Scientific Research Publishing

Journal of Service Science and Management, 2021, 14, 115-131
https://www.scirp.org/journal/jssm
ISSN Online: 1940-9907
ISSN Print: 1940-9893

Optimizing Blockchains Structures Based on Entropy and TOPSIS Model

Parisa Sabbagh

DISA-MIS Department, University of Salerno, Fisciano, Italy
Email: psabbagh@unisa.it

How to cite this paper: Sabbagh, P. (2021). Optimizing Blockchains Structures Based on Entropy and TOPSIS Model. *Journal of Service Science and Management, 14,* 115-131.
https://doi.org/10.4236/jssm.2021.141008

Received: July 8, 2020
Accepted: February 23, 2021
Published: February 26, 2021

Abstract

In this study, a new and applicable model is proposed which represents the opportunities of implementing blockchains into IoT framework in the sensor data acquisition. We aim to propose a simple perspective of blockchain in IoT which is flexible with high throughput in sensor networks. This method is based on probabilities that use the combination of the Entropy and TOPSIS model and the blockchain parameter configuration used in the proposed model in this study includes: Parameters, Block, Epoch Time, Consensus Protocol, Throughput Queuing, Network Topology and Containerization. We selected Hyperledger Fabric as our blockchain solution to deploy across our network with the use of Docker Swarm. Our framework leverages the containerization of Fabric so that the network can operate between the edge devices and the cloud in a single, private system. This model contains some areas in security, such as privacy policy and delay versus the value of the task to be processed.

Keywords

Blockchain, Internet of Things (IoT), Entropy, TOPSIS, Sensor Data

1. Introduction

Blockchain technology has shown promising application prospects, which is a distributed means of securing data in a way that is auditable, immutable, and fault-resistant. Blockchain introduced in 2009 (Nakamoto, 2008), which debut of Bitcoin served as a functional proof-of-concept and removed any necessary access for trusted third parties in the transaction with any strange people in the world. These transactions need to validate from banks, but Blockchains carried out by network peers. Blockchains are based on trust which created in a trust-less platform to act as rules. When it created, Bitcoin had some faults which occurred by heavy cost on computation, power, and memory for every full participant in a wide network of peers. As of May 2018, the Bitcoin ledger size surpassed 196 GB (BitInfoCharts, 2018). Also, other Bitcoin limits are well-documented in works such as (Croman et al., 2016; Eyal & Sirer, 2013). For instance, as mentioned by (Vukolić, 2016) in the early days of Bitcoin, the performance of its probabilistic proof-of-work (PoW) based consensus fabric, also known as the blockchain, was not a major issue. The situation today is radically different and the poor performance scalability of early PoW blockchains no longer makes sense. Also, (Croman et al., 2016) stated that the increasing popularity of blockchain-based cryptocurrencies has made scalability a primary and urgent concern. Also, (Eyal & Sirer, 2013) believe that Bitcoin mining is vulnerable since Bitcoin's mining protocol is not incentive-compatible. Higher revenues can lead new miners to join a selfish miner pool, a dangerous dynamic that enables the selfish mining pool to grow towards a majority. They show that at least 2/3 of the network needs to be honest to thwart selfish mining; a simple majority is not enough. These works and references notice that Bitcoin is very slow to process any transactions and it is not scalable when participants grow and may still be susceptible to a number of different attacks. On the other hand, the number of smart devices with wireless communication capabilities has risen to a scale of billions (Gartner, 2015). This kind of growth is to occur the manufacturing costs of electronics and will develop a smart device with sensors connected to the Internet. These sensors have a lot of tasks by using the Internet, but these sensors nodes which defined and determined in smart devices have some limitations, such as processing power, preventing them from running CPU intensive tasks locally, privacy policy, and security. Because of these limitations, these sensor nodes can offload their processing tasks to run in a cloud area server which can be faced with some other challenges, such as delay and low access and response time to users. This challenge can be solved by defining some edge devices as an intermediary between the end-users nodes which use smart devices and cloud. These edge devices can bring the local connection to the smart devices and challenges, such as delay and low access and respond time to users run in local when other tasks are offloading to the cloud. For solving this problem, a cloud and edge computing model should be improved to optimize delay and low access and respond to time to users.

By combining Blockchain to these edge-centric IoT systems, some other challenges will be taken which surveyed in (Dorri et al., 2016; Yeow et al., 2017). Because IoT is not expensive as Blockchain in power consumption, communication, computation, and memory usage, combining IoT and Blockchain supposed to be a creative work that is a distributed ledger. The traditional approach of centralized services has a single point of failure, but Blockchain does not have it. Every participating node has a copy of the Blockchain ledger, and changes to that ledger are very difficult to make without the proper consensus of some determined number of participants. By securing the IoT system with an optimized structure of Blockchain can bring decentralized management of devices. In addition, Blockchain's immutability can offer a way to audit and track data sources.

In this study, we aim to propose a simple perspective of blockchain in IoT which is flexible with high throughput in sensor networks. This model contains some areas in security such as privacy policy and delay versus the value of the task to be processed. Also, we propose framework trends present in the literature and how to implement them. In Section 2, an overview of the core Blockchain concept studied; Section 3, presented trends in the IoT framework; Section 4 discussed Blockchain parameters used in the proposed model; and Section 5 tried to describe applying Blockchain to an IoT system implementation.

2. Blockchain Overview

The name "Blockchain" itself refers to the structure of the Blockchain's data can be thought of as an immutable chain of events. Data, mostly in the form of transactions, are grouped together in a block. This block is then packaged with a reference to the previous block, which contains a reference to the block before that, etc. Blockchain is "distributed" because every participant in the network holds a copy of this append-only ledger. All participants, or peers, must agree on the state of the ledger and unauthorized changes to that ledger must be reasonably detectable. In reality, Blockchain technology requires 1) a distributed ledger among peers, 2) a consensus protocol to ensure that all peers have the same copy, and 3) a cryptographic infrastructure. Every other detail is determined by the desired application. *Since public blockchain platforms are open to the world, they can rapidly draw the attention of software development companies and communities to the strengths of blockchain technology* (Sicilia & Visvizi, 2019). *presently, the intricacy of supply chains and the issues that exist in the administration of this chain prompts an exercise in futility and cash. Utilizing new innovations, for example, blockchain will improve quality and diminish costs* (Troisi et al., 2020).

2.1. Consensus Protocols

We provide an overarching classification for consensus methods. In-depth protocol algorithms and mathematical proofs of robustness are beyond the scope of this paper. Consensus protocols have their own massive pool of research effort and are well-reviewed in other works: (Vukolić, 2016; Yeow et al., 2017; Christidis & Devetsikiotis, 2016; Cachin & Vukolic, 2017).

1) Lottery Election: Lottery election, such as Proof of Work (PoW), relies on the probability to "elect" a consensus leader who determines the order of incoming transactions, usually for a set amount of time. In Bitcoin, peer nodes append a nonce to a block and calculate the hash value. The resulting value must have some pre-determined number of leading zeros. Peers constantly hash new values in order to find the correct "answer." The first peer to find the right nonce broadcasts its results to the network, who verifies and appends the block organized by the winning leader. This temporary leader is the one that deter-

mines the order of transactions within its announced block. In this case, the difficulty is determined by the number of leading zeros and the leader is only the leader for one block. Other lottery election examples include Proof of Stake (PoS) and Proof of Elapsed Time (PoET) (Cachin & Vukolic, 2017). Although this is by no means an exhaustive list. Note that election difficulty and leader term lengths could be configurable parameters.

2) Majority Election: The majority election refers to a majority of peers voting on a particular value. Peers could vote to validate a transaction, or vote upon a block leader. The distinction here is that majority election does not rely on probability, but is instead far more communication-bound—the notable example being Practical Byzantine Fault Tolerance (PBFT) (Castro & Liskov, 2002). The consensus in this manner must take care to manage its upscaling to prevent significant communication overhead. The general trend is to create "round-robin" or subgroup voting pools to mitigate scaling issues (Mazières, 2016).

2.2. Implementation Differences

1) ***Read-Write Access***: Blockchain systems can be defined by what entities have read and write access to the ledger. As aforementioned, write access is append-only. If any peer can read the ledger, it is public. If the read access is limited, it is private. If any peer can append to the ledger, it is permissionless. If write access is limited, it is permission. Bitcoin serves as the prime example of a public, permissionless Blockchain network. Anyone can participate in the network and the ledger is open to the public. Anonymity is preserved by the use of public-private key pairs. Public networks tend to require computationally heavy consensus methods—or, in the case of Ethereum's hash method (Buterin, 2015), computation, and memory-intensive. This is mainly to protect against Sybil attacks and prevent double-spending (see II-C). A network like Sovrin (Windley & Reed 2018) is public and permission. Anyone has access to ledger information, but additions to the ledger can only be made by a specific set of participants. Permissioned Blockchain has the benefit of not requiring consensus methods as resource-intensive as permission-less systems since unauthorized parties could be revoked for not being part of a whitelist. Privacy is not a goal for Sovrin, which is an identity management system. Alas, even for Bitcoin's anonymous addressing, true privacy is not guaranteed on a public network (Ben-Sasson et al., 2014). Private, permission Blockchain like Hyperledger Fabric (Yang & Enyeart, 2020) target enterprise applications, where businesses may want the fault tolerance and self-management offered by Blockchain within a private network. Smaller networks reduce communication overhead but tend to be less secure as a result—large, public networks have the advantage of peer numbers, where a 51 percent majority attack is more difficult to execute.

2) ***Block Handling***: Block handling refers to methods that try to reduce Blockchain latencies, either in writing to the ledger or reading from it for transaction validation. These methods include block ordering, pruning, and ledger sharing. Block ordering can be handled in a number of ways. Bitcoin can result in forks when more than one peer concurrently broadcasts a valid block to append to the ledger. At that point, peers will continue to "race" against one another to find the next hash value, and once the longer chain is created, the other peers will adopt that chain. Ethereum mitigates this wasted effort by creating incentives to include so-called "orphaned" blocks into the main chain. Directed Acyclic Graphs (DAG) have also been suggested for block ordering to decrease ordering delay (Lewenberg et al., 2015; Popov, 2017). Whereas traditional Blockchain is mostly linear in structure, DAG Blockchain allows for more complex web chains. Block pruning has been suggested to reduce ledger size (Kokoris-Kogias et al., 2017), generally by reducing older blocks into a new "jumpoff" point from which the ledger can continue. As long as all peers reach consensus and agree to prune, the ledger can be collectively reduced. This method needs to carefully define at which point old data is considered "old enough." Ledger sharing is another method to reduce ledger read times. A system could provide service-specific Blockchain, such as in (Gencer et al., 2016). Transaction validations would avoid searching through unnecessary blocks in order to find relevant information, but be linked at common blocks for some set interval. Other proposed protocols with ledger sharing can be found in (Luu et al., 2016). Possible attack vectors on a Blockchain network. Which the peer node outside the inner shaded area is the only non-malicious actor presented in **Figure 1**.

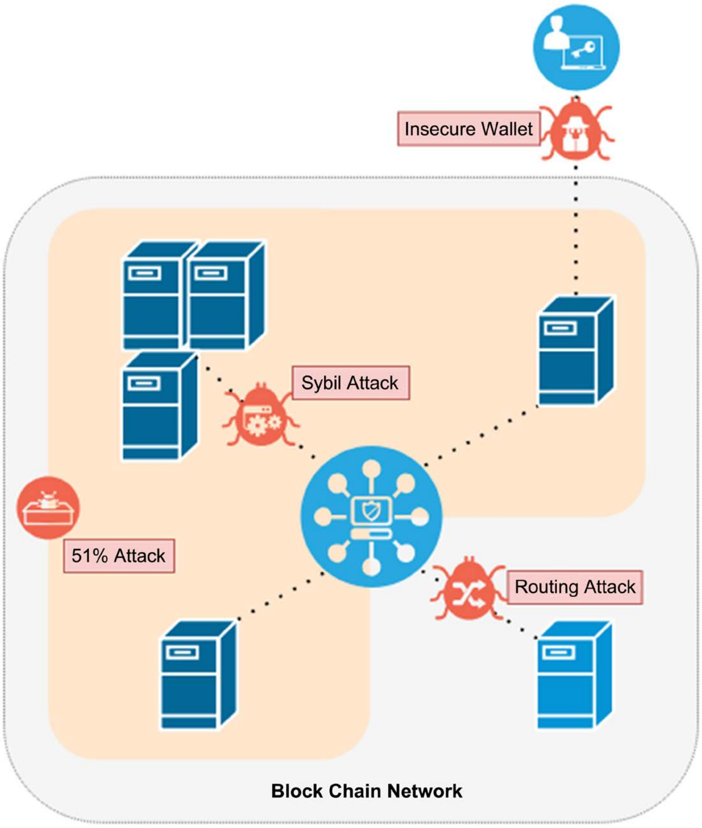

Figure 1. Possible attack vectors on a Blockchain network. In this case, the peer node outside the inner shaded area is the only non-malicious actor.

2.3. Common Attack Vectors

Common attack vectors involving Blockchain systems include Sybil attacks, 51% attacks, routing attacks, and insecure wallet implementations (**Figure 1**). Keeping these attacks in mind that are as follows help to remember the limitations and requirements of a Blockchain system.

1) ***Sybil Attack*****:** Sybil attacks refer to the generation of multiple virtual peers by a malicious party with the intent to influence a network. Permissioned Blockchain will throw out unauthorized requests to the network. Public networks such as Bitcoin and Ethereum utilize CPU-intensive algorithms to prevent such attacks, forcing participants to "vote" with computing power.

2) 51% ***Attack*****:** The 51% attack is similar to the Sybil attack, but more broadly refers to a malicious party controlling the majority of the network to influence the Blockchain. In a system that relies on computing resources to build a Blockchain, the majority pool will be able to dominate the ledger.

3) ***Routing Attacks*****:** Since Blockchain networks are heavily communication-dependent, routing attacks can affect block propagation, and thereby block ordering on more remote peers (Apostolaki et al., 2017). This can be especially detrimental to implementations of Blockchain that make use of timeouts to throw out potentially invalid or malfunctioning peers.

4) ***Insecure Wallet Implementations*****:** Probably the most publicized weaknesses of Blockchain networks are insecure wallet implementations (Boireau, 2018). Crypto-currency networks rely on public-private key infrastructure for transaction addressing. Possession of private keys provides proof of ownership of currency like bitcoin. These private keys can be managed by applications called wallets, which, if compromised, in turn, compromises the security of the linked currency. This vulnerability falls back upon user-end passwords and implementation design, not upon Blockchain technology itself.

2.4. Applications

Blockchain is inherently transactional, its functionality has been expanded to include smart contracts—code stored on the Blockchain that executes upon fulfillment of programmed criteria (Szabo, 1996). Instead of requiring some trusted third party to verify an agreement between parties, a contract stored on the Blockchain could automate specified transactions. Smart contracts provide a way to create a system that is, to some degree, self-managing. With smart contracts, blockchain networks can accomplish more than crypto-currency. They can provide proof of existence, intellectual property rights, public notaries, supply chain management (Christidis & Devetsikiotis, 2016; Bahga & Madisetti, 2016; Pureswaran & Brody, 2014; Crosby et al., 2016), any application that could benefit from an immutable record. Blockchain has been investigated for smart grids, smart homes (Andersen et al., 2017; Dorri et al., 2017), decentralized program applications (Buterin, 2015) database storage (McConaghy et al., 2016), and the list of potential uses continues to grow with every passing day.

3. Proposed Method

Though merging Blockchain and IoT is not without its challenges, a common theme in overcoming issues of delay and computation overhead is to utilize edge, fog, and cloud computing. Edge devices serve as gateway nodes for data aggregation and packaging while leveraging the more plentiful resources of the cloud. This emerging hierarchy of computing resources tends to follow the patterns shown in **Figure 2**: centralized, locally centralized, decentralized, or layered. Works such as (Stanciu, 2017) and (Liang et al., 2017) mention the need for edge devices and the leveraging of a cloud service for data persistence. Other suggestions like (Dorri et al., 2017) utilize a locally centralized design. Many approaches such as in (Aniello et al., 2017; Daza et al., 2017) suggest a layering of Blockchains to handle the delay. Edge-level devices would have a Blockchain tailored for faster verification and higher throughput of data, while higher levels could utilize more robust, yet slower consensus methods in a larger network of peers at the cloud level. The layers would be linked by some form of verification process depends on the specific framework's algorithms. **Figure 2** represented trend patterns for the blockchain-IoT framework. The use of edge computing in the IoT, as well as cloud and fog computing, can be seen on the IoT in **Figure 3**. **Figure 3** presents an overview of the architecture which is categorized into three layers, i.e. Edge, Fog, and Cloud. As represented in **Figure 3** data acquired by the sensors transmit to the edge devices. The edge layer transmits the data to the fog layer. Each fog node covers the small associated community and is responsible

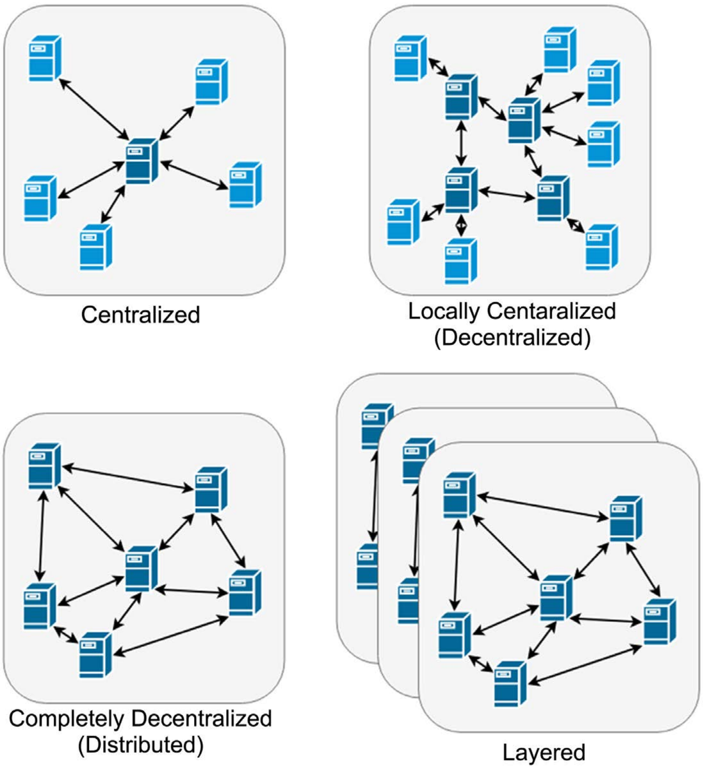

Figure 2. Trend Patterns for Blockchain-IoT Frameworks.

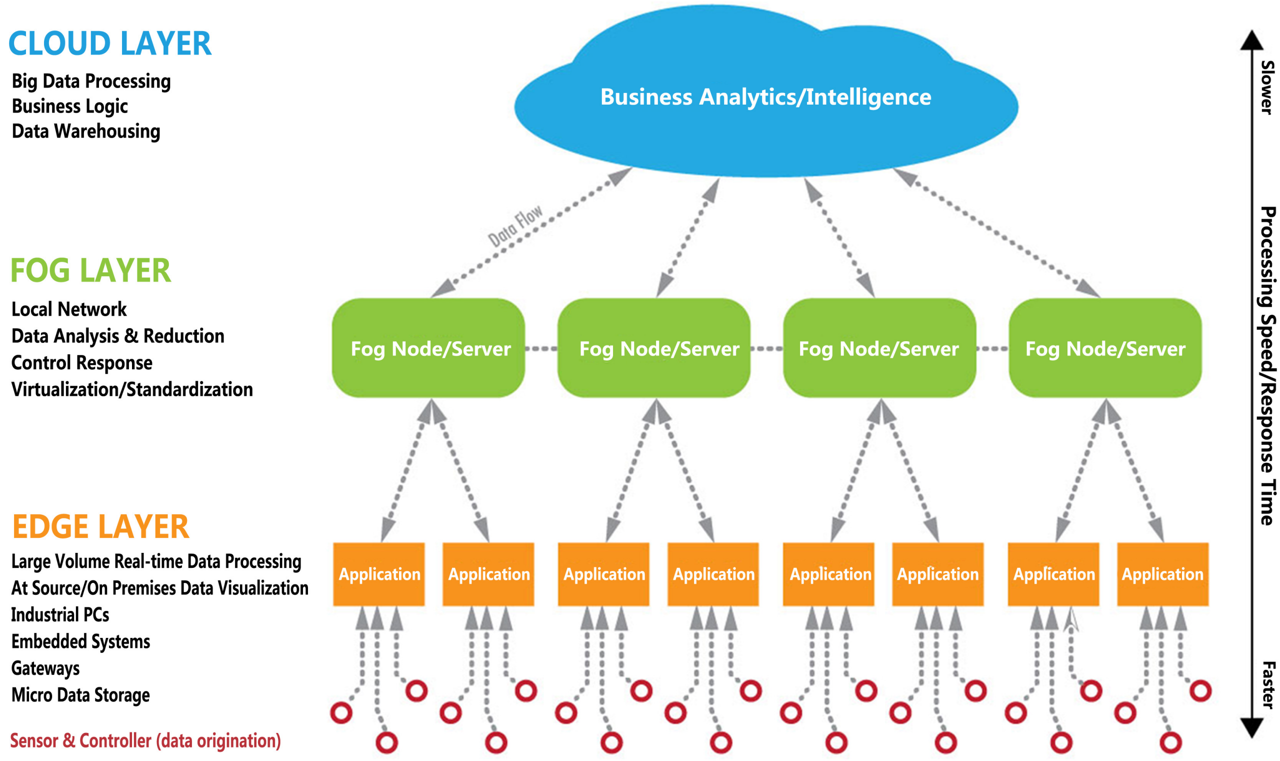

Figure 3. Cloud, fog and edge computing in IoT environment.

for data analysis and service delivery in a timely manner. The fog layer provides localization, while the cloud layer provides wide-area monitoring and control. A fog node can access the distributed cloud over the internet to flexibly deploy the application service and computing availability. As mentioned by Sharma et al. (2017) rather than transferring raw IoT data streams to the cloud, we can locally gather, categorize, and analyze data by deploying a number of fog nodes in the IoT network. Fog computing is an emerging computing model that brings computing abilities to the edge of the distributed IoT network. This distributed computing infrastructure includes a set of physical machines with high-performance capabilities that are linked to one another this can greatly mitigate traffic in the core network and potentially speed up the processing of large amounts of IoT data.

4. Proposed Model Parameters Configuration

IoT networks tend to produce large amounts of data that need to be analyzed for further action. This need incites requirements for the delay, throughput, authentication, integrity, and likely privacy while minimizing resource consumption. In an environment of numerous devices, a system also needs to be careful with the management of identities and cryptographic keys.

In this study, we aim to propose a simple perspective of blockchain in IoT which is flexible with high throughput in sensor networks. This model contains some areas in security such as privacy policy and delay versus the value of the task to be processed. In this section, we discuss the blockchain parameter configuration used in the proposed model that includes Parameters Block, Epoch Time, Consensus Protocol, Throughput Queuing, Network Topology, and Containerization. Blockchain parameters used in the proposed model seek aims to propose a simple perspective of blockchain in IoT which is flexible with high throughput in sensor networks. Also, contains some areas in security such as privacy policy and delay versus the value of the task to be processed.

1) ***Block Parameters***: Block size can be varied by restricting the number of transactions to each block, or placing a limit on data size. Data size will affect propagation times and communication channel overhead. Block intervals, the time allowed between block creation, can also be controlled. Bitcoin aims to create blocks roughly every 10 minutes (Abraham et al., 2016), whereas Ethereum will create a new block about every 15 seconds (Buterin, 2015). This affects the time it takes for a transaction confirmation since blocks that have been in the chain longer are more secure it would take more effort for a malicious party to change older blocks than newer ones.

2) ***Epoch Time***: Peer leadership for consensus methods could last longer than the lifetime of the most recent block. This is sometimes referred to as the epoch time, which could help reduce communications by reducing the count of elections required. On the other hand, a longer epoch time in the hands of a compromised node could prove disastrous.

3) ***Consensus Protocol***: Aniello et al. (2017) suggested using different consensus protocols at the edge versus at the cloud. Investigating changes in the consensus protocol at the edge layer could be worthwhile, especially if those changes are made to accommodate environmental changes. Such changes could involve noisier networks, spotty connections, or even momentary peer failures. To some degree, dynamic consensus parameters are already utilized in the Bitcoin network. Hashing difficulty is altered to ensure block intervals at a set rate, which is a response to peer activity. We could take this further and have a system respond to peer count, channel conditions, peer status, etc. Other difficulty controls for consensus systems are reviewed in (Kraft, 2016).

4) ***Throughput Queuing***: In the case of changing environments, end devices that are trying to submit data to the ledger may not be able to transmit at an optimal rate. If an end device is functioning but its gateway node loses connection, ideally the data should be preserved. Once the connection is restored, the peer may need to make use of some kind of queuing policy to prevent flooding the network with requests.

5) ***Network Topology***: Since Blockchain implementation is dependent upon the distribution of peers by necessity, having the actual network reconfigure in response to the environmental conditions could prove an interesting adaptation. This would work best in a permissions environment, else the system would risk a 51 % attack from malicious peers. Additionally, difficulties would arise with key and certificate distributions.

6) ***Containerization***: The suggestion to utilize containers in IoT-Blockchain systems has been investigated in (Stanciu, 2017), and indeed projects such as Hyperledger Fabric are closely coupled with Docker containers for pluggable operation. Container orchestration across environments could be handled with Docker Swarm or Kubernetes.

Table 1 represents the main parameters of this study to model optimizing blockchain in IoT system by using the Entropy and TOPSIS method. All of these parameters mentioned in Table 1 are used in the equation of this research which is placed in the form of a table to avoid redundancy of the description. The parameters and the indexes inside the table are taken from the proposed model relationships and mathematical model defined by the author.

At first, the confidentiality of data must be modeled. Hence, the local error is in the form of Equation (1) at the Blockchain in IoT system.

$$\epsilon_{e,t} = \frac{1}{n}\sum_{i=1}^{n}\epsilon_{i,e,t} = \frac{\left|r_{i,e,t} - s_{i,e,t}\right|}{\left|r_{i,e,t}\right| + \left|s_{i,e,t}\right|} \tag{1}$$

Table 1. Proposed model main parameters.

Description	*Mathematic Symbols*
Network graph	$\xi = \left(I \cup \{A\}.E\right)$
n data provider set	I
Data submitter (transmission)	A
Network communication set	E
Measurement number for e iteration	T_e
i raw data record in e iteration	$r_{i,e,t}$
i raw data provider in e iteration	$R_{i,e}$
Raw data domain	$\mathcal{R}$
Cumulative function	α
Summarizer function	$fs : \mathcal{R}^{T_e} \rightarrow S^{T_e}$
Data record summarizer	$s_{i,e,t}$
i summarizer date in e iteration	$S_{i,e}$
Summarizer domain data	S
e local error in iteration and t time	$\epsilon_{e,t}$
e global error in iteration and t time	$\varepsilon_{e,t}$
Data groups	$G \subset I$
Data groups numbers	m
Internal data integration groups	$a_{e,t}^{G}$
Data supplier (transmitter and receiver)	a, a_1, a_2
Local group error for G group	$\epsilon_{e,t}^{G}$
Total group error for G group	$\varepsilon_{e,t}^{G}$

In the above equation, each section is the difference between raw data and summary in the provider i (sender or receiver of data). A higher level local error in detecting attacks gives greater security and confidentiality. It should be noted that local error does not depend on cumulative function. The precision of the confidentiality of data in detecting attacks is calculated by general error which is in accordance with Equation (2).

$$\varepsilon_{e,t} = \frac{\left|\alpha\left(R_{e,t}\right) - \alpha\left(S_{e,t}\right)\right|}{\left|\alpha\left(R_{e,t}\right)\right| + \left|\alpha\left(S_{e,t}\right)\right|} \tag{2}$$

Given the fact that the confidentiality of data is possible under the conditions of Equation (1) and (2), the average difference between raw data $R_{e,t} = \left(r_{i,e,t}\right)_{i=1}^{n}$ and the summarized data $S_{e,t} = \left(s_{i,e,t}\right)_{i=1}^{n}$. The higher the overall error, the lower the response will be to maintain the confidentiality of information in communicating and detecting attacks in Blockchain in IoT system. To maintain the confidentiality of data when communicating devices, they must compute a cumulative distribution function between raw data and cumulative data, which is called a local group error whose relationship is as Equation (3).

$$\epsilon_{e,t}^{G} = \frac{\left|r_{i,e,t} - a_{e,t}^{G}\right|}{\left|r_{i,e,t}\right| + \left|a_{e,t}^{G}\right|}, i \in G \tag{3}$$

Similarly, the cumulative distribution function must be computed between aggregated data and cumulative data, which is called the sum of the group error by calculating with Equation (4).

$$\varepsilon_{e,t}^{G} = \sum_{i \in G} \frac{\left|s_{i,e,t} - a_{e,t}^{G}\right|}{\left|s_{i,e,t}\right| + \left|a_{e,t}^{G}\right|} \tag{4}$$

The calculation of the throughput in the Blockchain in IoT system is given by the Equation (5), the delay calculation is in the form of the Equation (6), and the bit error rate calculation is in the form Equation (7).

$$Throughput = Max_{Window_{Size}}^{f_w(t+1)} \times Delay^{f_w(t+1)} \times RTT^{N_{data}} \tag{5}$$

$$Delay_{D(n)} = \left(\frac{1}{\left(\frac{4}{\tan^2 \frac{\theta}{2}} \right)^{\frac{2}{f_w(t+1)}} \sqrt{a\left(N_{data}\right)}} \right) \tag{6}$$

$$BER = 1 - \left(1 - B_e\right)^{N_{data}} = 1 - \mathrm{e}^{N_{data} \log\left(1 - B_e\right)} + f_w\left(t+1\right) \tag{7}$$

In Equation (5), $Max_{Window_{Size}}$ is the maximum window size in the evaluation of sending and receiving data, which can be calculated after calculating the delay in Equation (6). RTT is the time it takes to send data in sweep along the way in the Blockchain in IoT system. It is worth noting that the calculation of delay has been done end-to-end.

5. Blockchain Implementation for the Proposed Framework

In this section we try to describe applying blockchain to an IoT system implementation as following:

***A. Hyperledger Fabric*:** We selected Hyperledger Fabric as our blockchain solution to deploy across our network with the use of Docker Swarm. Fabric is a private, permissions blockchain whose functions are closely coupled with Docker containers (Yang & Enyeart, 2015). The fabric also decouples transaction validation from ledger block ordering into separate containers. Peer nodes carry out validation and ledger maintenance. Orderer nodes handle consensus and block broadcasting to their peers. The Fabric equivalent to smart contracts is called "chain code". Figure 4 depicts the general framework that has been planned for use in our project. Our framework leverages the containerization of Fabric so that the network can operate between the edge devices and the cloud in a single, private system.

***B. Sensor Device Layer*:** Our setup uses sensors hardwired to Raspberry Pi devices. Multiple sensors can be connected to one Pi for the sake of clarity, Figure 4 shows one sensor per edge device.

***C. Edge Device Layer*:** We use Raspberry Pi devices for our Edge Device Layer. Each Pi is installed with a Fabric Peer Container equipped with a chain

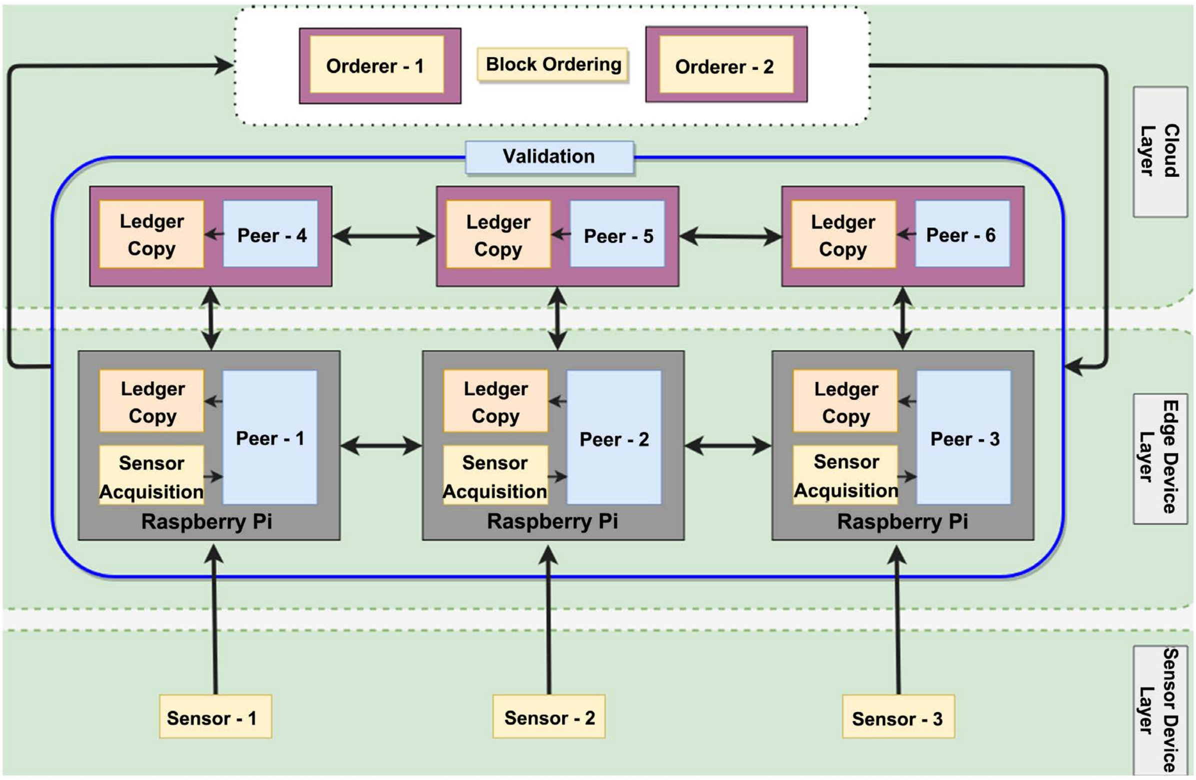

Figure 4. Data workflow of the proposed framework.

code that packages sensor data into proper transaction format for the ledger. Figure 4 shows several bidirectional arrows pointing between peers for cleaner representation; however, connections can be assumed between any two peers in the network.

***D. Cloud Layer*:** The Cloud Layer is where multiple peers can be deployed for fault-tolerance. The optimal number of backup peers will ultimately depend on the overall system environment. The Cloud Layer will also contain the Orderer nodes so that the Edge Device Layer does not have to handle the overhead of block-creation.

***E. Data Workflow*:** As represented in Figure 4, data acquired by the sensors transmit to the edge devices. The data is packaged by chain code hosted on the peer nodes. The peer that packages the respective data into a "transaction" will request validation from a configurable number of peers. The selected peers will be able to determine transaction validity by way of criteria that can be determined by the Blockchain designer, typically this will include transaction structure and peer signature authenticity. If the data transaction passes validation, the Peer that created the transaction will send it to the orderer nodes in the Cloud Layer. If the selected peers deem the data transaction invalid, it will not be sent to the orderer nodes. Once the orderer nodes receive enough transactions to create a block, or the Block Epoch time has expired, they will reach a consensus on the order of transactions they have received since the last block's creation. The block will be created and broadcast back to the peer nodes. Data acquisition frequency should be related to Epoch time in order to control overall latency from sensor detection to ledger access by an authorized client.

6. Conclusion

Combining blockchain with IoT provides a stable and robust decentralized way to manage the rapidly increasing number of networked devices. By reconfiguring some parameters of Blockchains, it can lead to enable these dynamic features to lead a system that is capable of enduring changing environments. Choosing what parameters of Blockchain in viewing mode and design in the process, helps to analyze management in a stand-alone system. By considering some security advantages and quality of services criteria such as processing power, preventing them from running CPU intensive tasks locally, privacy policy versus the value of the task to be processed to gain the best rates of throughput and delay, it will be possible to enhance the structure of blockchain in IoT systems.

The blockchain parameter configuration used in the proposed model in this study includes: Parameters Block, Epoch Time, Consensus Protocol, Throughput Queuing, Network Topology and Containerization. In this study, a new and applicable model is proposed which represents the opportunities for implementing blockchains into IoT framework in the sensor data acquisition. This method is based on probabilities that use the combination of the Entropy and TOPSIS model. We aim to propose a simple perspective of blockchain in IoT which is flexible with high throughput in sensor networks. This model contains some areas in security such as privacy policy and delay versus the value of the task to be processed.

The main contribution of the study is that we selected Hyperledger Fabric as our blockchain solution to deploy across our network with the use of Docker Swarm. Our framework leverages the containerization of Fabric so that the network can operate between the edge devices and the cloud in a single, private system. We selected Hyperledger Fabric which guarantees interoperability and confidentiality through the creation of channels, as well as the optimization of the timing of operations as our blockchain solution to deploy across our network with the use of Docker Swarm. The proposed framework leverages the containerization of Fabric so that the network can operate between the edge devices and the cloud in a single, private system. The architecture proposed in this paper lays the groundwork for further research in this area paving the way for new business models and novel, distributed applications.

For future studies, we propose that the new concept of "consensus games of IoT" that play a key role in the study of data trading under the framework of IoT associated with blockchain ecosystems would be used to establish a general framework for new business models and also novel distributed applications.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

Abraham, I., Malkhi, D., Nayak, K., & Ren, L. (2016). Solidus: An Incentive-Compatible Cryptocurrency Based on Permissionless Byzantine Consensus. arXiv:1612.02916 https://arxiv.org/abs/1612.02916v1

Andersen, M. P., Kolb, J., & Chen, K. (2017). *WAVE: A Decentralized Authorization System for IoT via Blockchain Smart Contracts*. Berkeley, CA: Electrical Engineering and Computer Sciences, University of California at Berkeley. https://www2.eecs.berkeley.edu/Pubs/TechRpts/2017/EECS-2017-234.pdf

Aniello, L., Baldoni, R. Gaetani, E., & Lombardi, F. (2017). A Prototype Evaluation of a Tamper-Resistant High Performance Blockchain-Based Transaction Log for a Distributed Database. *13th European Dependable Computing Conference*, Geneva, 4-8 September 2017, 151-154. https://ieeexplore.ieee.org/document/8123568 https://doi.org/10.1109/EDCC.2017.31

Apostolaki, M., Zohar, A., & Vanbever, L. (2017). Hijacking Bitcoin: Routing Attacks on Cryptocurrencies. *IEEE Symposium on Security and Privacy*, San Jose, CA, 22-26 May 2017, 375-392. https://www.semanticscholar.org/paper/Hijacking-Bitcoin%3A-Routing-Attacks-on-Apostolaki-Zohar/07db4d7b141081644b9cbb3e6d1f34c1bc80db24 https://doi.org/10.1109/SP.2017.29

Bahga, A., & Madisetti, V. (2016). Blockchain Platform for Industrial Internet of Things. *Journal of Software Engineering and Applications, 9,* 533-546. https://www.scirp.org/journal/paperinformation.aspx?paperid=71596 https://doi.org/10.4236/jsea.2016.910036

Ben-Sasson, E., Chiesa, A., Garman, C., Green, M. et al. (2014). Decentralized Anonymous Payments from Bitcoin. *IEEE Symposium on Security and Privacy*, Berkeley, CA, 18-21 May 2014, 459-474.
http://zerocash-project.org/media/pdf/zerocash-oakland2014.pdf
https://doi.org/10.1109/SP.2014.36

BitInfoCharts (2018). *Bitcoin (BTC) Statistics—Price, Blocks Count, Difficulty, Hash Rate, Value*. https://bitinfocharts.com/bitcoin/

Boireau, O. (2018). Securing the Blockchain against Hackers. *Network Security, 2018,* 8-11. https://www.sciencedirect.com/science/article/abs/pii/S1353485818300060
https://doi.org/10.1016/S1353-4858(18)30006-0

Buterin, V. (2015). *A Next-Generation Smart Contract and Decentralized Application Platform*.
https://www.semanticscholar.org/paper/A-Next-Generation-Smart-Contract-and-Decentralized-Buterin/0dbb8a54ca5066b82fa086bbf5db4c54b947719a

Cachin, C., & Vukolic, M. (2017). Blockchain Consensus Protocols in the Wild. arXiv:1707.01873 https://arxiv.org/abs/1707.01873

Castro, M., & Liskov, B. (2002). Practical Byzantine Fault Tolerance and Proactive Recovery. *ACM Transactions on Computer Systems, 20,* 398-461.
http://pmg.csail.mit.edu/papers/bft-tocs.pdf
https://doi.org/10.1145/571637.571640

Christidis, K., & Devetsikiotis, M. (2016). Blockchains and Smart Contracts for the Internet of Things. *IEEE Access, 4,* 2292-2303.
https://ieeexplore.ieee.org/document/7467408
https://doi.org/10.1109/ACCESS.2016.2566339

Croman, K., Decker, C., Eyal, I., Gencer, A. et al. (2016). On Scaling Decentralized Blockchains. In J. Clark, S. Meiklejohn, P. Ryan, D. Wallach, M. Brenner, & K. Rohloff (Eds.), *Financial Cryptography and Data Security. FC 2016. Lecture Notes in Computer Science* (Vol. 9604, pp. 106-125). Berlin, Heidelberg: Springer Berlin Heidelberg.
https://fc16.ifca.ai/bitcoin/papers/CDE+16.pdf
https://doi.org/10.1007/978-3-662-53357-4_8

Crosby, M., Machiappan, P., & Pattanayak, S. (2016). Blockchain Technology: Beyond Bitcoin. *Applied Innovation Review, No. 2,* 7-19.
https://j2-capital.com/wp-content/uploads/2017/11/AIR-2016-Blockchain.pdf

Daza, V., Pietro, R. D., & Klimek, I. (2017). CONNECT: Contextual Name Discovery for Blockchain-Based Services in the IoT. *IEEE International Conference on Communications*, Paris, 21-25 May 2017, 1-6. https://ieeexplore.ieee.org/document/7996641
https://doi.org/10.1109/ICC.2017.7996641

Dorri, A., Kanhere, S. S., & Jurdak, R. (2016). Blockchain in the Internet of Things: Challenges and Solutions. arXiv:1608.05187 https://arxiv.org/abs/1608.05187

Dorri, A., Kanhere, S. S., Jurdak, R., & Gauravaram, P. (2017). Blockchain for IoT Security and Privacy: The Case Study of a Smart Home. *2017 IEEE International Conference on Pervasive Computing and Communications Workshops*, Kona, HI, 13-17 March 2017, 618-623.
https://www.semanticscholar.org/paper/Blockchain-for-IoT-security-and-privacy%3A-The-case-a-Dorri-Kanhere/28fe6a3fab2f2097a6f9aac5ae9799577badf883
https://doi.org/10.1109/PERCOMW.2017.7917634

Eyal, I., & Sirer, E. G. (2013). Majority Is Not Enough: Bitcoin Mining Is Vulnerable. *Communications of the ACM, 61.*

https://www.cs.cornell.edu/~ie53/publications/btcProcFC.pdf

Gartner (2015). *Gartner Says 6.4 Billion Connected "Things" Will Be in Use in 2016, up 30 Percent from 2015*. https://www.gartner.com/en/newsroom/press-releases/2015-11-10-gartner-says-6-billion-connected-things-will-be-in-use-in-2016-up-30-percent-from-2015

Gencer, A. E., van Renesse, R., & Sirer, E. G. (2016). Service-Oriented Sharding with Aspen. arXiv:1611.06816 https://arxiv.org/abs/1611.06816

Kokoris-Kogias, E., Jovanovic, P., Gasser, L., Gailly, N., Syta, E., & Ford, B. (2017). *OmniLedger: A Secure, Scale-Out, Decentralized Ledger via Sharding*. IACR Cryptology ePrint Archive, 406. https://eprint.iacr.org/2017/406.pdf

Kraft, D. (2016). Difficulty Control for Blockchain-Based Consensus Systems. *Peer-to-Peer Networking and Applications, 9,* 397-413. https://link.springer.com/article/10.1007/s12083-015-0347-x https://doi.org/10.1007/s12083-015-0347-x

Lewenberg, Y., Sompolinsky, Y., & Zohar, A. (2015). Inclusive Block Chain Protocols. In R. Böhme, & T. Okamoto (Eds.), *Financial Cryptography and Data Security. FC 2015. Lecture Notes in Computer Science* (Vol. 8975, pp. 528-547). Berlin, Heidelberg: Springer. http://fc15.ifca.ai/preproceedings/paper_101.pdf https://doi.org/10.1007/978-3-662-47854-7_33

Liang, X., Zhao, J., Shetty, S., & Li, D. (2017). Towards Data Assurance and Resilience in IoT Using Blockchain. *IEEE Military Communications Conference*, Baltimore, MD, 23-25 October 2017, 261-266. https://ieeexplore.ieee.org/document/8170858 https://doi.org/10.1109/MILCOM.2017.8170858

Luu, L., Narayanan, V., Zheng, C. et al. (2016). A Secure Sharding Protocol for Open Blockchains. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, Vienna, October 2016, 17-30. https://dl.acm.org/doi/10.1145/2976749.2978389 https://doi.org/10.1145/2976749.2978389

Mazières, D. (2016). *The Stellar Consensus Protocol: A Federated Model for Internet-Level Consensus*. Stellar Development Foundation. http://www.scs.stanford.edu/17au-cs244b/notes/scp.pdf

McConaghy, T., Marques, R., Miller, A., & Jonghe, D. D. (2016). *BigchainDB: A Scalable Blockchain Database*. https://mycourses.aalto.fi/pluginfile.php/378362/mod_resource/content/1/bigchaindb-whitepaper.pdf

Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. https://bitcoin.org/bitcoin.pdf

Popov, S. (2017). *The Tangle*. IOTA. https://cointhinktank.com/upload/IOTA-2017.pdf

Pureswaran, V., & Brody, P. (2014). *Device Democracy: Saving the Future of the Internet of Things*. IBM Institute for Business Value. https://www.ibm.com/downloads/cas/Y5ONA8EV

Sharma, P. K., Chen, M.-Y., & Park, J. H. (2017). A Software Defined Fog Node Based Distributed Blockchain Cloud Architecture for IoT. *IEEE Access, 6,* 115-124. https://www.semanticscholar.org/paper/A-Software-Defined-Fog-Node-Based-Distributed-Cloud-Sharma-Chen/4722497018e8bc1270976ddd5bdd32901338dc52 https://doi.org/10.1109/ACCESS.2017.2757955

Sicilia, M. Á., & Visvizi, A. (2019). Blockchain and OECD Data Repositories: Opportunities and Policymaking Implications. *Library Hi Tech, 37,* 30-42. https://doi.org/10.1108/LHT-12-2017-0276

Stanciu, A. (2017). Blockchain Based Distributed Control System for Edge Computing. *21st International Conference on Control Systems and Computer Science*, Bucharest, 29-31 May 2017, 667-671. https://ieeexplore.ieee.org/document/7968630 https://doi.org/10.1109/CSCS.2017.102

Szabo, N. (1996). *Smart Contracts: Building Blocks for Digital Markets*. https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_2.html

Troisi, O., Maione, G., Grimaldi, M., & Loia, F. (2020). Growth Hacking: Insights on Data-Driven Decision-Making from Three Firms. *Industrial Marketing Management, 90,* 538-557. https://doi.org/10.1016/j.indmarman.2019.08.005

Vukolić, M. (2015). The Quest for Scalable Blockchain Fabric: Proof-of-Work vs. BFT Replication. In J. Camenisch, & D. Kesdoğan (Eds.), *Open Problems in Network Security. iNetSec 2015. Lecture Notes in Computer Science* (Vol. 9591, pp. 112-125). Cham: Springer. https://link.springer.com/chapter/10.1007/978-3-319-39028-4_9 https://doi.org/10.1007/978-3-319-39028-4_9

Windley, P., & Reed, D. (2018). *Sovrin: A Protocol and Token for Self-Sovereign Identity and Decentralized Trust*. Sovrin Foundation. https://sovrin.org/library/sovrin-protocol-and-token-white-paper/

Yang, B., & Enyeart, D. (2020). Welcome Hyperledger Fabric 2.0: Enterprise DLT for Production. Hyperledger.org. https://www.hyperledger.org/blog/2020/01/30/welcome-hyperledger-fabric-2-0-enterprise-dlt-for-production

Yeow, K., Gani, A., Ahmad, R. W. et al. (2017). Decentralized Consensus for Edge-Centric Internet of Things: A Review, Taxonomy, and Research Issues. *IEEE Access Journal, 6,* 1513-1524. https://ieeexplore.ieee.org/abstract/document/8168250