smart cities

Article

Cybersecurity in a Scalable Smart City Framework Using Blockchain and Federated Learning for Internet of Things (IoT)

Seyed Salar Sefati [1,2,*], Razvan Craciunescu [1], Bahman Arasteh [2,3], Simona Halunga [1,4], Octavian Fratu [1,4] and Irina Tal [5]

[1] Telecommunications Department, Faculty of Electronics, Telecommunications and Information Technology, National University of Science and Technology POLITEHNICA Bucharest, 060042 Bucharest, Romania
[2] Department of Software Engineering, Faculty of Engineering and Natural Science, Istinye University, Istanbul 34460, Türkiye
[3] Department of Computer Science, Khazar University, Baku AZ1096, Azerbaijan
[4] Academy of Romanian Scientists, 05044 Bucharest, Romania
[5] Lero, School of Computing, Dublin City University, D09 V209 Dublin, Ireland
* Correspondence: sefati.seyedsalar@upb.ro

Highlights:

What are the main findings?

- Implementation of blockchain enhances the security and scalability of smart city frameworks.
- Federated Learning enables efficient and privacy-preserving data sharing among IoT devices.

What are the implications of the main finding?

- The proposed framework significantly reduces the risk of data breaches in smart city infrastructures.
- Improved data privacy and security can foster greater adoption of IoT technologies in urban environments.

Abstract: Smart cities increasingly rely on the Internet of Things (IoT) to enhance infrastructure and public services. However, many existing IoT frameworks face challenges related to security, privacy, scalability, efficiency, and low latency. This paper introduces the Blockchain and Federated Learning for IoT (BFLIoT) framework as a solution to these issues. In the proposed method, the framework first collects real-time data, such as traffic flow and environmental conditions, then normalizes, encrypts, and securely stores it on a blockchain to ensure tamper-proof data management. In the second phase, the Data Authorization Center (DAC) uses advanced cryptographic techniques to manage secure data access and control through key generation. Additionally, edge computing devices process data locally, reducing the load on central servers, while federated learning enables distributed model training, ensuring data privacy. This approach provides a scalable, secure, efficient, and low-latency solution for IoT applications in smart cities. A comprehensive security proof demonstrates BFLIoT's resilience against advanced cyber threats, while performance simulations validate its effectiveness, showing significant improvements in throughput, reliability, energy efficiency, and reduced delay for smart city applications.

Keywords: blockchain; federated learning; IoT security; smart cities; data privacy

check for updates

Citation: Sefati, S.S.; Craciunescu, R.; Arasteh, B.; Halunga, S.; Fratu, O.; Tal, I. Cybersecurity in a Scalable Smart City Framework Using Blockchain and Federated Learning for Internet of Things (IoT). *Smart Cities* **2024**, *7*, 2802–2841. https://doi.org/10.3390/smartcities7050109

Academic Editors: Pierluigi Siano, Javier Prieto and Javier Parra Domínguez

Received: 14 August 2024
Revised: 26 September 2024
Accepted: 29 September 2024
Published: 1 October 2024

1. Introduction

Urban development projects around the world are increasingly focusing on creating smart cities [1], aiming to modernize infrastructure and improve public services with the help of the Internet of Things (IoT) [2]. IoT represents a network of technologies critical for developing smart city environments. It involves a broad network of sensors and devices that are interconnected and automated to enhance urban infrastructure efficiency and maximize the use of resources [3]. Smart cities utilize data from connected devices to improve sustainability and energy management, ultimately enhancing the quality of life for city residents [4]. Furthermore, IoT has a crucial role beyond city limits as well, enhancing connectivity in healthcare with smart devices and in agriculture through innovative farming technologies [5]. Unlike traditional networks that are mainly used for personal communication, IoT networks allow devices to share information with each other automatically, without human involvement [6]. This creates opportunities for advanced control systems, smart identification, accurate location tracking, and detailed monitoring, bringing in a new era of automated connectivity in both cities and rural areas [7].

The rapid development of smart cities has driven a growing reliance on IoT technologies to improve urban infrastructure, services, and overall quality of life [8]. However, significant challenges remain in addressing critical issues such as security, privacy, scalability, and efficiency in these systems [9]. While blockchain and federated learning (FL) have shown potential, most existing frameworks tend to focus on either security or privacy, without offering a comprehensive solution capable of handling the demanding requirements of large-scale, real-time IoT applications [10]. Additionally, the complexity of urban environments, where millions of interconnected devices generate vast amounts of diverse data, amplifies the need for innovative approaches that ensure secure data management, real-time processing, and privacy preservation [11]. Current IoT frameworks often suffer from problems like high energy consumption [12], limited scalability [13], and susceptibility to cyberattacks, particularly in smart city applications [14]. Moreover, the evolving nature of blockchain and FL technologies, coupled with the absence of standardized regulatory frameworks, adds to the complexity of their deployment in IoT systems [15]. The lack of solutions capable of addressing security, scalability, and privacy concerns simultaneously highlights a significant gap in the existing research [16]. Thus, there is a crucial need for novel frameworks that integrate these technologies, providing secure, scalable, and efficient solutions for smart city IoT systems [17]. Despite ongoing advancements, current frameworks still fail to effectively balance real-time data processing, decentralized security, and privacy mechanisms in handling the vast and heterogeneous datasets generated in urban environments.

Existing IoT frameworks often rely on either blockchain for secure data management or FL for privacy-preserving analytics [18], but they typically struggle to meet the scalability and efficiency demands of large-scale [19], real-time smart city applications [20]. In this paper, the Blockchain and Federated Learning IoT (BFLIoT) framework addresses this gap by integrating both technologies into a unified solution. In the first step, the system begins by collecting real-time data—such as traffic flow, environmental conditions, and public safety information—from various IoT devices. The data is normalized for consistency across different sources, encrypted to ensure privacy, and securely stored on the blockchain. To manage data access and security, the Data Authorization Center (DAC) employs a cryptographic framework, utilizing bilinear pairings and secure hash functions to generate secret and public keys. This ensures that only authorized users can decrypt and access the data. Furthermore, Edge Computing (EC) devices are deployed to handle local data processing, reducing the computational burden on central servers and enabling efficient management of the large data volumes typical in smart cities. FL is then applied to the encrypted data, allowing distributed IoT devices to collaboratively train machine learning (ML) [21] models while preserving local data privacy. The primary contributions of this paper are as follows:

- **Framework Design**: The BFLIoT framework represents the first comprehensive integration of blockchain and federated learning, addressing the unique security and scalability challenges of smart city IoT systems. The framework's security is underpinned by a foundational proof based on the intractability of the Discrete Logarithm (DL) problem, which demonstrates its robustness against sophisticated cyber threats.
- **Comprehensive Performance Analysis**: Extensive simulations are conducted to evaluate the framework's performance across various smart city applications, focusing

on key metrics of Quality of Service (QoS) such as throughput, reliability, and energy consumption. This analysis provides valuable insights into the practical viability of the BFLIoT framework in real-world scenarios.

- **Scalable and Efficient Data Processing**: The method optimizes the placement and operation of EC devices, enabling efficient local data processing and reducing the reliance on central servers. This scalability is crucial for handling the large data volumes typical of smart city environments, ensuring that the system maintains real-time processing capabilities.
- **Advanced Anomaly Detection Framework**: The framework includes a dynamic anomaly detection system that adapts to evolving data patterns. The globally refined model from FL enhances the accuracy of detecting irregularities and potential security threats, improving the overall security and reliability of smart city operations.
- **Formal Security Verification**: The BFLIoT protocol's security is thoroughly verified using ProVerif, a tool for the formal verification of cryptographic protocols. This formal analysis confirms the framework's resilience against a wide range of cyber threats, establishing a high level of confidence in its security architecture.

The paper is organized as follows: Section 2 reviews significant recent literature on IoT security challenges and the integration of blockchain with AI. Section 3 identifies the main challenges and strategic gaps from these studies and discusses the proposed classification model and segmentation. Section 4 presents the proposed method. Section 5 presents security proof with an enhanced encryption scheme, along with a formal analysis using ProVerif (**version** 2.05). Section 6 focuses on the performance analysis, detailing the simulation setup and its findings. Finally, Section 7 concludes the paper and suggests directions for future research.

2. Literature Review

This section critically reviews the existing literature on enhancing IoT security through blockchain-based FL. It highlights key advancements, identifies current challenges, and explores emerging trends that contribute to the framework of this study.

2.1. Evolution of IoT Security Challenges

The rapid expansion of the IoT has revolutionized various sectors, such as healthcare, agriculture, and urban development, by enabling enhanced connectivity and automation [22]. Despite these advancements, IoT networks face several significant security challenges due to the diverse and resource-constrained nature of IoT devices. Many IoT devices have limited computational power, memory, and energy resources, making them vulnerable to a wide range of security threats, including unauthorized data access, device tampering, and service interruptions [23]. Traditional security methods, which were originally designed for conventional computing systems, often struggle to protect these networks from sophisticated cyberattacks that exploit these vulnerabilities [24]. One of the primary challenges in IoT security stems from the dynamic nature of IoT networks. IoT environments are characterized by frequent changes in device connectivity, where devices can join or leave the network at any time [25]. This constant flux complicates efforts to maintain a secure, stable, and scalable system as new devices introduce additional attack vectors and complicate the overall security architecture. Moreover, the heterogeneity of IoT devices—ranging from simple sensors to complex actuators—further exacerbates security issues. Ensuring seamless interoperability and secure communication between these diverse devices is a persistent challenge for existing IoT frameworks. In addition to these device-level issues, the massive amounts of data generated in IoT networks, particularly in smart city applications, present significant challenges for data integrity, privacy, and real-time processing [1]. IoT networks are prone to data breaches and cyberattacks that target sensitive information, such as personal data, healthcare records, or city infrastructure information [12]. The lack of robust security mechanisms in traditional IoT frameworks makes it difficult to ensure secure data transmission and storage, especially in large-scale networks.

2.2. Blockchain and AI in IoT Security: A Review of Recent Studies

Merlec et al. [26] introduced a Smart Contract-enabled Context-Aware Access Control (SC-CAAC) scheme specifically designed for Blockchain-enabled IoT systems. This approach uses context-aware access control models together with smart contracts to manage access permissions in real time. Taking advantage of blockchain's features like immutability (data cannot be changed), transparency, and decentralization strengthens security and privacy without the need for a central authority. This helps build trust because all access control policies and decisions are permanently recorded on the blockchain. However, there are some challenges, such as the complexity of setting up and managing blockchain and smart contracts. Additionally, as the number of IoT devices and access control policies increases, scalability might become an issue, potentially slowing down transactions and increasing costs.

CheSuh et al. [27] proposed to employ Blockchain and ML to enhance QoS in IoT applications and to optimize parameters like delay and throughput. This comprehensive approach significantly boosts security, data integrity, and QoS accuracy. However, its complexity may pose scalability challenges in large IoT networks, and the high computational demands of ML and Blockchain could limit its feasibility in resource-constrained settings.

Kiran Ray et al. [28] propose an Ownership Transfer Protocol (OTP) for IoT devices that utilize Physically Unclonable Functions (PUF) and blockchain technology to ensure secure ownership transfer. This protocol allows tracking and tracing of the smart objects within the supply chain without requiring a Trusted Third Party (TTP) and supports Partial Ownership Transfer (POT) for temporary ownership changes. It leverages the immutable nature of blockchain for distributed environment support and authenticates devices and parties involved in the transfer process. The OTP was evaluated using the Ethereum blockchain and the Scyther tool for security verification, showing resistance against common attacks and optimal gas consumption. The OTP provides a decentralized solution for IoT ownership transfer, enhancing security and authentication without a TTP. The implementation of Ethereum has proved to be practical and energy efficient. However, PUF technology might introduce complexity in implementation and scalability challenges for large-scale IoT ecosystems.

Moreover, Li et al. [29] proposed a privacy-preserving bidirectional (PB) option for blockchain-enhanced logistics IoT. This scheme supports smart contracts for data access control, ciphertext-policy attribute-based encryption for privacy protection, and hash functions for data integrity detection. It introduces a logistics routing selection algorithm that takes into consideration time efficiency, transportation cost, and workload and features a bidirectional choice strategy to offer more human-like services to both customers and express delivery sites. The security and performance analysis shows that PB-IoT provides strong data privacy and supports bidirectional choices, making it a comprehensive approach for addressing the issues of logistics privacy and chaotic access control mechanisms in IoT. However, the complexity and scalability of integrating blockchain, encryption, and smart contracts might pose challenges, especially in larger logistics networks.

Vishwakarma and Das [30] presented a Blockchain-Based Integrated Security System (BLISS), a comprehensive security solution designed for IoT applications. This work focuses on enhancing cybersecurity by providing efficient mechanisms for the identification, authentication, confidentiality, and integrity of IoT devices and data exchanges. BLISS employs smart contracts on blockchain technology to create trustful clusters of IoT devices, facilitating secure data exchange without the need for a Trusted Third Party (TTP). The system is implemented on a combination of Raspberry Pi 4 and desktop PCs, demonstrating significant improvements in computation and energy consumption, with reduced storage and communication overhead compared to existing schemes. The security analysis confirms BLISS's resilience against various IoT-specific cyber threats. While BLISS provides a robust security framework for IoT applications, the reliance on blockchain technology might introduce challenges related to scalability and latency, particularly in large-scale IoT networks.

Also, Singh and Dwivedi [31] introduced a novel Blockchain-Based Secure Autonomous Vehicular IoT (SAVIoT) Architecture with Efficient Smart Contracts, aiming to enhance the cybersecurity of Autonomous Vehicles (AVs) by using blockchain technology for secure data sharing across AV networks. The implementation utilizes Solidity for smart contracts and the Ethereum platform, with Ganache and Truffle for blockchain deployment and MATLAB for analysis. This architecture ensures AV information and network integrity by enforcing predefined rules for data exchange through smart contracts, thus improving safety and reliability. It also enhances AV cybersecurity through decentralized, rule-based data sharing, reducing vulnerability to cyberattacks. However, the complexity of blockchain and smart contracts might present scalability challenges in extensive AV networks.

Khan, Bourouis [32] proposed a Blockchain Hyperledger-enabled Healthcare Industrial Internet of Things (BHIIoT) to boost data security in e-healthcare systems by addressing the limitations of centralized server-based architectures, such as node connectivity failures and issues with parallel data sharing. It introduces a secure, distributed structure that employs blockchain-distributed ledger technology and NuCypher threshold re-encryption, significantly enhancing data management, network resources, and overall trust within a peer-to-peer environment. This system automates key processes like authentication, logging, and transactions through chain codes and offers a scalable solution for optimizing the lifecycle of medical Wireless Sensor Networks (WSNs). However, the integration's complexity, the need for extensive evaluation in large-scale applications to confirm its benefits over traditional methods, and the requirement for continuous updates to combat emerging security threats present challenges, particularly in terms of scalability, interoperability, and maintaining cutting-edge security measures.

Hu Xiong et al. [33] introduced an advanced privacy-preserving authentication protocol for heterogeneous IIoT systems, leveraging a proxy resignature mechanism to facilitate secure communication between ID-based and certificateless-based cryptosystems. This protocol addresses critical security requirements, such as mutual authentication, user anonymity, resistance to modification, replay, and impersonation attacks, while ensuring perfect forward secrecy, nonrepudiation, and compatibility across heterogeneous environments. The protocol's security is rigorously validated under the extended Computational Diffie-Hellman (eCDH) assumption in the random oracle model. Notably, it demonstrates a low computational cost and reduced communication overhead compared to existing methods. However, the added complexity of achieving cross-domain communication introduces additional computational overhead, marking the primary limitation of the proposed approach.

Zhong et al. [34]. proposed an identity-based broadcast encryption (IBBE) scheme for VANETs to address redundancies in one-to-many communication between the Trust Authority (TA) and multiple vehicles. The scheme introduces IBBE technology to enable the TA to generate a single fixed-length ciphertext for a group of vehicles, thus reducing encryption overhead and improving efficiency. Additionally, a proxy server is incorporated to convert IBBE ciphertext into identity-based encryption (IBE) ciphertext, further reducing decryption costs for new vehicles joining the communication. The proposed method is evaluated through comprehensive security analysis and experimental results, demonstrating enhanced communication efficiency and reduced encryption redundancy. However, the use of a proxy server adds a layer of complexity, potentially introducing additional computational overhead.

2.3. Common Challenges and Strategic Gaps Identified across Studies

The analysis of blockchain's use in IoT security, as discussed in Section 2.2, reveals several recurring challenges and strategic gaps that hinder the effective integration of blockchain into IoT environments. Despite the promise of blockchain technologies, these limitations underscore the critical need for innovative solutions that can address scalability, efficiency, complexity, resource constraints, and interoperability in large-scale IoT applications, such as smart cities.

Scalability and Efficiency: The exponential growth of IoT networks—comprising billions of interconnected devices—has led to significant increases in data generation and transaction volumes. Traditional blockchain frameworks, particularly those relying on proof-of-work (PoW) mechanisms [30], struggle to handle this scale. Solutions like the SC-CAAC scheme [26] and the BHIIoT approach [32], while innovative, suffer from scalability bottlenecks, which result in delayed transactions and increased operational costs. These delays compromise the real-time functionality required by smart city applications, ultimately negating some of blockchain's core advantages, such as decentralization and transparency. The inability of current frameworks to efficiently scale for large, real-time IoT environments exposes a critical knowledge gap—the need for new architectures or consensus mechanisms that can handle large-scale IoT systems without compromising performance.

Complexity and Implementation Hurdles: Integrating blockchain into IoT networks introduces a high degree of complexity, particularly when combined with advanced technologies like ML and encryption. Solutions that aim to integrate these technologies often demand a robust technical infrastructure and require expertise in multiple domains. This creates a substantial barrier to entry, particularly in settings with limited resources. For instance, the BHIIoT framework [32], while addressing scalability, suffers from complexity in terms of deployment and ongoing management. This complexity makes it harder to use these solutions in real-world situations and slows down their wider adoption. The lack of simple and efficient frameworks shows the need for solutions that make it easier to set up and manage blockchain and IoT technologies.

Resource Constraints: IoT devices typically have limited processing power, memory, and energy resources, which make it challenging to directly implement blockchain solutions. For instance, systems like privacy-preserving logistics IoT [29] demonstrate how blockchain's cryptographic processes and ledger maintenance tasks can overwhelm IoT devices, leading to high energy consumption and computational demands. Although alternatives such as off-chain processing or lighter protocols have been proposed, they often come at the cost of reduced security or loss of blockchain's core benefits, such as immutability and transparency. This gap between the theoretical advantages of blockchain and its practical limitations in resource-constrained environments reveals the need for new solutions that can maintain security and efficiency while minimizing resource demands.

Interoperability and Standardization: The lack of interoperability between various blockchain platforms and IoT protocols presents another significant challenge. The heterogeneity of IoT devices and protocols, combined with the fragmented landscape of blockchain technologies, makes it difficult to achieve seamless communication across different systems. Existing frameworks, such as the BLISS security system [30], fail to fully address the need for standardized protocols and interfaces that would enable efficient cross-platform integration. This issue becomes particularly acute in large-scale IoT environments where diverse devices need to securely and efficiently share data. The absence of a unified framework that supports diverse IoT and blockchain technologies highlights a pressing need for solutions that promote interoperability and standardization.

Table 1 demonstrates the blockchain-enabled IoT innovations: strengths and limitations.

Table 1. Comparative analysis of blockchain-enabled IoT innovations: strengths and limitations.

Reference	Prior Studies	Advantages of Prior Studies	Disadvantages of Prior Studies	BFLIoT Contribution
[26]	SC-CAAC for Blockchain-IoT	Enhances security and privacy Promotes trust	Complex deployment Scalability and cost issues.	BFLIoT integrates federated learning for scalability and edge computing to reduce complexity.
[27]	Blockchain and ML for IoT QoS	Improves security and QoS Utilizes ML analytics.	Scalability issues. High computational demands.	BFLIoT optimizes energy efficiency and scalability through distributed edge computing and lightweight ML models.
[28]	OTP with PUF and blockchain	Secure flexible ownership transfer	PUF complexity. Scalability evolving m.	BFLIoT avoids hardware-based complexity by using cryptographic methods for secure and scalable data management.
[29]	PB-IoT for logistics	Enhances privacy; Supports humane services.	Integration complexity. High computational needs.	BFLIoT simplifies integration with a unified blockchain and federated learning model, reducing computational load.
[30]	BLISS for IoT	Improves cybersecurity; efficient mechanisms.	Scalability and latency in large network optimization are needed.	BFLIoT addresses latency by distributing processing across edge devices, ensuring real-time performance.
[31]	Secure AV IoT Architecture SAVIoT	Decentralized data sharing; improves safety.	Low scalability in AV networks. High resource demands.	BFLIoT's decentralized architecture handles larger networks efficiently, improving scalability and reducing latency.
[32]	BHIIoT for e-healthcare	Enhances data security and scalable.	Integration complexity; needs extensive evaluation.	BFLIoT's lightweight and scalable design simplifies deployment and supports real-time healthcare applications.
[33]	Computational Diffie-Hellman (eCDH)	Ensures secure communication between heterogeneous cryptosystems, low computation cost, and reduced communication overhead.	Extra computation cost due to cross-domain communication.	BFLIoT's recommend a low cost algorithm
[34]	Identity-Based Broadcast Encryption (IBBE) with Proxy Server for VANETs	Reduces encryption redundancy, improves communication efficiency, lowers decryption cost for new vehicles.	Complexity added by using a proxy server, introducing extra computational overhead	BFLIoT's reduces encryption redundancy in one-to-many communication

3. Problem Statement

As smart cities rapidly expand, they face complex challenges in managing traffic, monitoring the environment, and ensuring public safety. Traditional IoT systems that support these tasks are increasingly struggling with critical issues such as data privacy, security, scalability, and real-time efficiency [35]. With the rising number of IoT devices, the volume of data generated is becoming harder to process and analyze efficiently [36]. Current systems often lack the capability to provide secure, reliable connections and collaboration among the numerous IoT devices distributed across urban environments, resulting in inefficiencies and missed opportunities for optimized city management. Moreover, these conventional systems face difficulties in ensuring secure data transmission and privacy protection, especially as data travels through multiple points from sensors to end-users. This situation necessitates an advanced solution that can handle large-scale data in a secure, privacy-preserving manner while also being adaptable to the dynamic needs of urban infrastructures. To address these concerns, our proposed BFLIoT system integrates blockchain and federated learning to provide an enhanced, secure, and scalable solution. As illustrated in Figure 1, the system ensures end-to-end encryption, maintaining data integrity and security from the IoT sensors to the end-users while allowing efficient collaboration and data processing. This approach aims to overcome the limitations of traditional IoT systems, enabling smart cities to handle their data requirements more effectively and securely.

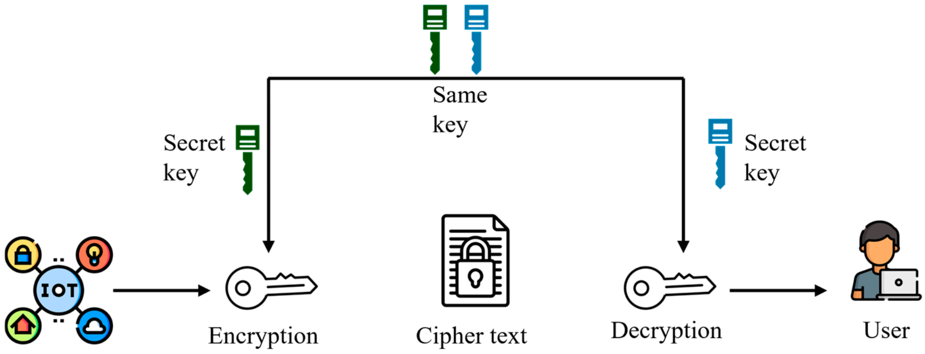

Figure 1. Conceptual framework for secure and scalable IoT integration in smart city infrastructure.

3.1. BFLIoT Framework and Segmentation

In the proposed BFLIoT framework, FL is a core component that enables decentralized model training across IoT devices. Rather than centralizing data from various sensors, FL allows each IoT device to train a local model on its own data, ensuring privacy. The local model updates are then shared with a central aggregator, where a global model is created and distributed back to the devices. This process enhances the system's predictive capabilities for tasks such as anomaly detection, traffic flow optimization, and public safety analysis, all without exposing sensitive data.

FL is learning patterns and trends within the data generated by the IoT devices. Each local model captures the unique environmental and operational characteristics of the IoT devices, such as traffic patterns in specific city regions or variations in air quality. The aggregation of these local models into a global model provides a holistic view of the city, improving the accuracy of predictions and decisions across the entire smart city infrastructure.

Necessity of FL in the BFLIoT Framework:

- **Privacy Preservation:** By keeping data local to the IoT devices, FL significantly enhances privacy. Sensitive data, such as personal or public movement patterns, is never transmitted to a central server, reducing the risk of data breaches and enhancing trust in the system.
- **Scalability:** With thousands of IoT devices continuously generating data, centralized systems would face significant challenges in handling this volume of information. FL decentralizes computation, minimizing the need for extensive server resources and allowing the system to scale effectively in large smart city environments.
- **Real-Time Anomaly Detection:** The BFLIoT framework is designed to adapt to changing conditions within the city. FL facilitates real-time updates to anomaly detection models, allowing the system to continuously improve its ability to detect cyber threats or system failures without compromising security.

Blockchain is a foundational component that ensures security, data integrity, and decentralization across the IoT ecosystem in smart cities. Blockchain plays a crucial role by providing a distributed ledger that stores encrypted data and model updates from IoT devices, enhancing trust, transparency, and resilience against cyber threats.

Necessity of Blockchain in the BFLIoT Framework:

- **Decentralized Data Storage and Integrity:** Blockchain stores encrypted data and FL model updates in an immutable, distributed ledger. Each IoT device contributes to the ledger by adding blocks containing encrypted data or model parameters. The distributed nature of Blockchain ensures that no single point of failure exists, enhancing the system's fault tolerance and security.
- **Tamper-Resistant and Transparent Transactions:** By using Blockchain, the BFLIoT framework guarantees that once data or model updates are recorded, they cannot be altered or tampered with. This immutability is critical for securing sensitive data and

ensuring that any malicious attempt to manipulate the system would be detectable by the decentralized network of nodes.

- **Secure Data Sharing through Smart Contracts:** Blockchain's smart contracts are leveraged to manage access control and automate data sharing across IoT devices. Smart contracts allow for automated execution of predefined rules, such as determining which entities can access certain data or model updates without relying on intermediaries. This automation not only ensures security but also improves system efficiency by reducing the need for manual intervention.
- **Validation of FL Updates:** Blockchain serves as the validation mechanism for FL model updates. Before updates are aggregated into the global model, they are verified and added to the Blockchain ledger, ensuring that only valid, secure contributions from authenticated devices are incorporated. This guarantees the trustworthiness of the learning process, preventing malicious data injections or model poisoning attacks.

This combination of FL and blockchain technology provides a robust, secure, and scalable solution for managing IoT systems in smart cities. This integration is detailed through a mathematical framework, demonstrating how data from IoT devices is processed, classified, and segmented to identify and mitigate security threats efficiently. Table A1 in the Appendix A shows the notation used further on for the mathematical description of classification and segmentation processes.

Given a dataset from IoT devices, let it be represented as Equations (1) and (2):

$$X = (x_1, x_2, \ldots, x_n) \tag{1}$$

$$\mathrm{Y} = (y_1, y_2, \ldots, y_m) \tag{2}$$

Here, X represents a vector containing data from all IoT devices. These vectors compile information such as device behavior, network traffic patterns, and resource usage. Y is a set of predefined categories used to classify the security status or threats to devices. These categories range from normal operation to various forms of compromised behavior, such as data exfiltration, unauthorized access, or malware infection. The classification model in Equations (3) and (4) f, which maps data points to their corresponding categories, is defined as follows:

$$f : X \to Y \tag{3}$$

$$y_i = f(x_i; \theta) \tag{4}$$

In our model, each data point x_i from the IoT in the dataset is mapped to a corresponding security status y_i using the classification function f. This mapping allows the model to classify different security statuses or threats for the devices, ranging from normal operation to various compromised behaviors such as data exfiltration, unauthorized access, or malware infection. Here, θ symbolizes the parameters of the ML model, which are tuned to optimize threat detection accuracy.

Equation (5) optimizes a loss function L is minimized, quantifying the difference between the predicted categories and the actual categories. The cross-entropy loss, commonly used for classification tasks, is defined as follows:

$$L(\theta) = -\textstyle\sum_{i=1}^{n} \sum_{j=1}^{m} z_{ij} \log(\hat{W}_{ij}) \tag{5}$$

Equation (6) indicate where z_{ij} is a binary indicator denoting if category j is the correct classification for observation i, and $\hat{W}_{ij}$ is the model's predicted probability of x_i belonging to category j. Segmentation is the process of dividing the dataset X into subsets $S_1, S_2, \ldots, S_p$ based on specific criteria, such as characteristics of the data or the categories predicted by the classification model.

$$g_{\text{seg}} : X \to S \tag{6}$$

This process aims to increase the homogeneity within segments and the heterogeneity between them. The segmentation function g assigns each data point x_i to a segment S_j, where j ranges from 1 to p (total number of segments). Thus, each segment is defined by Equation (7):

$$S_j = (x_i | g(x_i) = j \, , \, j = 1, 2, \ldots, p) \tag{7}$$

Optimization of the segmentation process involves minimizing the variance within each segment and maximizing the variance between segments, defined as follows for a segment S_j:

$$Var(S_j) = \frac{1}{|S_j|} \sum\nolimits_{x_i \in S_j} (x_i - \mu_j)^2 \tag{8}$$

The total inter-segment variance, denoted as Σ^2, is given by the following:

$$\Sigma^2 = \sum\nolimits_{j=1}^{p} |S_j| (\mu_j - \mu)^2 \tag{9}$$

Here, μ_j is the mean of the data points in the segment S_j, and μ represents the overall mean of the dataset. The classification model is optimized by adjusting θ to minimize the loss function $L(\theta)$, typically through gradient descent or its variants, to efficiently handle the large datasets and complex architectures common in IoT applications. The update rule for the parameters is given by Equation (10):

$$\theta^{(t+1)} = \theta^t - \alpha \nabla L(\theta^t) \tag{10}$$

where α is the learning rate, and $\nabla L(\theta^t)$ is the gradient of the loss function concerning the parameters at iteration t.

3.2. Why Blockchain over Traditional Encryption (e.g., HTTPS)?

In the proposed BFLIoT framework, Blockchain plays a fundamental role in securing data and ensuring transparency and trust, going beyond what traditional encryption methods like HTTPS provide. While HTTPS offers encryption for data in transit and at rest, it does not address key challenges related to data integrity, decentralization, trust, and tamper resistance, all of which are critical in a smart city IoT environment. This section explains why Blockchain is a superior choice over traditional encryption methods for this framework.

Immutability and Data Integrity: Traditional encryption methods like HTTPS secure data by encrypting it during transmission and storage, ensuring it is inaccessible to unauthorized parties. However, once the data reaches its destination, there is no built-in mechanism to prevent it from being altered or tampered with. Blockchain, on the other hand, provides an immutable ledger, meaning once data is recorded in a block, it cannot be altered or deleted without being detected. This ensures data integrity over time, making it especially valuable in environments where the authenticity and accuracy of data are paramount.

Blockchain Advantage: Blockchain guarantees immutability, ensuring that all transactions and data entries are tamper-proof and verifiable. This provides a level of trust and security that traditional encryption methods do not inherently offer.

Decentralization and Trust lessness: HTTPS relies on centralized servers to manage encryption keys and validate data exchanges. This creates single points of failure and requires trust in the central authority managing the server. In contrast, Blockchain is a decentralized system where multiple nodes in the network participate in validating transactions. This trustless environment ensures that no single entity has control over the data, and the system can continue to operate even if some nodes fail or are compromised.

Blockchain Advantage: The decentralized nature of Blockchain removes reliance on any single trusted party, providing greater resilience against failures and attacks and making the system more robust and reliable in a large-scale IoT environment.

Tamper-Resistant and Transparent Data: Encryption alone does not provide mechanisms for auditing or ensuring transparency. HTTPS can secure data during transmission, but it does not offer a method to track or verify the history of data once it has been exchanged. Blockchain's distributed ledger records every transaction in a transparent and traceable manner. Each transaction is linked to the previous one, creating a chronological chain of records that can be audited at any time, ensuring accountability.

Blockchain Advantage: Blockchain provides real-time transparency and traceability, allowing any participant to audit the history of data and ensure that no unauthorized changes have been made. This is particularly important in smart city IoT systems, where regulatory compliance and data accountability are critical.

Consensus-Based Validation: With HTTPS, while data is encrypted during transmission, it is up to the central server or authority to verify the validity of the data. This can lead to vulnerabilities if the central server is compromised. Blockchain uses a consensus mechanism (such as Proof-of-Stake) to validate transactions before they are recorded. This ensures that only valid, authenticated data is added to the Blockchain, providing a much stronger validation process than simple encryption.

Blockchain Advantage: The consensus mechanisms in Blockchain ensure that all data added to the ledger is verified and trusted, preventing malicious actors from injecting false or harmful data into the system.

Smart Contracts for Automated Access Control: Traditional encryption methods like HTTPS do not offer automated mechanisms for controlling how and when data is accessed. Blockchain enables the use of smart contracts, which are self-executing pieces of code that can enforce rules and policies. These smart contracts can automatically grant or deny access to data based on predefined conditions, ensuring secure, automated data management without the need for manual intervention.

Blockchain Advantage: Smart contracts provide a higher level of automation and security in managing data access, reducing human errors, and increasing the overall efficiency of the system. They also help in enforcing security policies dynamically.

3.3. Mathematical Formulation for QoS in BFLIoT Systems

A structured and professional mathematical formulation has been developed to establish a connection between IoT data characteristics and QoS parameters within the BFLIoT system context. This involves defining a series of mathematical expressions and optimization problems aimed at capturing the operational efficiency, security, and scalability of the system. Through this approach, a rigorous analysis and optimization of the system's performance can be achieved, ensuring a comprehensive understanding and enhancement of its capabilities.

Data throughput: In the context of the IoT, throughput refers to the efficiency and speed at which transactions and data exchanges are processed within the network, which is crucial for maintaining real-time learning and decision-making capabilities across a vast array of connected devices. Equation (11) defines the throughput:

$$\tau = \frac{\sum_{i=1}^{n} |d_i|}{T} \tag{11}$$

- τ: Represents the throughput of the system, typically measured as the amount of data processed per unit of time.
- $\sum_{i=1}^{n} |d_i|$: represents the summation of the absolute values of data (d_i) processed, from $i = 1$ to n, where i indexes each data transaction or piece of data processed, and n is the total number of transactions or data pieces processed in the given period.
- T: Represents the total period over which the throughput is measured. This could be in seconds, minutes, hours, etc., depending on the context of the measurement.

Energy efficiency: In a blockchain-integrated FL framework for the IoT, managing energy consumption involves optimizing computational processes, communication protocols, and data handling to ensure the system's sustainability, efficiency, and cost-effectiveness despite the inherent challenges posed by the diversity and energy limitations of IoT devices. We define energy efficiency as the ratio between the consumed energy (E_c) and the total quantity of data processed, as expressed by Equation (12):

$$E = \frac{E_c}{\sum_{i=1}^{n} |d_i|} \tag{12}$$

Reliability: In the context of the IoT, it represents the system's ability to operate correctly and consistently over time without failures. This metric is crucial in distributed networks where consistent operation is vital for data integrity, user trust, and overall system performance. Equation (13) defines the reliability:

$$R = e^{-\lambda t} \tag{13}$$

- R stands for the system's reliability over time, indicating the probability of failure-free operation throughout a specific period t.
- λ represents the failure rate of the system, which quantifies the frequency of failures per unit of time. A lower failure rate corresponds to higher reliability.

Latency: Latency, in the context of networked systems, typically shows the time necessary for a data packet to travel from its source to its destination. In the framework of a secure and scalable blockchain-integrated framework for FL across the Internet of Things (IoT), latency is a critical metric that can significantly impact the overall performance and responsiveness of the system. The equation for latency can be represented as follows:

$$LT = \frac{D}{S} + P \tag{14}$$

- LT denotes the latency, measured as the total time taken for a data transaction.
- D represents the distance traveled by the data packet, which can be the physical distance between devices in an IoT environment.
- S is the speed of the data transmission, which can be influenced by the medium of transmission (e.g., fiber optics, wireless) and the bandwidth of the network.
- P accounts for the processing time required at each node the data packet encounters, including delays introduced by routing decisions, data processing, and any queuing that may occur within the network infrastructure or the blockchain system itself.

4. Proposed Method

In this smart city framework, blockchain plays a crucial role in ensuring security, transparency, and data integrity. It acts as a decentralized, unchangeable ledger that securely stores encrypted IoT data and FL model updates, preventing unauthorized access or tampering. By removing the need for a central authority, blockchain reduces the risk of cyberattacks and creates a trustless system where all interactions are verifiable and transparent. The integration of blockchain with FL means that encrypted model updates from IoT devices are validated and securely stored, making them immutable. Additionally, smart contracts automate data access permissions and enforce rules, improving efficiency.

The innovation here comes from combining Blockchain and Federated Learning. While FL keeps raw data private by processing it locally, Blockchain adds an extra layer of security by ensuring all updates are verified and protected from manipulation. This hybrid approach offers a more scalable and secure solution for smart city infrastructures, tackling key issues like data security, privacy, and scalability more effectively than traditional centralized systems. By leveraging both technologies, the framework delivers a secure, transparent, and efficient way to manage the vast amount of data generated in smart cities.

In a smart city environment, a sophisticated BFLIoT system is designed to optimize traffic flow, monitor environmental health, and ensure public safety, all while safeguarding data privacy and integrity. The system incorporates a plethora of IoT devices that collect data continuously. This data is then securely integrated into a blockchain, analyzed through

ML models on edge devices, and refined via FL to improve city-wide operational efficiency and security. The notations used in the mathematical description are summarized in Table A2 in the Appendix A. In this section, we will describe the proposed method step by step.

Step 1: Data Collection and Processing

IoT devices collect varied data, including traffic patterns, air quality indexes, and public space occupancy rates. This data is initially processed to normalize and encrypt before it's recorded on the blockchain. Normalization ensures that data from different sources is on a uniform scale, facilitating accurate analysis. Encryption protects the data's privacy and integrity before it is securely recorded on the blockchain. The normalization process can be represented mathematically as Equation (15):

$$X_{\text{norm}} = \frac{X - \mu_X}{\sigma_X} \tag{15}$$

Step 2: Cryptographic Foundations

The DAC sets up a cryptographic framework to ensure the security of IoT data through encryption and controlled access mechanisms. This framework is crucial for protecting data from unauthorized access and cyber threats, employing advanced encryption techniques and secure hash functions. It enables the safe transmission and storage of sensitive information across the IoT ecosystem, ensuring data integrity and confidentiality.

- The choice of a bilinear map e a pairing function over cryptographic groups G and G_T, along with the choice of a prime number q, underpins the robustness of our encryption scheme, enabling secure interactions within the framework. Additionally, specifying four secure hash functions for distinct aspects of the encryption process further tailors our cryptographic measures to address diverse security requirements, ensuring comprehensive data protection across the IoT infrastructure.
- The allocation of hash functions within our cryptographic framework plays an important role in enhancing data security and integrity. Four secure hash functions are defined for their specific roles in the encryption process:
    - $H_1 : (0,1)^* \rightarrow \mathbb{Z}_q^*$ maps binary strings to integers within $\mathbb{Z}_q^*$, facilitating secure numerical operations.
    - $H_2 : (0,1)^* \rightarrow G$ transforms binary strings into elements of the cryptographic group G, ensuring that data can be securely embedded within this group.
    - $H_3 : G_T \rightarrow (0,1)^*$ performs the inverse operation, converting group elements back into binary strings, which is essential for data retrieval and processing.
    - $H_4 : (0,1)^* \rightarrow (0,1)^*$ is designed to maintain data integrity, providing a reliable mechanism for verifying the unaltered state of data throughout the encryption and decryption processes.

The DAC generates a secret key s from $\mathbb{Z}_q^*$ and computes the public key $P_{\text{pub}} = g^s$, essential for the encryption of IoT data. This public key plays a critical role in the encryption of IoT data, ensuring that only those with proper authorization can decrypt and access the information.

The normalized IoT data (X_{norm}) is hashed using H_4, to produce a hash value for integrity verification. The encrypted data ($Data_{\text{encrypt}}$) includes both the original normalized data and its hash value, ensuring that any tampering can be detected. Thus, the encrypted data is expressed as Equation (16):

$$Data_{\text{encrypt}} = \text{Encrypt}\Big(X_{\text{norm}} || H_4(X_{\text{norm}}), P_{\text{pub}}\Big) \tag{16}$$

This allows for both the original normalized data and its hash value to be encrypted together using the public key P_{pub}. The encrypted data C is securely uploaded to the blockchain, ensuring the integrity and confidentiality of the IoT information.

Step 3: Integration and Secure Operation of EC

In our system, the key challenge lies in securely integrating and managing a network of sensors and EC devices. In our simulations, we used 200 sensors, and 10 EC devices are addressed. The strategic placement of EC devices ensures efficient data processing from approximately 20 sensors each, optimizing load distribution and enhancing real-time data processing capabilities. The foundation of our security model is established based on a rigorous registration and authentication process for each EC device. Initiated through the secure transmission of a unique identifier by each EC device to the DAC, this process is formally represented as Equation (17):

$$EC \rightarrow DAC : EID_D \tag{17}$$

where the notation $EC \rightarrow DAC$ denotes that the EC device is securely sending data to the DAC, and EID_D represents the unique identifier of the EC device. The secure transmission of EID_D involves the EC device sending its unique identifier through an encrypted communication channel to the DAC. This identifier is used to authenticate the device and ensure that only authorized EC devices can access sensitive data. Once the DAC receives and verifies EID_D, it can register the EC device within the cryptographic framework, enabling subsequent secure communication. This authentication mechanism is crucial for ensuring that only verified EC devices are allowed to participate in the network, thus protecting the system against unauthorized access and potential security vulnerabilities. The EID_D facilitates identification and plays a crucial role in the subsequent stages of secure data encryption, transmission, and processing. Upon successful authentication, the DAC generates a unique, long secret key for each EC device via a secure procedure, establishing a key pair for secure data exchanges, as shown in Equation (18):

$$lsk_{esb} \leftarrow \text{Random}(f), \quad LPK_{esb} = g^{lsk_{esb}} \tag{18}$$

Here, lsk_{esb} represents the long secret key for the EC device, and LPK_{esb} denotes the public key component of the key pair, generated through a cryptographic function g. Using their unique secret keys, EC devices encrypt data before integration into the blockchain, subsequently enabling the secure and verifiable addition of new data blocks:

$$Data_{\text{encrypt}} = \text{Encrypt}(\text{data}, lsk_{esb}) \tag{19}$$

Store C in blockchain:

$$B_{\text{new}} = \text{Hash}(B_{\text{old}} \mid\mid \text{Hash}(Data_{\text{encrypt}})) \tag{20}$$

where C represents the encrypted data given in Equation (20), B_{old} denotes the previous block in the blockchain, and B_{new} signifies the newly added block. EC devices retrieve the encrypted dataset from the blockchain using access credentials, decrypt the data, and process it to derive actionable insights, as shown in Equations (21) and (22):

$$C_{\text{retrieve}}(EC) = \text{Blockchain}(Access\_Credentials_{EC}) \tag{21}$$

$$D_{\text{processed}} = \text{Process}_{EC}(\text{Decrypt}(Data_{\text{encrypt}}, lsk_{EC})) \tag{22}$$

where $C_{\text{retrieve}}(EC)$ is the process of retrieving encrypted data by the EC device using its access credentials, and $D_{\text{processed}}$ represents the data processed by the EC device after decryption with its private key lsk_{EC}. The rate of change of data flow from sensors to EC devices denoted as ΔD_{flow}, is analyzed using Equation (23) to identify sudden spikes:

$$\Delta D_{flow} = \frac{dD}{dt} \tag{23}$$

Moreover, the integrity of data transmission paths is verified through contour integrals, ensuring that the data's path through the network maintains its integrity from the source to the destination. This is mathematically represented as Equation (24):

$$\oint_C F \cdot ds = 0 \tag{24}$$

Furthermore, the concept of proportional perpendicularity is used to optimize the placement of EC devices relative to the sensors they manage, ensuring that data transmission paths are as direct as possible, minimizing latency and maximizing efficiency. Mathematically, this relationship can be defined as Equation (25):

$$\overrightarrow{r}_{EC} \propto \overrightarrow{r}_{sensor} \perp \overrightarrow{d}_{optimal} \tag{25}$$

where $\overrightarrow{r}_{EC}$ and $\overrightarrow{r}_{sensor}$ are the position vectors of the EC device and sensor, respectively, and $\overrightarrow{d}_{optimal}$ represents the direction vector that is perpendicular to the most efficient data transmission path.

Step 4: Federated Learning on Encrypted Data

The FL process in our system ensures the secure handling and updating of models using encrypted data. This approach allows multiple EC devices to collaboratively learn a shared prediction model while keeping all the training data local, thereby enhancing privacy and security. Encrypted datasets are processed locally on each device, and only model updates are shared across the network, ensuring sensitive information remains protected. Through this method, our system leverages the collective intelligence of distributed devices without compromising the confidentiality of the underlying data.

The formulas describe the encryption and key generation process within federated learning, ensuring security and privacy by using cryptography hash functions and random selections within a defined group, as well as generating session keys based on public and private keys. The security and privacy of the FL process are established through the following Equations (26) and (27):

$$EID_{D,b} = H_2(EID_b), \quad \omega \leftarrow_R Z_q^* \tag{26}$$

$$RSK_{a,b} = LPK_{esb}^{lsk_{esb}}, \quad S_{a,b}, y_b = g^{r_{a,b}}, \quad r_{a,b} = H_1(RSK_{a,b}, t_{\text{seq}}) \tag{27}$$

In the context of IoT, ω represents a randomly chosen value that strengthens the security of the encryption and key generation process for federated learning. The notation $\omega \leftarrow_R Z_q^*$ indicates that ω is randomly selected from the multiplicative group of integers modulo q, ensuring unique and unpredictable session keys that protect the privacy of IoT device data. Equation (27) describes the generation of shared key and secure values using public–private key pairs, cryptographic hash functions, and sequence identifiers to establish secure communication channels and authenticate data exchange. Encrypted data batches for model updates are represented as follows:

$$Data_{\text{encrypt}} = \text{Encrypt}(X_{\text{batch}}, \text{key1}) \quad h_{ed} = H_4(Data_{\text{encrypt}}) \tag{28}$$

$$C_0 = H_1(\text{Index}, k, t_{\text{seq}}, h_{ed}) \tag{29}$$

Equation (28) encrypts a data batch and then computes a hash value for the encrypted data, verifying its integrity through a secure hash function. Equation (29) generates a secure value using various identifiers, a key, and the hash of the encrypted data, which together help authenticate the data and ensure its secure association with an index and sequence number.

In our FL framework, gradient descent plays a crucial role in optimizing the shared model directly on encrypted data. This process allows each participating EC device to compute gradients based on its local dataset, without exposing the raw data to the network.

These local gradients are then securely aggregated to update the global model. This approach ensures that model updates are informed by the collective data of all devices, enhancing learning efficacy while preserving data privacy. The use of encrypted data for gradient computation and model updates represents a significant advancement in secure, distributed ML.

$$\theta_{(t+1)} = \theta_t - \eta \nabla L(\theta_t, \text{Decrypt}(Data_{\text{encrypt}}(lsk_{csb}))) \tag{30}$$

$$\Theta_{(t+1)} = \text{Aggregate}\Big((\Delta^{\theta^i}_{(t+1)})_{i \in ESA}\Big) \tag{31}$$

$$\mathcal{ED}_0 = \text{Encrypt}\Big(\Theta_{(t+1)}, k\Big), \quad \text{Blockchain write}(\mathcal{ED}_0) \tag{32}$$

Equations (30)–(32) describe updating model parameters using gradient descent with decrypted data, aggregating these updates across multiple devices, and securely encrypting the global parameters for storage on the blockchain. The FL paradigm is fundamentally designed to leverage data from multiple devices without compromising the privacy of the individual data sources. This approach needs a mathematical framework that can both symbolize the aggregation of data from diverse sources and facilitate a global optimization process that respects the privacy-preserving constraints of the system.

The aggregation of data from multiple devices, while preserving the privacy of individual datasets, presents a unique challenge. Traditional data aggregation methods that combine data into a single dataset are not suitable, as they may compromise data privacy. To address this, we utilize the mathematical concept of the disjoint union of datasets:

$$\bigcup\nolimits_{i=1}^{N} D_i \tag{33}$$

This representation allows us to conceptualize the collective contribution of data from each device as part of a unified learning process without physically pooling the data together. The disjoint union symbolizes the coexistence of datasets in the FL model, ensuring that each dataset remains distinct and secure. Furthermore, the global optimization goal in FL necessitates a method that can seamlessly integrate the contributions of all devices toward improving the shared model. This integration must be continuous and respect the privacy of the data. To achieve this, we formulate the 481 optimization goal as shown in Equation (34):

$$\theta^* = \text{argmin} \int_{\bigcup D_i} L(\theta, x)\, dx \tag{34}$$

This equation represents the continuous optimization of the model parameters, θ, by minimizing the integrated loss function, L, across the disjoint union of datasets, $\bigcup_{D_i}$. The integral here symbolizes a holistic evaluation of the model's performance over the aggregated data while maintaining the privacy and integrity of each dataset.

Step 5: Secure Handling of Gradients

In the FL process, the focus shifts to the secure handling of gradients to ensure that data privacy is preserved during the model aggregation phase. Gradients, which are derived from the local data on each EC device, contain sensitive information about the data itself. To safeguard this information, gradients are encrypted before their aggregation across the network. This encryption step is crucial for maintaining the confidentiality of each participant's data while still allowing the collective learning process to benefit from the insights contained within these gradients. By securely handling gradients in this manner, the system ensures that data privacy is maintained, while the integrity of the FL process is upheld.

$$G_{\text{local},i} = \nabla L(\theta_t, D_i) \tag{35}$$

$$Data_{\text{encrypt}} = \text{Encrypt}(G_{\text{local}}; PK_{\text{FL}}) \tag{36}$$

$$G_{\text{aggregated}} = \frac{1}{N} \sum\nolimits_{i=1}^{N} Data_{\text{encrypt}} \tag{37}$$

Equations (35)–(37) outline the steps in a FL process where local gradients derived from individual EC devices are first calculated, then encrypted using a public key to ensure data privacy, and finally aggregated across the network to update the global model. This sequence ensures that sensitive data remains confidential while allowing for the collaborative refinement of the model through securely shared insights.

Step 6: Global Model Update and Deployment

After the secure aggregation of encrypted gradients from all participating EC devices, these aggregated gradients are used to update the global model. This step ensures that the global model learns from the entire network's data without directly accessing or exposing any individual dataset. The updated global model then undergoes a secure deployment process back to each EC device. This deployment is carried out in such a manner that the integrity and confidentiality of the global model are preserved, ensuring that only authorized devices can access and utilize the updated model for further data processing and insights generation. This cyclical process of updating and deploying the global model allows for continuous learning and adaptation across the network, enhancing the system's overall intelligence and responsiveness to new data patterns and insights.

$$\theta_{\text{global}}^{\text{new}} = \theta_{\text{global}} + \Delta\theta \tag{38}$$

$$\Delta\theta = -\eta \cdot G_{\text{decrypted}} \tag{39}$$

$$Data_{\text{encrypt}} = \text{Encrypt}(\theta_{\text{global}}^{\text{new}}; PK_{\text{EDi}}) \tag{40}$$

$$\theta_{\text{global}}^{\text{new}} = \text{Decrypt}(Data_{\text{encrypt}}; SK_{\text{EDi}}) \tag{41}$$

Equations (38)–(41) describe the cycle of updating and securely managing the global model within a FL system. First, the global model parameters are updated by adding the decrypted gradient changes Equations (38) and (39), then these updated parameters are encrypted Equation (40) and sent back to each EC device, where they are decrypted Equation (41) to ensure that only authorized devices can access the updated model for ongoing learning and data processing.

Step 7: Anomaly Detection Framework

This framework facilitates the collective learning achieved through the FL process, enabling each EC device to utilize the updated global model for identifying anomalies in their local data streams. The model's ability to detect anomalies is a direct result of the diverse data it has been trained on, allowing for a robust and nuanced understanding of what constitutes normal behavior and what may be considered an anomaly. This anomaly detection framework is crucial for proactive monitoring and maintenance within IoT ecosystems, where the early detection of irregular patterns can prevent potential system failures or security breaches. By employing the global model in this capacity, the system enhances its operational efficiency and security posture, ensuring that anomalies are identified and addressed promptly. The real-time aspect of this anomaly detection underscores the dynamic and responsive nature of the system, which is continuously updated to reflect the latest data insights and threat intelligence.

$$Data_{\text{encrypt}} = \text{Encrypt}(H_4(X_{\text{norm}}), P_{\text{pub}}) \tag{42}$$

Equation (42) illustrates the encryption process used to secure the normalized data X_{norm} within the anomaly detection framework.

Specifically, $Data_{\text{encrypt}} = \text{Encrypt}\Big(H_4(X_{\text{norm}}), P_{\text{pub}}\Big)$ shows that the hash of the normalized data, $H_4(X_{\text{norm}})$, is encrypted using the public key P_{pub}. This step ensures that the data remains confidential and tamper-proof as it is used by each EC device to detect anomalies, thereby enhancing the security and integrity of the FL process in the IoT ecosystem.

In our anomaly detection framework, the dynamic threshold $\underline{T}$ plays a critical role in accurately identifying normal and anomalous data points, leveraging the global model's probability outputs. By dynamically adjusting the threshold based on the mean $\mu_{\text{anomalies}}$ and standard deviation $\sigma_{\text{anomalies}}$ of detected anomalies, the system ensures heightened sensitivity to evolving data patterns. This adaptability is crucial for maintaining detection accuracy in diverse and evolving operational environments, as it significantly reduces false positives and false negatives. Thus, the implementation of a dynamic threshold enhances the system's reliability and responsiveness, enabling prompt and effective responses to potential threats or operational anomalies. Classification of data points based on model probability output relative to threshold $\underline{T}$:

$$A(x) = \begin{cases} 1 & \text{if } p\left(x; \theta_{\text{global}}\right) \leq \underline{T} \\ 0 & \text{otherwise} \end{cases} \tag{43}$$

Dynamic threshold determination:

$$\underline{T}_{\text{new}} = \mu_{\text{anomalies}} + \lambda \times \sigma_{\text{anomalies}} \tag{44}$$

Step 8: Enhancing Federated Learning

This phase is essential for ensuring that the collaborative learning process across the network of EC devices remains efficient and upholds the highest standards of privacy and security. By encrypting model updates, before they are shared, we safeguard sensitive information from potential intercepts and unauthorized access. The aggregation process combines these updates to improve the global model, while ensuring that individual data contributions remain confidential. This step is critical in fostering a secure, collaborative environment where EC devices can contribute to collective intelligence without compromising their data or the integrity of the learning process. This enhancement of the FL process is instrumental in advancing the system's capabilities, facilitating a more robust, secure, and effective deployment of ML models across distributed networks.

$$\theta_{\text{local},i} = \text{Encrypt}(\nabla L(\theta_{\text{local},i}, D_i); PK_{\text{FL}}) \tag{45}$$

$$\theta_{\text{global}}^{\text{new}} = \theta_{\text{global}} + \frac{1}{k}\sum\nolimits_{k=1}^{K} \text{Decrypt}\left(G_{\text{encrypted}}^{k}, SK_{\text{FL}}\right) \tag{46}$$

Equations (45) and (46) in the FL process involve securely encrypting local gradients at each EC device before transmission and subsequently decrypting and aggregating these encrypted gradients to update the global model. This ensures that individual data remains confidential while collectively enhancing the model, promoting a secure environment for collaborative machine learning across distributed networks.

Step 9: Consensus Mechanism and Model Integration

This consensus mechanism is employed to achieve agreement among the participating EC devices on the validity of the aggregated model updates before they are committed to the blockchain. This step is important for maintaining a tamper-proof record of model evolution, ensuring that only verified and collectively agreed-upon updates enhance the global model. Integrating these updates into the blockchain secures the learning process against malicious attempts to alter the model and promotes transparency among participants.

$$\Theta_{\text{aggregated}} = \frac{1}{k}\sum\nolimits_{k=1}^{K} \text{Decrypt}(G_{\text{encrypted}}^{k()}, SK_{\text{FL}}) \tag{47}$$

$$B_{\text{validated}} = \text{Consensus}(B_{\text{current}}, \Theta_{\text{aggregated}}) \tag{48}$$

$$B_{\text{new}} = B_{\text{validated}} \oplus \Theta_{\text{aggregated}} \tag{49}$$

$$\Theta_{\text{global}}^{\text{new}} = \text{Broadcast}(B_{\text{new}}) \tag{50}$$

Equations (47)–(50) outline the process for securely updating a global model in a blockchain-based FL system. First, encrypted gradients from each EC device are decrypted and aggregated to form an updated global model, ensuring that all updates are secure and derived from authenticated sources. A consensus mechanism then confirms the validity of the aggregated updates before they are permanently recorded in a new blockchain block, which is subsequently broadcast to all network participants to synchronize the updated model across the system.

Step 10: Advanced Model Deployment and Data Decryption

This final step ensures that all participating EC devices receive the latest version of the global model, enabling them to leverage improved algorithms for data processing and anomaly detection. The deployment is conducted in a manner that secures the model against unauthorized access, maintaining the confidentiality of the collective intelligence developed through the FL process. Additionally, a complex data decryption process is introduced at this stage, allowing EC devices to securely decrypt and utilize the processed data. This decryption process is crucial for maintaining the privacy and security of the data as it is transmitted back to the devices for actionable insights. By incorporating advanced cryptographic techniques, the system ensures that only authorized devices can access the decrypted information, safeguarding against potential security breaches. Through this sophisticated deployment and decryption framework, the system enhances its capability to provide secure, accurate, and actionable insights across the network, driving informed decision-making and efficient operations.

$$\Theta_{\text{global}}^{\text{new}} \overset{\text{Deploy}}{\rightarrow} EC_i, \forall i \in \text{Network} \tag{51}$$

Data decryption process by EC devices, as shown in Equations (52)–(54):

$$\text{Verify} : H_3(C_{\text{enc}} \parallel (k_{\text{indexed}} \parallel t_{esa}) \parallel C_0 \parallel H_1(SK_{bi} \parallel Serv \parallel t_{esb})) = C_{\text{dec}} \tag{52}$$

$$\text{Decrypt} : C_{\text{dec}} = H_3\left(e(C_0, T_{K_{b_i}}) \parallel e(g, C_0)^{H_1(SK_{bi}\|Serv\|t_{esb})}\right) \tag{53}$$

$$\text{OriginalData} = \text{Decrypt}(C_{\text{dec}}, SK_{\text{bi}}) \tag{54}$$

In the proposed method, the updated global model, Θ global_new, is securely deployed to all EC devices within the network, ensuring each device is equipped with the latest algorithms for enhanced data processing and anomaly detection. This deployment process is fortified with advanced cryptographic measures to prevent unauthorized access and maintain the confidentiality of the collective intelligence developed through federated learning. Following deployment, the data decryption process begins, where a verification step using cryptographic hashes confirms the integrity of the encrypted data before any decryption occurs. Subsequently, the data is decrypted using complex cryptographic functions involving bilinear pairings and additional hash operations, tailored to ensure that only authorized devices can access the original data. This sophisticated deployment and decryption framework enhances the system's capability to provide secure, accurate, and actionable insights, thereby supporting informed decision-making and efficient operations across the network. Algorithm 1 presents the proposed method.

Algorithm 1: Proposed Method for a single node

Input: Collection of IoT Devices D, Set of Edge Computing Devices EC, Data Authorization Center (DAC)
Output: Securely processed data with high Quality of Service (QoS)

```
> Cryptographic Setup for DAC
if DAC. Setup Complete () == False then
    DAC. Initialize Cryptographic Parameters()
    for each hash Function in (H1, H2, H3, H4) do
        DAC. Configure (hash Function)
    end for
    DAC. Generate Public Key ()
end if

> Data Processing for Each IoT Device (Single Node)
for each device in D do
    if device. HasData () then
```

```
        Raw Data = device. Collect Data ()
        Normalized Data = Normalize Data (Raw Data)
        Encrypted Data = Encrypt Data(Normalized Data, DAC. PublicKey)
        Blockchain. Store(Encrypted Data)
    else
        Continue
    end if
end for

> Integration and Secure Operation of Edge Computing Devices (EC)
for each EC_device in EC do
    if EC_device. Is Registered With(DAC) == False then
        ECID = EC_device. Generate Unique Identifier()
        EC_device.Register(ECID, DAC)
    end if
    Key Pair = DAC. Generate Secure Key For (EC_device)
    EC.Store Key Pair(Key Pair)
end for

> Federated Learning on Encrypted Data
GM = Initialize Global Model()
for Round = 1 to Number Of Rounds do
    Local Models = []
    for each EC_device in EC do
        Encrypted Data = EC_device. Fetch Encrypted Data From Blockchain()
        if EncryptedData != None then
            Local Model = EC_device. TrainModel On Encrypted Data()
            Local Models. Append (Local Model)
        end if
    end for
    GM = Aggregate Models (Local Models)
end for

> Secure Handling of Gradients
for each EC_device in EC do
    Gradients = EC_device. Compute Encrypted Gradients()
    if Gradients. Is Valid() then
        Secure Gradients = Encrypt(Gradients, PublicKey_FL)
        Blockchain. Store (Secure Gradients)
    end if
end for
Aggregated Gradients = Aggregate Encrypted Gradients From Blockchain()
GM = Update Global Model (GM, Aggregated Gradients)

> Anomaly Detection Framework
for each EC_device in EC do
    Encrypted Data = EC_device. Fetch Encrypted Data From Blockchain()
    Decrypted Data = Decrypt(Encrypted Data, EC_device. Private Key)
    Anomalies = Detect Anomalies (Decrypted Data, GM)
    EC_device. Report Anomalies(Anomalies)
end for

> Consensus on Model Updates and Blockchain Integration
if Reach Consensus On (GM) then
    Blockchain. Update Global Model(GM)
else
    Log Error("Consensus not reached")
end if

> Advanced Model Deployment and Data Decryption
for each EC_device in EC do
    Deploy (GlobalModel_new, EC_device)
    Encrypted Data = EC_device. Retrieve Encrypted Data()
    if Verify Integrity (Encrypted Data) then
        Decrypted Data = Decrypt (Encrypted Data, EC_device. Private Key)
        EC_device. Process Data (Decrypted Data)
    end if
end for
```

5. Security Proof with Enhanced Mathematical Rigor

Security proofs serve as the backbone of cryptographic protocols [37], providing rigorous validation of a system's resilience against sophisticated cyber threats. This is particularly crucial in protecting the confidentiality, integrity, and availability of data within the ever-expanding digital landscape. The following sections delve into an advanced security proof for the BFLIoT system, emphasizing the intractability of the DL problem and its implications for cryptographic robustness. We assume the existence of a cyclic group G of prime order q, where the DL problem is presumed to be hard. Let g be a generator of G. The difficulty of finding $\log_g h$ for any $h \in G$ underpins the security of our cryptographic constructs. Under the assumption that the DL problem is intractable within group G, the BFLIoT system exhibits robust security against adaptive chosen-message attacks in the random oracle model, covering aspects of data encryption, blockchain integration, and FL processes. The proof employs the game-based approach [38], detailing interactions between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ within a polynomially bounded environment. We leverage cryptographic primitives and constructs, ensuring they collectively resist potential adversarial strategies aimed at compromising the system's integrity, confidentiality, and availability

Table A3 in the Appendix A shows the notations of the security proof and mathematical rigor.

5.1. Encryption Scheme and Security

The BFLIoT encryption mechanism is constructed as follows:

Key Generation

$\mathcal{C}$ selects $s \in \mathbb{Z}_q$ uniformly at random and sets the public key as $P_{\text{pub}} = g^s$. The secrecy of s is crucial for the scheme's security. Consider the encryption process for a message $m \in \mathbb{Z}_q$ given by Equation (55).

$$C = g^r || (m \oplus H(g^{rs})) \tag{55}$$

where $r \in \mathbb{Z}_q$ is selected randomly and uniformly for each encryption, and H is a cryptographic hash function acting as a random oracle. This H is the same as H_3 from the previous section, which serves as the specific hash function for encryption. The security of this encryption scheme can be analyzed through the following steps:

Semantic Security Indistinguishability under chosen-plaintext attack (IND-CPA)

The scheme aims to achieve indistinguishability under chosen plaintext attacks. This property ensures that an adversary cannot distinguish between the encryptions of two messages of their choice, even if they are allowed to choose the messages themselves [39].

Reduction to DL Problem

We claim that, if an adversary $\mathcal{A}$ can break the IND-CPA security of our scheme, then we can construct an algorithm $\mathcal{B}$ that solves the DL problem, thus contradicting our hardness assumption. Assume $\mathcal{A}$ is an adversary that can distinguish between the encryptions of two messages m_0 and m_1 with a non-negligible advantage. $\mathcal{A}$ chooses $m_0, m_1 \in \mathbb{Z}_q$ and sends them to the challenger. The challenger, simulating the role of $\mathcal{B}$, is given a DL challenge (g, g^x), where x is unknown. $\mathcal{B}$ must use $\mathcal{A}$ to solve this challenge. $\mathcal{B}$ simulates the encryption oracle for $\mathcal{A}$ using g^x as the public key. When $\mathcal{A}$ requests the encryption of a message, $\mathcal{B}$ generates a random r, computes g^r, and uses the random oracle model to simulate $H(g^{rs})$, even without knowing s. This simulation leverages the random oracle property of H.

1. Eventually, $\mathcal{A}$ outputs a guess for the encryption of m_0 or m_1. Since $\mathcal{B}$ can simulate the encryption oracle without knowing s (only using g^x), any advantage $\mathcal{A}$ has in distinguishing the encryptions directly translates into $\mathcal{B}$'s ability to compute g^x.
2. If $\mathcal{A}$ succeeds with a non-negligible advantage, $\mathcal{B}$ uses this advantage to solve the DL problem, contradicting our assumption that the DL problem is hard.

Random Oracle Model

The security of the scheme also critically relies on the hash function H being modeled as a random oracle. This idealization means that H behaves as a truly random function, where the output for each new input is indistinguishable from a random value from its output domain. The use of H in the encryption scheme ensures that the ciphertext component $(m \oplus H(g^{rs}))$ is secure against chosen plaintext attacks, as the output of H cannot be predicted or manipulated by an adversary.

Security Argument

We claim that the scheme is IND-CCA secure under the DL assumption. Specifically, given a DL challenge (g, g^x), an adversary $\mathcal{A}$'s ability to distinguish between encryptions of two chosen plaintexts implies the ability to solve for x, contradicting the DL assumption.

Federated Learning and Differential Privacy

Incorporating FL within the BFLIoT framework introduces unique challenges, particularly in ensuring the privacy and security of distributed model training. We use a differentially private mechanism, where each participating device adds noise to its model update before aggregation. Formally, for a local gradient G_i, the noise-adjusted gradient is as follows:

$$\overset{\sim}{G_i} = G_i + \mathcal{N}(0, \sigma^2 I) \tag{56}$$

where $\mathcal{N}(0, \sigma^2 I)$ denotes Gaussian noise. This ensures that the aggregated model adheres to (ϵ, δ)-differential privacy, significantly mitigating the risk of data leakage through model updates. By adding this Gaussian noise, the aggregated model adheres to differential privacy, significantly mitigating the risk of data leakage through model updates. This technique ensures that even if an adversary gains access to the aggregated model, they cannot infer sensitive information about the individual data points that contributed to the model.

Blockchain Integration and PoW

The integrity and non-repudiation of transactions within the BFLIoT system are maintained through blockchain technology. A critical component of this integration is the Proof-of-Stake (PoS) consensus mechanism, replacing the previously mentioned PoW due to its energy efficiency and scalability. The PoS mechanism is formalized as follows: Validators are chosen to create new blocks and validate transactions based on the number of coins they hold and are willing to "stake" as collateral. This mechanism significantly reduces the computational work required compared to PoW, making it more suitable for IoT environments with limited resources. The security of PoS hinges on the economic incentives and penalties designed to ensure honest behavior among validators. A critical component of this integration is the PoW consensus mechanism, which we formalize as follows: Let HF be a cryptographic hash function. A valid PoW is N such that

$$\text{HF}\ (N||block) < target \tag{57}$$

where *block* represents the data content and *target* defines the difficulty. The security of PoW hinges on the preimage resistance of *HF*, ensuring that finding a valid nonce requires computational work proportional to the difficulty.

5.2. Security Analysis

Confidentiality of Data Transmission and Storage

In modern IoT systems, particularly in complex environments such as smart cities, ensuring the security of data transmission, storage, and device communication is crucial. The BFLIoT framework integrates blockchain and FL to address key security challenges, safeguarding data from unauthorized access, tampering, and various forms of cyberattacks. This section provides a detailed analysis of the security mechanisms implemented in the BFLIoT system, covering aspects such as confidentiality, access control, anonymity, and resistance to common threats like replay, modification, and impersonation attacks. By comparing these features with other IoT security frameworks, we demonstrate the robustness of the BFLIoT system in maintaining secure and reliable operations in dynamic IoT environments.

The BFLIoT system ensures the confidentiality of data through encryption, both during storage and transmission, utilizing the equation:

$$C = \text{Encryp}(H_4(X_{\text{norm}}), P_{\text{pub}}) \tag{58}$$

In this equation, the encryption applies to the hash of the normalized data X_{norm}. This ensures that both the data and the hash are secured, preventing unauthorized access during storage and transmission.

Robust Access Control Mechanism

The system's access control is enforced through a two-tiered approach, ensuring secure data access as indicated by the function:

$$RSK_{a,b} = LPK_{esb}^{lsk_{esb}} \tag{59}$$

Anonymity Through Pseudonyms

The scheme introduces anonymity by assigning dynamically updated pseudonyms:

$$PID_i = LPK_i + (LPK_{esb})^{lsk_i} \tag{60}$$

Unlinkability of Device Requests

Each request incorporates a unique, anonymous secret key (ask_i) and pseudonym (PID_i), making it unlinkable: PID_i for subsequent messages remains unlinkable.

Replay Attack Resistance

The inclusion of a timestamp (t_i) in messages sent by devices enables the EC to determine the message's freshness:

If (t_i) falls within an acceptable range, the system can reject replayed messages.

Modification Attack Resistance

If a message fails cryptographic verification, it is discarded, thereby thwarting modification attacks. If message verification fails, the message is discarded.

Impersonation Attack Resistance

Unauthorized entities cannot impersonate legitimate devices or users, safeguarding against impersonation attacks. Unauthorized entities cannot generate a legitimate signature. Table 2 provides a comparison of security analysis across different IoT systems, including the proposed BFLIoT method and other systems such as SAVIoT [31], eCDH [33], IBBE [34], Federated Average (FedAvg) algorithm and DeepChain.

Table 2. Comparison-based security analysis.

Criteria	Proposed Method	SAVIoT	eCDH	IBBE	FedAvg	DeepChain
Confidentiality of Data Transmission and Storage	✓	×	✓	✓	✓	✓
Robust Access Control Mechanism	✓	✓	×	✓	✓	✓
Anonymity Through Pseudonyms	✓	×	×	×	×	×
Unlinkability of Device Requests	✓	×	×	×	×	×
Replay Attack Resistance	✓	×	✓	×	✓	✓
Modification Attack Resistance	✓	✓	×	✓	×	✓
Impersonation Attack Resistance	✓	×	✓	✓	✓	✓

5.3. Formal Analysis Using ProVerif

In the development of our smart city BFLIoT system, where the robustness and reliability of communication channels and data integrity are critical, we employed ProVerif to conduct a rigorous formal analysis [40]. ProVerif is an advanced automated tool designed specifically for the cryptographic verification of protocols, enabling us to model and validate the security attributes of our system comprehensively.

We constructed a detailed model of the BFLIoT system within ProVerif, encapsulating the intricate interactions between honest entities and potential adversaries. The model is comprehensive, incorporating the definitions of cryptographic primitives such as hash functions, symmetric and asymmetric encryption/decryption algorithms, and bilinear pairings. These elements are fundamental in securing communications between IoT devices within our blockchain-enabled framework. The model extends to delineate the full sequence of protocol operations, from the initial registration of devices through to the intricate processes of data transmission and subsequent blockchain integration. This extensive modeling ensures a holistic analysis that covers all potential security facets.

The verification process initiates with the definition of a threat model, outlining potential adversaries and their capabilities. Subsequently, we specify a series of security-related queries to assess properties such as confidentiality, authentication, integrity, and non-repudiation. ProVerif evaluates these properties by either proving each query or identifying counterexamples that reveal potential vulnerabilities. The results from ProVerif are summarized as follows, demonstrating the resilience of our BFLIoT system against the modeled threats:

Verification summary:

(1) Query not attacker(s) is true.
(2) Query not attacker(lsk(i)) is true.
(3) Query not attacker(lpk(i)) is true.
(4) Query not attacker(ask(i)) is true.
(5) Query not attacker(lsesb(i)) is true.
(6) Query not attacker(lskb(i)) is true.
(7) Query not attacker(m(i)) is true.
(8) Non-interference RIDi is true.
(9) Query inj-event(endES_Veri) ==> ==> inj-event(endSDi_Sig) is true.

These results demonstrate that our BFLIoT system's security properties, as modeled in ProVerif, withstand the assumed adversary model. For instance, the verification that "not attacker(s)" is true indicates that the adversary cannot deduce the secret key 's' from any observable communications or computations. Furthermore, the result "Non-interference RIDi is true" confirms that the system maintains non-interference regarding the RIDi variable, implying that operations involving RIDi do not interfere with other parts of the system, thereby upholding privacy and security guarantees.

Lastly, the injective agreement, denoted by the following *inj-event(endES_Veri) ==> inj-event(endSDi_Sig)*, verifies that if the end of an event related to the Edge Server verification is observed, then the corresponding start event for the Sensor Device signature must have occurred. This establishes a causal link between the events, ensuring that a verification event on the server side corresponds to an actual signature event on the device side, thus validating the integrity and authenticity of the communication process. Figure 2 illustrates the ProVerif verification process used to analyze the security properties of the BFLIoT system. The process begins with defining the protocol in Pi calculus and specifying the security properties to be verified. The Horn Clause Generator converts these definitions into logical formulas, which are then analyzed by the Trace Solver. Depending on whether attack traces are found, the outcomes indicate if the security properties are true, false, or unproven. This thorough analysis ensures that the BFLIoT system can withstand various modeled adversarial threats, confirming the robustness and integrity of the communication protocols and data security mechanisms within the system.

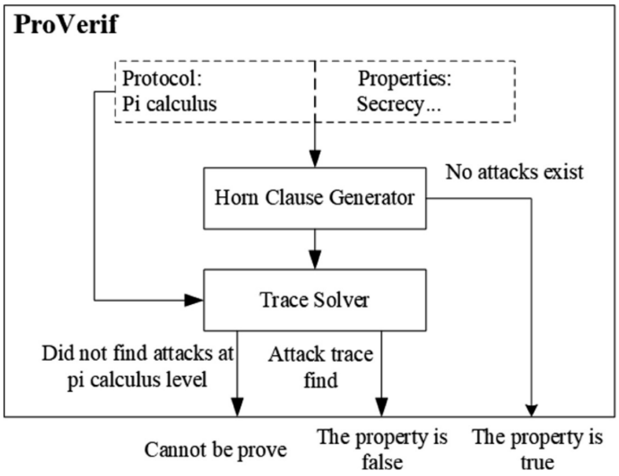

Figure 2. ProVerif verification process.

6. Performance Analysis

To conduct a pragmatic assessment of our method's accessibility, we introduce a security infrastructure using constrained embedded apparatus, referred to here as SMART, for sensor acquisition. This framework, featuring a modest 4 kB of data storage and 128 kB of program storage, efficiently establishes a mutable foundation of trust for distant embedded systems while maintaining security. It is designed to create a secure code execution environment that is completely segregated from pre-installed software, including the operating system, safeguarding against unauthorized alterations to critical code segments. Our simulation was conducted on a workstation with an Intel Core i3-4170 CPU (3.7 GHz), 8 GB of RAM, running Ubuntu 16.04 LTS, using the OMNet++ version 5.2 simulation toolkit. OMNet++ is a modular, component-based, open-source discrete event simulation software principally used for simulating various network types and their associated protocols. Cryptographic operation durations were obtained through experimental runs on a system using VC++ 6.0, coupled with the Pairing-Based Cryptography (PBC) library. For a fair comparison of all examined security mechanisms at an equivalent level of security—represented by the 1024-bit RSA encryption standard—we employ a bilinear map $e : G_1 \times G_1 \to G_2$, with G_1 being a group of order q generated by a base point P on a supersingular elliptic curve $y^2 = x^3 + x$, characterized by an embedding degree of 2 over a 64-bit field, $|G_1| = 128$ bytes and $|\mathbb{Z}_q^*| = 20$ bytes. The symbols $Cost_{\text{hash}}$, $Cost_{\text{pair}}$, $Cost_{\text{exp1}}$, $Cost_{\text{exp2}}$, and $Cost_s$ are designated to represent the computational costs. Table 3 shows the simulation parameters. Table 4 shows the symbol of computational cost for each function.

Table 3. Simulation parameters.

Parameters	Description
Deployment area	1000 m × 400 m
Number of users	2
Number of the cloud server	1
Number of sensors	40, 80, 120, 160, 200
Communication range of cloud server	2000 m
Communication range of sensors	20 m
Simulation time	2400 s
Cryptographic Library	VC++ 6.0 with Pairing-Based Cryptography (PBC) library
Security Standard	1024-bit RSA encryption

Table 4. Computational cost for each function.

Parameters	Description
$Cost_{\text{hash}}$	Computational cost for hash functions, essential for data integrity, digital signatures, random number generation, and hash tables.
$Cost_{\text{pair}}$	The computational cost for bilinear pairing operations is used in advanced cryptographic systems like identity-based encryption and zero-knowledge proofs.
$Cost_{\text{exp1}}$	Computational cost of exponentiation in G_1, a group on a supersingular elliptic curve, crucial for frequent cryptographic operations.
$Cost_{\text{exp2}}$	Computational cost of exponentiation in G_2, another group used for various cryptographic operations in different settings.
$Cost_s$	The computational cost for symmetric encryption/decryption, where the same key encrypts and decrypts data, is noted for its efficiency.

Performance Evaluation

In evaluating the BFLIoT system, emphasis was placed on its ability to handle transactions under different operational loads, a crucial aspect for smart city applications where data traffic can vary significantly. To assess this, the system was tested at two distinct rates: a moderate transaction rate of 328.4 Transactions Per Second (TPS) and a significantly higher rate of 3208.3 TPS. These rates were chosen to reflect typical scenarios in smart city environments, where systems must maintain efficiency under both regular and peak loads.

The performance results indicate that at a moderate rate of 328.4 TPS, the BFLIoT system achieved a throughput of 326.2 TPS, corresponding to a completion rate of approximately 99.33%. Under the higher load of 3208.3 TPS, the system maintained near-perfect performance, achieving 3208.2 TPS with a completion rate of 99.997%. The completion rate is calculated as follows:

$$Completion\ Rate_{328.4\ TPS} = \frac{326.2}{328.4} \approx 99.33\%,$$

$$Completion\ Rate_{3208.3\ TPS} = \frac{3208.2}{3208.3} \approx 99.997\%,$$

These results demonstrate that the BFLIoT system is highly efficient, maintaining performance levels well above 95% across both moderate and high transaction rates. This indicates that the system is capable of scaling effectively, meeting the demands of smart city infrastructures without significant performance degradation. The superior performance at the higher transaction rate can be attributed to the system's decentralized architecture, which leverages blockchain and federated learning. As transaction loads increase, the BFLIoT framework optimizes resource allocation and processing efficiency, resulting in better resource utilization and throughput. This scalability is particularly important for smart city applications, where large volumes of data need to be processed in real-time to ensure the smooth operation of urban services.

In the evaluation of the proposed SAVIoT architecture [31], focus was given to its performance in securing autonomous vehicular networks under different operational scenarios. The SAVIoT system's capability to handle secure transactions and data exchanges was examined at two distinct operational rates: A moderate rate of 210.5 TPS and a significantly higher rate of 2015.7 TPS. The performance results demonstrated that the actual throughput achieved at these rates was 208.3 TPS and 2015.4 TPS, respectively. The throughput completion rates are calculated as follows:

$$Completion\ Rate_{328.4\ TPS} = \frac{208.3}{210.5} \approx 99.045\%,$$

$$Completion\ Rate_{3208.3\ TPS} = \frac{2015.4}{2015.7} \approx 99.985\%,$$

In the evaluation of the eCDH [33] protocol within the IIoT environment, the focus was primarily on the protocol's ability to facilitate secure and private communication across heterogeneous systems. The eCDH method was assessed for its capability to authenticate transactions and data exchanges, considering operational scenarios that are critical for IIoT applications. The operational rates tested were a moderate rate of 150.4 TPS and a higher rate of 1584.6 TPS. The outcomes demonstrated that the actual throughput achieved at these rates was 148.7 TPS and 1583.9 TPS, respectively. The throughput completion rates are calculated as follows:

$$Completion\ Rate_{150.4\ TPS} = \frac{148.7}{150.4} \approx 98.867\%,$$

$$Completion\ Rate_{3208.3\ TPS} = \frac{1583.9}{1584.6} \approx 99.956\%,$$

In the analysis of the proposed IBBE [34] scheme within VANETs, attention was devoted to its innovative approach to minimizing redundancy in communications between a TA and multiple vehicles. The IBBE method's efficiency in handling broadcast encryption was scrutinized, particularly its capacity to manage secure, one-to-many message dissemination. Operational tests focused on two scenarios: a moderate operational scenario with a rate of 100.2 TPS and a more demanding scenario at 1024.5 TPS. The results indicated that the actual throughputs were 98.9 TPS and 1024.1 TPS, respectively. Also, Table 5 provides a summary of the metrics of the schemes.

$$Completion\ Rate_{100.2\ TPS} = \frac{98.9}{100.2} \approx 98.7\%,$$

$$Completion\ Rate_{3208.3\ TPS} = \frac{1024.1}{1024.5} \approx 99.961\%,$$

Table 5. Summary of the metrics of the schemes.

Metric	BFLIoT	SAVIoT	DeepChain	FedAvg	eCDH	IBBE
Moderate Rate Throughput	326.2	208.3	180.2	170.1	148.7	98.9
High-Rate Throughput	3208.2	2015.4	1700.3	1600.5	1583.9	1024.1
Throughput Completion Rate (Moderate, %)	99.330	99.045	98.972	98.914	98.867	98.700
Throughput Completion Rate (High, %)	99.997	99.985	99.962	99.960	99.956	99.961
Latency for Main Operation (Seconds)	0.069	0.082	0.087	0.090	0.095	0.120

In the evaluation of DeepChain within the context of smart city applications, the performance was assessed at two distinct rates: a moderate rate of 180.3 TPS and a significantly higher rate of 1602.5 TPS. The results revealed that DeepChain achieved a throughput of 178.9 TPS at the moderate rate and 1601.3 TPS at the higher rate. The completion rates were calculated as follows:

$$Completion\ Rate_{180.3\ TPS} = \frac{178.9}{180.3} \approx 99.22\%,$$

$$Completion\ Rate_{1602.5\ TPS} = \frac{1601.3}{1602.5} \approx 99.92\%,$$

Similarly, the Federated Average (FedAvg) algorithm was evaluated under the same operational conditions. At a moderate transaction rate of 170.2 TPS, FedAvg achieved a throughput of 169.1 TPS, while at a higher rate of 1590.8 TPS, it attained 1589.4 TPS. The completion rates for FedAvg were:

$$Completion\ Rate_{170.2\ TPS} = \frac{169.1}{170.2} \approx 99.36\%,$$

$$Completion\ Rate_{1590.8\ TPS} = \frac{1589.3}{1590.8} \approx 99.91\%,$$

Initial Configuration:

This configuration represents the baseline scenario with a moderate load on the system. The times for encryption, re-encryption, and decryption are calculated based on fundamental operations like message processing, hashing, and group theory-based multiplication. Table 6 presents the notations and descriptions for the comparative analysis of computational overhead in cryptographic operations across various BFLIoT scenarios.

Table 6. Notation and their descriptions.

Notation	Description
T_{enc}	Encrypt a message.
$T_{\text{re-enc}}$	Re-encrypt a message using proxy re-encryption
T_m	Time to process a message
T_e	Basic encryption time
T_h	Hashing time
T_a	Authentication time
T_{bp}	Base proxy re-encryption time
T_{mtp}	Message to proxy conversion time
T_{gtmul}	Group theory-based multiplication time

- Encryption time T_{enc} is calculated as $6T_m + T_e + 6T_h + 2T_a \approx 38.789$ ms. This calculation includes the time to process six messages, perform basic encryption, hash six times, and authenticate twice.
- Re-encryption time $T_{\text{re-enc}} = 3T_{bp} + T_m + 2T_{gtmul} \approx 38.451$ ms. Proxy re-encryption uses three base proxy operations, message processing, and two group theory-based multiplications.
- Decryption time T_{dec} is $4T_{bp} + (n+3)T_m + T_{mtp} + 4T_{gtmul} + 3T_h + (n+1)T_a \approx 325.895$ ms. Decryption involves multiple base proxy operations, message processing, message-to-proxy conversion, several group theory-based multiplications, hashing, and authentication steps. Here, n refers to the number of messages being decrypted, adding complexity to the process.

Second Configuration:

This configuration modifies the re-encryption and decryption processes, streamlining some parts to slightly improve performance in specific areas.

- Re-encryption time $T_{\text{re-enc}}$ is $2T_{bp} + T_e + 2T_m + T_{gtmul} \approx 38.091$ ms. This process reduces the number of base proxy operations from three to two. It also adjusts encryption time and message processing to optimize the re-encryption time.
- Decryption time T_{dec} is $2T_{bp} + T_{gtmul} \approx 108.874$ ms. Decryption in this configuration is simplified, involving only two base proxy operations and one group theory-based multiplication. This results in a much faster decryption time compared to the initial configuration.

Third Configuration:

This setup focuses on further reducing the encryption time while maintaining acceptable levels of re-encryption and decryption performance.

- Encryption time T_{enc} is $T_m + T_h + T_{gtmul} \approx 5.651$ ms. The encryption process is highly optimized, with minimal message processing, one hashing operation, and one group theory-based multiplication, resulting in a much faster encryption time.
- Re-encryption Time $T_{\text{re-enc}}$ is $2T_{bp} + 3T_m + 2T_e + T_h + T_a + 2T_{gtmul} \approx 48.723$ ms. Although encryption is faster, re-encryption remains more complex, involving two base proxy operations, three message processing steps, two encryption operations, one hashing operation, one authentication step, and two group theory-based multiplications.
- Decryption time T_{dec} is $2T_{bp} + T_{mtp} + 2T_{gtmul} \approx 112.276$ ms.

Decryption is slightly more complex than in the second configuration but still optimized. It requires two base proxy operations, message-to-proxy conversion, and two group theory-based multiplications.

Degraded Configuration 1:

This configuration highlights increased times due to heavier processing, perhaps due to system overload, larger message sizes, or less efficient components.

- Encryption time $T_{\text{enc}} = 6T_m + T_e + 6T_h + 2T_a \approx 48.789$ ms. Encryption becomes slower, possibly due to increased message size or processing demands.
- Re-encryption time $T_{\text{re-enc}}$ is $3T_{bp} + T_m + 2T_{gtmul} \approx 48.451$ ms.

Encryption becomes slower, possibly due to increased message size or processing demands.

- Decryption time T_{dec} is $4T_{bp} + (n+3)T_m + T_{mtp} + 4T_{gtmul} + 3T_h + (n+1)T_a \approx 425.895$ ms.

Decryption performance lags significantly, suggesting either larger messages or an increased number of operations per message.

Degraded Configuration 2:

This configuration further emphasizes the slowdown, reflecting worse performance metrics compared to previous setups.

- Re-encryption time $T_{\text{re-enc}}$ is $2T_{bp} + T_e + 2T_m + T_{gtmul} \approx 48.091$ ms. Despite reductions in base proxy operations, re-encryption remains slow.
- Decryption time T_{dec} is $2T_{bp} + T_{gtmul} \approx 208.874$ ms.

Decryption performance continues to deteriorate due to slower group theory-based multiplication or base proxy operations.

Degraded Configuration 3:

This configuration presents the slowest times across all phases, indicating further system inefficiencies.

- Encryption time T_{enc} is $T_m + T_h + T_{gtmul} \approx 15.651$ ms.
- Re-encryption time $T_{\text{re-enc}}$ is $2T_{bp} + 3T_m + 2T_e + T_h + T_a + 2T_{gtmul} \approx 58.723$ ms,
- Decryption time T_{dec} is $2T_{bp} + T_{mtp} + 2T_{gtmul} \approx 212.276$ ms.

Degraded Configuration 4:

The paper introduces an efficient and privacy-preserving authentication protocol designed for heterogeneous systems to secure communication between ID-based and certificate-based cryptosystems. Despite its focus on security, the system experiences a substantial degradation in performance across encryption, re-encryption, and decryption processes.

- Encryption time T_{enc} is calculated as $6T_m + T_e + 6T_h + 2T_a \approx 68.789$ ms.

The encryption time shows a marked degradation compared to earlier configurations, suggesting increased processing demands due to message size and additional cryptographic operations such as hashing and authentication.

- Re-encryption time $T_{\text{re-enc}}$ is $3T_{bp} + T_m + 2T_{gtmul} \approx 68.451$ ms.

Re-encryption also reflects a notable performance decrease, with proxy-based re-encryption and group theory-based multiplication slowing down the process significantly.

- Decryption time T_{dec} is $4T_{bp} + (n+3)T_m + T_{mtp} + 4T_{gtmul} + 3T_h + (n+1)T_a \approx 625.895$ ms.

Re-encryption also reflects a notable performance decrease, with proxy-based re-encryption and group theory-based multiplication slowing down the process significantly. Figure 3 shows the overhead of cryptography with another algorithms in same scenarios.

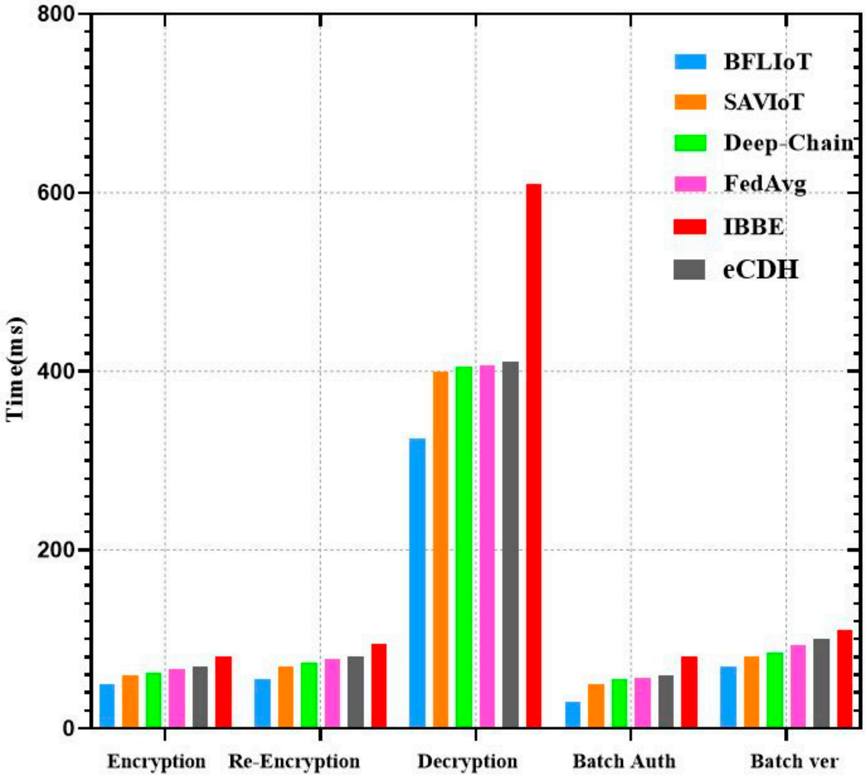

Figure 3. Comparative Analysis of Computational Overhead in Cryptographic Operations across BFLIoT Scenarios.

In assessing the reliability of various IoT security methodologies, it was observed that the BFLIoT approach consistently outperformed others, as indicated by the reliability scores across different sensor counts. The formula utilized to ascertain the reliability at any sensor count is given by $R = e^{-\lambda t}$, where R is the reliability score, λ is the failure rate, and t is the period considered.

For BFLIoT, at the sensor count of 200, the reliability score is 0.93, implying that the system exhibits a 93% probability of uninterrupted functioning over 10 h. This can be attributed to the decentralized nature of federated learning, coupled with the robustness of blockchain technology. Specifically, the failure rate λ is inversely related to the number of sensors; more sensors lead to a distributed and redundant data acquisition approach, enhancing the fault tolerance of the system:

$$\lambda = -\frac{1}{t}\ln(R)$$

For BFLIoT, the calculated failure rate λ_{BFLIoT} at 200 sensors is lower compared to other methods, indicative of a more resilient framework.

Figure 4 clearly illustrates how each security methodology performs under varying sensor densities, crucial for understanding their effectiveness in larger, more complex IoT networks.

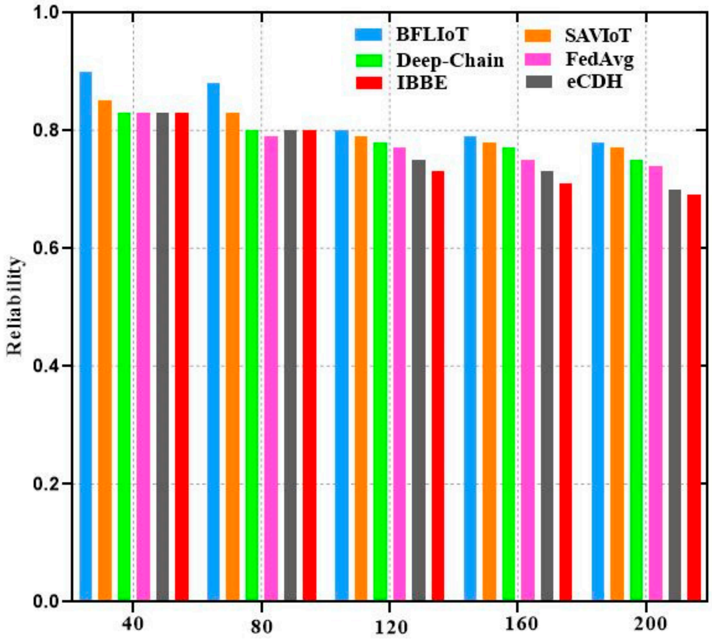

Figure 4. Reliability in differnet number of sensors.

The energy consumption results, as illustrated in Figure 5, reveal the comparative analysis of our proposed BFLIoT method against traditional security approaches and a scenario without security. Notably, the 'No security' approach demonstrates the lowest energy consumption across all sensor counts, which can be attributed to the absence of any encryption or advanced data processing overhead. The lack of security measures implies that there is no additional computational burden, resulting in minimal energy utilization. However, this also means that the data is susceptible to various cyber threats, rendering the system vulnerable to attacks.

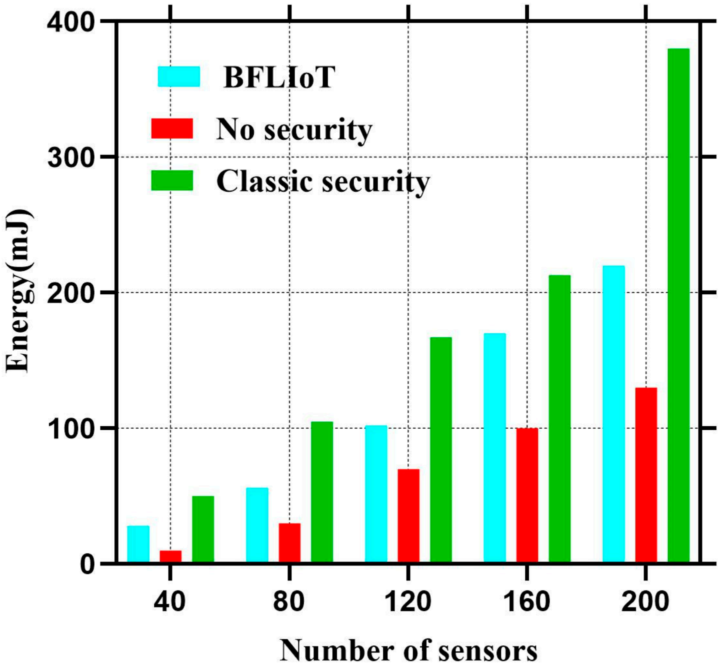

Figure 5. Energy consumption of different security methods as a function of the number of sensors.

On the other hand, the 'Classic security' method, which includes traditional security measures such as AES (Advanced Encryption Standard) for symmetric encryption, RSA (Rivest–Shamir–Adleman) for asymmetric encryption, and secure key exchange protocols like Diffie–Hellman, exhibits the highest energy consumption. These cryptographic operations are known to be computationally intensive. Performing symmetric/asymmetric encryption and decryption, key exchange protocols, and digital signatures for each data transmission and reception significantly increases the overall energy expenditure. While this method provides a high level of security, it is not optimized for scenarios where energy efficiency is a critical factor, such as in IoT devices with limited battery life.

Our proposed BFLIoT method is positioned between these two extremes. It leverages a hybrid approach that incorporates lightweight cryptographic operations and efficient FL processes, aiming to balance security and energy efficiency. By offloading some of the computational tasks to edge devices and utilizing blockchain for immutable record-keeping, BFLIoT reduces the energy consumption related to data encryption and transmission while still maintaining a robust security posture. Furthermore, the FL approach allows for localized model updates without the need to transmit large volumes of data, thus conserving energy. The result is a middle-ground solution that upholds data integrity and confidentiality while being more energy-conscious than classical security methods.

Figure 6 presents a comparative analysis of latency across various algorithms—BFLIoT, Deep-Chain, IBBE, SAVIoT, FedAvg, and eCDH—measured at different sensor counts (40, 80, 120, 160, and 200). It is evident that the BFLIoT system consistently outperforms other algorithms, demonstrating the lowest latency across all sensor counts, indicative of its efficient data handling and processing capabilities. In contrast, FedAvg and eCDH exhibit the highest latency, suggesting that they are less efficient in managing data transmission and processing in an IoT environment. As the number of sensors increases, most algorithms show a trend of increased latency, though BFLIoT maintains relatively stable and lower latency, highlighting its suitability for real-time smart city applications where low latency is crucial.

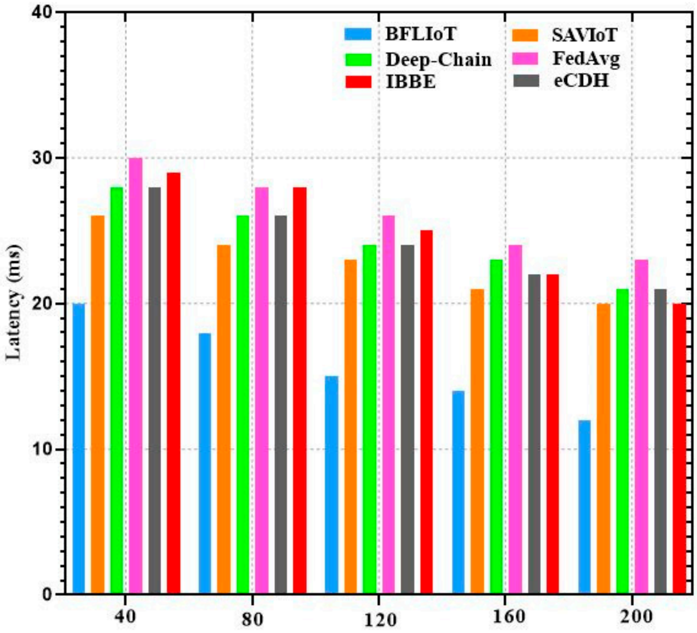

Figure 6. Latency in different numbers of nodes.

Figure 7 displays the model's accuracy, where both training and validation accuracy demonstrate significant fluctuations between approximately 0.80 and 0.96 across the epochs. This pattern reflects the dynamic nature of the IoT data, with accuracy stabilizing at higher values after the initial training phases. The fluctuations in the accuracy are consistent with the expected variability in data from heterogeneous IoT devices within smart cities. This behavior highlights the system's adaptive capacity, allowing for real-time data processing while maintaining a satisfactory accuracy rate. The BFLIoT framework's integration of FL ensures that the training data remains distributed across IoT nodes, thereby preserving privacy and improving scalability without sacrificing performance.

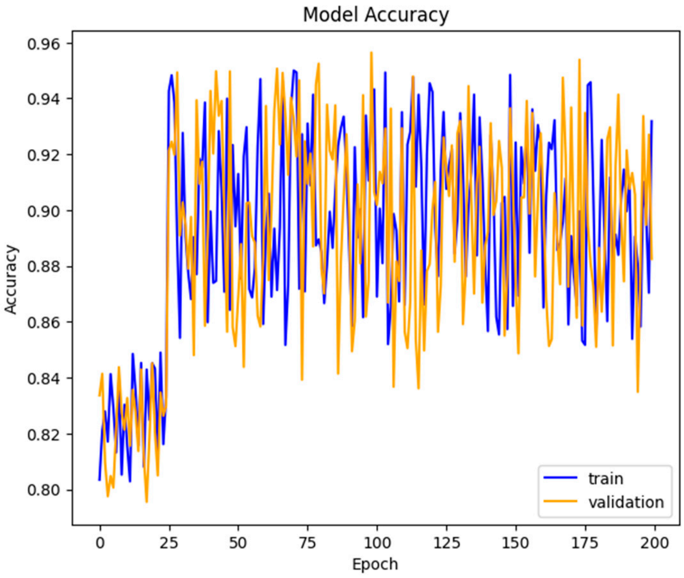

Figure 7. Model accuracy in different epochs.

Figure 8 depicts the model's loss, where both training and validation loss follow a similar trajectory. The loss decreases sharply during the initial 25 epochs, stabilizing with minor fluctuations for the remainder of the training process. This indicates that the model quickly converges to a low-error state, and subsequent fluctuations reflect the minor adjustments necessary to fine tune the model over time. These fluctuations correspond to the iterative updates typical of Federated Learning, where model parameters are optimized based on distributed data streams. The consistent reduction in loss validates the efficacy of the BFLIoT system in ensuring secure and scalable data handling without compromising model performance.

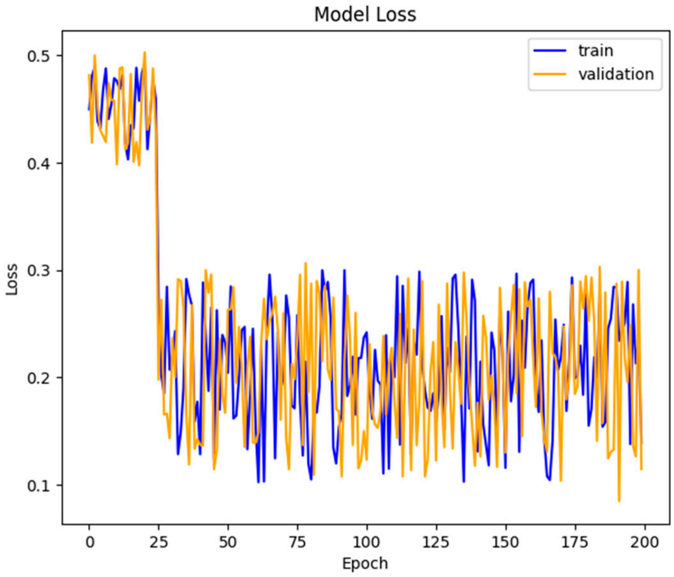

Figure 8. Model loss in different epochs.

7. Conclusions and Future Work

The BFLIoT framework presents a groundbreaking solution to enhance the security, scalability, and efficiency of IoT systems within smart city applications. By integrating blockchain technology with federated learning, BFLIoT decentralizes data processing, enabling real-time, collaborative data analysis while preserving privacy. This approach addresses the risks associated with centralized data repositories and ensures data integrity through the blockchain's immutable ledger system. Performance evaluations and security validations using ProVerif confirm BFLIoT's effectiveness in maintaining high security and operational standards, making it a viable model for modern urban infrastructures. The novelty of BFLIoT lies in its unique combination of blockchain and FL technologies to tackle the specific challenges of IoT environments. Unlike traditional centralized data processing systems, BFLIoT leverages the decentralized nature of blockchain to provide enhanced security and data integrity. FL ensures data privacy by keeping data localized on edge devices, with only model updates shared, significantly reducing data exposure risks compared to conventional cloud-based solutions. Compared to existing frameworks, BFLIoT offers superior scalability by distributing the data processing load across multiple nodes, avoiding bottlenecks and latency issues common in centralized architectures. The integration of ProVerif for security validation underscores the rigorous verification process BFLIoT has undergone, highlighting its robustness against potential threats.

Future work will expand BFLIoT's applicability beyond smart cities to sectors such as healthcare, agriculture, and industrial IoT, adapting to the unique challenges of these environments. Additional research will focus on enhancing energy efficiency by exploring advanced cryptographic methods and efficient data communication techniques to support the sustainability of IoT ecosystems. The BFLIoT system addresses the limitations of current IoT frameworks and offers a secure, scalable, and privacy-preserving solution. It represents a significant advancement in IoT technology, poised to impact various sectors and enhance urban infrastructure management.

Author Contributions: Conceptualization, S.S.S. and R.C.; methodology, S.S.S.; software, S.S.S.; validation, S.S.S., R.C. and B.A.; formal analysis, S.S.S. and I.T.; investigation, S.S.S. and I.T.; resources, S.S.S.; data curation, S.S.S.; writing—original draft preparation, S.S.S.; writing—review and editing, S.H. and O.F.; visualization, S.S.S.; supervision, O.F.; project administration, O.F.; funding acquisition, O.F. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Project "Mobility and Training for beyond 5G ecosystems (MOTOR5G)" funded by the European Union's Horizon 2020 Program under the Marie Skłodowska Curie Actions (MSCA) Innovative Training Network (ITN) under Grant 861219. The present work is also supported to the H2020-MSCA-RISE "Research Collaboration and Mobility for Beyond 5G Future Wireless Networks (RECOMBINE)" project with GA no. 872857.

Data Availability Statement: The code used in this study will be uploaded to GitHub upon acceptance of the paper. A link to the repository will be included in the final version of the paper.

Conflicts of Interest: The authors declare that there are no conflicts of interest regarding the publication of this paper.

Appendix A

Table A1. Notation in IoT device data classification model.

Notation	Description
X	Set of feature vectors from IoT devices, used to extract relevant information for security analysis.
x_i	Feature vector for the i-th device, encompassing data points like device behavior and network interactions.
Y	Set of predefined categories representing various potential security statuses of the devices.
f	Classification model mapping feature vectors X to categories Y, vital for determining security threat levels
y_i	The category assigned to the i-th device's data, indicating the security status as determined by model f
θ	Parameters of the classification model, are tuned to optimize threat detection accuracy.
L	Cross-entropy loss function for classification, measuring the model's performance in accurately classifying device data.

Table A1. *Cont.*

Notation	Description
z_{ij}	A binary indicator if category j is the correct classification for observation i, is used for training accuracy.
$\hat{W}_{ij}$	Predicted probability that i belongs to category j, indicating the likelihood of each security status.
n	Total number of IoT devices represented in the dataset, influencing the model's complexity and scalability
m	Total number of security categories, which define the granularity of threat assessment.
$S_1, S_2, \ldots, S_p$	Segments of the dataset, each defined by unique criteria to enhance model learning and detection capabilities.
g_{seg}	Segmentation function that assigns data points to segments based on their characteristics, improving model efficiency.
S_j	Segment containing data points categorized under security status j, used for focused analysis.
$g(x_i)$	The function g serves as the segmentation function, assigning each device's feature vector x_i to a specific segment S_j which groups data points based on similar characteristics such as behavior, security risk levels, or operational patterns.
$\text{Var}(S_j)$	Intra-segment variance for segment S_j indicating the consistency of data within a segment.
μ_j	Mean of data points within segment *S_j*, helping in normalization and comparison of segments.
p	P represents the total number of segments into which the dataset is divided.
μ	The overall mean of the dataset provides a baseline for comparing segment deviations.
$\theta^{(t+1)}$	Updated model parameters after iteration t+1, reflecting learning and adaptation to new data.
α	Learning rate used in the optimization algorithm, balancing speed, and accuracy of convergence.
$\nabla L(\theta^t)$	The gradient of the loss function concerning parameters θ at iteration t, guiding model updates.

Table A2. Notation of the proposed method.

Notation	Description
X_{norm}	Normalized data ensures uniform scale across IoT device inputs, crucial for accurate traffic and environmental analysis.
μ_X	Represents the mean of IoT device data, essential for assessing average traffic conditions or environmental quality.
σ_X	Standard deviation, indicating variability in IoT data, is useful for detecting anomalies in traffic or environmental conditions.
e	A bilinear map crucial for secure multi-party computations in traffic and environmental data sharing among IoT devices.
G, G_T	Cryptographic groups used for secure data operations, ensuring that traffic and environmental data remain tamper-proof.
q	Prime number defining the order of cryptographic groups, foundational for the security parameters of IoT data exchanges.
H_1	Hash function that maps binary strings to integers within $\mathbb{Z}_q^*$. Hash function that securely maps device identifiers to cryptographic values, protecting device identity in a smart city network.
$\mathbb{Z}_q^*$	$\mathbb{Z}_q^*$ refers to the set of all nonzero integers modulo q. In cryptographic terms, this represents the multiplicative group of integers modulo q excluding zero.
H_2	Hash function that maps binary strings to elements of group G. Hash function for securely embedding IoT data within cryptographic groups, vital for preserving data integrity.
H_3	Hash function that maps elements of G_T back into binary strings. Converts group elements back to strings, facilitating the secure retrieval of encrypted traffic and environmental data.

Table A2. *Cont.*

Notation	Description
H_4	Hash function used for verifying data integrity. Ensures data integrity by verifying that traffic and environmental data have not been altered post-encryption.
P_{pub}	Public key used for encryption and signature verification. Public key for encrypting IoT data, allowing secure data exchange across the smart city network.
g	Generator of the cryptographic group G. Generator of the cryptographic group, fundamental to the creation and management of encryption keys in IoT security.
s	Secret scalar chosen from $\mathbb{Z}_q^*$, used in public key generation. Secret key component in cryptographic operations, critical for maintaining secure communication between IoT devices.
C	Cipher text of encrypted data.
EID_D	Unique identifier of an IoT device, used for secure network authentication. Unique identifier for each IoT device, ensuring secure and authenticated device operations in smart city infrastructure.
$lsk_{esb} \leftarrow \text{Random}(f)$	Long secret key generated securely for each device through a random process. Randomly generated long secret key for each IoT device, enhancing the security of device-specific operations.
$LPK_{esb} = g^{lsk_{esb}}$	Long-term public key derived from the long sec ret key using generator g. Used in securing device communications within the IoT network.
Store C in blockchain	Encrypted data C is securely stored in the blockchain. Ensuring immutable recording of traffic and environmental data.
$B_{\text{new}} = \text{Hash}(B_{\text{old}})$	New blockchain block created from the hash of the previous block, a critical step for maintaining a secure, verifiable record of IoT data transactions.
$C_{\text{retrieve}}(EC)$	Method for IoT devices to securely access encrypted data from the blockchain, crucial for Edge Computing (EC).
$D_{\text{processed}}$	Represents the IoT data post-decryption, used for actionable insights into traffic flow and environmental conditions.
$\text{Decrypt}_{lsk_{EC}}(C)$	Decryption of cipher text C by the IoT device using its long secret key.
ΔD_{flow}	Measures the change in data flow, critical for monitoring variations in traffic density or environmental sensor outputs.
$\frac{dD}{dt}$	Rate of change of data, important for understanding trends in traffic congestion and environmental conditions over time
$\oint_C$	Contour integral over a closed path C, used in the context of data flow integration. Integral used to ensure the completeness and integrity of data paths in the IoT network.
F	Vector field integrated over path C, typically representing data flow or force fields. Represents forces or flows in vector fields, useful in simulations of traffic patterns and environmental dispersion models
ds	Differential path element along the contour C.
$\vec{r}_{EC}$	Position vector of the IoT device in the network. Essential for optimizing sensor placements and ensuring effective data coverage in a smart city.
$\propto$	Symbol indicating proportionality, used in algorithms that adjust IoT device operations based on traffic and environmental data scales.
$\vec{r}_{sensor}$	Position vector of the sensor relative to the IoT device. Sensor position vectors, key to strategically deploying environmental and traffic monitoring sensors for optimal data collection.
$\perp$	Denotes perpendicularity between vectors, used in optimal path calculations. Indicates perpendicularity in data transmission paths, crucial for minimizing interference and maximizing the efficiency of data flow in IoT networks.
$\vec{d}_{optimal}$	Optimal direction vector for data transmission between IoT device and sensor.
$EID_{D,b}$	Encrypted device identifier for IoT device b, ensuring confidentiality and integrity.
ω	Random element selected from $\mathbb{Z}_q^*$, used in cryptographic operations.
$RSK_{a,b}$	Rendezvous Secret Key, a cryptographic key for secure interactions between entities a and b.

Table A2. *Cont.*

Notation	Description
$S_{a,b}$	Secret session key for communication between devices *a* and *b*. Encrypted communication, vital for maintaining data confidentiality in IoT interactions.
y_b	Public ephemeral value associated with device *b* during a session.
t_{seq}	*t* stands for timestamp. *Seq*: Indicates that this timestamp is part of a sequence, which could be used to order events, messages, or data packets chronologically.
$r_{a,b}$	Nonce used once in a session between devices *a* and *b* for enhanced security.
ED	Encrypted Data, key for protecting sensitive information in traffic and environmental monitoring systems.
ENC_k	Represents the encryption function. The subscript *k* indicates that this function uses the cryptographic key *k* to perform the encryption.
(X_{batch})	This is the batch of data that is being encrypted. In the context of IoT systems or any large-scale data processing environment like a smart city BFLIoT system, data is often processed in batches for efficiency.
h_{ed}	Hash of encrypted data, providing a unique fingerprint for verification without revealing content. Providing a checksum to verify data integrity before decryption in IoT systems.
C_0	Initial commitment in cryptographic protocols, ensuring integrity and non-repudiation.
$\theta_{(t+1)}$	Updated model parameters after iteration $t+1$, typically in a learning or optimization context. Updated model parameters in machine learning algorithms, essential for adapting traffic control and environmental prediction models to new data.
η	Learning rate, controlling the update magnitude in optimization processes. Determining the speed and effectiveness of updates to IoT data processing models.
$\nabla L(\theta_t, \text{Dec}_{lsk_{csb}}(ED))$	Gradient of the loss function with respect to θ_t, calculated on decrypted data. Key for refining machine learning models based on secure IoT data.
$\Theta_{(t+1)}$	Aggregated model parameters after updates from all devices at iteration $t+1$. Critical for enhancing the collaborative intelligence of IoT devices in smart city applications.
$\Delta^{\theta^i}_{(t+1)}$	Parameter updates from the *i*-th device, contributing to the overall model update.
ESA	Set of all participating EC devices in the federated learning network. Central to distributed data processing and decision-making in smart cities.
$\mathcal{ED}_0$	Encrypted version of the aggregated model parameters for secure transmission.
$\bigcup_{i=1}^{N} D_i$	Disjoint union of datasets from *N* devices, representing data aggregation while preserving privacy. Maintaining data privacy while enabling comprehensive analysis in federated learning.
θ^*	Optimal model parameters obtained from minimizing integrated loss across aggregated data.
$\int_{UD_i} L(\theta, x)\, dx$	Integral of the loss function over the disjoint union of datasets, indicating continuous optimization.
$G_{\text{local},i}$	Local gradient computed on the *i*-th device, derived from local data and model parameters.
$\nabla L(\theta_t, D_i)$	Gradient of the loss function for local model parameters θ_t on dataset D_i.
PK_{FL}	Public key used in federated learning for encrypting data, ensuring participant data confidentiality.
N	Total number of devices participating in the federated learning network, indicative of the scale of collaborative data processing in smart city infrastructure.
$\frac{1}{N}$	Used to average aggregated values across all devices, essential for balancing model updates in federated learning systems.
$G_{\text{encrypted}}$	Encrypted local gradients, secured with public key PK_{FL}. Securing detailed traffic and environmental data during collaborative learning processes.
$G_{\text{aggregated}}$	Aggregated encrypted gradients, averaged across all participating devices. Averaged to update global models without compromising the privacy of individual IoT data inputs.
$\theta_{\text{global}}^{\text{new}}$	Updated global model parameters after applying aggregated gradient changes.

Table A2. *Cont.*

Notation	Description
$\Delta\theta$	Change to be applied to the global model, based on the decrypted aggregated gradients.
$\theta_{\text{encrypted}}^{\text{new}}$	Encrypted updated global model, ready for secure transmission to devices.
$\theta_{\text{global}}^{\text{new}}$	Decrypted updated global model, ready for deployment on edge devices.
SK_{Edi}	Secret key of a specific EC device, used to decrypt transmitted data.
PK_{Edi}	Public key associated with an EC device, used for encrypting data before transmission.
$H_4(X_{\text{norm}})$	Hash function applied to normalized data, part of security checks for anomaly detection.
$A(x)$	Anomaly detection function that classifies data points as normal or anomalous based on a threshold T. Crucial for identifying deviations in traffic patterns and environmental conditions in real-time.
$p(x;\theta_{\text{global}})$	Probability output of the global model for a data point x, used to assess anomaly status against threshold T.
T	Threshold for classifying data points in anomaly detection, adjusted dynamically.
T_{new}	Updated dynamic threshold for anomaly detection based on statistical measures of detected anomalies. Recalibrated to maintain accuracy as traffic and environmental conditions evolve.
$\mu_{\text{anomalies}}$	Mean of detected anomalies, used in dynamic threshold calculation.
$\sigma_{\text{anomalies}}$	Standard deviation of detected anomalies, used in dynamic threshold calculation.
λ	Scaling factor applied to the standard deviation in the calculation of the new threshold T_{new}.
$\theta_{\text{local},i}$	Encrypted local model update of the i-th device, using public key PK_{FL}. Ensuring secure and personalized adaptation to localized data conditions.
$E_{pk}^{-1}(G_{\text{encrypted}}^{k})$	Decryption of encrypted gradients, part of the global model update aggregation.
$\Theta_{\text{aggregated}}$	Aggregated model updates from participating devices, prior to consensus validation.
$\frac{1}{k}$	This represents the multiplicative inverse of the number k, used to calculate an average. In this context, it appears there might be a typographical error or confusion, as k is also used as the variable of summation. It is more conventional to see it as $\frac{1}{k}$ when computing an average, where K is the total number of items over which the sum is calculated.
$B_{\text{validated}}$	Blockchain record after consensus validation of the aggregated updates. Ensuring that all device contributions are authenticated and the model update is secure
B_{new}	Updated blockchain record incorporating the new validated updates.
$\Theta_{\text{global}}^{\text{new}}$	Broadcasted new global model state, synchronized across all network participants.
$\overset{\text{Deploy}}{\rightarrow} EC_i$	Represents the deployment process of $\Theta_{\text{global}}^{\text{new}}$ to each EC device, denoted by EC_i, across the entire network.
C_{enc}	C Stands for ciphertext, which is the output of an encryption process. *enc* Indicates that the ciphertext has been encrypted, specifying the state of the data as being securely encoded.
k_{indexed}	k Represents a cryptographic key, which is used for encryption, and decryption. *Indexed*: Implies that the key is part of a collection or series of keys, each uniquely identified by an index.
t_{esa}	The notation t_{esa} represents a specific timestamp in data communication contexts.
SK_{bi}	*SK* Stands for "Secret Key," which is used for decrypting data that has been encrypted with the corresponding Public Key. *bi* serves as an identifier and index for a particular device.
t_{esb}	t Stands for timestamp. *Esb* stands for a specific protocol.
C_{dec}	C Stands for ciphertext which is the data in its encrypted form. *Dec* Indicates that the ciphertext has been decrypted, specifying the state of the data as having been transformed from its secure, encoded format to its plaintext format.
$T_{K_{b_i}}$	T stands for Transformation. K: Represents a Key used in the cryptographic operation. b i user b and has an identifier i.

Table A3. Notation of security proof with enhanced mathematical Rigor.

$\mathcal{A}$	Represents the adversary in the cryptographic proof, trying to compromise the system. Testing the system's resilience against potential security breaches.
$\mathcal{C}$	The challenger in cryptographic games simulates the protocol to validate the security measures of the IoT system.
s	A secret value selected uniformly at random from $\mathbb{Z}_q$, used as the exponent in the key generation to create the public key. Used in the cryptographic key generation process to secure IoT device communications.
P_{pub}	The public key in the BFLIoT system, derived as g^s, where g is a generator of the group G.
m	Represents a message that is an element of $\mathbb{Z}_q$, involved in the encryption process.
r	A random value is chosen uniformly from $\mathbb{Z}_q$ for each encryption process.
C	The ciphertext resulting from the encryption scheme combines a power of g and the message masked with a hash output.
H	A cryptographic hash function is used as a random oracle, ensuring the randomness of the hash output used in the encryption.
$\mathcal{B}$	An algorithm constructed to solve the DL problem using the adversary's ability to break the encryption scheme.
G_i	Local gradient from a device in the federated learning process.
$\widetilde{G_i}$	Noise-adjusted gradient ensuring (ϵ, δ)-differential privacy by adding Gaussian noise $\mathcal{N}(0, \sigma^2 I)$.
HF	The cryptographic hash function used in the PoW mechanism within the blockchain integration.
n	Nonce in the PoW, a number that miners adjust to solve the hashing challenge.
$\mathcal{N}(0, \sigma^2 I)$	Gaussian noise was added to the local gradient to ensure differential privacy.

References

1. Sefati, S.S.; Arasteh, B.; Halunga, S.; Fratu, O.; Bouyer, A. Meet User's Service Requirements in Smart Cities Using Recurrent Neural Networks and Optimization Algorithm. *IEEE Internet Things J.* **2023**, *10*, 22256–22269. [CrossRef]
2. Sefati, S.S.; Halunga, S. Ultra-reliability and low-latency communications on the internet of things based on 5G network: Literature review, classification, and future research view. *Trans. Emerg. Telecommun. Technol.* **2023**, *34*, e4770. [CrossRef]
3. Ullah, A.; Ahmad, J.; Arif, M.; Javed, A.; Alazab, M.; Khan, M.S. Smart Cities: The Role of Internet of Things and Machine Learning in Realizing a Data-Centric Smart Environment. *Complex Intell. Syst.* **2024**, *10*, 1607–1637. [CrossRef]
4. Miranda, R.; Filippoupolitis, A.; Oliff, W. Revolutionising the Quality of Life: The Role of Real-Time Sensing in Smart Cities. *Electronics* **2024**, *13*, 550. [CrossRef]
5. Farooq, M.S.; Riaz, S.; Abid, A.; Abid, K.; Naeem, M. A Survey on the Role of IoT in Agriculture for the Implementation of Smart Farming. *IEEE Access* **2019**, *7*, 156237–156271. [CrossRef]
6. Sefati, S.S.; Halunga, S. Mobile sink assisted data gathering for URLLC in IoT using a fuzzy logic system. In Proceedings of the 2022 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), Sofia, Bulgaria, 6–9 June 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 379–384.
7. Javed, A.R.; Usman, M.; Raza, M.; Khan, A.A.; Noor, K.; Din, S. Future Smart Cities: Requirements, Emerging Technologies, Applications, Challenges, and Future Aspects. *Cities* **2022**, *129*, 103794. [CrossRef]
8. Alfandi, O.; Patel, A.; Santos, A.; Bennett, F. A Survey on Boosting IoT Security and Privacy through Blockchain: Exploration, Requirements, and Open Issues. *Clust. Comput.* **2021**, *24*, 37–55. [CrossRef]
9. Farahani, B.; Firouzi, F.; Luecking, M. The Convergence of IoT and Distributed Ledger Technologies (DLT): Opportunities, Challenges, and Solutions. *J. Netw. Comput. Appl.* **2021**, *177*, 102936. [CrossRef]
10. Qu, Y.; Uddin, M.P.; Gan, C.; Xiang, Y.; Gao, L.; Yearwood, J. Blockchain-enabled federated learning: A survey. *ACM Comput. Surv.* **2022**, *55*, 1–35. [CrossRef]
11. Sefati, S.S.; Haq, A.U.; Craciunescu, R.; Halunga, S.; Mihovska, A.; Fratu, O. A Comprehensive Survey on Resource Management in 6G Network Based on Internet of Things. *IEEE Access* **2024**, *12*, 113741–113784. [CrossRef]
12. Henry, R.; Herzberg, A.; Kate, A. Blockchain Access Privacy: Challenges and Directions. *IEEE Secur. Priv.* **2018**, *16*, 38–45. [CrossRef]
13. Sefati, S.S.; Halunga, S. Data forwarding to Fog with guaranteed fault tolerance in Internet of Things (IoT). In Proceedings of the 2022 14th International Conference on Communications (COMM), Bucharest, Romania, 16–18 June 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1–5.
14. Omar, A.S.; Basir, O. Identity Management in IoT Networks Using Blockchain and Smart Contracts. In Proceedings of the 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Halifax, NS, Canada, 30 July–3 August 2018; IEEE: Piscataway, NJ, USA, 2018.
15. Kharche, A.; Badholia, S.; Upadhyay, R.K. Implementation of Blockchain Technology in Integrated IoT Networks for Constructing Scalable ITS Systems in India. *Blockchain Res. Appl.* **2024**, *5*, 100188. [CrossRef]

16. Sisi, Z.; Souri, A. Blockchain Technology for Energy-Aware Mobile Crowd Sensing Approaches in Internet of Things. *Trans. Emerg. Telecommun. Technol.* **2024**, *35*, e4217. [CrossRef]
17. Jyoti, A.; Yadav, V.; Prakash, A.; Kumar Jha, S.; Rahul, M. Blockchain for Big Data: Approaches, Opportunities and Future Directions. *Recent Adv. Electr. Electron. Eng. (Former. Recent Pat. Electr. Electron. Eng.)* **2024**, *17*, 229–243. [CrossRef]
18. Malik, K.R.; Samad, A.; Anjum, A.; Saeed, Y.; Han, K. A Methodology for Real-Time Data Sustainability in Smart City: Towards Inferencing and Analytics for Big-Data. *Sustain. Cities Soc.* **2018**, *39*, 548–556. [CrossRef]
19. Sefati, S.; Mousavinasab, M.; Zareh Farkhady, R. Load balancing in cloud computing environment using the Grey wolf optimization algorithm based on the reliability: Performance evaluation. *J. Supercomput.* **2022**, *78*, 18–42. [CrossRef]
20. Wan, S.; Chen, Z.; Lin, K.; Wang, X.; Wu, Y.; Di, Y. To Smart City: Public Safety Network Design for Emergency. *IEEE Access* **2017**, *6*, 1451–1460. [CrossRef]
21. Mahesh, B. Machine Learning Algorithms-A Review. *Int. J. Sci. Res. (IJSR)* **2020**, *9*, 381–386. [CrossRef]
22. Aldoseri, A.; Al-Khalifa, K.N.; Hamouda, A.M. AI-Powered Innovation in Digital Transformation: Key Pillars and Industry Impact. *Sustainability* **2024**, *16*, 1790. [CrossRef]
23. He, P.; Zhou, Y.; Qin, X. A Survey on Energy-Aware Security Mechanisms for the Internet of Things. *Future Internet* **2024**, *16*, 128. [CrossRef]
24. Amoo, O.O.; Shithili, S.; Ramanujan, R. Cybersecurity Threats in the Age of IoT: A Review of Protective Measures. *Int. J. Sci. Res. Arch.* **2024**, *11*, 1304–1310. [CrossRef]
25. Kvak, K.; Straka, M. The Use of the Internet of Things in the Distribution Logistics of Consumables. *Appl. Sci.* **2024**, *14*, 3263. [CrossRef]
26. Merlec, M.M.; In, H.P. SC-CAAC: A Smart Contract-Based Context-Aware Access Control Scheme for Blockchain-Enabled IoT Systems. *IEEE Internet Things J.* **2024**, *11*, 19866–19881. [CrossRef]
27. CheSuh, L.N. Improve Quality of Service for the Internet of Things Using Blockchain & Machine Learning Algorithms. *Internet Things* **2024**, *26*, 101123.
28. Kiran, M. Blockchain Based Secure Ownership Transfer Protocol for Smart Objects in the Internet of Things. *Internet Things* **2024**, *25*, 101002.
29. Li, K. Privacy-Preserving Scheme with Bidirectional Option for Blockchain-Enhanced Logistics Internet of Things. *IEEE Internet Things J.* **2024**, *11*, 20562–20574. [CrossRef]
30. Vishwakarma, L.; Das, D. BLISS: Blockchain-Based Integrated Security System for Internet of Things (IoT) Applications. *Int. J. Inf. Secur.* **2024**, *23*, 1649–1665. [CrossRef]
31. Singh, D.; Dwivedi, R.K. Designing Blockchain Based Secure Autonomous Vehicular Internet of Things (IoT) Architecture with Efficient Smart Contracts. *Int. J. Inf. Technol.* **2024**, 1–17. [CrossRef]
32. Khan, A.A. Data Security in Healthcare Industrial Internet of Things with Blockchain. *IEEE Sens. J.* **2023**, *23*, 25144–25151. [CrossRef]
33. Xiong, H.; Wu, Y.; Jin, C.; Kumari, S. Efficient and privacy-preserving authentication protocol for heterogeneous systems in IIoT. *IEEE Internet Things J.* **2020**, *7*, 11713–11724. [CrossRef]
34. Zhong, H.; Zhang, S.; Cui, J.; Wei, L.; Liu, L. Broadcast encryption scheme for V2I communication in VANETs. *IEEE Trans. Veh. Technol.* **2021**, *71*, 2749–2760. [CrossRef]
35. Gervais, A. On the Security and Performance of Proof of Work Blockchains. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016.
36. Sefati, S.S.; Fartu, O.; Nor, A.M.; Halunga, S. Enhancing Internet of Things Security and Efficiency: Anomaly Detection via Proof of Stake Blockchain Techniques. In Proceedings of the 2024 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), Osaka, Japan, 19–22 February 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 591–595.
37. Iqbal, M.A. A Review on Internet of Things (IoT): Security and Privacy Requirements and the Solution Approaches. *Glob. J. Comput. Sci. Technol.* **2016**, *16*, 1–9.
38. Datta, A.; Halpern, J.Y.; Mitchell, J.C.; Roy, A.; Sen, S. A symbolic logic with concrete bounds for cryptographic protocols. *arXiv* **2015**, arXiv:1511.07536.
39. Wang, J. Game-Based Low Complexity and Near Optimal Task Offloading for Mobile Blockchain Systems. *IEEE Trans. Cloud Comput.* **2024**, *12*, 539–549. [CrossRef]
40. Benrebbouh, C.; Mansouri, H.; Cherbal, S.; Pathan AS, K. Enhanced secure and efficient mutual authentication protocol in iot-based energy internet using blockchain. *Peer-to-Peer Netw. Appl.* **2024**, *17*, 68–88. [CrossRef]

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