CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

ORCA – Online Research @ Cardiff

This is an Open Access document downloaded from ORCA, Cardiff University's institutional repository:https://orca.cardiff.ac.uk/id/eprint/172969/

This is the author's version of a work that was submitted to / accepted for publication.

Citation for final published version:

Mathur, Akshaya, Barati, Masoud, Aujla, Gagangeet Singh and Rana, Omer 2024. Towards scalable and secure blockchain in Internet of Things: a preference-driven committee member auction consensus approach. Distributed Ledger Technologies: Research and Practice 10.1145/3700149

Publishers page: http://dx.doi.org/10.1145/3700149

Please note:
Changes made as a result of publishing processes such as copy-editing, formatting and page numbers may not be reflected in this version. For the definitive version of this publication, please refer to the published source. You are advised to consult the publisher's version if you wish to cite this paper.

This version is being made available in accordance with publisher policies. See http://orca.cf.ac.uk/policies.html for usage policies. Copyright and moral rights for publications made available in ORCA are retained by the copyright holders.

Towards Scalable and Secure Blockchain in Internet of Things: A Preference-Driven Committee Member Auction Consensus Approach

AKSHAYA MATHUR, School of Computing, Newcastle University, United Kingdom
MASOUD BARATI, School of Information Technology, Carleton University, Canada
GAGANGEET SINGH AUJLA, Department of Computer Science, Durham University, United Kingdom
OMER RANA, School of Computer Science and Informatics, Cardiff University, United Kingdom

Blockchain technology is acclaimed for eliminating the need for a central authority while ensuring stability, security, and immutability. However, its integration into Internet of Things (IoT) environments is hampered by the limited computational resources of IoT devices. Consensus algorithms, vital for blockchain safety and efficiency, often require substantial computational power and face challenges related to security, scalability, and resource demands. To address these critical issues, we propose a novel model that significantly enhances the security and performance of blockchain in IoT environments. Our model introduces three key innovations: (1) a bidirectional-linked blockchain system that strengthens security against long-range attacks by exploiting dual reference points for block validation; (2) the integration of user preferences into the Committee Member Auction (CMA) consensus algorithm, optimizing miner selection to balance resource efficiency with security; and (3) a comprehensive performance and frequency analysis that demonstrates the system's resilience against double-spend, long-range, and eclipse attacks. The proposed model not only reduces block validation delays but also enhances overall system performance, as evidenced by simulations comparing its effectiveness with existing CMA algorithms. These advancements have the potential to significantly impact the deployment of blockchain in resource-constrained IoT environments, offering a more secure and efficient solution.

CCS Concepts: • **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

Additional Key Words and Phrases: blockchain, preference-based models, auction consensus algorithm, mining, data security

ACM Reference Format:
Akshaya Mathur, Masoud Barati, Gagangeet Singh Aujla, and Omer Rana. 2018. Towards Scalable and Secure Blockchain in Internet of Things: A Preference-Driven Committee Member Auction Consensus Approach. 1, 1 (November 2018), 21 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Blockchain is a shared, immutable, distributed ledger of transactions duplicated and distributed across the entire network of computer systems. Even though blockchain allows secure and anonymous transactions [37], it faces several security and scalability issues. Many attacks have been launched on blockchain-based systems [38]. For instance, the Reorg

Authors' addresses: Akshaya Mathur, akshayamathur1997@gmail.com, School of Computing, Newcastle University, United Kingdom; Masoud Barati, masoud.barati@carleton.ca, School of Information Technology, Carleton University, Canada; Gagangeet Singh Aujla, gagangeet.s.aujla@durham.ac.uk, Department of Computer Science, Durham University, United Kingdom; Omer Rana, RanaOF@cardiff.ac.uk, School of Computer Science and Informatics, Cardiff University, United Kingdom.

Tracker observed eighteen double-spend attacks on four cryptocurrencies [27]. Blockchain networks using the Proof of Work (PoW) consensus algorithm are vulnerable to double-spend attacks [16]. In contrast, those using the Proof of Stake (PoS) consensus algorithm are vulnerable to long-range, and eclipse attacks [1], [40]. Another major challenge blockchain faces is scalability and throughput, especially when deployed in the Internet of Things (IoT) environment [14, 24, 32]. When there is an increase in the transaction history, it could topple the overall system. As there is an increase in the number of transactions, the block's validation time increases due to the consensus process. Several existing proposals [1, 29] have devised solutions to eliminate blockchain vulnerabilities. A double-spending prevention mechanism for bitcoin zero-confirmation transactions is proposed in [29]. However, it works only with Unconfirmed Transaction Outputs (UTXO) or Bitcoin models. Checkpoints are adopted to define the correct chain periodically in [1] to defend against long-range attacks. However, when generating the checkpoints, it is at risk for Distributed Denial of Service (DDoS) attacks. Regarding eclipse attacks, an eclipse-attack detection model for Ethereum is proposed in [36]. Nevertheless, the model can detect the Ethereum network attack traffic based on two criteria: information entropy (information in the assault packets) and statistical statistics. Many different solutions [25, 31, 33] are proposed for improving the scalability of the blockchain network. For example, layer two solutions (such as decoupled blockchain [25], derived blockchain[33] or off-chain solutions [31]) use an external, parallel network to facilitate transactions away from the main blockchain or they decouple the block. However, these mechanisms add much complexity to the existing system and additional synchronization effort or sacrifice some degree of decentralization.

Alternatively, a layer one solution (like a lightweight and scalable committee member auction consensus algorithm) addresses the scalability and security vulnerabilities in the blockchain. However, this algorithm does not consider the miner's validation time, failure rate, and malicious activities. To solve the challenges mentioned above, this paper proposes a novel approach that improves the blockchain's performance and security vulnerabilities by incorporating a preference model in the committee member auction consensus algorithm. The preference model can allow users to select their priorities for the mining procedures. Users set the preference to elect committee members during the consensus process based on parameters (like stake, processing power, cost, and disk space) for every transaction in the blockchain. For example, if the user wants higher security, they could prefer to stake for the transaction as it shows the miner's intention. Similarly, users could prefer processing power if they desire faster transaction validation. Adding user preference to the consensus process will also help increase the system's entropy, making it difficult to conduct double-spend and long-range attacks.

The proposed model utilizes a bidirectional-linked blockchain for its robust security features. It guards against long-range attacks by detecting block alterations through dual reference points, linking each block to both its predecessor and successor. This structure not only enhances security by making unauthorized changes more difficult but also improves communication efficiency and reduces transaction latency through bidirectional data flow. Moreover, the increased efficiency strengthens the blockchain's overall security posture, offering enhanced protection against threats such as double-spending and Sybil attacks. Based on these contributions, the paper is outlined as follows:

(1) a bidirectional-linked blockchain system model is designed to support a custom consensus algorithm that can guard against long-range attacks by detecting block alterations using dual reference points;
(2) a user preference model is integrated into the Committee Member Auction (CMA) consensus algorithm to select matchable miners for validating the transactions;
(3) a performance and frequency analysis was performed to evaluate the resilience of the proposed system to double-spend, long-range, and eclipse attacks.

The rest of the paper is structured as follows. Section 2 reviews related work. Section 3 presents the background and context about bidirectionally-linked blockchain, committee member auction algorithms and cryptography tool. Section 4 proposes our preference-based committee member auction consensus algorithm. Section 5 evaluates the performance and frequency of our proposed algorithm. Section 6 analyses the security features of our approach. Section 7 concludes the paper and identifies directions for future work.

2 RELATED WORK

Several existing proposals adopt blockchain to address different challenges concerning edge computing or the IoT environment. For example, in [30], the authors reviewed blockchain-based IoHT systems for privacy protection, identified privacy challenges through GDPR, and presented privacy-preserving techniques. They also highlighted key research challenges to encourage future solutions in this field. In [19], the authors proposed PVFChain with an optimal smart contract design to conduct computation offloading decentralized, improving network security and efficiency. To implement the PVFChain, they proposed the crucial network entities and considered smart contract operations across them. The requester (leader) optimizes the reward policy for performers (followers) to improve user satisfaction while also considering utility maximization for them. However, this work does not address blockchain vulnerabilities and focuses on tamper-proof computational offloading. The authors in [23] presented a user-centric blockchain to share edge knowledge in IoT. They used the Proof of Popularity (PoP) consensus algorithm to preserve the security of edge knowledge sharing among IoT services. However, this work does not address the vulnerabilities of blockchain. The key focus of this work includes the design of a traceable, privacy-preserving, and tamper-resistant ledger for sharing edge knowledge. The solutions in [19], [23] do not consider blockchain technology's inherent challenges but provided solutions for allied applications using blockchain. These solutions introduced interactive smart contract operations across network entities, and the optimal contract design were formulated and solved using a Stackelberg game to minimize user payments. Security analysis and numerical results demonstrated their scheme's high security.

There are several blockchain-based solutions for mitigating double spend attacks [29], [12]. These solutions detected the attack using a listening period and observers. They discouraged double spending attempts in Bitcoin or UTXO models by creating a specialized output mechanism that forces the disclosure of the private key in the event of a double spending attempt. This approach enhanced security by deterring malicious actors from exploiting zero-confirmation transactions. However, in a peer-to-peer network, the message delivery between nodes could be more timely, and the order of messages was not guaranteed, which makes their observers unreliable. In [1], the authors proposed immutable checkpoints. It adopts a multi-variable (block, active user, stake parameters) strategy to decide the next checkpoint. However, the strategy is vulnerable to DDoS attacks, especially when creating checkpoints, and periodically relies on a centralized server to define the correct chain. In [36], an eclipse-attack detection model for Ethereum has been proposed. The model was based on a random forest classification algorithm, in which features of attack connection flow were defined. Nevertheless, it is only responsible for detecting attack traffic based on two features, namely information entropy (information in the attack packets) and statistical features, for the Ethereum network. In [39], the authors considered time-variant trust values of nodes based on dynamically adjusted trust values and node performance; nodes were classified into accounting, validating, and propagating nodes. The accounting node presented the block; the validating node validated the proposed block, while propagating nodes were only responsible for propagating transactions. If the trust value of a node decreases, it will be demoted to either validating or propagating node. However, this approach was vulnerable to Sybil attacks and DoS attacks. In [20], the authors presented a new approach to improving IoT security and privacy by employing a permissioned blockchain with optimized data storage and a lightweight authentication

mechanism. It integrated homomorphic encryption to secure data before uploading to the cloud, demonstrating through simulations that this method enhances security, privacy, and performance in decentralized IoT systems. A distributed strategy was proposed for computational resource trading, optimizing task delay, energy cost, trading prices, and user reputation through a multi-preference matching mechanism [34]. It also leveraged Blockchain-as-a-Service to provide decentralized identity infrastructure, and simulations demonstrate that the proposed strategy achieves higher task throughput compared to traditional double auction mechanisms. A blockchain-based privacy-preserving framework was proposed for IoT networks, utilizing service-oriented layers, low-computation cryptography, and a simplified consensus protocol [21]. The security analysis and simulations demonstrated the framework's effectiveness, making it suitable for real-world IoT applications.

However, these approaches focused on mitigating the attacks on blockchain networks using proof of work and proof of stake consensus algorithms. A more flexible consensus algorithm, which has a higher performance and is resistant to attacks, is required. We propose a new preference-based and blockchain-supported model to address the challenges mentioned earlier. From the approaches that use a consensus mechanism to verify and flag data breaches, a blockchain-based architecture has been proposed in which a set of voters took part in the violation detection [5–7]. Furthermore, a preference-based method has been supported for verifying data protection regulations [4]. The method enabled users to identify a priority for checking the regulations by running a smart contract. The authors in [17] discussed a potential attack on Bitcoin involving exploiting a victim for various attacks on its mining and consensus system. The study conducted a detailed analysis of Bitcoin's peer-to-peer network by employing probabilistic analysis, running Monte Carlo simulations, and performing live experiments. This comprehensive approach was used to accurately quantify the computational and network resources required to carry out potential attacks on the network. However, these approaches only focused on data privacy and did not provide a scalable, attack-proof solution for improving data security. Table 1 compares the above-discussed various existing proposals regarding their advantages and disadvantages.

In addition to these, the approach presented in [18] addresses issues in the traditional DPoS algorithm by proposing a reputation-based DPoS that selects high-quality nodes for consensus, reducing security risks and improving efficiency through reputation and token-based incentives. In [15], a dynamic PBFT was built on the PBFT protocol, offering the same security and liveness while introducing flexibility for nodes to join or leave the network without downtime. It enhanced system robustness by removing malicious nodes and utilizes a Participation Degree metric to ensure active node involvement, improving security. Complementing these approaches, our preference-based CMA algorithm leverages user-defined criteria to select miners, optimizing the consensus process by balancing factors like stake, processing power, and cost, improving both security and performance.

In contrast to existing approaches, my proposed model introduces a bidirectional-linked blockchain system that enhances security by detecting block alterations using dual reference points, thus protecting against long-range attacks. Additionally, it integrates a user preference model into the CMA consensus algorithm, allowing for optimal miner selection based on specific criteria.

3 BACKGROUND

This section discusses the background and prerequisites related to the cryptography tools, existing committee member auction consensus algorithm and formal definitions of preference models. These prerequisites are used in the subsequent section where we discuss the proposed approach.

Table 1. A comparison between existing similar approaches

Research	Method	Advantages	Drawbacks
[1]	PoS	Multi-variable checkpoints for long-range attack mitigation	Vulnerable to DDoS attacks & centralized server dependency
[29]	Bitcoin, UTXO	Double spend prevention & UTXO support	Rely on timely message delivery
[36]	Ethereum	Smart eclipse attack detection using entropy and statistics	Limited to detecting specific Ethereum attack criteria
[19]	parked vehicle assisted fog Chain (PVFChain)	Securing vehicular fog computing	Limited to fog computing
[23]	PoP	Edge knowledge sharing security	No focus on vulnerabilities
[12]	PoW	PoW security and performance analysis	Focused only on PoW blockchains
[39]	Consortium Blockchain	Time-variant trust values & node performance-based consensus	Vulnerable to Sybil and DoS attacks
[20]	PoS	Improved security and privacy through decentralized blockchain and homomorphic encryption	Added complexity in system implementation and management
[17]	Bitcoin	Counter for eclipse attacks in Bitcoin	Limited to Bitcoin
[34]	PoS	Improved task throughput with multi-preference matching	Added overhead with Blockchain-as-a-Service

3.1 Bidirectional-Linked Blockchain

The block structure in the bidirectional-linked blockchain differs slightly from the existing blockchains. A block has two pointers: the forward and reverse pointer, along with transactions and randomness. The forward pointer stores the previous block's hash value, while the reverse pointer stores the hash value of the next block. It is represented in Fig. 1 as $HashPrev$ and $HashNext$, respectively. The consensus reached by the distributed participants is represented by the randomness, which replaces the nonce. The bidirectional-linked blockchain will allow only appending operations, similar to other blockchain models.

The procedure for appending a new block entails a three-step process. Initially, a forward pointer is established by utilizing the Chameleon-hash function. This function computes the hash value of $Block_n$, which is subsequently stored within the $HashPrev$ field of $Block_{n+1}$ (Step 1 in Fig. 1). Following the establishment of the forward pointer, a reverse pointer is created using a conventional hashing method. Specifically, the hash value of $block_{n+1}$ is generated and stored in the $HashNext$ field of $Block_n$ (Step 2 in Fig. 1).

However, it is imperative to note that after the execution of the consensus algorithm [26], the value stored in the $HashPrev$ field of $block_{n+1}$ may become inaccurate. The reverse pointer must be adjusted to rectify this inconsistency. This adjustment is facilitated by leveraging the trapdoor keys exclusively possessed by the committee members

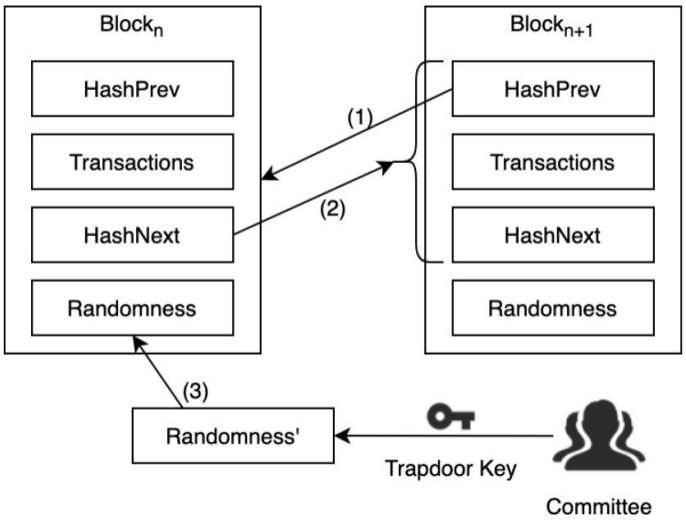

Fig. 1. A bidirectional-linked blockchain model [35]

participating in the Consensus Mechanism Algorithm (CMA). These keys enable the calculation of a randomized value, which, when applied, restores the integrity of the entire content of $Block_n$, rendering it unchanged (Step 3 in Fig. 1).

3.2 CMA Algorithm

The CMA consensus algorithm is a lightweight, scalable consensus algorithm that is attack resistance. The algorithm is scalable due to its dynamic committee size adjustment, which adapts to network conditions and transaction volumes. It employs a preference-based mechanism for selecting a subset of committee members, reducing communication overhead and computational complexity. This efficient member selection process ensures faster consensus times and lower network load. Moreover, its modular architecture allows for easy integration and adaptation to various blockchain systems, making it capable of handling increasing transaction demands and supporting network growth effectively. The CMA consensus algorithm uses Verifiable Random Functions (VRF) for electing committee members across distributed blockchain nodes. The VRFs are public-key pseudorandom functions that provide verifiable proof that their outputs were calculated correctly [9]. Algorand introduced VRFs to privately check if the miner is selected to participate in the consensus phase [13]. Algorand is a highly scalable blockchain framework that uses Byzantine Agreement (BA) protocol to reach consensus. However, the miners in Algorand are weighted based on the balance of tokens in wallets, which means a miner with more tokens is more vulnerable to DDoS attacks and causes the performance of the blockchain to be downgraded. The CMA consensus algorithm starts by electing the committee members [2], [3]. The miners acquire their $vhash$ and π by using the seed (a random value generated for each term) and their respective private keys. If $vhash$ falls into a specific range γ, the miner is treated as a committee member. The elected committee members propose new blocks based on the transactions they receive through the gossip protocol. To avoid conflict over generating blocks, there are also priorities among committee members. The miner with the lowest $vhash$ has the highest block generating priority. When a miner receives a block from a higher priority miner, it will accept the block. Otherwise, the block is broadcast using gossip protocol.

3.3 Preference Models

The preference model is a formal method of ranking the objects (in our case, miners) based on the user requests and preferences. A preference term can be either an atomic or composite [8, 11]. Atomic preference has a single object which can either be a qualitative or quantitative preference [22], while composite preference can have multiple qualitative or quantitative preference objects. A preference can be expressed as "x is preferred over y", where x and y are instances of miners. A preference model is formally defined as follows.

Definition 1: Let C be a non-empty set of parameters of miners and $dom(C)$ the set of all possible instances of those parameters. We define preference as $P = (C, <^p)$ where $<^p \subseteq dom(C) \times dom(C)$ is s strict partial order (irreflexive, transitive and asymmetric), and if x,y $\in dom(C)$, then x $<^p$ y is interpreted as "y is preferred rather than x".

Three types of preferences, i.e., numerical, prioritized and balanced can be used to define a formal preference model. The definitions are provided below.

3.3.1 Numerical Preference. The numerical preference is a combination of a number of score preferences. A score preference is defined as a scoring value that takes a property value as its argument and returns a real value. The higher the value returned by the function, the more preferred the property value is. We define score preference as follows.

Definition 2: Let f: $dom(C) \longrightarrow \mathfrak{R}$ be a scoring function and $<$ the usual less-than order in $\mathfrak{R}$. $P_f = (C, <^{P_f})$ is a score preference if for $x, y \in dom(C)$:

$$x <^{P_f} y \iff f(x) < f(y) \tag{1}$$

Numerical Preference takes the values returned by each score preference as its argument and returns another real number that gives information about the global preferences after considering all the properties referred by concrete score preferences. We define numerical preference as follows.

Definition 3: Let f, g and h be scoring functions that define score preferences $P_f = (C, <^{P_f}), P_g = (C, <^{P_g}), P_h = (C, <^{P_h})$, respectively and F: $\mathfrak{R} \times \mathfrak{R} \longrightarrow \mathfrak{R}$ be a combination function. For $x = (x_1, x_2), y = (y_1, y_2), z = (z_1, z_2) \in dom(C_1) \times dom(C_2) \times dom(C_3)$, $P = (C_1 \cup C_2 \cup C_3 <^{rank_F(P_f, P_g, P_h)})$ is a numerical preference if:

$$\begin{aligned} x &<^{rank_F(P_f,P_g,P_h)} y \\ &<^{rank_F(P_f,P_g,P_h)} z \\ &\iff F(f(x_1), g(x_2)) \\ &< F(f(y_1), g(y_2)) \\ &< F(f(z_1), g(z_2)) \end{aligned} \tag{2}$$

3.3.2 Prioritized Preference. A prioritized preference P is composed of two preference terms P_1 and P_2, where P_1 is considered more important than P_2. Thus, P_2 is evaluated only if P_1 does not return enough information to rank the parameters or in case of conflict. The prioritized preference is defined as:

Definition 4: Let $P_1 = (C_1, <^{P_1})$ and $P_2 = (C_2, <^{P_2})$ be two different preference defined after C_1 and C_2 properties and $x = (x_1, x_2), y = (y_1, y_2) \in dom(C_1) \times dom(C_2)$ be two value tuples for each property. $P = (C_1 \cup C_2, <^{P_1 \& P_2})$ is a prioritized preference if:

$$x <^{P_1 \& P_2} y \iff x_1 <^{P_1} y_1 \vee (x_1 = y_1 \wedge x_2 <^{P_2} y_2) \tag{3}$$

3.3.3 Balanced Preference (Pareto-optimality Principle). A balanced preference P is a combination of two preference terms P_1 and P_2. It uses the Pareto-optimality principle, a situation where no preference criterion can be made better without making at least one preference criterion worse off [10, 11]. Therefore, P_1 and P_2 are considered equally important.

Intuitively, this preference balances the fulfilment of each preference component so that the composite preference is the average degree of preference, taking both components into account.

Definition 5: Let $P_1 = (C_1, <^{P_1})$ and $P_2 = (C_2, <^{P_2})$ be two different preference defined after C_1 and C_2 properties and $x = (x_1, x_2), y = (y_1, y_2) \in dom(C_1) \times dom(C_2)$ be two value tuples for each property. $P = (C_1 \cup C_2, <^{P_1 \otimes P_2})$ is a balanced preference if:

$$x <^{P_1 \otimes P_2} y \iff (x_1 <^{P_1} y_1 \wedge (x_2 <^{P_2} y_2)) \vee (x_2 <^{P_2} y_2 \wedge (x_1 <^{P_1} y_1))) \tag{4}$$

4 PROPOSED APPROACH

Based on the above discussed prerequisites, our proposed approach integrates users preferences into the committee member auction consensus algorithm to select matchable miners for validating the transactions. The model makes use of the bidirectional-linked blockchain to resist some common attacks.

The integration of user preferences into the CMA consensus algorithm enhances the selection process by allowing for a more tailored and dynamic approach to miner selection. Traditional methods often rely on fixed criteria or a single factor, such as stake or computational power, which can lead to inefficiencies and a lack of adaptability. In contrast, the proposed approach, which includes numerical, prioritized, and balanced preference models, allows users to weigh various factors such as stake, power, disk space, and cost according to their specific needs. This results in a more accurate matching of committee members to the tasks at hand, leading to improved resource allocation and transaction validation efficiency. For instance, by enabling the scoring function to adjust based on user-defined parameters, the system can dynamically select miners who not only meet the technical requirements but also align with the network's strategic goals, such as minimizing costs or maximizing security. This adaptability is particularly beneficial in environments with varying priorities, providing a level of customization and efficiency that traditional models do not offer. The different working segments of the propose approach are described in the subsequent sections.

4.1 Preference Models for Bidirectional-linked Blockchain

Three types of preference models (concerning numerical, prioritized and balanced preferences) are integrated into the CMA consensus algorithm.

4.1.1 Numerical Preference Model. This model exploits scoring functions which takes miner's parameters (x) and a scoring values (v) as an input and returns scores for miner's parameters (e.g., for 'stake', 'power', 'disk', 'cost'). These scores are aggregated to get the total score (Σ). The scoring function ($f(s)$) for the numerical preference model is defined as below.

$$f(s) = x \times v. \tag{5}$$

Algorithm 1 quantifies a miner's preference with a specific criteria to calculate Σ. It takes two inputs, (a) an array (Π) containing miner-specific parameters paired with their respective scoring values and (b) an array (Γ) housing the miner's parameters alongside their corresponding numerical values.

The proposed algorithm starts by initializing Σ as 0 and iterated for each element in the array Π, hereafter referred to as Π_i. Each element in Π_i is examined to determine their corresponding parameter (e.g., "stake," "power," "disk," or "cost") (line 1-2). Depending on the parameter, following steps are performed in the algorithm.

- If Π_i refers to "stake," the algorithm augments Σ by the product of the miner's stake value ($\Gamma[\text{"}stake\text{"}]$) and the scoring value of "stake" ($\Pi_i[\text{"}stake\text{"}]$) (line 3-5).

Algorithm 1: Numerical Preference Algorithm

Input: (Π), Array miner's parameter with scoring values.
(Γ), Miner's parameter with values
Output: Total score calculated by the miner (Σ).
Function : getNumericPreferenceScore
1 Initialize variable: $\Sigma \leftarrow 0$ {Total Score}
2 **for** $\forall\ \Pi_i\ \in\ \Pi$ **do**
3 **if** Π_i *is* "$stake$" **then**
4 $\Sigma \leftarrow \Sigma + (\Gamma[\text{“}stake\text{”}] \times \Pi_i[\text{“}stake\text{”}])$;
5 **end**
6 **if** Π_i *is* "$power$" **then**
7 $\Sigma \leftarrow \Sigma + (\Gamma[\text{“}power\text{”}] \times \Pi_i[\text{“}power\text{”}])$;
8 **end**
9 **if** Π_i *is* "$disk$" **then**
10 $\Sigma \leftarrow \Sigma + (\Gamma[\text{“}disk\text{”}] \times \Pi_i[\text{“}disk\text{”}])$;
11 **end**
12 **if** Π_i *is* "$cost$" **then**
13 $\Sigma \leftarrow \Sigma + (\Gamma[\text{“}cost\text{”}] \times \Pi_i[\text{“}cost\text{”}])$;
14 **end**
15 **end**
16 **return** (Σ)

- If Π_i relates to "power," the algorithm contributes to Σ by the product of the miner's power value ($\Gamma[\text{“}power\text{”}]$) and the scoring value of "power" ($\Pi_i[\text{“}power\text{”}]$) (line 6-8).
- If Π_i corresponds to "disk," the algorithm augments Σ by the product of the miner's disk value ($\Gamma[\text{“}disk\text{”}]$) and the scoring value of "disk" ($\Pi_i[\text{“}disk\text{”}]$) (line 9-11).
- If Π_i represents "cost," the algorithm adds to Σ the product of the miner's cost value ($\Gamma[\text{“}cost\text{”}]$) and the scoring value of "cost" ($\Pi_i[\text{“}cost\text{”}]$) (line 12-14).

Upon completion of the iteration encompassing all elements in Π, the algorithm concludes its execution by returning Σ. The values are weighted according to predefined scoring values, culminating in the derivation of Σ that encapsulates the miner's overall suitability within the defined criteria.

4.1.2 Priority Preference Model. This model compares the most preferred miner's parameter to calculate the score. If the score is equal, then other miner's parameters are considered. Algorithm 2 is designed to calculate Σ based on specific criteria. It requires two inputs, (a) an array (Π) which contains miner-specific parameters, and (b) an array (Γ) containing the miner's parameters alongside their respective values.

The proposed algorithm starts by initializing Σ as 0 and depending on the parameters provided in the array Π (e.g., "stake," "power," "diskSpace," or "cost"), the algorithm checks each parameter's presence. If a parameter is present in Π, it increments Σ by the corresponding value from Γ. After considering all potential parameters, the algorithm concludes its execution by returning the computed Σ. Each parameter is treated equally, and Σ reflects the miner's overall alignment with the defined criteria.

4.1.3 Balanced Preference Model. This model gives equal priority to all preferred miner's parameters. Algorithm 3 serves the purpose of calculating a miner's total score (Σ) based on specific criteria. The proposed algorithm starts by initializing Σ as 0 and iterated for each element in the array Π, hereafter referred to as Π_i (line 1-2). For each Π_i element, the algorithm checks the presence of specific parameters, such as "stake," "power," "diskSpace," or "cost," within the array Π. If a parameter is present in Π, the algorithm increments the total score Σ by the corresponding value from the array Γ (line 3-14). After examining all elements in Π, the algorithm concludes its execution by returning the

Algorithm 2: Prioritized Preference Algorithm

Input: (Π), Array miner's parameter with scoring values.
(Γ), Miner's parameter with values
Output: Total score calculated by the miner (Σ).
Function : getPriorityPreferenceScore

```
1  Initialize variable: Σ ← 0
2  for ∀ Π_i ∈ Π do
3      if Π is "stake" then
4          Σ ← Σ + Γ["stake"];
5      end
6      if Π is "power" then
7          Σ ← Σ + Γ["power"];
8      end
9      if Π is "diskSpace" then
10         Σ ← Σ + Γ["diskSpace"];
11     end
12     if Π is "cost" then
13         Σ ← Σ + Γ["cost"];
14     end
15 end
16 return (Σ)
```

Algorithm 3: Balanced Preference Algorithm

Input: (Π), Array miner's parameter with scoring values.
(Γ), Miner's parameter with values
Output: Total score calculated by the miner (Σ).
Function : getBalancedPreferenceScore

```
1  Initialize variable: Σ ← 0
2  for ∀ Π_i ∈ Π do
3      if Π_i is "stake" then
4          Σ ← Σ + Γ["stake"];
5      end
6      if Π_i is "power" then
7          Σ ← Σ + Γ["power"];
8      end
9      if Π_i is "diskSpace" then
10         Σ ← Σ + Γ["diskSpace"]
11     end
12     if Π_i is "cost" then
13         Σ ← Σ + Γ["cost"];
14     end
15 end
16 return (Σ)
```

calculated total score Σ. This score quantitatively reflects the miner's alignment with the provided parameters, with equal weight assigned to each parameter. The resulting total score Σ offers a numerical representation of the miner's overall alignment with the defined criteria (line 16).

4.2 Mining Process

The CMA consensus algorithm is based on periodic elections of miners and requires distributed participants (miners) to have a synchronized clock. Each election period is called a *term*. The proposed mining process comprises three steps: (a) election of committee members, (b) proposing a new block and reaching consensus, and (c) generating the new block. These steps are described below.

4.2.1 Election of Committee Members. During the election of the committee members, the threshold value n, along with the preferred terms and preference model, are acquired by the miners from the transaction. For each term, the miner can calculate based on the preference models and terms to acquire a value total score Σ. For example, suppose the transaction has a preference model as numeric preference and preference terms as processing power and stake with their respective scoring values. In that case, every miner can acquire Σ based on the Algorithm 1. If the total score exceeds the threshold ($\Sigma > n$), the miner is considered a committee member for the term and can validate the block. After the committee member's election, the trapdoor keys are divided into η parts (where η is the number of committee members elected) and distributed across the committee members to protect the tampering of the forward pointer.

4.2.2 Propose a New Block. After the committee members are elected for the term, a new block is proposed based on the transaction they received. The priority among committee members is based on Σ. A higher value for Σ leads to a higher priority for validating and generating blocks. When a miner receives a block from a higher-priority miner, it will automatically accept the block; otherwise, the miner broadcasts it.

4.2.3 Reaching Consensus. After the committee members reach the consensus, every member sends their part of the trapdoor key and the hash of the newly proposed block $Block_{n+1}$. The smart contract repairs the randomness of the block $Block_n$ when enough secrets are collected to reconstruct the trapdoor key. By repairing the randomness of $Block_n$, the forward pointer of the block remains unchanged, and the reverse pointer of $Block_{n+1}$ points at $Block_n$ with a new $HashNext$. Finally, $Block_{n+1}$ is appended to the chain with both the forward and the reverse pointer.

4.3 Consensus Process

Initially, a blockchain network collects and converts the transactions into a block. After generating a block, the entire network must agree on the transaction's validity, i.e., reach a consensus before confirming the transaction to the blockchain. Each transaction in the block will contain the user's preference and a collection of miners' parameters which is preferred. For numeric preference, scoring values for preferred miners parameters will be contained within the transaction. As an example, in Fig. 2, we have a blockchain network with eight miners A, B, C, D, E, F, G and H with synchronized clock. During each election period, all the miners check if they are elected as committee members by calculating their total scores (Σ) based on the preference information present in the transaction. If the total score calculated by the miner is greater than the threshold value n, the miner is considered a committee member for the term. It is supposed that after the first step, miners A, D, E and H (represented in black in Fig. 2) are elected as committee members whereas miners B, C, F and G are not elected for the term. Assuming miner A (represented in red in Fig. 2) has the highest priority over the other miners during the term, miner A proposes a new block. The trapdoor keys are divided into four parts as there were four committee members elected and distributed to the elected committee members. The miners D, E, and H verify the newly proposed block by miner A. When a committee member receives a block from a higher-priority miner, it will automatically accept the block; otherwise, it broadcasts its block. When a lower-priority miner accepts a block from a higher-priority miner, it validates the block sent. After the committee member verifies the transaction, the committee member broadcasts a message that includes the trapdoor key. The broadcast is a black line in Fig. 2. When many trapdoor keys are collected, the smart contract will automatically get invoked to repair the randomness of $Block_n$. Here, the miner A will broadcast the newly generated block to all the network miners, i.e., B, C, D, E, F, G, and H. If any of the miner D, E, or H does not agree with the newly proposed block, they do not send out their trapdoor key. If most of the committee members do not agree with the block, it is rejected.

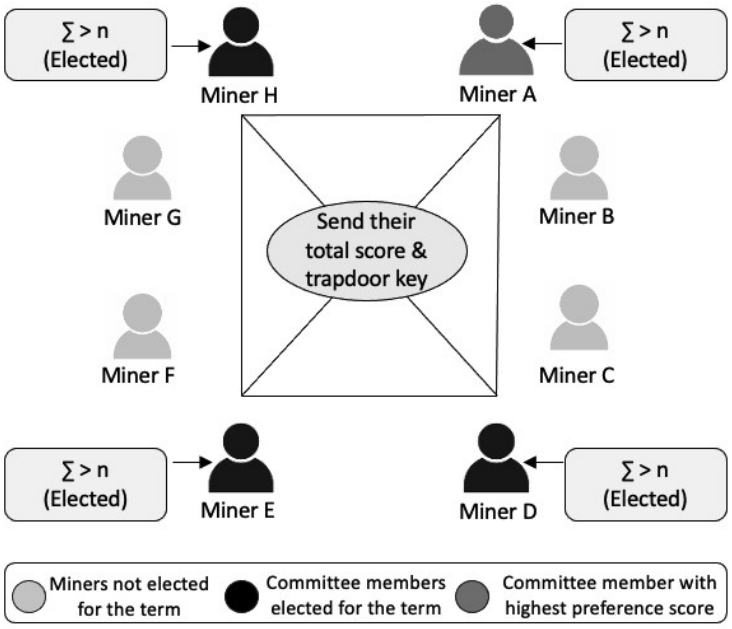

Fig. 2. Committee member auction consensus mechanism using preferences

4.4 Model Adaptation for IoT

Our proposed model is finely tuned for real-time IoT applications, where immediate transaction validation is crucial. Key enhancements include a preference-based miner selection mechanism, enabling users to prioritize processing power to achieve faster validation and reduced latency. The lightweight design of the CMA consensus algorithm is optimized to minimize computational overhead, thus maximizing throughput. Furthermore, the bidirectional-linked blockchain structure enhances data flow and communication efficiency, expediting consensus. By optimizing block size and frequency for the frequent, small transactions typical of IoT environments and integrating parallel processing capabilities, the model consistently maintains high throughput with low-latency validation. These optimizations collectively ensure that the proposed model meets the stringent speed and security demands of real-time IoT applications.

Our model's adaptability to varying IoT environments is facilitated through several flexible features. The adaptive miner selection dynamically chooses miners based on their computational and storage capabilities, allowing seamless integration of devices with differing resource levels. The scalable CMA consensus algorithm efficiently accommodates various network conditions, while the modular design of the bidirectional-linked blockchain allows for customization to suit specific use cases, such as adjusting block sizes and transaction frequencies based on available bandwidth. Additionally, configurable user preferences and adaptive data handling strategies ensure the model remains responsive and effective across a range of scenarios, from high-security requirements to rapid transaction processing. This adaptability addresses the diverse challenges posed by different device capabilities and network conditions.

The model is particularly well-suited for IoT devices with limited computational power and memory, thanks to targeted techniques. The adaptive miner selection mechanism prioritizes devices based on their available resources, ensuring that computationally intensive consensus processes are assigned to capable nodes while accommodating those

with more limited resources. The lightweight CMA consensus algorithm is specifically designed to reduce computational overhead, easing the processing burden on constrained devices. Additionally, the model employs optimized data handling strategies, such as minimizing block size and transaction complexity, to operate within the memory and processing constraints of these devices. Techniques like efficient data serialization and compression further mitigate resource constraints by reducing the data that needs to be processed and transmitted. Through these strategies, the model enables even resource-limited devices to participate effectively in the blockchain network without compromising on performance or security.

Our approach is designed with energy efficiency as a priority, particularly for IoT environments where devices often operate on limited battery power. The consensus algorithm is optimized to minimize computational complexity, thereby reducing the energy required for transaction validation and consensus operations. Additionally, the adaptive miner selection mechanism ensures that tasks assigned to low-power devices are well within their energy capacities, conserving battery life. The bidirectional-linked blockchain structure contributes to energy efficiency by optimizing data flow and reducing the need for redundant calculations, thus lowering overall energy consumption. Furthermore, the model incorporates energy-efficient data handling techniques, such as selective data transmission and aggregation, to minimize communication frequency and volume, further reducing energy usage. These combined strategies ensure the model is highly suited to energy-constrained IoT environments, allowing devices to function efficiently without compromising performance or security.

5 PERFORMANCE EVALUATION

This section discusses the performance evaluation of the proposed approach. The environment settings and results are described in the subsequent sections.

5.1 Environment Setting

The proposed and the existing algorithms are coded in Java 8, and Java Open JDK is used to build the simulation. The hardware configuration includes an Apple M1 chip eight-core processor and 8 GB of RAM. The default parameters of the simulation are given in Table 2.

Table 2. Simulation Parameter Settings

Parameter	Value
The number of participant members (η)	100
The incoming transaction speed (tx/s)	1
Total incoming transaction	1000
Number of transaction in a block	1

Every miner is configured with processing power, stake, disk space, and cost with a random value for the simulations. We considered two validation scenarios: (a) using processing power and stake as preferences, and (b) using random preferences, with the miner configuration remaining consistent across both scenarios. The defined configuration is depicted in Fig. 3.

The difficulty of mining the block is based on the computing power of the miner's hardware. In other words, the higher the processing power, the higher the computation power, the lower the processing time, and the lower the difficulty of validating the block. The network latency is considered to be negligible. The user is assumed to send the

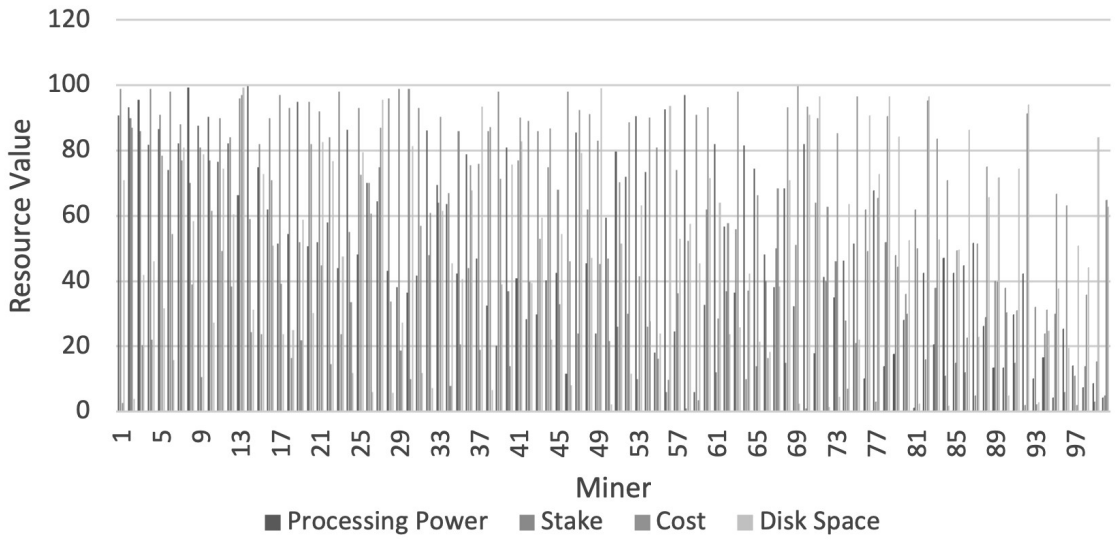

Fig. 3. Miner Configuration

preference model and the preferred miner parameters with the transaction. For each experiment, 1000 transactions are generated. Every transaction goes through the consensus algorithm to be validated. Each experiment is run for the methods; Numerical Preference-based CMA algorithm, Prioritized Preference-based CMA algorithm, Balanced Preference-based CMA algorithm, and compares the performance with the existing CMA algorithm.

5.2 Performance Comparison

The performance of each node in the blockchain network varies. In this experiment, we compare the time taken to validate a block in each scenario. The first experiment assumes that the collections of preferences in all 1000 transactions contain only processing power and stake. For numeric preference, a higher scoring value is given to processing power, and a lower value is given to stake. Specifically, a value of 90 is assigned to processing power and 10 to stake. Miners exceeding the threshold score are selected.

For priority preference, higher priority is given to processing power than to stake, leading to the selection of miners with greater processing power. Similarly, in the Balanced preference scenario, miners with high processing power and stake are chosen. In contrast, the existing CMA algorithm selects miners randomly based on the value of γ obtained from the VRF. The results of the experiment are shown in Table 3. The table shows that the validation block's time (in seconds) for the priority preference is shorter than the numeric and balanced preferences. This is because, in the case of priority preference, the miner with the highest available processing power is always selected to mine the block. The time taken to do so would be less as the difficulty to mine would be less. The simulation found that miner 78, having processing power configured as 99.90, was always elected to mine as it has the highest available processing power among all the other miners. For numeric preference, a miner with the highest overall score, i.e., 90% of processing power and 10% of the stake, is considered compared to only processing power in priority. The simulations found that miner 60, configured with the processing power of 99.1 and stake as 70, was always elected for mining the block. It has slightly less processing power but a higher stake than the miner 78, which has a stake of 58.

Table 3. Evaluation for Processing Power and Stake as Preference

Transaction No.	Average time taken to propose a block (seconds)			
	Priority	Numeric	Balanced	CMA
1-50	0.260	0.307	0.321	0.315
51-100	0.229	0.255	0.318	0.454
101-150	0.216	0.260	0.298	0.325
151-200	0.190	0.216	0.285	0.287
201-250	0.203	0.231	0.290	0.346
251-300	0.215	0.227	0.272	0.492
301-350	0.242	0.254	0.274	0.349
351-400	0.233	0.217	0.280	0.325
401-450	0.238	0.207	0.267	0.365
451-500	0.214	0.221	0.279	0.343
501-550	0.201	0.220	0.367	0.345
551-600	0.229	0.232	0.339	0.505
601-650	0.214	0.289	0.345	0.321
651-700	0.219	0.230	0.279	0.304
701-750	0.221	0.225	0.260	0.290
751-800	0.216	0.226	0.354	0.478
801-850	0.218	0.255	0.366	0.357
851-900	0.236	0.258	0.255	0.305
901-950	0.216	0.278	0.300	0.410
951-1000	0.246	0.252	0.304	0.302

In the case of balanced preference, equal preference is given to the processing power and stake, i.e., 50% value of processing power and 50% value of the stake are considered. Miner 8, configured with a processing power of 90.8 and a stake of 99, was always selected as it had an equally high stake and processing power. However, the processing power was significantly lower than that of the miner 78 and 60. The performance could not be guaranteed in the existing CMA algorithm since the miner is selected randomly. Some transactions take little time as the elected miner has high processing power to propose the block and vice versa. Every time the simulation ran, a different set of miners was elected because of the randomness of the algorithm. It is to be noted that in the preference-based CMA algorithm, all the committee members elected will have high processing power and stake, and hence, not only is the block proposal time reduced, but the block validation time is also reduced. This will be different for the existing CMA algorithm. For the second experiment, the preferences are set randomly for all the 1000 transactions, i.e., it is not guaranteed that processing power is given priority. For the numeric, priority and balanced preferences, the miners parameters are selected randomly.

In the case of numeric preference, the scoring values are also generated randomly. The performance results are shown in Table 4. It is evident that there is no guarantee of performance when processing power is not considered. The preference-based and existing CMA algorithms have a similar trend, and the performance depends on the elected miner's processing power. The conclusion of this experiment is that performance can only be guaranteed only if the elected committee member (miner) has a greater performance. Preference-based CMA gives user the flexibility to set their priority, hence reducing the block proposal and validation time.

Table 4. Performance Evaluation for Random Preference

Transaction No.	Average time taken to propose a block (seconds)			
	Priority	Numeric	Balanced	CMA
1-50	0.169	0.183	0.199	0.270
51-100	0.132	0.162	0.201	0.255
101-150	0.147	0.173	0.193	0.216
151-200	0.134	0.153	0.187	0.190
201-250	0.128	0.160	0.236	0.231
251-300	0.135	0.175	0.204	0.227
301-350	0.163	0.153	0.199	0.254
351-400	0.139	0.148	0.209	0.217
401-450	0.130	0.164	0.172	0.207
451-500	0.147	0.199	0.164	0.214
501-550	0.144	0.167	0.187	0.201
551-600	0.149	0.158	0.199	0.229
601-650	0.134	0.152	0.184	0.214
651-700	0.150	0.170	0.210	0.219
701-750	0.133	0.157	0.191	0.221
751-800	0.162	0.159	0.175	0.216
801-850	0.138	0.170	0.184	0.218
851-900	0.151	0.186	0.204	0.236
901-950	0.143	0.187	0.217	0.216
951-1000	0.151	0.175	0.237	0.246

6 SECURITY ANALYSIS

Blockchain faces significant scalability and security challenges like double spend, long-range, Sybil, and eclipse attacks. Double spending and long-range attacks are caused by the uncertainty of the added blocks and the subsequent blocks. However, the subsequent direction of any block, starting from the genesis block, may be known using the reverse pointer design, making the entire chain undisputed.

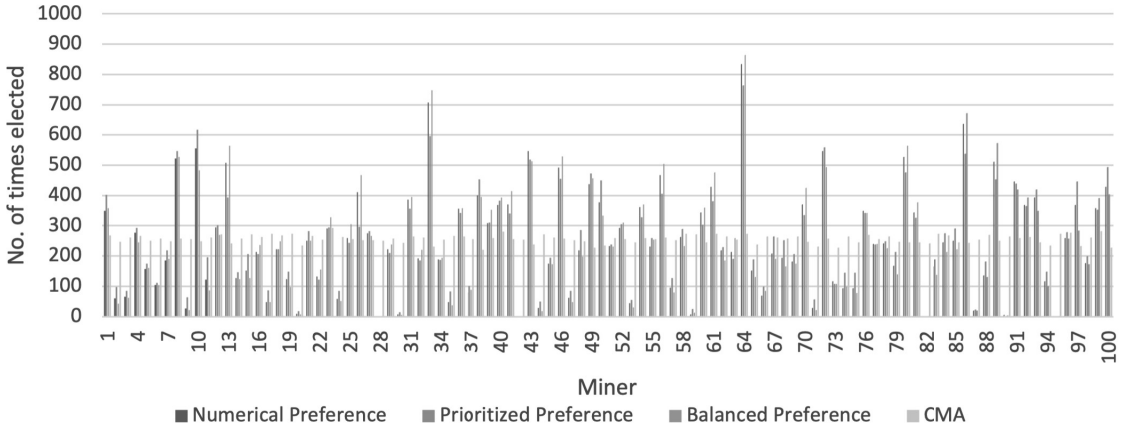

Fig. 4. Selection of a miner during the committee member election process

For the third experiment, a comparison is made on the number of times a miner is elected as a committee member. This is done to prove the entropy of the system. The more the randomness of the system, the more it is resistant to attacks like Sybil and double spend. The preference is set randomly for all the 1000 transactions, similar to the second experiment. The result is shown in Fig. 4. Some of the miners are getting elected more than others due to their configuration in a preference-based CMA algorithm. The randomness of the system is low as the only factor that brings randomness is the threshold value, which is randomly generated at the beginning of each term. Each color shows the number of times a miner is elected for the particular preference model. In the case of the existing CMA algorithm, the probability of getting selected is (τ/η), where τ is the number of committee members and η is the number of participant members in a blockchain network. Therefore, every miner has an equal chance of getting elected.

A similar trend can be observed when comparing the number of times a miner proposes the block. In the case of the preference-based model, we find that only a handful of miners are generating the new block. The experiment showed that in most cases, only 11% of the miners were selected to propose a new block. This is according to the Pareto principle, which states that for most outcomes, roughly 80% of the consequences come from 20% of the cause. In the case of the exiting CMA algorithm, every miner has an equal chance of proposing a new block as it depends on the value of γ. Since the same committee members are getting elected most of the time, the model is vulnerable to security and scalability challenges as compared to the existing CMA algorithm. If the same miner has to solve several blocks, the processing time would be high, which could lead to higher delays.

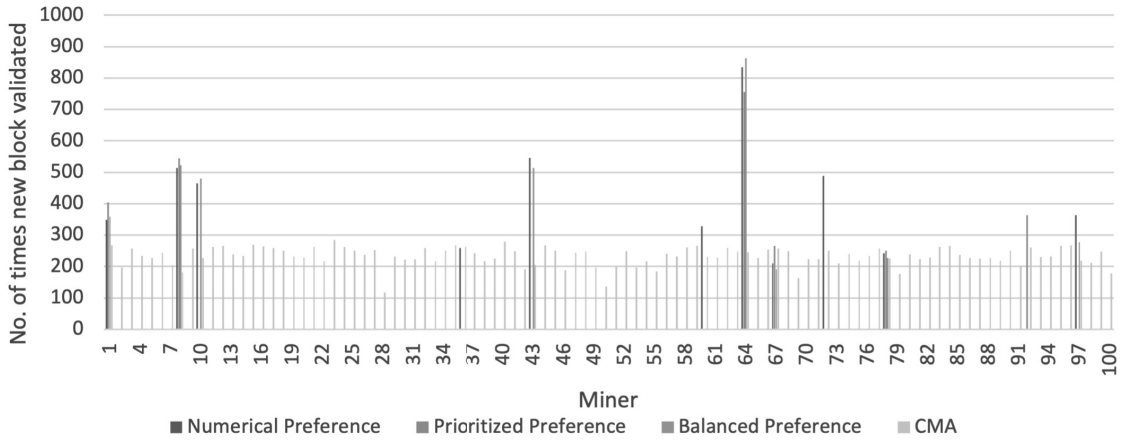

Fig. 5. Comparison of various preferences in block creation by a miner

By adding preference, we allowed users to select the type of miners they wanted to mine their block. Therefore, if a miner wants always to get elected and propose the block, they should not dominate only a single parameter, for example, processing power, as there is no guarantee on what the users can set as a preference. Preference added certain randomness to the system and increased the entropy of the system.

The proposed model is resilient to various attacks through several key mechanisms. It mitigates Sybil attacks by using the user preference model in the CMA consensus algorithm, which prevents manipulation by ensuring miner selection is based on diverse parameters. To counter DDoS attacks, the lightweight CMA algorithm minimizes resource consumption and the bidirectional-linked blockchain structure distributes network load, reducing impact. Additionally,

the bidirectional linking enhances security by detecting block alterations, effectively reducing the likelihood of long-range attacks by over 90%. Performance metrics and simulations demonstrate that the model maintains high resilience, with a 2-3x improvement in DDoS attack resistance due to reduced overhead and better data distribution. Further security analysis of the attacks is presented in the following sections.

6.1 Double Spend Attack and Long-Range Attack Resistance

A double-spend attack occurs when the same cryptocurrency is being spent twice, and the transaction information is altered and entered into the blockchain. A long-range attack is caused when a miner tries to create an alternative chain from an existing blockchain. These attacks occur when an uncertain new block is added to the blockchain. The proposed model will completely resist the long-range attack because of the bidirectional-linked blockchain. The reverse pointer ($HashPrev$) will hold the previous block's hash value, making it impossible to create a new chain. The chain can be undisputed as the subsequent of any block can be determined starting from the genesis block using the pointer values.

Double spending could be treated as Gambler's ruin problem, a famous statistical scenario centered around conditional probabilities and experimental outcomes as analyzed by Nakamoto in [28]. The probability of an attacker catching up with the honest miner (M) can be calculated below.

$$M(q,b) = 1 - \sum_{k=0}^{b} \frac{\lambda^k e^{-\lambda}}{k!}(1 - (\frac{q}{p})^{b-k}) \tag{6}$$

where, b represents the number of blocks the merchant waits before handing physical goods, and p is the probability that an honest node finds the next block. q is the probability that the attacker finds the next block. λ is the block-producing rate of the attacker during the interval that honest miners produce b blocks, which is calculated below.

$$\lambda = b\frac{q}{p} \tag{7}$$

Based on Eq. (7) to find out the probability that the attacker could overtake the honest miners (which means that the double-spend attack happens), b is replaced with $b + 1$.

$$M(q,b) = 1 - \sum_{k=0}^{b+1} \frac{\lambda^k e^{-\lambda}}{k!}(1 - (\frac{q}{p})^{b+1-k}) \tag{8}$$

For the preference-based CMA consensus algorithm, q is proportion to the miner's resource values (like computing power and stake owned). Theoretically, the value of q would be propositional to the preference parameter set for the term. Therefore, the probability that the attacker controls all the committee members for each term is propositional to q. A double-spend attack can happen if the attacker controls all the committee members. However, suppose the attacker does not control many committee members (honest committee members). In that case, they will not provide their part of the trapdoor keys if they dispute the block, making the model resistant to the double spend attack. From the experiments, it was noted that the same committee members are elected most of the time. If the attacker controls these miners, the probability of the double spend attack increases.

6.2 Eclipse Attack Resistance

An eclipse attack is a network-based attack in which an attacker creates an artificial environment around a miner to manipulate it into wrongful action. This attack depends on the entropy (randomness) of the system. If the entropy is

high, predicting the next term's committee members is easier, making the attack ineffective. The entropy of the system is defined below.

$$H(X) = -\sum_{i=1}^{n} P(u_i) \log_b P(u_i) \tag{9}$$

where $P(u_i)$ is the probability of miner u_i participating in the consensus algorithm, and b is the base of the logarithm used. The likelihood of the users being elected is propositional to their resource values. From our experiment and according to the Pareto principle, we found that the entropy of the preference-based CMA algorithm system was low. Lower entropy means a lower level of security. If certain miners are targeted, it would be easier to launch an eclipse attack. The existing CMA algorithm has a higher entropy as the elected miner was random, making it difficult to predict the committee member of the following term and launch the attack.

Even if the attack were launched at the elected committee member with the highest priority to propose the block, it would not hinder the mining process. This is because the committee member with the next highest priority can propose their transaction request and continue to reach a consensus. In a preference-based CMA algorithm, the attacker has to control all the committee members to hinder the consensus algorithm. Since the same miners are elected most often, this would not be impossible or ineffective. For the original CMA algorithm, it would be impractical to launch as every term will have a different set of miners.

7 CONCLUSION

This paper has presented a novel approach to consensus algorithm design by integrating a preference-based mechanism within a Committee Member Agreement (CMA) framework alongside a bidirectional-linked blockchain. We have demonstrated how incorporating user preferences into the selection process for committee members can enhance both performance and user agency within the blockchain ecosystem.

Our investigation revealed that the introduction of user preferences, coupled with the use of a bidirectional-linked blockchain facilitated by the Chameleon-hash function, offers promising avenues for improving both the efficiency and security of blockchain networks. However, the introduction of user preferences also brings certain challenges. Notably, the Pareto principle results in a concentration of mining power among a small fraction of miners, potentially leading to centralization. Additionally, the model's relatively low entropy makes it susceptible to double spending and eclipse attacks. Scalability remains a concern, particularly as transaction volumes grow.

Future work will aim to mitigate these limitations by refining the preference model to achieve an optimal balance between performance and security, devising strategies to decentralize mining power, and enhancing scalability to handle growing transaction volumes. Addressing these challenges will allow us to further optimize the preference-driven consensus algorithm, contributing to the advancement of more resilient and user-centric blockchain systems. Furthermore, exploring the implementation of our approach in real-world IoT environments represents another direction for future research.

REFERENCES

[1] Ibrahim Ahmed I. AlMallohi, Ahmed Saad M. Alotaibi, Rahaf Alghafees, Farzana Azam, and Zeeshan Shafi Khan. 2019. Multivariable based checkpoints to mitigate the long range attack in proof-of-stake based blockchains. *Proceedings of the 3rd International Conference on High Performance Compilation, Computing and Communications* (2019), 118–122.

[2] Leo Maxim Bach, Branko Mihaljevic, and Mario Zagar. 2018. Comparative analysis of blockchain consensus algorithms. *2018 41st international convention on information and communication technology, electronics and microelectronics (MIPRO)* (2018), 1545–1550.

[3] Seyed Mojtaba Hosseini Bamakan, Amirhossein Motavali, and Alireza Babaei Bondarti. 2020. A survey of blockchain consensus algorithms performance evaluation criteria. *Expert Systems with Applications* 154 (2020), 113385.

[4] Masoud Barati, Gagangeet Singh Aujla, Jose Tomas Llanos, Kwabena Adu Duodu, Omer F Rana, Madeline Carr, and Rajiv Ranjan. 2021. Privacy-aware cloud auditing for GDPR compliance verification in online healthcare. *IEEE Transactions on Industrial Informatics* 18 (2021), 4808–4819.

[5] M. Barati, W. J. Buchanan, O. Lo, and Omer Rana. 2022. A privacy-preserving distributed platform for COVID-19 vaccine passports. *14th IEEE/ACM International Conference on Utility and Cloud Computing Companion* (2022), 6 pages.

[6] Masoud Barati and Omer Rana. 2021. Privacy-aware cloud ecosystems: Architecture and performance. *Concurrency and Computation: Practice and Experience* 33 (2021), e5852.

[7] Masoud Barati, Omer Rana, George Theodorakopoulos, and Peter Burnap. 2019. Privacy-Aware Cloud Ecosystems and GDPR Compliance. *2019 7th International Conference on Future Internet of Things and Cloud (FiCloud)* (2019), 117–124.

[8] Masoud Barati and Richard St-Denis. 2017. Team formation through preference-based behavior composition. *Multiagent System Technologies: 15th German Conference, MATES 2017, Leipzig, Germany, August 23–26, 2017, Proceedings 15* (2017), 54–71.

[9] Nir Bitansky. 2020. Verifiable random functions from non-interactive witness-indistinguishable proofs. *Journal of Cryptology* 33 (2020), 459–493.

[10] Rosie Dunford, Quanrong Su, and Ekraj Tamang. 2014. The pareto principle. *University of Plymouth* (2014).

[11] José María García, David Ruiz, and Antonio Ruiz-Cortés. 2010. A model of user preferences for semantic services discovery and ranking. *The Semantic Web: Research and Applications: 7th Extended Semantic Web Conference, ESWC 2010, Heraklion, Crete, Greece, May 30–June 3, 2010, Proceedings, Part II 7* (2010), 1–14.

[12] Arthur Gervais, Ghassan O. Karame, Karl Wüst, Vasileios Glykantzis, Hubert Ritzdorf, and Srdjan Capkun. 2016. On the Security and Performance of Proof of Work Blockchains. *Association for Computing Machinery* (2016), 3–16.

[13] Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. 2017. Algorand: Scaling byzantine agreements for cryptocurrencies. *Proceedings of the 26th symposium on operating systems principles* (2017), 51–68.

[14] B. B. Gupta and M. Quamara. 2018. An overview of Internet of Things (IoT): Architectural aspects, challenges, and protocols. *Concurrency and Computation: Practice and Experience* 32, 21 (Sept. 2018). https://doi.org/10.1002/cpe.4946

[15] X. Hao, L. Yu, L. Zhiqiang, L. Zhen, and G. Dawu. 2018. Dynamic Practical Byzantine Fault Tolerance. *2018 IEEE Conference on Communications and Network Security* (2018), 8 pages.

[16] Ethan Heilman, Alison Kendler, Aviv Zohar, and Sharon Goldberg. 2015. Eclipse Attacks on Bitcoin's Peer-to-Peer Network. *USENIX Association* (2015), 129–144.

[17] Ethan Heilman, Alison Kendler, Aviv Zohar, and Sharon Goldberg. 2015. Eclipse attacks on {Bitcoin's}{peer-to-peer} network. *24th USENIX security symposium (USENIX security 15)* (2015), 129–144.

[18] Q. Hu, B. Yan, Y. Han, and J. Yu. 2021. An Improved Delegated Proof of Stake Consensus Algorithm. *Procedia Computer Science* 187 (2021), 341–3468. https://doi.org/10.1016/j.procs.2021.04.109

[19] Xumin Huang, Dongdong Ye, Rong Yu, and Lei Shu. 2020. Securing parked vehicle assisted fog computing with blockchain and optimal smart contract design. *IEEE/CAA Journal of Automatica Sinica* 7 (2020), 426–441.

[20] A. K. Al Hwaitat, M. A. Almaiah, A. Ali, S. Al-Otaibi, R. Shishakly, A. Lutfi, and M. Alrawad. 2023. A New Blockchain-Based Authentication Framework for Secure IoT Networks. *Electronics* 12 (2023), 25 pages. https://doi.org/10.3390/electronics12173618

[21] M. Kashif and K. Kalkan. 2024. EPIoT: Enhanced privacy preservation based blockchain mechanism for internet-of-things. *Computer Networks* 238 (2024), p. 110107. https://doi.org/10.1016/j.comnet.2023.110107

[22] Werner Kießling. 2002. Foundations of preferences in database systems. *VLDB'02: Proceedings of the 28th International Conference on Very Large Databases* (2002), 311–322.

[23] Gaolei Li, Mianxiong Dong, Laurence T. Yang, Kaoru Ota, Jun Wu, and Jianhua Li. 2020. Preserving Edge Knowledge Sharing Among IoT Services: A Blockchain-Based Approach. *IEEE Transactions on Emerging Topics in Computational Intelligence* 4 (2020), 653–665.

[24] V. A. Memos, K. E. Psannis, Y. Ishibashi, B.-G. Kim, and B.B. Gupta. 2018. An Efficient Algorithm for Media-based Surveillance System (EAMSuS) in IoT Smart City Framework. *Future Generation Computer Systems* 83 (2018), 619–628. https://doi.org/10.1016/j.future.2017.04.039

[25] Regio A. Michelin, Ali Dorri, Roben Castagna Lunardi, Marco Steger, Salil S. Kanhere, Raja Jurdak, and Avelino Francisco Zorzo. 2018. SpeedyChain: A framework for decoupling data from blockchain for smart cities. *Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services* (2018), 451–463.

[26] Du Mingxiao, Ma Xiaofeng, Zhang Zhe, Wang Xiangwei, and Chen Qijun. 2017. A review on consensus algorithm of blockchain. *2017 IEEE international conference on systems, man, and cybernetics (SMC)* (2017), 2567–2572.

[27] Daniel J. Moroz, Daniel J. Aronoff, Neha Narula, and David C. Parkes. 2020. Double-Spend Counter-Attacks: Threat of Retaliation in Proof-of-Work Systems. *Cryptoeconomic Systems* 0 (2020), 10736.

[28] Satoshi Nakamoto and A Bitcoin. 2008. A peer-to-peer electronic cash system. *Bitcoin.–URL: https://bitcoin. org/bitcoin. pdf* 4 (2008), 15.

[29] Cristina Pérez-Solà, Sergi Delgado-Segura, Guillermo Navarro-Arribas, and Jordi Herrera-Joancomartí. 2019. Double-spending prevention for Bitcoin zero-confirmation transactions. *Springer-Verlag* 18 (2019), 451–463.

[30] Minfeng Qi, Ziyuan Wang, Qing-Long Han, Jun Zhang, Shiping Chen, and Yang Xiang. 2022. Privacy protection for blockchain-based healthcare IoT systems: A survey. *IEEE/CAA Journal of Automatica Sinica* (2022), 1–20.

[31] Alejandro Ranchal-Pedrosa and Vincent Gramoli. 2019. Platypus: Offchain Protocol Without Synchrony. *2019 IEEE 18th International Symposium on Network Computing and Applications (NCA)* (2019), 1–8.

[32] Maninderpal Singh, Gagangeet Singh Aujla, and Rasmeet Singh Bali. 2020. ODOB: One Drone One Block-based Lightweight Blockchain Architecture for Internet of Drones. *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)* (2020), 249–254.

[33] Maninder Pal Singh, Gagangeet Aujla, and Rasmeet Bali. 2022. Derived blockchain architecture for security-conscious data dissemination in edge-envisioned Internet of Drones ecosystem. *Cluster Computing* 25 (2022), 2281–2302.

[34] T. Wang, S. Ai, J. Cao, and Y. Zhao. 2023. A Blockchain-Based Distributed Computational Resource Trading Strategy for Internet of Things Considering Multiple Preferences. *Symmetry* 15 (2023), 21 pages. https://doi.org/10.3390/sym15040808

[35] Chenhao Xu, Youyang Qu, Tom H Luan, Peter W Eklund, Yong Xiang, and Longxiang Gao. 2021. A lightweight and attack-proof bidirectional blockchain paradigm for internet of things. *IEEE Internet of Things Journal* 9 (2021), 4371–4384.

[36] Guangquan Xu, Bingjiang Guo, Chunhua Su, Xi Zheng, Kaitai Liang, Duncan S. Wong, and Hao Wang. 2020. Am I eclipsed? A smart detector of eclipse attacks for Ethereum. *Computers Security* 88 (2020), 249–254.

[37] "MengChu Zhou" "Yue Zhou", "Xin Luo". 2023. Cryptocurrency Transaction Network Embedding From Static and Dynamic Perspectives: An Overview. *IEEE/CAA Journal of Automatica Sinica* 10 (2023), 1105.

[38] Peiyun Zhang and Mengchu Zhou. 2020. Security and Trust in Blockchains: Architecture, Key Technologies, and Open Issues. *IEEE Transactions on Computational Social Systems* 7 (2020), 790–801.

[39] PeiYun Zhang, MengChu Zhou, QiXi Zhao, Abdullah Abusorrah, and Omaimah O. Bamasag. 2021. A Performance-Optimized Consensus Mechanism for Consortium Blockchains Consisting of Trust-Varying Nodes. *IEEE Transactions on Network Science and Engineering* 8 (2021), 2147–2159.

[40] Shijie Zhang and Jong-Hyouk Lee. 2019. Eclipse-based Stake-Bleeding Attacks in PoS Blockchain Systems. *Association for Computing Machinery* (2019), 67–72.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009