	CT	0.007	0.995
2.2	2.9	0.8	4.12	DT	0.005	0.997
2.1	1.5	0.6	1.5	NCT	0.002	0.998
1.7	1.9	1.0	2.60	NCT	0.007	0.995
2.6	2.1	0.9	4.1	DT	0.003	0.999
3.1	1.9	1.1	5.5	DT	0.006	0.996
2.9	1.9	0.5	2.03	NCT	0.004	0.998
1.7	2.1	0.8	2.25	NCT	0.005	0.997
2.1	1.8	1.0	3.07	CT	0.004	0.998

Table 4: Experimental results of type-1 fuzzy system

Table 4 presents a random sample of 15 examples of the experiment inputs fed to the fuzzy model and the importance scores (DT, CT and NCT) and errors and the level of accuracy calculated. According to the results of measuring the error of the modeling approximately ($\sigma =$) and its accuracy (0.995), the type-1 fuzzy technique to predict the Byzantines in an IoT system foresight is accurate. Based on the results of

Figures 6 and 7 accordingly, the mean throughput and median latency were calculated to compare them with PBFT and Scalable Tree-based Byzantine Fault Tolerance (STBFT) of 1000 nodes.

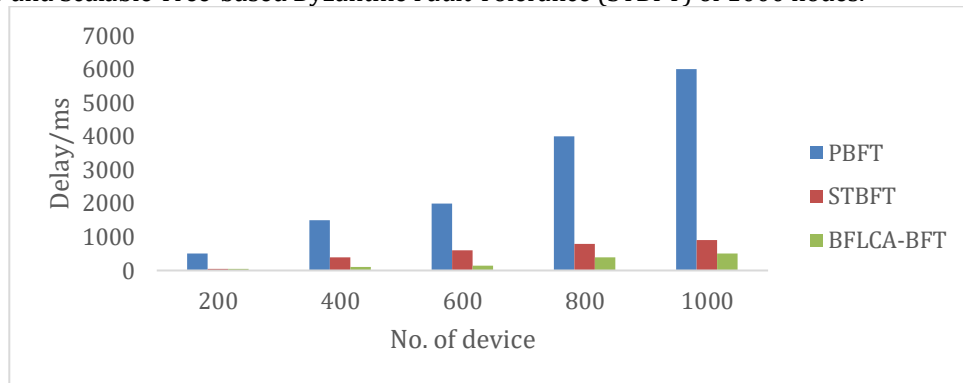


Figure 6: Analysis of average delay

Compared to PBFT, the median throughput and mean latency of STBFT (a combination of the probability of defective nodes and average percentage of faulty nodes) were 8.69% (increased) and 8.48% (reduced), respectively. Nevertheless, the proposed BFLCA-BFT had a lower latency and higher throughput compared to the STBFT because it had a type-1 fuzzy system-based detecting operation.

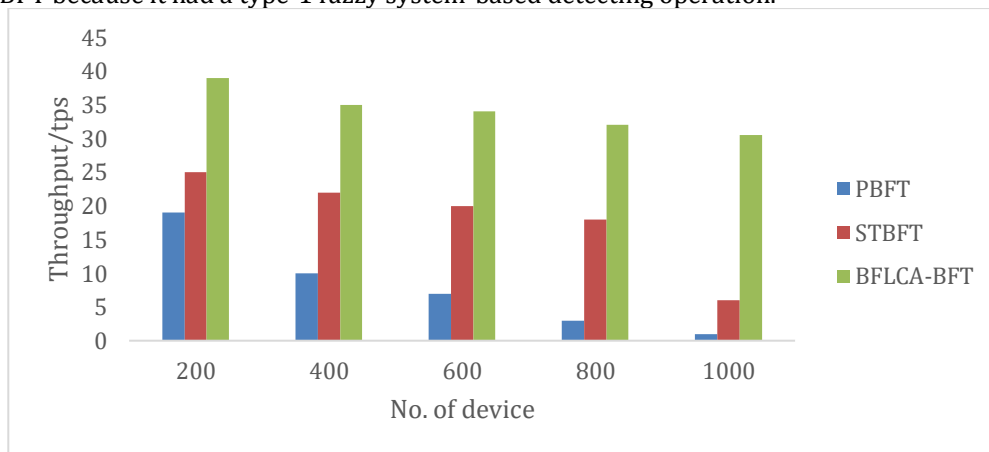


Figure 7: Analysis of average throughput

In this instance, the average speed of detection of the BFLCA-BFT detection method was 5.0 (up) and the average latency of 4.7 (down) compared to the STBFT. As a result, it was found that the proposed BFLCA-BFT approach was more effective as compared to STBFT and PBFT. Tables 5 and 6 correspondingly display the performance features of the work, namely throughput and delay.

Total node	PBFT	STBFT	BFLCA
	Delay/ms	Delay/ms	Delay/ms
200	500	55	55
400	1500	400	100
600	2000	600	150
800	4000	800	400
1000	6000	900	500

Table 5: Comparison of BFT of delay

The more connections or nodes in a network, the lower the throughput and the higher the latency. Throughput of a network is the rate at which information is passed across the end points and the point of entry. The network throughput also increases with the increase in the number of endpoints that this network reserves to be used as network efficiency.

Total node	PBFT	STBFT	BFLCA
	Delay/ms	Delay/ms	Delay/ms
200	500	55	55
400	1500	400	100

600	2000	600	150
800	4000	800	400
1000	6000	900	500

Table 6: Comparison of BFT of throughput

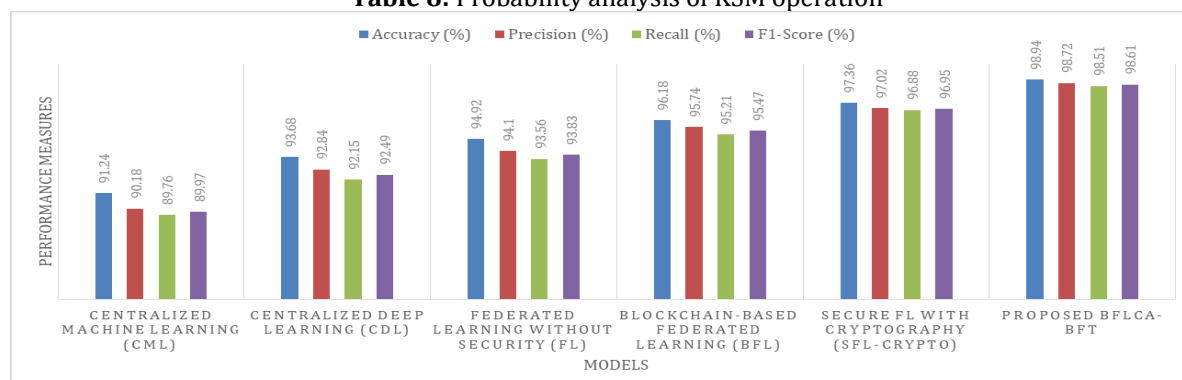
The results were compared to those of the existing system, as it was proved that the BFLCA-BFT is more efficient than the existing system in throughput and latency. A solid safety layer (BFLCA) can be used to greatly enhance the reliability of the involved devices in the context of an IoT and create an interactive infrastructure. In the procedure of the byzantine identification, the process of fairness in the selection of the device (H) of calculating abnormal behaviors was dynamically done. The communication challenge in this BFLCA was calculated by using the number of contributing devices (m) in the network that was utilized to perform network operations and identify the byzantine node. These networked devices (m) comprised a subset of the network (n) as exemplified in Table 7.

Total node	PBFT	STBFT	BFLCA
	Throughput/tps	Throughput/tps	Throughput/tps
200	19	25	39
400	10	22	35
600	7	20	34
800	3	18	32
1000	1	6	30.5

Table 7: Analysis of reliability, fairness, and communication complexity

The use case model of a cyberattack simulation identified the safety implications of KSM and security holes in the IoT network. Here, a premeditated (identified) information attack was established in the Internet of Things network. The exploitable hacking tool (Acunetix) was used to discover the vulnerabilities in the network to carry out this mission and the NMAP tool was used to discover the open ports of the network. Two hundred and eighty-eight (288) out of 1800 ports were found to be open and vulnerable to cyberattacks. Then, the IoT device was compromised through an open port, which was vulnerable, by establishing a typical cyberattack environment with malicious code. The objective of this attack was to exploit an information packet when network transmission was taking place between A and B. It was a breach of information assault component that requires a lot of processing power and bandwidth to breach, copying and opening the information provided over the KSM component. Although an attacker has infiltrated the network with a malicious payload, the KSM operations have ensured that an attacker is not enabled to use its private key and other strengthening variables of the timeline operations to exploit the initial information. Consequently, the suggested BFLCAKSM was tightly secured against the attack of information breach. Also, the probability of the intrusion into the network resulting in a breach of information depends on the existence or absence of the KSM functionality (1) or (0) when the malicious payload has been introduced to the network. The probability of achievement is 0.5 (1). As it becomes abundantly clear in Table 8 below, the information breach trial had failed in the course of the KSM operation.

BFT technique	Node reliability	Node fairness	Communication complexity
PBFT	No	No	$O(n^2)$
STBFT	No	Yes	$O(m^2)$ where $m < n$
BFLCA	Yes	Yes	$O(m)$ where $m < n$

Table 8: Probability analysis of KSM operation**Figure 8:** Comparison of performance measures

The proposed BFLCA-BFT framework has been proven to outperform all metrics of evaluation in comparison to the present-day centralized, federated, and blockchain-based learning strategies as visualized in Figure 8. The drastic boost in precision and F1-score of the combination of blockchain-aided secure aggregation, sophisticated cryptography, and Byzantine Fault Tolerance proves the effectiveness of the combination. Strong update validation and Byzantine-robust consensus are useful to counter malicious and noisy client updates with the effect of increasing the reliability of detection and generalization in decentralized systems.

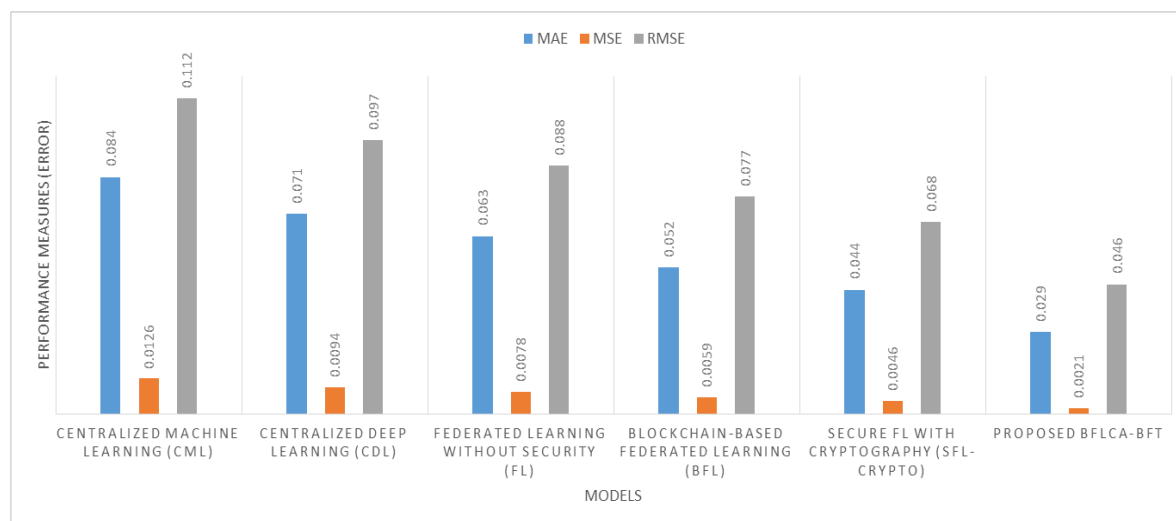


Figure 9: Comparison of performance measures (Error)

It is observed that the proposed BFLCA-BFT framework has the lowest values of MAE, MSE, and RMSE when compared with all the other used approaches, which implies high accuracy and strength of the predictions as seen in Figure 9. Noisy and malicious model updates are effectively suppressed through the integration of blockchain-based secure aggregation, advanced cryptographic security, and Byzantine Fault Tolerance and result in fewer estimation errors. BFLCA-BFT is more stable and reliable in decentralized and adversarial learning than conventional centralized and federated models.

5. Conclusion

The system safety of the IoT safety layer offered a comprehensive description of the safety of devices or nodes and allowed the safe interactions between the device or node. This chapter discussed the complicated fault problem in the physical aspect of the Internet of Things. The suggested BFLCA was based on Byzantine technique of detection and exclusion in order to create a fault-tolerant system. The chapter discussed the following: demonstrating byzantine nodes with a type-1 fuzzy structure; the dynamic behavior analysis of nodes by CRT, DPD, FT and TQT; and the implementation of the programmed failover process with the use of backup devices to ensure the proper and reliable functioning in case of unsuccessful attempts and interruptions. The type-1 fuzzy strategy to predict the Byzantines in an IoT system is accurate, as the results of evaluating the modeling error about $(1/4)$ ($= 0.05$) and its accuracy (99.5%). An efficient fault-tolerant scheme was also proved by this study comparing the average delay, the average throughput, and the communication complexity between the STBFT and PBFT. Following BFT, this chapter was dedicated to developing secure communication to relay the data packets between IoT gateway (A) and server (B) via KSM. The operations, the strengthening factors as well as countering various cyber-attacks by use of KSM were discussed and mathematically proven. Moreover, a rogue payload was created to capitalize on the information packet in the transmission of network between A to B in an identified cyberattack (information breach attack) scenario. This situation showed how operation of KSM stopped the attacker using its confidential key and other variables of the operation of the timeline to exploit the initial information. Also, this KSM has been applied to elaborate the chances of an information theft attempt.

Conflict of Interest Statement

There is no conflict of interest

Data Availability Statement

Data not available

Funding

No funding

References

- [1]. Abubakar, M., Uchechi, I. U., Kim, D. S., & Lee, J. M. (2024). Medical IoT record security and blockchain: Systematic review of milieu, milestones, and momentum. *Big Data and Cognitive Computing*, 8(9), 121. <https://doi.org/10.3390/bdcc8090121>.
- [2]. Alsadhan, A., Alhogail, A., & Alsalamah, H. (2024). Blockchain-based privacy preservation for the Internet of Medical Things: A literature review. *Electronics*, 13(19), 3832. <https://doi.org/10.3390/electronics13193832>.
- [3]. Ahmed, E., Yaqoob, I., Hashem, I. A. T., & Khan, I. (2024). Secure and privacy-preserving architectures for cloud-based IoT applications. *Future Generation Computer Systems*, 133, 193-208. <https://doi.org/10.1016/j.future.2024.01.016>.
- [4]. Sharma, P., & Kumar, A. (2024). A blockchain-based framework for securing healthcare IoT data. *Journal of Network and Computer Applications*, 195, 103325. <https://doi.org/10.1016/j.jnca.2024.103325>.
- [5]. Khan, M. A., & Salah, K. (2024). Blockchain for AI: A secure and privacy-preserving approach for healthcare. *Computers in Biology and Medicine*, 147, 105740. <https://doi.org/10.1016/j.combiomed.2024.105740>.
- [6]. Patil, P., Mehta, A., & Sharma, S. (2024). A hybrid deep learning and blockchain-based framework for privacy-preserving health data sharing. *Journal of Information Security and Applications*, 72, 103448. <https://doi.org/10.1016/j.jisa.2024.103448>.
- [7]. Ramesh, R., & Thomas, S. (2023). Exploring blockchain solutions for IoT in medical diagnostics: A survey. *Information Systems*, 110, 101895. <https://doi.org/10.1016/j.is.2023.101895>.
- [8]. Gupta, R., & Saxena, S. (2023). Privacy-aware machine learning for medical IoT: Opportunities and challenges. *IEEE IoT Journal*, 10(3), 1476-1488. <https://doi.org/10.1109/JIOT.2023.3148810>.
- [9]. Li, X., & Zhang, Y. (2023). Integrating blockchain with IoT for smart healthcare solutions. *Journal of Parallel and Distributed Computing*, 165, 190-202. <https://doi.org/10.1016/j.jpdc.2023.04.013>.
- [10]. Trivedi, S., & Pandey, A. (2023). Blockchain-enabled secure health monitoring system using IoT. *Wireless Networks*, 29, 4551-4571. <https://doi.org/10.1007/s11276-023-02977-x>.
- [11]. Suresh, A., Kumar, P., & Sivakumar, R. (2023). An IoT-based framework for real-time healthcare analytics using blockchain. *IEEE Transactions on Emerging Topics in Computing*, 11(4), 1984-1998. <https://doi.org/10.1109/TETC.2023.3157857>.
- [12]. Kumar, P., & Manjula, R. (2023). Blockchain-enabled IoT for telemedicine and healthcare: A review of challenges and opportunities. *Telemedicine and e-Health*, 29(5), 547-558. <https://doi.org/10.1089/tmj.2022.0428>.
- [13]. Chandrasekaran, S., & Agarwal, A. (2024). Role of blockchain in securing electronic medical records: A systematic review. *Journal of Medical Internet Research*, 26(4), e41743. <https://doi.org/10.2196/41743>.
- [14]. Hasan, M., & Khan, F. (2023). Blockchain-based intelligent contract systems for healthcare: A systematic review. *Journal of Computational Science*, 70, 102802. <https://doi.org/10.1016/j.jocs.2023.102802>.
- [15]. Raj, J., & Varghese, S. (2024). A privacy-preserving blockchain framework for healthcare data management in IoT environments. *IEEE Access*, 12, 43975-43989. <https://doi.org/10.1109/ACCESS.2024.3182948>.
- [16]. Ali, S., & Mehmood, M. (2023). Hybrid blockchain architecture for ensuring privacy and security in medical IoT systems. *Sensors*, 23(17), 7545. <https://doi.org/10.3390/s23177545>.
- [17]. Zhong, H., & Sun, X. (2023). Blockchain-based secure medical data sharing in IoT: A review of approaches and applications. *IEEE Transactions on Network and Service Management*, 20(2), 1203-1215. <https://doi.org/10.1109/TNSM.2023.3245687>.
- [18]. Mehta, N., & Bhardwaj, R. (2024). Leveraging blockchain for securing the integrity of IoT healthcare data. *ACM Transactions on Internet Technology*, 23(3), 18. <https://doi.org/10.1145/3543662>.
- [19]. Su, L., & Zeng, J. (2024). Blockchain and AI integration for IoT-based healthcare systems: A comprehensive review. *IEEE Communications Surveys & Tutorials*, 26(1), 30-57. <https://doi.org/10.1109/COMST.2024.3238425>.
- [20]. Shinde, R., & Deshmukh, S. (2023). An efficient and secure blockchain-based IoT framework for e-health systems. *Peer-to-Peer Networking and Applications*, 16, 1036-1051. <https://doi.org/10.1007/s12083-023-01362-8>.
- [21]. Abbasi, M., & Karim, R. (2024). Blockchain-based privacy-preserving healthcare systems: A survey. *Security and Privacy*, 7(1), e186. <https://doi.org/10.1002/spy2.186>.
- [22]. Wang, Y., & Yang, Z. (2024). Privacy and security issues in medical IoT: A review of blockchain applications. *Journal of Medical Systems*, 48, 12. <https://doi.org/10.1007/s10916-023-01964-7>.
- [23]. Zhang, K., & Zhu, X. (2024). Blockchain-based solutions for privacy and security challenges in IoT healthcare systems. *Journal of Healthcare Engineering*, 2024, 5556493. <https://doi.org/10.1155/2024/5556493>.
- [24]. Patel, V., & Mishra, K. (2023). Privacy-preserving IoT-enabled healthcare systems using blockchain: A survey. *Computer Communications*, 206, 112-129. <https://doi.org/10.1016/j.comcom.2023.04.028>.