electronics

MDPI

Article

Enabling a Secure IoT Environment Using a Blockchain-Based Local-Global Consensus Manager

Saleh Alghamdi [1,*], Aiiad Albeshri [1] and Ahmed Alhusayni [2]

[1] Faculty of Computer Sciences, King Abdulaziz University, Jeddah 21589, Saudi Arabia; aaalbeshri@kau.edu.sa
[2] Faculty of Computer Sciences, Umm-Al Qura University, Makkah 24211, Saudi Arabia; aehusayni@uqu.edu.sa
* Correspondence: salghamdi1268@stu.kau.edu.sa

Abstract: The Internet of Things (IoT) refers to the network of interconnected devices that can communicate and share data over the Internet. The widespread adoption of smart devices within Internet of Things (IoT) networks poses considerable security challenges for their communication. To address these issues, blockchain technology, known for its decentralized and distributed nature, offers potential solutions within consensus-based authentication in IoT networks. This paper presents a novel approach called the local and global layer blockchain model, which aims to enhance security while simplifying implementation. The model leverages the concept of clustering to establish a local-global architecture, with cluster heads assuming responsibility for local authentication and authorization. Implementing a local private blockchain facilitates seamless communication between cluster heads and relevant base stations. This blockchain implementation enhances credibility assurance, strengthens security, and provides an effective network authentication mechanism. Simulation results indicate that the proposed algorithm outperforms previously reported methods. The proposed model achieved an average coverage per node of 0.9, which is superior to baseline models. Additionally, the lightweight blockchain model proposed in this paper demonstrates superior capabilities in achieving balanced network latency and throughput compared to traditional global blockchain approaches.

Keywords: blockchain; Internet of Things; clustering; Byzantine Fault Tolerant; consensus protocols

Citation: Alghamdi, S.; Albeshri, A.; Alhusayni, A. Enabling a Secure IoT Environment Using a Blockchain-Based Local-Global Consensus Manager. *Electronics* **2023**, *12*, 3721. https://doi.org/10.3390/electronics12173721

Academic Editor: Miin-shen Yang

Received: 1 August 2023
Revised: 22 August 2023
Accepted: 31 August 2023
Published: 3 September 2023

1. Introduction

The term "Internet of Things" (IoT) describes a scenario in which a wide variety of gadgets, automobiles, appliances, and other things are linked together and equipped with applications, sensors, and network connections [1]. Through the use of the internet, these gadgets may share information with one another and with centralized databases. The IoT environment enables these devices to communicate, interact, and share information, leading to increased automation, efficiency, and connectivity in various domains. The IoT environment typically involves a combination of hardware, software, and network infrastructure. It includes devices such as sensors, actuators, gateways, embedded systems, communication protocols, cloud computing platforms, and data analytics tools. These components work together to enable seamless communication, data exchange, and interoperability among different devices and systems [2].

A distributed IoT environment refers to an Internet of Things (IoT) system where the network and computing resources are decentralized and distributed across multiple locations or devices. In this environment, the IoT infrastructure does not rely on a central system or cloud-based platform for data processing and decision-making. Instead, the intelligence and processing capabilities are distributed among the devices themselves or within localized edge computing nodes. In a distributed IoT environment, devices at the network's edge (e.g., sensors, actuators, and gateways) are equipped with processing power and storage capabilities. These devices can perform data collection, preprocessing,

analysis, and even decision-making tasks locally without relying heavily on cloud resources. This approach offers several advantages, including reduced latency, improved reliability, enhanced privacy and security, and efficient bandwidth utilization [3].

There are too many issues with the distributed IoT environment, such as connectivity, data management, privacy, and security, of which securing a distributed IoT environment is the most challenging due to the decentralized nature of the system. Ensuring the identity and trustworthiness of devices within a distributed IoT environment is crucial [4]. Device authentication mechanisms, such as certificates or cryptographic keys, need to be implemented to verify the identity of devices and prevent unauthorized access. Addressing these issues requires a comprehensive security strategy that encompasses device-level security, secure communication protocols, access control mechanisms, and ongoing monitoring and updates. Following industry best practices and standards is crucial to building a secure and resilient distributed IoT environment.

In recent years, it has been observed that blockchain technology and consensus protocols can play a significant role in securing IoT environments due to their immutable and tamper-resistant data. Blockchain provides a decentralized and distributed ledger that can store IoT data in a transparent and immutable manner. By storing data in blocks linked using cryptographic hashes, the integrity and immutability of the data can be ensured [5]. This helps prevent data tampering and unauthorized modifications. Consensus protocols, such as Proof of Work (PoW) or Proof of Stake (PoS), ensure agreement and trust in a blockchain network. These protocols enable nodes in the network to reach a consensus on the validity of transactions and the state of the blockchain. By employing consensus mechanisms, IoT environments can ensure data integrity and prevent malicious activities [6].

When used in the IoT, blockchain technology helps solve the problem of centralization while also allowing for the safe flow of data across IoT nodes, even in untrusted situations. Nevertheless, most existing blockchain consensus algorithms are too computationally intensive and cannot be scaled to meet the ever-increasing demands for efficiency and responsiveness in IoT devices. Additionally, IoT nodes may be attacked in several different ways. Hence, developing a reliable reputation evaluation scheme is important in establishing a secure and trustworthy IoT ecosystem for specific applications [7–9].

In this study, the distributed IOT environment is split into various clusters, enabling each cluster to autonomously choose consensus nodes within its territory. For this purpose, a real-time, updated reputation value approach is used. The selected nodes took part in a global consensus to suggest blocks. Additionally, the research improved the overall security of the IOT environment by introducing a federated process that included both secure global and local consensus. The coordination of both consensuses was delegated to two different leaders to lessen the burden on the leader of the local node. The suggested system also includes synchronizing both consensuses, preventing energy consumption by interdependent client transactions.

Research Contribution

The key contributions of the proposed work are as follows:

- To enhance the accuracy of the IoT environment, an approach is proposed that combines two consensus protocols: local consensus and global consensus. This integration aims to improve the overall reliability and agreement within the system;
- The proposed system incorporates a profile-based secure blockchain data structure at the global level and an additional blockchain-based data structure at the local level;
- An innovative synchronization method is introduced to facilitate coordination between global and local consensus, ensuring trust and preventing double-spending when dealing with local and global coordination;
- The blockchain system incorporates a rotating leader scheme and utilizes a Byzantine Fault Tolerant (BFT) leader election method to designate separate leaders for local and global consensus. This strategic approach effectively alleviates the workload on the local leader node, distributing the responsibilities more efficiently;

- Lastly, we conducted a comprehensive performance evaluation of our proposed algorithm and compared it against previous approaches. We obtained empirical results through extensive experimentation, highlighting our proposed algorithm's distinct advantages. Our algorithm demonstrates superior performance compared to other approaches.

The structure of this paper is as follows: In Section 2, a comprehensive overview of the most closely related work is provided, offering a detailed description and analysis. Section 3 is dedicated to the design of the protocol and its underlying algorithms, where specific attention is given to providing better security to the IoT environment. Section 4 includes simulation results and insights from the established procedure. Section 6 concludes the work by summarizing the main results and contributions.

2. Related Work

This section discusses the background of the core concepts as well as the current research that has been carried out in the domain of the said topic.

2.1. Internet of Things (IoT)

"Internet of Things" refers to the interconnection of "smart" devices, which can include both mechanical and digital machinery, items, and people and are capable of exchanging data over a network without the participation of a human. Figure 1 shows the typical IoT system. On a more macro scale, IoT applications include things like smart cities, smart homes, and smart healthcare systems, among others. The following are some of the most important parts [10] of the Internet of Things ecosystem:

- Sensors and Smart Devices: These are the physical things outfitted with sensors, actuators, and connections. Smartphones, wearables, smart home gadgets (thermostats, lighting, security cameras), industrial sensors, and even cars are some of the examples;
- Connection: To connect to the internet or local networks, IoT devices use a variety of communication protocols. Some examples are Wi-Fi, Bluetooth, cellular networks (3G, 4G, or 5G), Zigbee, Z-Wave, and LoRaWAN;
- Processing and Analysis: IoT devices use internal sensors to gather data about their surroundings. This data's processing and analysis occur locally on the device or in the cloud. It is possible to use advanced analytics and machine learning techniques to glean insightful information from the gathered data;
- Storage: The cloud is essential to IoT systems because it offers scalability, processing power, and storage. IoT devices frequently upload their data to the cloud for analysis and storage. Additionally, remote device management, firmware updates, and connections with other programs and services are all made possible by cloud platforms.

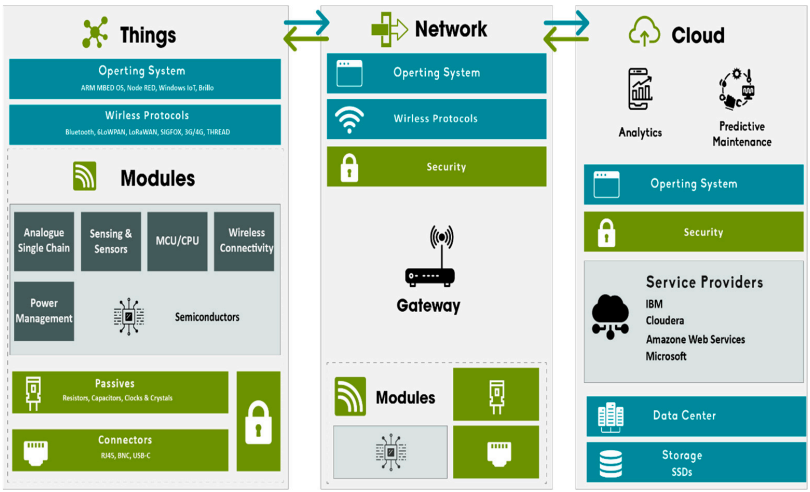


Figure 1. The typical IOT architecture, adopted from [1].

2.2. Security Issues in IoT

The wide-ranging diversity among IoT devices and their limited security visibility give rise to significant concerns regarding trust and security [11]. The absence of standardized trust mechanisms and essential security standards, combined with the scarcity and inadequate management of firmware updates for IoT devices, creates security vulnerabilities that attract malicious actors. Additionally, the IoT ecosystem lacks transparent identification and privacy solutions that grant individuals control over the identity data of their devices. Furthermore, there is a lack of security training and protocol implementation for both individuals and devices, along with ineffective information sharing with CERTs/CSIRTs (Computer Emergency Response Teams/Computer Security Incident Response Teams).

In theory, current blockchain solutions hold promise for bolstering the security and trustworthiness of IoT networks. However, practical restrictions, such as the extra processing resources needed owing to sophisticated encryption, limit their usefulness and provide issues that may be solved by improving certain blockchain network components.

2.3. The Blockchain Technology

Blockchain technology was first developed and used by a well-known cryptocurrency, Bitcoin, in 2008 [12]. It is a peer-to-peer network-based decentralized ledger technology. The usual blockchain paradigm is seen in Figure 2. Under this model, each node in the blockchain network keeps its copy of the ledger up-to-date. The blockchain primarily relied on cryptocurrencies to detect and handle the double-spending problem. IoT networks are only one of the many new areas where blockchain technology is being explored for its potential to secure and centrally record digital transactions.

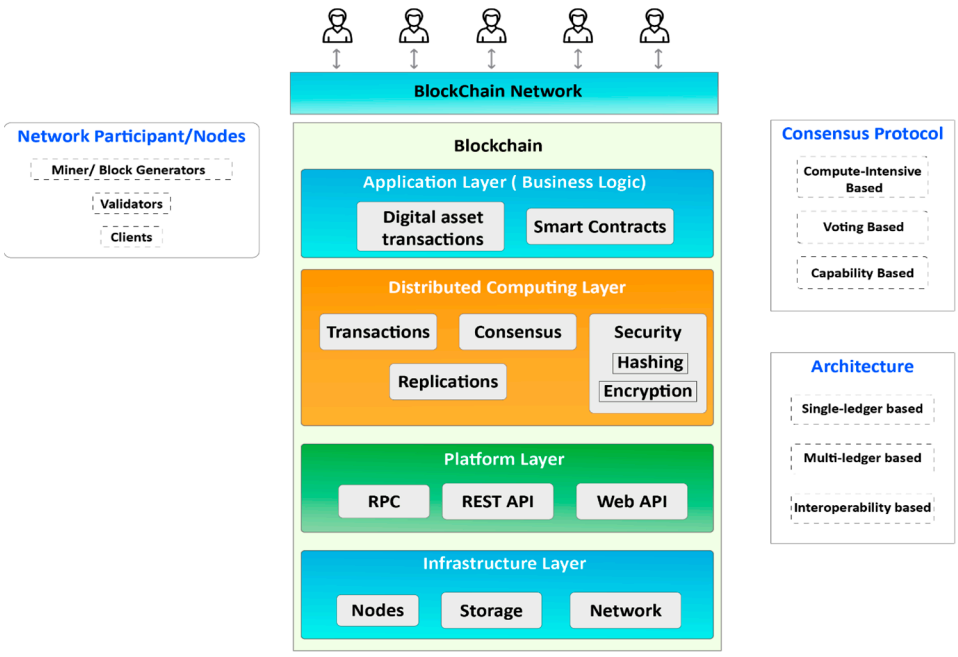


Figure 2. The blockchain architecture.

The ledger of a blockchain is composed of blocks, with each block consisting of two components. The first component represents the transaction, which needs to be stored in a database, and it can take various forms, such as a patient record, network traffic log, and commodity transaction, among others. On the other hand, the second component contains header information, including the hash of the current transaction, the combined previous hash, and a timestamp.

Thus, storage in this manner results in a linked chain block with a sequence, as shown in Figure 3. A new transaction will also first contribute to a specific block if it begins. Second, following previously established standards, miners confirm the block contains the

transaction. A group of miners uses a consensus technique to validate the transactions after verification. After successful validation, the confirmed transaction is prepared to be added to the blockchain ledger.

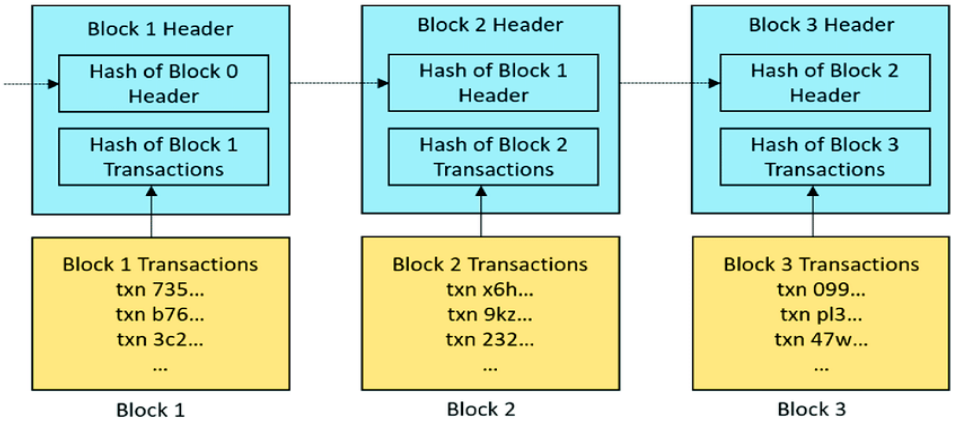


Figure 3. Block inter-links mechanism in blockchain.

2.4. Consensus Algorithms

In the realm of blockchain systems, a consensus mechanism refers to a program designed to foster widespread agreement on the current state of a ledger. The schematic diagram of the blockchain architecture is shown in Figure 4. In contrast, Table 1 presents the standard consensus algorithms that are most suitable for networks with many users and operations. The utilization of consensus mechanisms brings about numerous advantages for distributed ledgers, blockchains, and cryptocurrencies, as they can effectively replace the relatively slow processes of human auditors and verifiers. True decentralization and security are achieved in the two variations of consensus algorithms, namely public and permissionless blockchains. However, the need to ensure both privacy and security in their consensus processes results in the consumption of significant resources, leading to scalability challenges. Proof of Work (PoW) and Proof of Stake (PoS) are prime examples of such algorithms. Conversely, private and permissioned blockchains opt to sacrifice a portion of their decentralization, but they can still offer scalability, portability, and safety.

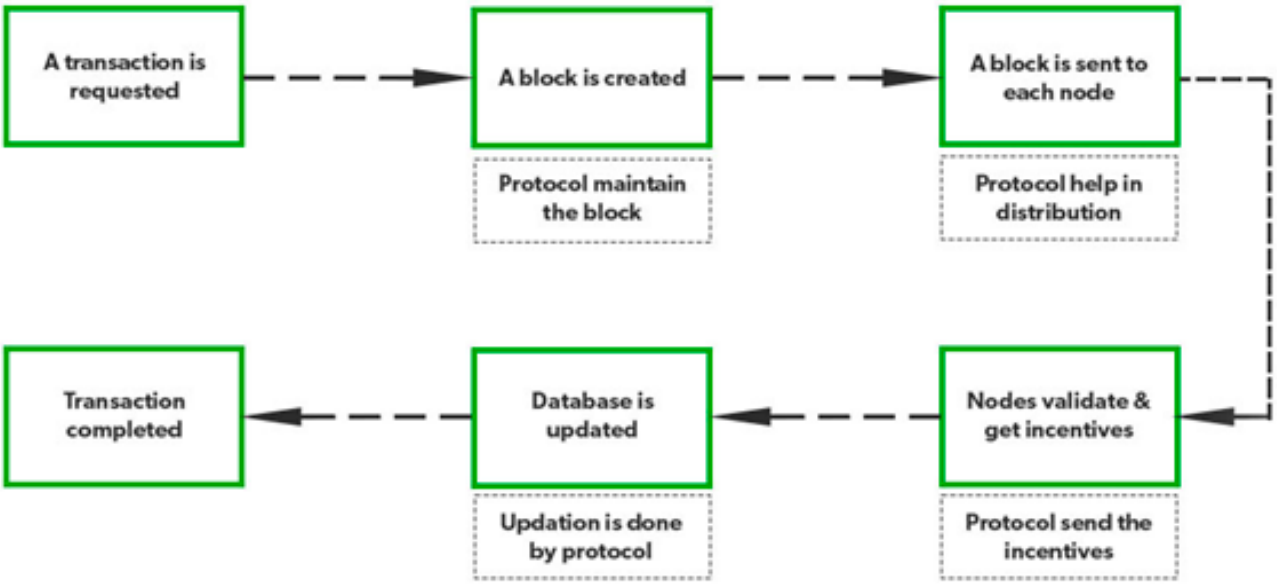


Figure 4. The blockchain protocol architecture.

Table 1. Existing consensus algorithms [13].

Reference	Merits	Demerits
Proof of Work (PoW)	More Secure Provides decentralized power and control	Required more power and electricity Fail to cover a wide area network
Proof of Stake (PoS)	Energy efficient Fast processing Mechanism	Less decentralization as compared to PoW Low Security
Delegated Proof of Stake	Faster in processing as compared to PoW and PoS Less expensive Low hardware requirements	Less resilience because of the decentralization strategy More suspects able to attack
Transaction as Proof of Stake (TPoS)	More secure due to the combined contribution of all nodes Easy adaptable algorithms	Speed degradation due to the combined contribution of nodes. Failed against short fork in the blockchain
Ripple	Path-dependent and un-editable chain helps fast transaction. Low power consumption as compared to PoW	Required to maintain node list which is difficult to manage Highly decentralize
Proof-of-Personhood	Remove the disadvantages of POW and POS	Expensive

This is made possible by employing consensus algorithms like Paxos, Raft, and PBFT, which impose less demanding security requirements due to their permissioned nature and the identities they employ [13].

PoW is not as energy-intensive as before. Paxos has been the preferred protocol for achieving consensus in distributed networks, particularly in permissioned blockchains. Over the past decade, it has practically become synonymous with consensus, leading to numerous implementations based on or influenced by Paxos. However, a consensus method called Raft has emerged to enhance Paxos' comprehension, correctness, and performance. Raft focuses heavily on leader elections and aims to make Paxos easier to understand and execute. Another consensus technique, Practical Byzantine Fault Tolerance (PBFT), was initially developed in the late 1990s. It provides fault tolerance for asynchronous systems by allowing up to $(n - 1)/3$ faults, where n is the total number of alternations.

2.5. Blockchain Models for Secure IoT Environment

This section provides a comprehensive discussion of the existing secure IOT environment. Due to the limited availability of time and space, some additional literature has been shown in Table 2. Sachi Nandan Mohanty et al. [14] express that current models of BC do not suit IOT because of model complexity, limited support for scalability, and the overhead of high bandwidth and latency. The authors have put forward a highly efficient and lightweight integrated blockchain (ELIB) model to address the specific requirements for IoT suitability. They conducted tests by implementing this model in a smart home environment to validate its applicability in various IoT settings. To overcome the limitations of resource-constrained smart homes, the authors introduced a centralized manager responsible for generating shared keys, facilitating data transportation, and managing incoming and outgoing requests. The ELIB model they propose establishes an overlay network that allows the integration of highly capable resources with a public blockchain, ensuring dedicated security and privacy. The authors used a lightweight consensus method in their proposed ELIB model. These optimizations were used in order to achieve the best possible performance.

Moreover, the authors conducted thorough simulations in different scenarios, taking into account processing speed, energy consumption, and overhead. They compared the results of their model with the baseline technique and demonstrated that the ELIB model achieved a remarkable 50% reduction in total processing time while consuming the least

amount of energy possible (0.07 mJ). The authors asserted that the ELIB model outperforms the baseline technique when evaluated against multiple criteria.

Yongfeng Qian et al. [15] also discussed that the Internet of Things (IoT) has advanced quickly, providing customers with significant convenience in a variety of areas, including smart homes, smart transportation, and more. They pointed out that it could potentially pose security issues, though. The authors used the three IoT layers of perception, network, and application in order to address this difficulty, followed by the accompanying security issues for each layer. Afterward, they suggested a high-level security management scheme based on blockchain for various IoT devices across their whole life cycles.

IoT security remains a significant challenge that requires further investigation. To instill trust in IoT systems, IoT authentication is essential, as emphasized by Osama A. et al. [16] The authors argue that centralized authentication methods have proven ineffective for cross-domain authentication and fail to accommodate the expanding IoT networks. The authors propose a hybrid architecture for IoT systems that combines centralized and blockchain-based authentication to address these issues. The proposed approach establishes edge servers for centralized authentication for interconnected IoT devices. The authors emphasize the need for efficient authentication, which entails minimizing the utilization of IoT resources. Therefore, lightweight cryptographic algorithms were implemented by the authors. To illustrate the proposed architecture, a local Ethereum blockchain network was employed. Before concluding, the authors conducted a side-by-side comparison of the proposed method with centralized and blockchain-based authentication systems. They assert that their approach provides considerable benefits to IoT devices in terms of the total computation cost, the amount of time required for execution, and the amount of power used.

To address concerns with trustworthiness, Punam Prabha presented a hybrid consensus mechanism (HCM) based on blockchain that is used in electronic healthcare systems (EHS) [17]. The purpose of the suggested HCM is to fulfill the role of a reputation module based on the actions of the blocks. HCM also includes five algorithms, one for each of the following: creation, validation, handling of forks (if any), generation of Merkle trees, and reward/punishment modules.

In their study [18], Sharda Tiwari explored a blockchain- and internet-based real-time medical management system that is protected using cryptography. The authors described how electronic healthcare systems are expanding significantly today. Since various patients have different data, it is necessary to record the data that is retrieved from IoT devices in a database for future reference. They expressed that the IoT is vulnerable to privacy and security breaches because it lacks inherent security measures, as this information is crucial for a certain patient. The doctor cannot identify the patient's real issue and cannot treat the patient properly if this data is manipulated by an intrusive party, which results in significant injury to the patient. The authors suggested that there should be a security mechanism for patient data using blockchain. To eliminate this anonymous data access, the authors created an IoT-based prototype using blockchain technology to keep the patient data private.

Table 2. Literature review of existing state-of-the-art secure IOT models.

Reference	Methodology (Protocol Used)	Limitations
Subhi Alrubei et al. [19]	HDPoA Honesty-based Distributed Proof of Authority, Scalability mechanism	Did not deploy over a large area with more nodes. Did not test in different IoT contexts, to fully evaluate the performance of HDPoA.
Abdella, J. et al. [20]	HiCoOB: Hierarchical Concurrent Optimistic Blockchain Consensus Protocol, Concurrent execution of transactions	The inability to provide assistance for mobile energy users and producers.

Table 2. *Cont.*

Reference	Methodology (Protocol Used)	Limitations
Pabitha, P. [21]	ModChain: A blockchain infrastructure that is hybridized, secure, and scalable, designed specifically for the IoT	Did not cater addition of false data in chain effectively.
Kaur, M. [22]	DPoAC: Delegated Proof of Accessibility, Data privacy and Encrypted communication	Large-scale experimentation is missing to ensure the validity of the proposed protocol. A comparison of the proposed protocol's performance with existing protocols is missing.
Xu, R. et al. [23]	µDFL: Micro chained Decentralized Federated Learning, Performance and scalability	Validation of the proposed µDFL in real-world FL applications is missing. Evaluation of performance and security on various attacks.
Alhejazi, M. et al. [24]	WMCA: Weighted Majority Consensus Algorithm, Security,	A protection mechanism for the master node from cyber attacks is missing. As master node will be the only node to calculate and maintain the weights for each minder.

Jawad Ali et al. [25] proposed a technique to address the issue generated by the nature and proliferation of IoT systems. These features make the devices vulnerable to a wide range of threats. Their method included observing how blockchain-supported IoT devices interacted with other networks to determine their trustworthiness. In order to extract each device's activity and analyze behavior using deep machine learning techniques, they built a unique behavior monitor and implemented it on a chosen node. And they employed Trusted Execution Technology (TEE), which can be used to set up a safe place for the blockchain's most important applications to run and store their data. In the evaluation phase, the authors concluded with an analysis of data from various IoT devices that a Mirai attack had impacted [25]. They claimed that their suggested method has more strength in terms of detection speed and accuracy than other models available.

From the above discussion, it has been observed that the selection of consensus algorithms depends on various factors such as scalability, energy efficiency, interoperability, and integration with existing IoT protocols and standards. Because blockchain platforms like Bitcoin and Ethereum blockchains are computationally costly, have large bandwidth overheads, and experience IoT delays, lightweight consensus algorithms are necessary for IoT to embrace blockchain [26]. In order to overcome this challenge, a suitable consensus protocol must be created by taking into account IoT issues, including a lack of security, disparate device standards, limited device memory, and a vast volume of data. Alternative to this, some IoT security issues may be resolved by a distributed blockchain, according to a novel blockchain-based solution disclosed in [27], which demonstrated a decentralized blockchain-based identity management system that protects patients' confidentiality and privacy when they get remote medical treatment. Using blockchain for medical IoT to protect privacy is a suggestion made in [28]. The storage problem for industrial IoT was solved by offering a hierarchical blockchain storage structure (ChainSplitter), in which most of the blockchain is hosted on cloud services. Lastly, lightweight algorithms are required for IoT blockchain to overcome power and processing time constraints [29,30].

3. Proposed Model

This section discusses the core methodology of the proposed model. The suggested network model seeks to give IoT networks a dependable, trustworthy security mechanism. The distributed scenario of the proposed work is divided into different layers and is described in the below subsection. The overall architecture of the proposed secure IoT is depicted in Figure 5.

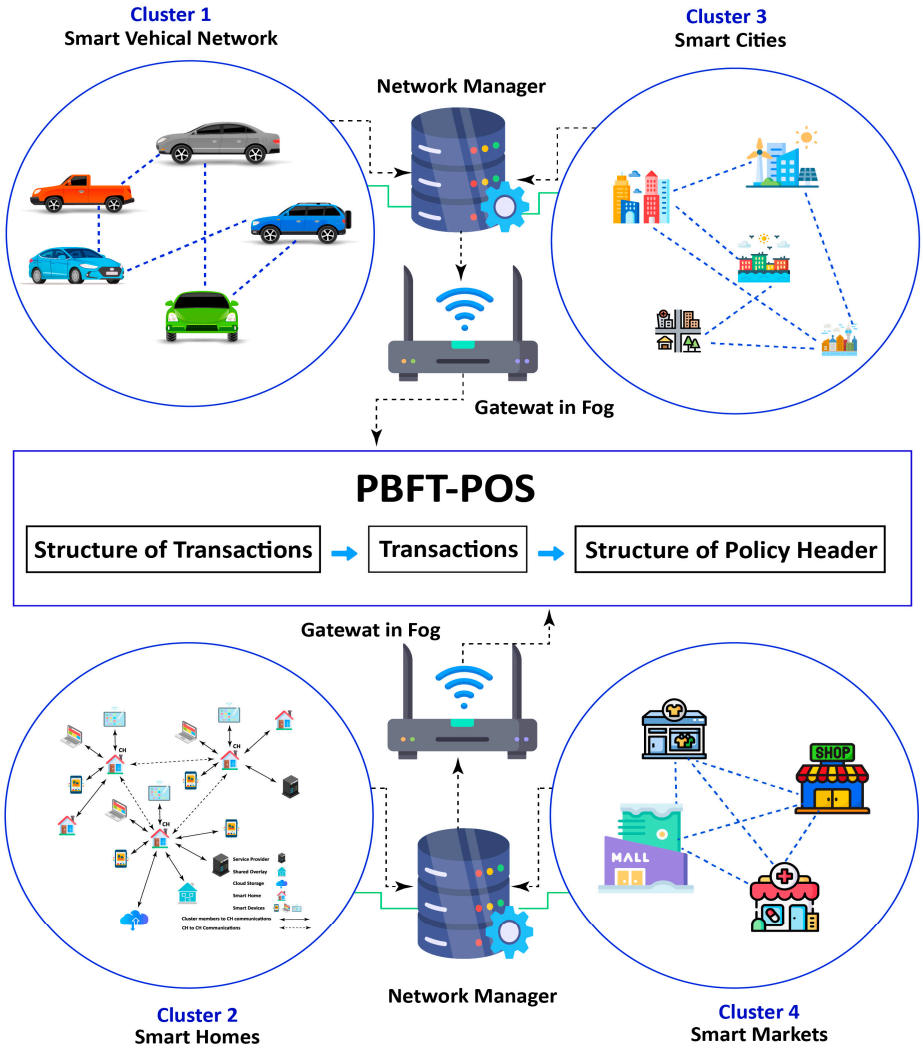


Figure 5. The cluster-based secure IoT environment.

3.1. The Cluster Formation

This layer discusses the detailed description of the distributed IOT environment where all the IOT devices are connected concerning their geographic distribution, locality, functionality, and cooccurrence. All the connected IoT devices are structured into a coherent network for effective communication, coordination, and data exchange. This is referred to as a network cluster. The key steps of this section are cluster formation and cluster head selection for a secure environment and smooth data flow between devices.

3.1.1. Cluster Formation Steps

The clustering formation has been done using Prim's algorithm [31] and the standard K-Mean algorithm [8]. A comprehensive graph is initially constructed, where each vertex represents an individual Internet of Things (IoT) device. The weight assigned to each edge is determined based on the Euclidean distance [31] between the corresponding devices.

The basic reason for using Euclidean distance is that in our case, almost 98% of the IoT devices are static in nature and directly connected, like IoT devices in smart cities, smart hospitals, smart schools, etc., and only 1% or 2% are concrete in nature, such as moving cars, that have a very rare impact on the performance of the proposed work. However, the cluster formation of the moving sensors, like cars, has been computed by collecting sensor data from vehicles in motion. This consists of information from GPS, accelerometers, and temperature sensors. The data is preprocessed to ensure a consistent format and eliminate noise and outliers. The pertinent sensor data features like velocity,

acceleration, and coordinates were also obtained for clustering. Each data point (sensor reading) is represented as a feature vector comprising the selected features. Then we determine the Euclidean distance between every pair of vector features. Each element (i, j) of the distance matrix will represent the Euclidean distance between data points i and j. However, the Euclidean distance might not fully capture temporal variations in such data. But this procedure performs well due to the rare case of moving IoT. A weighted graph that is precisely composed of the presently active nodes is constructed for each cluster individually. Later on, an iterative process is employed to construct a minimum-spanning tree (MST) using Prims's algorithm for each weighted graph generated in the preceding step. Algorithm 1 shows Prim's algorithm for constructing a minimum-spanning tree. In this algorithm, T is the minimum spanning tree, S is the resultant vertices set, V is the given vertices, d is weight, u and v are the subsets of S and W1; W2 is the cost of edges.

Algorithm 1: Cluster Formation using Prim's Algorithm-based MST

```
MST-PRIM (G, s)
    T ← φ
    S ← φ
    S ← S U {s}
    V ← V-{s}
    while V ≠ φ do
        d←∞
        for each vertex u ε S do
          for each vertex v ε Adj[u] do
            if (d > w(u,v))
              then d ← w(u,v)
            w1 ← v
          w2 ← u
        V ← V-{w1}
      S ← SU{w1}
    T ← T U {(w1, w2)}
  Return T
```

If a cluster consists of more than one active node, the MST is formed as a tree that minimizes the total weight of its edges. The selection is made from the available spanning trees within the graph. After the cluster formation process, the cluster head (CH) selection is computed. The fitness value for each prospective candidate is calculated during this procedure, and the best CH is selected based on this value. We take into account every crucial quality that a CH should have and, using a fitness function, merge them into a feature. The definition of the fitness function is given after a description of the crucial parameters.

3.1.2. Cluster Head Selection

Following the clustering process, every node is a candidate for the CH position. Based on the idea of a hybrid algorithm, the CH is considered to be the node that secures and increases the network's lifespan. The key parameters for the CH selection are intra-cluster communication cost, total local distance of the nodes to CH, and node distance to other clusters. Equation (1) determines each potential candidate's fitness value by following the parameters.

$$\mathrm{CH} = \mathrm{Max}\ \left(\sum_{i=1}^{n} dij + \sum_{i=1}^{n} wij + \sum_{i=1}^{n} vij\ \right) \tag{1}$$

where *dij* is the distance between nodes, *wij* is the edge cost, and *vij* is the cluster-to-cluster distance.

Each node in the cluster with a fitness value greater than the fitness values of the other nodes in the cluster is selected to serve as the CH for that cluster. As a result, a node is considered the best candidate for a CH if its distance to other adjacent nodes alive in MST

is the lowest, while its remaining energy and number of neighbors are the largest. In other words, this kind of node has the highest chance of becoming a CH.

A lightweight session key is allotted once devices successfully authenticate themselves with linked cluster head (CH) nodes and begin communicating with one another. CHs are responsible for carrying out the validation of the session key duration in order to make the authorization and authentication processes easier to complete. In order to improve scalability and support a wider variety of device types, the CH nodes are responsible for locally managing registration services, authentication management, and authorization. CH nodes are responsible for monitoring the locally customized registration process that is used to add new devices to the network.

Authentication of nodes can be accomplished by the distribution of cryptographic keys or through the use of session keys, and edge computing provides cryptographic choices that need less power. Figure 6 shows the authentication and encryption between CH and any node. CH nodes may offer long-term cached session keys as an alternative method when IoT devices have limited resources. CH nodes are responsible for assigning lightweight session keys to other nodes to authenticate such nodes to the network and maintain the authorization of enrolled nodes as approved entities. It has been suggested that symmetric keys and compact cryptography may help overcome the scalability challenges and resource limits that are associated with IoT devices.

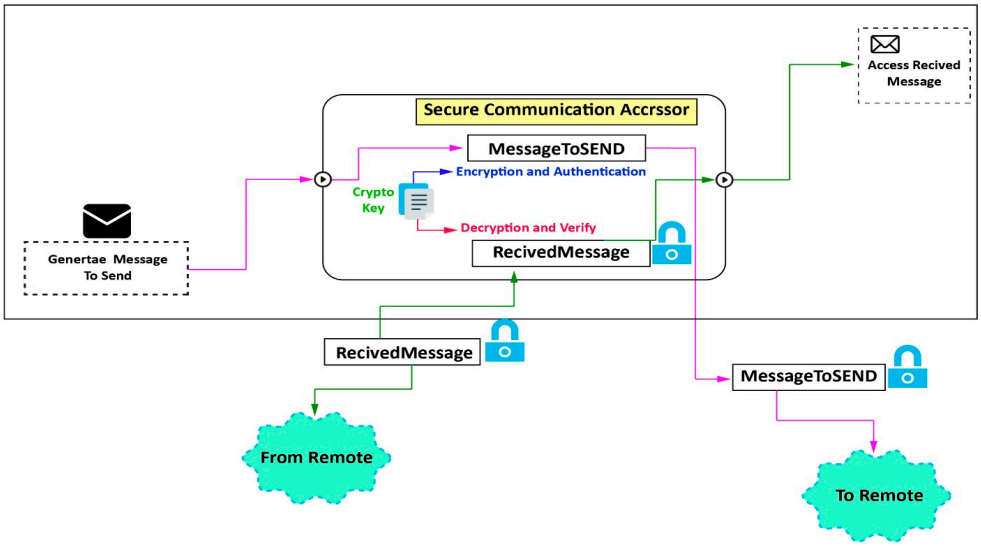


Figure 6. Secure authentication using crypto key between CH and any node.

Four different tasks are carried out by CH nodes:

- The registration of a new node to the system as a new entity;
- The distribution and designation of session keys (cryptographic keys);
- The administration and initiation of communications;
- Management and establishment of safe communications.

The distribution keys undergo encryption through symmetric key-wrapping [26]. A session key, a symmetric key with a unique identifier and a designated duration of validity, safeguards each communication. Secure communications encompass the administration of cryptographic keys (credential management) to enable message encryption, authentication, and decryption. As a result, certain CH nodes are in charge of controlling cryptographic keys. The summary of the permission process is shown in Figure 3.

3.2. Inter-Cluster Symmetry

All the chosen cluster head (CH) nodes collect and transmit data to the upper layer, usually called the base station, where the global blockchain is implemented. In the second

layer, all nodes operate a decentralized private, hybrid local blockchain, facilitating consensus through a predefined consensus algorithm. The POW algorithm [27], initially devised prior to the emergence of cryptocurrencies, found its early application in combating spam. It typically comprises hashing and validation functions. The hash function takes as input variable-length characters or files and produces a fixed-length hash value, rendering it irreversible due to the abandonment of raw and intermediate data during the hashing process. The hash function generates highly random results to ensure equitable competition among nodes. Validation functions establish a range of hash values that adhere to predefined rules, allowing for the measurement of the computational workload. These validation function ranges are dynamically adjusted to mitigate issues stemming from fluctuations in network hash rates, thereby ensuring the stable and seamless operation of the POW algorithm.

However, the POW algorithm is computationally intensive, consuming substantial energy and resources. To address this concern, we propose integrating the POW algorithm with the Delegated Proof of Stake (DPoS) algorithm [28]. DPoS, a variant of the Proof of Stake (PoS) system, employs a consensus mechanism wherein network users vote for delegates responsible for validating blocks. Research indicates that delegated proof-of-stake mechanisms offer security, speed, and efficiency advantages, streamlining blockchain operations.

In our proposed local consensus algorithm, we establish a network of 101 nodes, with the nodes garnering the highest votes being elected as witness nodes. The list of witness nodes is updated every 24 h, similar to the DPoS approach. If a witness node is found to generate blocks at a low rate or exhibit malicious behavior, it loses its credibility and witness identity. To handle such a situation, the consensus process of our proposed local algorithm encompasses three modules: the selection of a specific number of consensus nodes, achieving consensus on block verification, and demotion of malicious nodes. Figure 7 illustrates the structure of the proposed local consensus algorithm. An algorithm is shown in Algorithm 2. Here, N(c) represents the set of all consensus nodes. N(t) represents the trading node. N(a) represents the candidate nodes, and N(w) represents the wetness nodes.

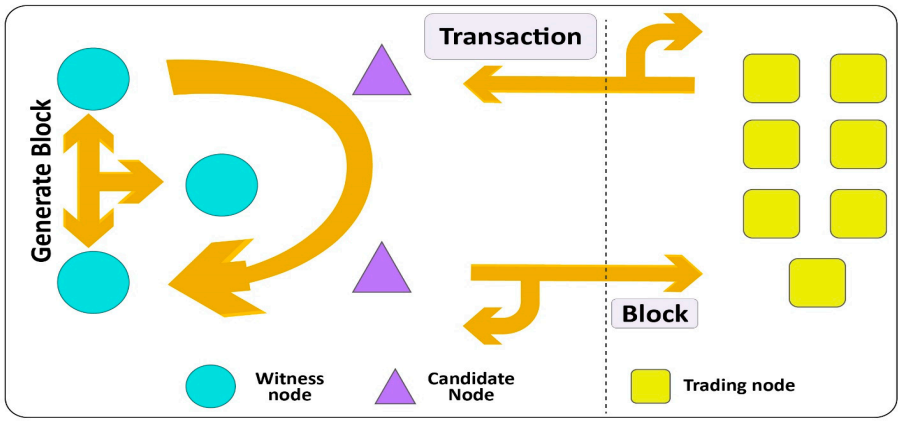


Figure 7. Working of the hybrid local blockchain consensus algorithm.

- The consensus node selection module: All nodes are given specific roles to perform different tasks in the blockchain system. These nodes are categorized primarily into consensus nodes, which consist of witness nodes and candidate nodes, and trading nodes;
- The consensus module: This module handles the whole flow from block generation to block verification, ensuring agreement among the participating nodes;
- The malicious node downgrading module: In the event of detecting a malicious node, the proposed local algorithm switches to the module designed to downgrade the malicious node.

Here, embedded vector data structures within the blockchain can store and organize data efficiently. This can enhance the performance and scalability of the blockchain by

allowing for faster data retrieval and manipulation. The embedded vector data structure ensures that data is stored directly within the blockchain, maintaining the decentralized and immutable nature of the system.

The network architecture under consideration must account for secure communication between CH (Cluster Head) nodes while managing resource limitations and decentralized distribution of IoT network nodes. Effective implementation of the blockchain processing procedure, which requires a significant amount of computational power and takes a significant amount of time, is not ideally suited for use within an IoT system. Consequently, the proposal provides an approach to data transfer based on blockchain technology that is compact, confidential, decentralized, and relies on distributed consensus. Within the IoT system, it is essential to account for the limitations imposed by devices with limited resources. To address this concern, a lightweight cryptographic mechanism will be implemented.

Algorithm 2: POW/DPoS hybrid Local Consensus Algorithm

```
POW/DPoS (D, nonce)
    Broadcast (nonce, D)
    N(j): HASH(NASH(PreBlockHead), nonce)
    While (HASH(PreBlockHead), nonce) > D)
        Nonce ← T nonce + 1
        N ← N(i)
        Broadcast (N(c))
        N(T) ← N ← N(c)
    endwhile
  Return N(w), N(a)
End
```

3.3. The Global Level Symmetry

This layer is made up of a decentralized network of nodes, which are referred to as cluster owners, and they collaborate with one another. In a way similar to that of a cloud server, its major responsibility is to supervise the administration of devices, the creation of data, and the processing of requests. Even though they could be constrained in terms of power and processing capacity, the trustworthy nodes that make up this layer have access to considerable computational resources. With the support of the global blockchain, more robust asymmetric cryptography algorithms are recommended at this level. In order to meet these criteria, it has been determined that implementing elliptic curve cryptography (ECC) is one of the potential solutions that might work for this layer. The global consensus algorithm is based on a hybrid of PBFT and POS. PBFT is a distributed consistency algorithm founded on the principle of state machine replication, wherein each node is required to sign messages sent and cannot alter messages from other nodes. Upon receiving a client request, the subsequent request is executed only after the previous request has been successfully completed through network-wide broadcasting. In the PBFT algorithm, all nodes operate under the same configuration, with a single master node and the remaining nodes functioning as replica nodes. The master node arranges client requests in a specific order and forwards them to the replica nodes.

To address certain limitations, such as inadequate system scalability and excessive communication overhead observed in the PBFT algorithm, we propose an integration of PBFT with the Proof of Stake (PoS) consensus algorithm. PoS is a widely used consensus protocol in blockchain technology that determines the selection of users responsible for validating new blocks of transactions, thereby earning rewards for accurate validation. The PBFT-POS algorithm presented in this study incorporates dynamic calculation of workload values for nodes, assessing their participation in each consensus round. The nodes' respective workload levels are considered when selecting participants for the consensus process. A combination of node workload values and a verified random function is used in order to improve the process of selecting primary nodes. This makes the selection of primary nodes

unpredictable while also lowering the possibility that dishonest nodes will be picked as main nodes. In addition, the PBFT agreement protocol is improved by choosing primary nodes that have a higher degree of dependability. These enhancements are dependent on the selection of the main nodes. This improvement considerably reduces the amount of time and communication burden needed to reach an agreement.

The global hybrid model mechanism is presented in Figure 8, replacing conventional multi-to-multiple communication with a more efficient one-to-multiple communication paradigm. This is achieved by selecting a primary node that is deemed more trustworthy than others. In addition to this, the PBFT-POS consensus algorithm integrates the procedure for changing views into the standard procedure for reaching an agreement in a smooth manner. Traditionally, in the PBFT consensus algorithm, the view change protocol is executed only in the presence of abnormal situations and when a sufficient number of nodes anticipate switching to a different view. In contrast, the WRBFT consensus algorithm selects a new consensus node for the next view after each consensus round, regardless of the success or failure of the previous round. A timeout mechanism is used to ensure the occurrence of the view change. This is done to ensure the liveliness of the algorithm. The Merkle Tree data structure can be utilized in the global consensus algorithm to enhance its efficiency and security. Here, a Merkle Tree can be employed to represent the system's state or transaction log. Each node in the Merkle Tree contains a hash value that represents a portion of the data. The root node of the Merkle Tree, known as the Merkle root, holds a hash value that represents the entire data set. The workings of the Global Consensus Algorithm workload are described in Algorithm 3, outlining the step-by-step processes involved in achieving a global consensus in the system. W represents node workload values, where N stands for the total number of nodes. Each node starts with an initial workload value of WV_{init} and node V is designated as the primary node. $WV_{inilowt}$ and WV_{high} are threshold values.

Algorithm 3: PBFT-POS hybrid Global Consensus Algorithm

PBFT-POS (W, WV_{init}, WV_{low}, WV_{high}, M)
T $\leftarrow WV_{low}$+ $(WV_{init} - WV_{inilowt})/2$;
For j $\leftarrow$ 0 to M $-$ 1 do
→ **If** W[j] > WV_{high}
→ → W[j] $\leftarrow$ Random (WV_{init}, WV_{high})
→ **Else if** W[j] $\leftarrow WV_{init}$
End if
End for
Return W;

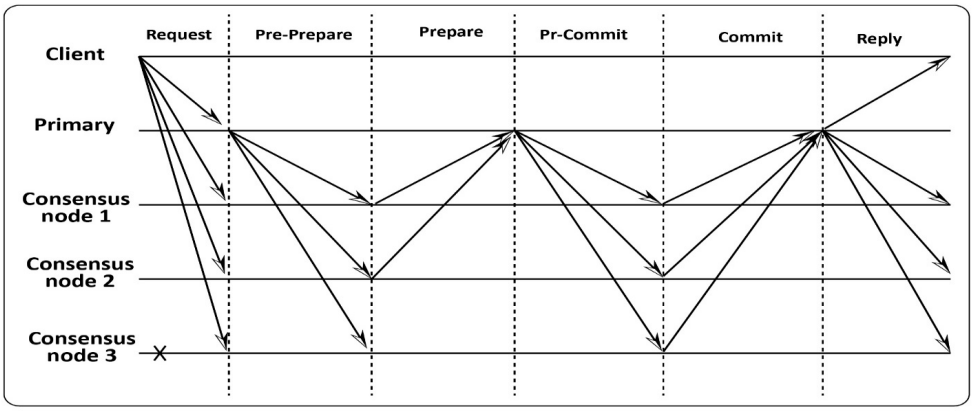


Figure 8. Working process of the global consensus algorithm.

A novel hybrid algorithm is proposed to enhance the existing two-stage confirmation of PBFT by introducing an additional stage, resulting in a consensus process of four stages.

The process is visually represented in Figure 1, and the specific steps of the consensus process are described below:

1. Request: Client C sends a request message to the consensus network, which is represented by the notation 'REQUEST, o, t, c'. In this case, "o" stands for the particular operation that was requested, "t" stands for the timestamp that was associated with the request, and "c" stands for the identification of the client;
2. Preparing in advance: upon receiving an authenticated request message, the main node designates a sequence number seq to this request within the current view v. It then disseminates a pre-prepared message to all participating consensus nodes;
3. Prepare: Once the consensus node has received the verified pre-prepare message, it will then send a prepared message to the main node;
4. Pre-commit: In the pre-commit phase, the main node logs any verified prepared messages it receives;
5. Commit: Once the consensus node has been given the verified pre-commit message, it will then carry out the particular action that has been requested. After that, it will send a commit message to the main node;
6. Reply: Following the reception of verified commit messages, the main node will carry out the action that has been requested and will then send a reply message to the client as well as to the other consensus nodes. It is an indication that the request has been properly carried out if the client gets a valid answer from the main node.

The shared trust between two CH (Cluster Head) nodes and their associated BS (Base Station) nodes is strengthened by the blockchain-based architecture that was presented for use in cooperating with approved organizations. When an entity or node in the Internet of Things participates in communication with nodes located in various clusters, this also increases trust. During the process of node communication, the blockchain-based system fixes addresses since it makes use of the execution of smart contracts. Without having to depend on predetermined addresses or domain names, cluster heads in the architecture that has been suggested are able to connect with edge devices and carry out smart contract operations.

4. Implementation

Simulation models were utilized in two distinct scenarios, each corresponding to a specific level of the proposed work, to showcase the viability and effectiveness of the presented model. The initial setup encompasses clustering, CH nodes, APIs, IoT devices, and an organizational structure, with each component fulfilling specific roles within the network. A comprehensive blockchain deployment simulator was employed to evaluate the implementation of Ethereum and HLF metrics. The Layer-3 simulation model runs on a dedicated workstation, functioning as the BS server for the secure ledger applications.

The IBM Cloud, which provides hosting services for development frameworks and platforms, was utilized for implementing the IoT devices. The Node-Red server facilitated interactions between IoT devices and servers, utilizing the Constrained Application Protocol (CoAP). The IBM IoTWatson platform was equipped with simulated physical nodes that were connected to the appropriate IBM Cloud Foundry services. To ensure the security of Layer-2 operations, a secure and lightweight permission-based blockchain framework was employed, and the IoT server was organized using a virtual environment that included multiple virtual nodes.

The experimental configuration of the authentication process consisted of four peers and one orderer node, all of which ran as Docker images inside containers created by Docker. Hosting services were provided by the Linux Foundation for the authentication (v1.4) blockchain technology, which is both open-source and free to use. The operating system used was Ubuntu Linux 18.4 LTS, which was clocked at 3.4 GHz and had 16 GB of RAM. The CPU that was utilized was an Intel Core i7-3770. Docker-Compose (version 1.17.0) performed the functions of an integrated development environment (IDE), making

configuring Docker images and containers easier. The Docker engine (version 19.03.8) managed the Docker environment.

Setting up a smart contract and deploying it on the peers' nodes with authorized data storage allowed a block of transactions to be recorded in the blockchain ledger. The composer-playground offered a web interface for users to manage transactions and assets and create and use smart contracts. The Composer Command Line Interface (CLI) was made available to developers to install, implement, and run smart contracts with the related specifications. The peers' instances of CouchDB were set to facilitate the effective execution of complicated queries on the transaction logs while simultaneously preserving the state data. The JavaScript Object Notation (JSON) was used for the modeling of the Chain Code (CC). By contacting the CC using APIs, the client application was able to get access to the state database and carry out operations such as putting, getting, and deleting data. A REST server was put up to assist the many processes that take place on a blockchain. This server provides RESTful APIs that web clients and virtual devices may access. Thanks to these APIs, users could submit transactions by means of HTTP requests using either the GET or the POST methods. The Fabric client application was hosted on the REST server, which made it possible to communicate with the HLF network by using the Google Remote Procedure Calls (gRPC) framework. Every node in the network had its own copy of the ledger, which included both the transaction log and a record of any changes to the status of the system. The data pertaining to the state adhered to a key-value pair format that included versioning. Blocks were cryptographically connected to one another to ensure their integrity, and ledger changes were stored in reverse chronological order.

We constructed a Layer Three simulation model to analyze the Ethereum and Hyperledger global networks from the perspective of their throughput as well as their latency. The simulation model carried out blockchain application execution on a workstation that also served in the capacity of BS server. Our simulated environment made it possible for us to evaluate these parameters under similar settings by having us submit both networks to a workload that was produced for them. The testing environment consisted of a distributed configuration that consisted of both blockchain networks combined. The workstation used to develop the simulation models was equipped with 16 gigabytes of memory and a CPU that was an Intel Core i7-3770 running at 3.4 gigahertz. For sake of simplicity, the Ethereum network was designed to consist of a single mining node. Please refer to Section 5 for a more in-depth analysis of the results and suggestions derived from the experiment.

5. Results and Discussion

This section describes the evaluation and experimental outcomes of the proposed hybrid blockchain model. The performance of the proposed work has been tested in three dimensions: cluster formation-based, transaction management-based, and security-based. Based on the experimental evaluation, it has been determined that the proposed technique is much better than the existing cutting-edge methods. A detailed description of each dimension is discussed in the below sub-sections.

5.1. The Cluster Formation Result

The first experiment is about analyzing the clustering formation algorithm discussed in Section 3.1. A network environment for IoT devices was simulated in order to examine the performance of the suggested clustering algorithms, as illustrated in Figure 9. The devices in these clusters are connected via Wi-Fi, Bluetooth, 5G, etc. Further, it is assumed that there is a common access point that enables cluster-to-cluster communication, and that access point is also capable of resolving the compatibility issue by using an appropriate plug-in to enable any device, such as Bluetooth. Moreover, the range of the access point is 100 m, and it is assumed that this length is enough to cover the supposed IoT environment. But this distance can be further extended by using extenders in addition to the ace points. It has 300 nodes that were randomly placed over a 2-D network. In order to compare the

findings, MATLAB 2018a was used because it provided a dependable environment for clustering methods and made it simple to simulate algorithms.

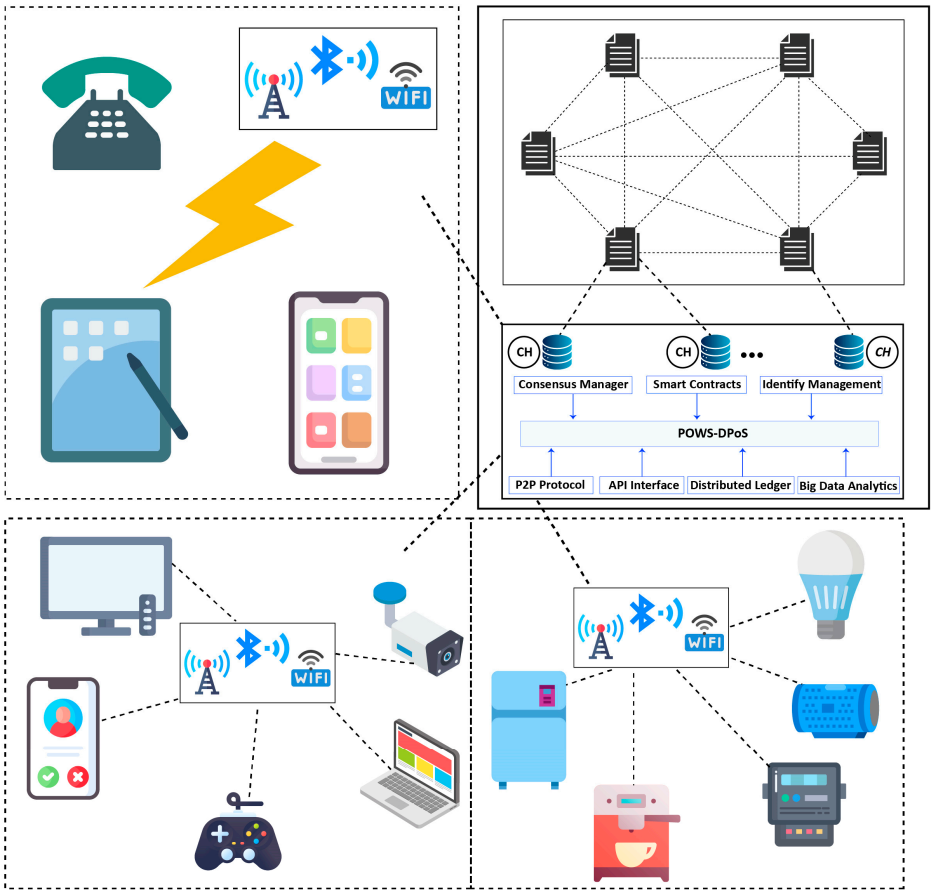


Figure 9. Cluster formation and cluster head selection.

5.1.1. Performance Matrices

Three standard measures have been chosen from the work of A Shokouhifar et al. [30]: each cluster's network load, distance, and area coverage.

5.1.2. Baseline Methods

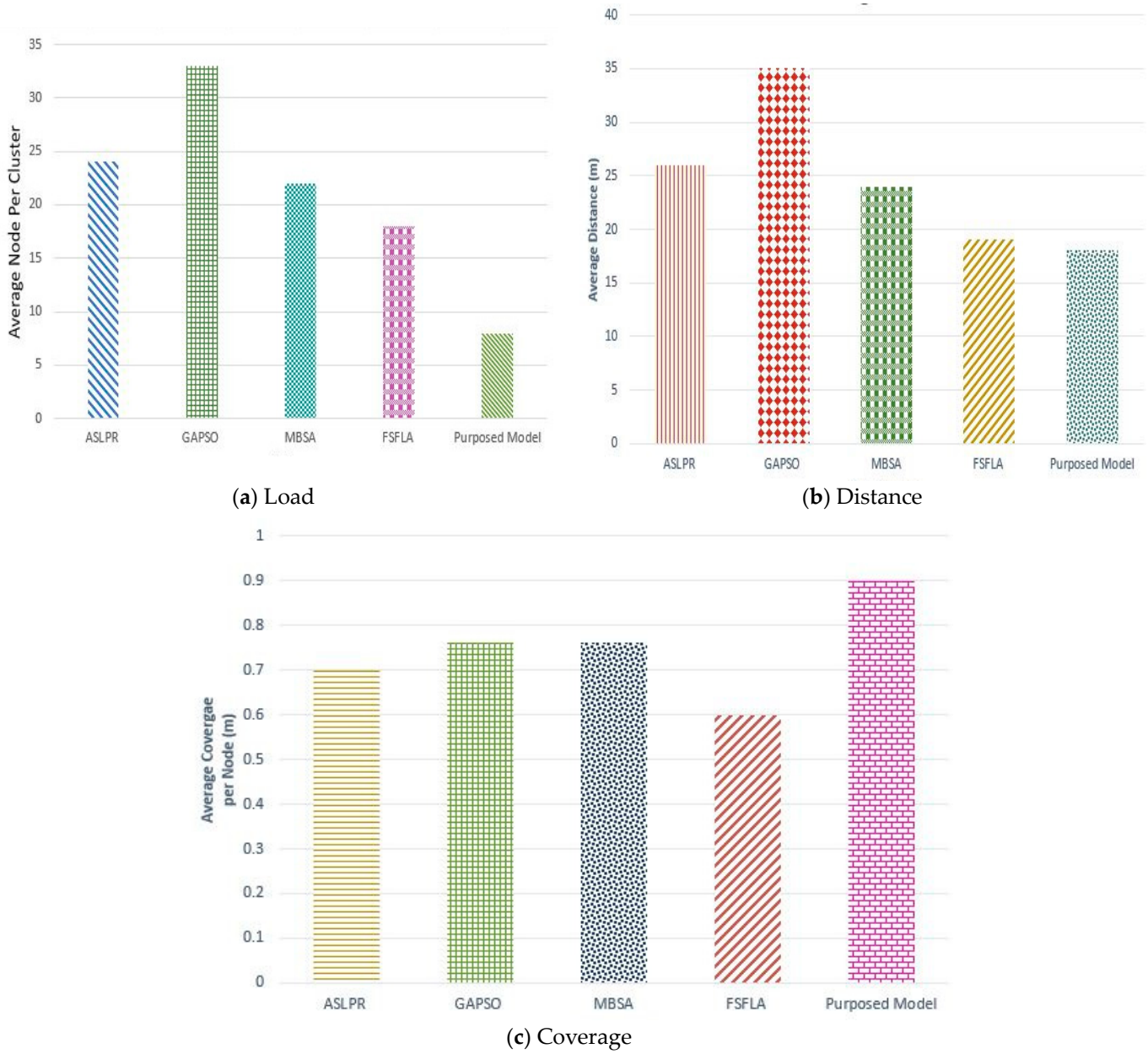
Three different baseline methods have been selected to benchmark the efficiency of the proposed algorithm:

- ASLPR [32]: This method of cluster formation and cluster head selection is based on genetic algorithm and a specifically designed approach for wireless sensor network (WSN);
- GAPSO [33]: This is another benchmark method for clustering and cluster head selection based on particle swarm optimization and a genetic algorithm-based hybrid model;
- MBSA: A multi-layer blockchain-based secure architecture for IoT environments in which cluster formation is made from genetic algorithm;
- FSFLA [34]: A fuzzy logic rules-based protocol is defined for cluster formation. Node-level energy and inter-cluster distance were the key parameters for adding cluster nodes.

5.1.3. Results

The simulation results indicate that the proposed clustering is effective and more efficient at reducing distances and overall network energy. Figure 10a–c illustrates how the suggested clustering model outperforms the other algorithms by reducing the amount of

traffic on the network, shrinking the distances between nodes, and, as a result, increasing the covered area of the network.



(**a**) Load (**b**) Distance

(**c**) Coverage

Figure 10. Comparison of the proposed method with baseline methods (**a**): Load, (**b**) distance, and (**c**) coverage.

5.2. Security Analysis of the Proposed Hybrid Blockchain Model

This section discusses the evaluation of the key contribution, which is the hybrid consensus protocol for security. The proposed technique has been compared with different benchmarked methods by taking various security baseline measures.

5.2.1. Performance Matrices

The key performance matrices for security have been adopted from the work of Demidenko et al. [22]. The details of each parameter are as follows:

- Privacy: The very first security challenge is privacy, in which the blockchain system keeps track of contracts between different entities. So, there needs to be a data disclo-

sure assessment. The identity of an item is encrypted, and its IoT address is used as a pseudonym in the blockchain;

- Authentication: It is also known as Web3, which is a proper blockchain-based authentication mechanism that, in essence, allows different nodes to communicate with each other;
- Flexibility And Heterogeneity: The ability to provide a uniform environment where different individual blockchains work together to achieve more scalability;
- Scalability: the network's capacity to support more nodes and perfect transactions in case of any node failure.

5.2.2. Baseline Method

To evaluate the performance of the proposed model, we compare it to the baseline models listed below and the descriptive data presented in Table 3.

Table 3. Comparative analysis of the proposed work with existing standard models.

Ref.	IoT Environment	Security Challenges				Consensus	Blockchain
		Privacy	Ht. & Flex.	Authentication	Scalability		
[33]	E-health	✔	✖	✖	✔	PoW	Public
[34]	Smart Grids	✖	✔	✔	✔	PoW/PoS	Private
[35]	Vehicular	✖	✔	✖	✖	PBFT	Consortium
[36]	Cloud Comp.	✔	✖	✔	✖	PoS/PoC	Consortium
[37]	Industrial IoT	✔	✔	✖	✖	PoW	Public
[38]	Smart Factory	✖	✔	✖	✖	PoS	Consortium
Proposed	Smart Cities	✔	✔	✔	✔	PoW/DPoS	Consortium

- Christian et al. [35] demonstrate a new blockchain mechanism to protect cloud-hosted healthcare data. They also describe the practical difficulties and the importance of their proposed research;
- Kamanashis et al. [36]: This article presents a security architecture for a smart city that blends a new blockchain model with smart city devices to protect and secure communication;
- Jiawan et al. [37]: This work proposed a consortium blockchain and smart contract technology to provide secure data storage and sharing in vehicle edge networks. In addition, this research suggests a reputation-based data-sharing mechanism to assure high-quality data sharing among vehicles;
- Hong et al. [38]: This article presents a blockchain-based distributed consensus in which data contribution frequency and energy contribution quantity are used to establish proof of work. It also covers the security solutions for vehicular interactions in EVCE computing;
- Zhetao et al. [39]: This work proposed a consortium blockchain technology to develop a safe energy trading system called energy blockchain. They also suggest a credit-based payment structure to facilitate quick and frequent energy trading. In addition, an optimal pricing technique for credit-based loans based on the Stackelberg game is given;
- Qinghua et al. [40]: Present their research in establishing the origin chain system, which uses blockchain technology to trace the origin of items across complex supply chain environments that necessitate a transparent, tamper-proof metadata architecture.

5.2.3. Result

The comparison of metric values in Table 3 demonstrates that the proposed IoT symmetric consensus model based on Hyperledger hybrid blockchain produces better performance than the earlier works documented in scholarly research. The proposed model can tackle privacy, scalability, authentication, and flexibility under a single umbrella structure.

6. Conclusions

This paper introduces a novel security model for IoT devices operating within multi-hop networks, leveraging blockchain's distributed technology. The proposed model presents a practical approach to implementing decentralized blockchain applications for enhancing IoT network security. The paper provides a detailed discussion of the system implementation and elaborates on how the blockchain-based model can enhance IoT system authentication and authorization. The local-global layer model enhances network security, reduces processing load, and minimizes network latency and load. The proposed implementation significantly improves integrity and security by leveraging the peer-to-peer nature of blockchain communication and mapping it to device-to-device communication in cellular systems.

Moreover, the solution addresses various IoT security challenges, including privacy, authentication, heterogeneity, flexibility, and network scalability. The paper compares the proposed hybrid clustering algorithm with existing models through simulation studies, demonstrating superior performance across several metrics, such as network load, network coverage, and distances. Additionally, the research evaluates the performance of a multi-layer blockchain-based framework, highlighting the effectiveness of a lightweight blockchain compared to the global blockchain Ethereum. The authors plan to focus on deploying a practical and scalable testbed, configured as a proposed framework of IoT devices, to further study, analyze, and compare the performance in real-world environments. In the future, the proposed technique can be further enhanced concerning the transaction factor of the blockchain. Moreover, new similarity measures and clustering algorithms could be utilized to obtain better results.

Author Contributions: Conceptualization, methodology, software, writing—original draft preparation, formal analysis, investigation, S.A.; validation, Supervision, A.A. (Aiiad Albeshri); resources, Funding acquisition, A.A. (Ahmed Alhusayni). All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

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

References

1. Boyes, H.; Hallaq, B.; Cunningham, J.; Watson, T. The industrial internet of things (IIoT): An analysis framework. *Comput. Ind.* **2018**, *101*, 1–12. [CrossRef]
2. Vitturi, S.; Zunino, C.; Sauter, T. Industrial Communication Systems and Their Future Challenges: Next-Generation Ethernet, IIoT, and 5G. *Proc. IEEE* **2019**, *107*, 944–961. [CrossRef]
3. Minoli, D. IoT Applications to Smart Campuses and a Case Study. *EAI Endorsed Trans. Smart Cities* **2017**, *2*, 153483. [CrossRef]
4. Abdulhamid, A.; Kabir, S.; Ghafir, I.; Lei, C. An Overview of Safety and Security Analysis Frameworks for the Internet of Things. *Electronics* **2023**, *12*, 3086. [CrossRef]
5. McCutcheon, W.; Pappa, A.; Bell, B.A.; McMillan, A.; Chailloux, A.; Lawson, T.; Mafu, M.; Markham, D.; Diamanti, E.; Kerenidis, I.; et al. Experimental Verification of Multipartite Entanglement in Quantum Networks. *Nat. Commun.* **2016**, *7*, 13251. [CrossRef]
6. Cheng, Y.; Chen, W.; Fan, W.; Huang, W.; Yu, G.; Liu, W. IoTFuzzBench: A Pragmatic Benchmarking Framework for Evaluating IoT Black-Box Protocol Fuzzers. *Electronics* **2023**, *12*, 3010. [CrossRef]
7. Stanley, M.; Gui, Y.; Unnikrishnan, D.; Hall, S.R.G.; Fatadin, I. Recent Progress in Quantum Key Distribution Network Deployments and Standards. *J. Phys. Conf. Ser.* **2022**, *2416*, 012001. [CrossRef]
8. Qiu, X.; Yao, D.; Kang, X.; Abulizi, A. Blockchain and K-Means Algorithm for Edge AI Computing. *Comput. Intell. Neurosci.* **2022**, *2022*, 1–13. [CrossRef]
9. Sobecki, A.; Barański, S.; Szymański, J. Privacy-Preserving, Scalable Blockchain-Based Solution for Monitoring Industrial Infrastructure in the Near Real-Time. *Appl. Sci.* **2022**, *12*, 7143. [CrossRef]
10. Plageras, A.P.; Psannis, K.E.; Stergiou, C.; Wang, H.; Gupta, B.B. Efficient IoT-Based Sensor BIG Data Collection–Processing and Analysis in Smart Buildings. *Future Gener. Comput. Syst.* **2018**, *82*, 349–357. [CrossRef]

11. Guo, J.; Xiong, Q.; Yang, M.; Zhao, Z. A Double-Compensation-Based Federated Learning Scheme for Data Privacy Protection in a Social IoT Scenario. *Comput. Mater. Contin.* **2023**, *76*, 827–848. [CrossRef]
12. Saad, M.; Bin Ahmad, M.; Asif, M.; Khalid Khan, M.; Mahmood, T.; Tag Eldin, E.; Abdel Hameed, H. Blockchain and IIoT Enabled Solution for Social Distancing and Isolation Management to Prevent Pandemics. *Comput. Mater. Contin.* **2023**, *76*, 687–709. [CrossRef]
13. Alsaqqa, S.; Almajali, S. Blockchain Technology Consensus Algorithms and Applications: A Survey. *Int. J. Interact. Mob. Technol.* **2020**, *14*, 142. [CrossRef]
14. Mohanty, S.N.; Ramya, K.C.; Rani, S.S.; Gupta, D.; Shankar, K.; Lakshmanaprabu, S.K.; Khanna, A. An Efficient Lightweight Integrated Blockchain (ELIB) Model for IoT Security and Privacy. *Future Gener. Comput. Syst.* **2020**, *102*, 1027–1037. [CrossRef]
15. Qian, Y.; Jiang, Y.; Chen, J.; Zhang, Y.; Song, J.; Zhou, M.; Pustišek, M. Towards Decentralized IoT Security Enhancement: A Blockchain Approach. *Comput. Electr. Eng.* **2018**, *72*, 266–273. [CrossRef]
16. Meinert, E.; Alturkistani, A.; Foley, K.A.; Osama, T.; Car, J.; Majeed, A.; Van Velthoven, M.; Wells, G.; Brindley, D. Blockchain Implementation in Health Care: Protocol for a Systematic Review. *JMIR Res. Protoc.* **2019**, *8*, e10994. [CrossRef]
17. Prabha, P.; Chatterjee, K. Design and Implementation of Hybrid Consensus Mechanism for IoT Based Healthcare System Security. *Int. J. Inf. Technol.* **2022**, *14*, 1381–1396. [CrossRef]
18. Wankhade, V.R. Adoption of Blockchain Based Smart Application in Machine Learning. *Int. J. Res. Appl. Sci. Eng. Technol.* **2021**, *9*, 600–605. [CrossRef]
19. Alrubei, S.M.; Ball, E.A.; Rigelsford, J.M.; Willis, C.A. Latency and Performance Analyses of Real-World Wireless IoT-Blockchain Application. *IEEE Sens. J.* **2020**, *20*, 7372–7383. [CrossRef]
20. Abdella, J.; Tari, Z.; Mahmud, R.; Sohrabi, N.; Anwar, A.; Mahmood, A. HiCoOB: Hierarchical Concurrent Optimistic Blockchain Consensus Protocol for Peer-to-Peer Energy Trading Systems. *IEEE Trans. Smart Grid* **2022**, *14*, 3927–3943. [CrossRef]
21. Pabitha, P.; Priya, J.C.; Praveen, R.; Jagatheswari, S. ModChain: A Hybridized Secure and Scaling Blockchain Framework for IoT Environment. *Int. J. Inf. Technol.* **2023**, *15*, 1741–1754. [CrossRef]
22. Kaur, M.; Gupta, S.; Kumar, D.; Verma, C.; Neagu, B.-C.; Raboaca, M.S. Delegated Proof of Accessibility (DPoAC): A Novel Consensus Protocol for Blockchain Systems. *Mathematics* **2022**, *10*, 2336. [CrossRef]
23. Xu, R.; Chen, Y. µDFL: A Secure Microchained Decentralized Federated Learning Fabric Atop IoT Networks. *IEEE Trans. Netw. Serv. Manag.* **2022**, *19*, 2677–2688. [CrossRef]
24. Alhejazi, M.M.; Mohammad, R.M.A. Enhancing the Blockchain Voting Process in IoT Using a Novel Blockchain Weighted Majority Consensus Algorithm (WMCA). *Inf. Secur. J. A Glob. Perspect.* **2021**, *31*, 125–143. [CrossRef]
25. Khezr, S.; Yassine, A.; Benlamri, R. Towards a Secure and Dependable IoT Data Monetization Using Blockchain and Fog Computing. *Clust. Comput.* **2022**, *26*, 1551–1564. [CrossRef]
26. Xu, H.; Zhang, L.; Onireti, O.; Fang, Y.; Buchanan, W.J.; Imran, M.A. BeepTrace: Blockchain-Enabled Privacy-Preserving Contact Tracing for COVID-19 Pandemic and Beyond. *IEEE Internet Things J.* **2021**, *8*, 3915–3929. [CrossRef]
27. Kumar, R.N. Comparative Study of Proof of Work (PoW) and Delegated Proof of Stake (DPoS) Blockchain Consensus Algorithm. *Int. J. Res. Appl. Sci. Eng. Technol.* **2021**, *9*, 650–654. [CrossRef]
28. Rashid, M.M.; Choi, P.; Lee, S.-H.; Kwon, K.-R. Block-HPCT: Blockchain Enabled Digital Health Passports and Contact Tracing of Infectious Diseases like COVID-19. *Sensors* **2022**, *22*, 4256. [CrossRef]
29. Rustam, F.; Raza, A.; Ashraf, I.; Jurcut, A.D. Deep Ensemble-based Efficient Framework for Network Attack Detection. In Proceedings of the 2023 21st Mediterranean Communication and Computer Networking Conference (MedComNet), Ponza, Italy, 13–15 June 2023; pp. 1–10.
30. Ezhilarasi, M.; Gnanaprasanambikai, L.; Kousalya, A.; Shanmugapriya, M. A novel implementation of routing attack detection scheme by using fuzzy and feed-forward neural networks. *Soft Comput.* **2023**, *27*, 4157–4168. [CrossRef]
31. Rustam, F.; Raza, A.; Ashraf, I.; Jurcut, A.D. Prim Algorithm Approach to Improving Local Access Network in Rural Areas. *Int. J. Comput. Theory Eng.* **2011**, *3*, 413–417. [CrossRef]
32. Shokouhifar, M.; Jalali, A. A New Evolutionary Based Application Specific Routing Protocol for Clustered Wireless Sensor Networks. *AEU-Int. J. Electron. Commun.* **2015**, *69*, 432–441. [CrossRef]
33. Wang, S.; Li, H.; Chen, J.; Wang, J.; Deng, Y. DAG Blockchain-Based Lightweight Authentication and Authorization Scheme for IoT Devices. *J. Inf. Secur. Appl.* **2022**, *66*, 103134. [CrossRef]
34. Fanian, F.; Kuchaki Rafsanjani, M. Memetic Fuzzy Clustering Protocol for Wireless Sensor Networks: Shuffled Frog Leaping Algorithm. *Appl. Soft Comput.* **2018**, *71*, 568–590. [CrossRef]
35. Esposito, C.; De Santis, A.; Tortora, G.; Chang, H.; Choo, K.-K.R. Blockchain: A Panacea for Healthcare Cloud-Based Data Security and Privacy? *IEEE Cloud Comput.* **2018**, *5*, 31–37. [CrossRef]
36. Hakak, S.; Khan, W.Z.; Gilkar, G.A.; Imran, M.; Guizani, N. Securing Smart Cities through Blockchain Technology: Architecture, Requirements, and Challenges. *IEEE Netw.* **2020**, *34*, 8–14. [CrossRef]
37. Kang, J.; Yu, R.; Huang, X.; Wu, M.; Maharjan, S.; Xie, S.; Zhang, Y. Blockchain for Secure and Efficient Data Sharing in Vehicular Edge Computing and Networks. *IEEE Internet Things J.* **2019**, *6*, 4660–4670. [CrossRef]
38. Liu, H.; Zhang, Y.; Yang, T. Blockchain-Enabled Security in Electric Vehicles Cloud and Edge Computing. *IEEE Netw.* **2018**, *32*, 78–83. [CrossRef]

39. Li, Z.; Kang, J.; Yu, R.; Ye, D.; Deng, Q.; Zhang, Y. Consortium Blockchain for Secure Energy Trading in Industrial Internet of Things. *IEEE Trans. Ind. Inform.* **2017**, *14*, 1. [CrossRef]
40. Lu, Q.; Xu, X. Adaptable Blockchain-Based Systems: A Case Study for Product Traceability. *IEEE Softw.* **2017**, *34*, 21–27. [CrossRef]