RPoC: An efficient and scalable consensus algorithm for SCM applications

Aaliya Sarfaraz, *Student Member, IEEE*, Ripon K. Chakraborty, *Member, IEEE*,
and Daryl L. Essam, *Member, IEEE*,

Abstract—The growing interest in blockchain technology has gained a lot of attention in supply chain management (SCM) and sparked the quest for decentralized, scalable, efficient, and trustworthy consensus schemes. Traditional blockchains rely on computationally expensive consensus mechanisms with low throughput and high latency. This paper conducts a performance evaluation of several existing consensus protocols to illustrate blockchain's shortcomings in terms of consensus and propose a new consensus algorithm: Reputation-based proof-of cooperation (RPoC). The RPoC algorithm uses a layered architecture to segment the nodes that participate in the consensus phase, in order to improve scalability and efficiency while maintaining trust among peers. Layered design addresses the issues of flexibility and scalability, as well as breaking down the extensive mining process into segments. Rather than choosing a few nodes for mining, the proposed consensus process involves all network nodes, making it more efficient, decentralized, and scalable. Through extensive theoretical analysis and experimentation, the suitability of the proposed algorithm is well grounded in terms of scalability and efficiency.

Index Terms—Blockchain, Supply chain, Consensus algorithm, Efficiency, Decentralization

I. INTRODUCTION

BLOCKCHAIN has gained a lot of recognition in the Supply Chain Management (SCM) domain recently, mostly due to the emergence of digitization and growth of the industry 4.0 context across industries. The emergence of Bitcoin [1], has further fuelled this recognition. Although cryptocurrencies have been the most well-known use of blockchain technology, several researchers have also identified the usage of blockchain and cryptocurrencies in different supply chain applications [2] [3]. However, integrating blockchain technology into traditional SCM is a significant challenge, particularly with the absence of any tailored consensus algorithms to tackle or embed within supply chain problems [4].

The blockchain architecture validates information through a consensus mechanism among network nodes, removing the need for intermediaries. The consensus mechanism ensures a tamper-proof environment and makes sure that the information stored is reliable and valid [5]. In a blockchain, all nodes must agree on the current state of the ledger, making it difficult for adversaries to insert tampered blocks. The consensus algorithm is the most significant component of a blockchain system as its efficiency significantly influences each blockchain's overall

efficiency [5]. With the growing adoption of blockchain in the SCM domain, a number of consensus algorithms have been developed to meet these requirements. When it comes to implementing a blockchain solution for a SCM application, choosing the right consensus algorithm is critical. Since consensus is the essence of blockchains, it is utilized to foster trust among stakeholders. Therefore, the influence of a mining node on a supply chain system's operation will be crucial. This involves the creation of more extensive consensus algorithms that can efficiently choose miners and scale to different numbers of nodes.

Many challenges continue to affect blockchain technology, including insufficient transactions per second (TPS), transaction latency, and decentralization [6]. The throughput of existing blockchains is relatively low due to the complex consensus process, for instance, in a public blockchain with the proof-of-work (PoW) consensus algorithm, all nodes must perform hash calculations and are only allowed to broadcast their blocks after spending a great deal of energy for their computation [7]. As a matter of fact, high consensus latency, low throughput, and high energy consumption make it difficult to use PoW in complex or large supply chain systems. Delegated Proof Of Stake (DPoS) [8] is an improved and optimized version of Proof of Stake (PoS) that reduces the time delay, increases consensus throughput, and offsets for the lack of accountability by majority elections. A blockchain system's nodes are divided into three groups by DPoS: witnesses, delegates, and workers. However, since DPoS retains a degree of centralization, there are significant security risks [8]. Practical Byzantine Fault Tolerance (PBFT) [9] was designed to be high-performance with low overhead in an asynchronous system and reaches consensus after a series of steps. Hyperledger Fabric [10] and Zilliqa [11] are two projects that presently employ PBFT. The PBFT algorithm is fast at processing transaction requests, but the overhead of communications limits its scalability [12]. Moreover, due to the necessity of defining the number of nodes before starting to operate, this algorithm is difficult to use in a SCM setting. In Ripple [13], nodes accumulate the unconfirmed transactions and publish them as candidate sets. The candidate sets that receive more votes than a fixed threshold proceed to the next round. This procedure is repeated until a collection receives at least 80 % of the votes cast by the nodes, at which point it is added to the ledger. Unlike bitcoin [14], where coins are distributed randomly to miners, Ripple is pre-mined, and is thus highly centralized and not a suitable candidate for SCM applications [9].

Proof of Importance (PoI) [15] is an advanced consensus

The authors are with the School of Engineering and Information Technology, University of New South Wales, ACT, Australia (e-mail: a.sarfaraz@student.unsw.edu.au; r.chakraborty@adfa.edu.au; d.essam@adfa.edu.au).

mechanism that eliminates the disadvantage of the wealthy being even wealthier. Each node's importance scale decides which nodes are qualified to add a block to blockchain. However concern with this is that nodes would accumulate as many coins as possible to reap the benefits of block formation, so this behavior concentrates capital and reduces transaction activity. In Proof-of-Capacity (PoC) [6], a miner's storage is prioritized over hashing power. The aim of this mechanism is to reduce the amount of computing energy used. Instead of computing the hash in every block, PoC allows the list of potential solutions to be stored even before a block is mined. PoC is scalable, efficient, and cost-effective, however, with the rise of cloud providers and large corporations, the mining process is becoming increasingly centralized and monopolized [9]. Considering all those shortcomings in existing consensus algorithms, a proper and customized consensus algorithm should be designed for typical SCM problems, particularly to resolve the TPS, latency and centralization issues. The majority of current consensus algorithm research is focused on improving mainstream consensus algorithms, even though only a few are relevant to SCM, which are highlighted in the literature review section.

While the dynamic SC sector has enormous development potential, it is challenged by several other SCM issues. The Bullwhip Effect (BWE) [16] is one of them and has been discussed in the literature in recent years. BWE occurs by order oscillations at each SC stage. Blockchain can mitigate BWE by providing real time information and coordination among stake holders. It is crucial to share appropriate demand data throughout a SC, because it may help the upstream echelons with resource and material scheduling. Furthermore, inventory requirements might be directly linked to inconsistencies between demand over time and actual demand fulfillment. In this research we offset the traditional SCM phenomenon of BWE with BC by providing total visibility and exchanging demand data across all stakeholders. This would keep business transactions tamper-proof and available to stakeholders without the need for a centralized control body, as long as business practices and negotiated data processing contracts between firms were followed.

To address the aforementioned challenges, we first choose a few existing proofs based (PoW, DPOS) and voting based (PoI, PoC, Ripple, BPFT) consensus algorithms to test their performance in a proposed blockchain based SCM framework. Based on the performance of those existing consensus algorithms, the second layer of this work proposes a reputation-based consensus mechanism by redesigning some of the existing approaches, while complementing their strength and eliminating some of their weaknesses. We name this proposed approach as reputation-based Proof-of-Coordination (RPoC) consensus mechanism that reaches consensus by coordinating between two layers of nodes. The first layer consists of high authority nodes which are chosen based on a combination of their reputation score and verified identity, while the second layer is made up of subordinate nodes who are selected using a random selection algorithm and grouped in clusters with a master node for each. By the performance evaluation (see section V), each of the six existing consensus algorithms decreases

blockchain efficiency by limiting blockchain throughput and increasing transaction latency. Whereas RPoC is made up of layers, each with its own set of nodes operating in parallel, thus increasing efficiency, decreasing latency, and eliminating the centralization issue.

This paper provides a blockchain enabled SCM framework to provide visibility and coordination along with a blockchain consensus processes. We propose a two-layer consensus algorithm that combines a reputation and random selection algorithm, appropriate for the SCM. The following are the paper's key contributions:

- i An information sharing framework is implemented based on a permissioned blockchain for a complex SC scenario. The use of BC technology in SCM is considered in terms of mitigating BWE.
- ii A detailed analysis of several existing decentralized consensus algorithms is presented.
- iii A reputation-based consensus algorithm has been proposed by combining the advantages of existing algorithms and throughput, scalability, and latency were verified and validated for the improved algorithm.
- iv The proposed algorithm is compared to the existing consensus algorithms and significantly improves TPS and scalability for SCM applications.

The rest of this article is laid out as follows. Section II offers a comprehensive literature review and section III presents a taxonomy of consensus algorithms and their underlying properties. Section IV introduces the RPoC consensus algorithm and section V, presents the computational results and discussion along with different performance comparisons. The conclusions are drawn in section VI.

II. RELATED WORK

In this section, we review the literature regarding different consensus algorithms. In Section II-A, we discuss existing blockchain-based solutions for SCM applications, accompanied by an analysis of existing consensus algorithms in Section II-B.

A. BLOCKCHAIN BASED SCM

SCM networks often grow exponentially, requiring fast communication among various stakeholders. Improved product traceability, better data visibility and efficient data sharing among diverse stakeholders are a few key aspects for an efficient SCM ecosystem. Consequently, blockchain is currently being used in different supply chain problems to increase transparency, auditability, and visibility. Abeyratne and Monfared [17] recognized the potential of blockchain technology in attaining consensus in low-trust settings, where limited trust between organizations and customers were enhanced by allowing product tracing and providing transparency for manufacturing processes. Later, Gao et al. [18] argued that keeping a global picture of a supply chain is challenging due to the diverse engagement of systems and suggested the use of blockchain technology to improve the situation. Industry and academia have paid close attention to the use of blockchain technology in the food supply chain. Walmart [19],

for example, is a business Endeavor that uses blockchain technology to bring transparency to a decentralized food supply ecosystem by digitizing the supply chain process. Traditional supply chain procedures are progressively being influenced by blockchain technology. Maersk [20] is another example of a blockchain-enabled digital shipping platform for a global supply chain.

ProductChain [21] is a provenance architecture for food supply chains that allows for blockchain-based food monitoring and tracing. A similar approach was proposed by Wang et al. [22] for product tracing and transferring and was based on blockchain using smart contracts. Apart from the food chain use cases, the underlying blockchain technology has a significant impact on the usability of many approaches in real-world deployments. For instance, Wang et al. [23] examines the application of blockchain in a construction supply chain in a feasibility study. Similarly, Rahmanzadeh et al. [24] recommended using a blockchain in combination with fuzzy sets to make tactical supply chain policy decisions. van Engelenburg et al. [25] designed a blockchain architecture to evaluate if it can reduce information asymmetry while preserving sensitive data. They concluded that blockchain technology can help parties strike a balance between inventory control and flexibility, however data privacy issues exist.

Furthermore, in supply chains, achieving adequate scalability features is another challenge for all blockchain enabled systems, because huge volumes of transactions leads to increased blockchain sizes and may cause a blockchain to reach its transaction throughput limits. Lin et al. [26] prototype a food safety tracking system that uses blockchain, smart contract, and Electronic Product Code Information Services (EPCIS). To resolve the data expansion issue, they limit the amount of information from a single node by collaborating on on-chain and off-chain data management. Likewise, many researchers have been actively investigating different features of blockchain. [26] employed a smart contract instead of traditional transaction records to protect critical business information. Furthermore, Bader et al. [27] introduced PrivAccChain, a privacy-preserving architecture for optimizing multi-hop information retrieval while maintaining stakeholder accountability in supply chains. Despite all these recent works and attentions, blockchain research in the SC domain is still in its early stages, which has yet to overcome significant obstacles in terms of scalability, throughput, latency, and trust.

B. CONSENSUS ALGORITHMS

In the previous subsection, we explored blockchain applications in SCM. Blockchain relies on consensus algorithms to reduce computing costs, latency, and throughput. Therefore, in this section we will analyze existing techniques for achieving consensus. Consensus algorithms are critical for improving and automating business and vendor customer logistics between various stakeholders in SCM. It is critical in accelerating the delivery of a manufactured product with greater performance while reducing costs and time. According to the literature, most consensus algorithms are created specifically for cryptocurrency [7] [28]. However, the trend is shifting,

and SCM is embracing blockchain for a variety of reasons, including traceability and efficiency, security, and trust.

Chainweb [29] scaled conventional PoW by running numerous parallel chains. Each network chain mines the same coin, which may be moved between chains via a simple payment verification and thus achieved better throughput and shorter latency. Later, Li et al. [30] solved communication issues in BPFT, by presenting a multi-layer PBFT consensus mechanism to increase scalability. Existing consensus mechanisms, such as PoW, necessitate computational resources in a distributed network, making it difficult for SCM applications to maintain consistency and scalability. Litke et al. [31] discussed SCM's feasibility and identified various decentralized consensus mechanisms that can be utilized in SCM-based applications. The authors concluded that despite existing consensus methods, there is still a need for consensus algorithms that are versatile enough to accommodate a wide range of applications and use cases for various supply chain ones. The study [32] suggested a consensus approach, Proof-of-Trust (PoT), which is applicable for the online service industry. It uses a hybrid blockchain design that incorporates a consortium blockchain into an open, public service network, bringing accountability to the business. This consensus algorithm uses RAFT leader election [33] and Shamir's secret sharing techniques [34], and picks transaction validators based on their trust values. later, Zhu et al. [35] improved PoT by combining the crowd sourcing scenario with blockchain to enable distributed governance and accountability. The algorithm used a subjective logic reputation mechanism and selected nodes with high trust. Only a small number of nodes were allowed to create blocks, participate in verification, and claim crowd sourcing assignments.

Tsang et al. [36] formulated a consensus protocol and proposed blockchain-based food trace-ability systems in which blocks were mined by stakeholders rather than designated miners in SCM. Instead of using any proof, the proposed consensus algorithm chooses stakeholders to verify and validate in blockchain on a probabilistic basis. However, that proposed consensus process is limited by the fact that it is focused only on the scenario of food traceability, rather than other supply chain applications. Yusuf et al. [37] proposed a blockchain-based network, which addressed supplier issues in a vegetable supplier's SCM. The crash fault-tolerant consensus algorithm was used to validate the proposed blockchain-based network. The authors claimed that this consensus algorithm was more appropriate and feasible for the vegetable supply chain.

The literature above emphasizes the importance of blockchain in SCM, and at the same time it highlights the requirement of improving system efficiency and information accuracy to meet the needs of evolving supply chain operations. Blockchain based SCM has been found to be promising in terms of making information sharing more effective. However, due to the consensus mechanism, directly implementing blockchain in SCM for efficiency is impractical. To fill the current research gap, this paper proposes a blockchain consensus algorithm that is explicitly for SCM.

III. BLOCKCHAIN CONSENSUS ALGORITHMS

A blockchain is a distributed system based on a network of peers to verify and validate transactions. Maintaining a consensus process is critical to ensure that all participants agree on the ledger's status. Furthermore, as blockchain is decentralized, there is no central authority to verify that all distributed node ledgers are identical. To ensure that ledgers in different nodes are consistent, several protocols have been designed. Using a consensus system allows peers to reach a decision based on certain rules. The consensus mechanisms can be divided based on how participants in the verifying network agree with one another regarding the ledger they maintain. Therefore, we classify blockchain consensus algorithms into two categories: proof-based consensus algorithms and voting-based consensus algorithms.

A. Proof-Based Consensus Algorithms

The basic idea behind a proof-based consensus algorithm is that among several nodes entering the network, the node that performs sufficient proof will be granted the right to append a new block to the chain and obtain some reward. Proof-based consensus algorithms are more suitable for public blockchains and difficult to implement in SCM due to broad consensus latency and high energy usage [33].

1) Proof of work: [9]

In this approach, the miners do many calculations in order to solve a mathematical puzzle, and the puzzle is solved using a Hash algorithm SHA-256 [38]. A typical PoW block consists of: hash of previous and current block, a transaction record and a nonce. To reach consensus, miners seek hash values that are equal to or lower than a "Target hash". Whenever a miner finds a solution, it announces the block to the entire network, where all other miners check the hash value to verify it. If the block is validated, the miner gets the reward for mining and every other node in the network will add this block to their chain. Although the PoW algorithm has a high level of decentralization and security, it confronts challenges with the mining process, as well as resource and time consumption. In addition, the speed and success rate of the hash function are largely dependent on the processing capabilities of the hardware of the miner. Because of the aforementioned limitations, it is unsuitable for big networks like SCM, that require great efficiency.

2) Delegated proof-of-stake: [8]

This consensus algorithm is an improved version of Proof-of-Stake, where nodes are allowed to choose validators to validate blocks by voting. The following is how DPoS works: Minors are referred to as delegates and selected delegates are rotated through the network from time to time, and they deliver blocks in a predetermined sequence. When there are fewer delegates in the network, it is easier for them to organize themselves according to the designated time slot. When delegates publish invalid transactions, the rest of the token holders vote them out and new delegates are chosen. Users may delegate their voting power to other users they trust to vote for them. Since the number of validators is small, the network can be easily organized, and validators can decide what is the best time to

publish blocks. However, restricting the number of validators would result in a centralized network. Despite its scalability, energy conservation, and low-cost transactions, its application in SCM is limited because of its semi-centralized nature.

3) Proof-of-Importance: [15]

PoI is a consensus algorithm developed in response to critiques of the Proof-of-Stake (POS) algorithm. According to the PoS paradigm, a node's ability to mine block transactions is proportional to the number of coins they own. This strategy would incentives miners to save their coins rather than spending them while making the rich richer. However, with PoI, each node is assigned an importance value, and nodes are chosen for mining based on that value. This approach maintains blockchain's decentralization while also striking a balance between channelling funds in wallets and circulating them out. PoI enables certain nodes to mine blocks that support the chain's infrastructure rather than focusing solely on computational and value aspects. Since no mining is needed, the PoI algorithm is fast, energy-efficient, and secure.

4) Proof-of-Capacity: [6]

In the PoC consensus algorithm, free hard drive space is required, so miners have to invest their money in large capacity hard disks. The PoC technique allows mining devices to validate transactions using their available hard drive space instead of by energy consumption. PoC works by saving a list of solutions on the hard drive before it starts mining. It can produce several large datasets (known as plots) on a hard disk during work. The more plots a node has, the more chances it has to match the required hash value, resulting in a higher possibility of winning the mining reward. If a hard drive solves the previous block's problem the fastest, it wins the block.

B. Voting-Based Consensus Algorithm

In voting-based consensus algorithms, all nodes in a network will participate in validating transactions and maintain the ledger together. Before appending new blocks to the ledger, they exchange their results with a voting-based consensus algorithm. It is preferred in private and consortium blockchain and is too slow and inefficient for SCM applications.

1) Ripple: [13]

Ripple divides network nodes into two types: servers and clients. Servers are responsible for the consensus process and clients can only move funds. Each server contains its own list of nodes which is called Unique Node Lists (UNLs). The importance of UNL to the server cannot be overstated. When deciding whether to pack a transaction into the ledger, the server queries the UNL nodes, and if the received agreements exceed 80%, the transaction is packed into the ledger. The ledger for a node will remain correct as long as the number of defective nodes in a UNL is less than 20%.

2) Practical byzantine fault tolerance (PBFT): [9]

In PBFT, all nodes must vote to mine a block, and consensus is reached when more than two-thirds of the nodes approve a block and can usually tolerate one-third of the network's performance. The whole process can be divided into three phases: pre-prepared, prepared and commit. A primary will

be chosen according to certain rules in each round and in each round, a new block is decided. In each round, a node advances to the next phase if it receives votes from more than two-thirds of all nodes. As a result, when running the PBFT algorithm, the nodes in the entire network must be specified. However, a constant number of nodes in an SCM application is not guaranteed. The PBFT algorithm cannot be a perfect alternative for SCM due to the unknown number of network nodes. Unlike other consensus algorithms, this approach does not require any asset to be staked, and thus allows for faster and more cost-effective consensus. Its advantages include energy efficiency and high throughput, while its drawbacks include a lack of scalability and storage latency due to the network's need to wait for all nodes' votes.

Table II summarizes the above algorithms along with their advantages and disadvantages.

IV. SYSTEM MODEL AND CONSENSUS SCHEME

In this section, we demonstrate how we employed our Reputation-based Proof-of-coordination approach to build a supply chain architecture that minimized the bullwhip effect. In addition, the selection of consensus nodes and the block confirmation mechanism in our RPoC are detailed.

A. Blockchain based SCM framework

This section describes a blockchain-based information sharing framework for a complex Supply Chain (SC) scenario that minimizes BWE. The proposed system architecture is depicted in Figure 1.

Figure 1: Blockchain based SCM Architecture

In the proposed model, manufacturing and non-manufacturing stakeholders are part of a multi-tier supply network. In addition to vertical information sharing and cooperation, this method requires horizontal communication between stakeholders on the same SC tier. Suppliers and producers who use BC will collaborate by exchanging demand data and stock levels. The collaboration can be done through a permissioned BC, so only those who are members of the SC have access. It's easy to measure the effect of demand data because all layers of the supply chain share the same demand data and inventory policy. The system model and assumptions are given below:

1) *Assumptions:* The following assumptions are provided in the SC model.

- The retailer tracks the demand of the end consumer and places orders at the top tier (e.g., distributor or manufacturer) using the (Q, R) inventory policy.
- Any number demanded by the retailer can be generated indefinitely by the producer.
- Out of stock orders are not lost at any point; instead, they become backlogs that will be executed as soon as the inventory is replenished.
- The actual demand cannot be predicted ahead of time.
- Orders might be positive or negative, and cancellations are permitted.

The credentials are obtained from a Certificate Authority (CA), which consists of a set of public and private keys as well as a digital signature. If a situation occurs, a CA has access to their individual identities and may disclose the true identities of the stakeholders and their relationships. The following is an overview of our blockchain-coordinated SCM framework.

- i Stakeholders will be assigned a minimum reputation score after receiving keys and joining the network.
- ii When an order arrives at the retailer, the stock inventory is initialized. To calculate the demand deviation, the demand quantity is reported in the blockchain. The supplier then determines whether the demand quantity can be met based on the current inventory level. If the inventory is greater than or equal to the demand quantity, the demand quantity is then removed from the inventory. Alternatively, if demand exceeds supply, the order will be sent to the upper echelon (e.g., distribution centre or manufacturer).
- iii When the relative inventory amount is less than or equal to the reorder point (ROP), an order is placed at the next higher echelon. Request ID and order quantity are among the details sent to the next higher echelon. The next upper echelon sends back information on this order's lead time, and information about the order, including order ID, date of release, lead time, and order quantity, which are also documented on the BC.
- iv An order is shipped when the delivery date arrives. The upper echelon would also receive information about the order ID and the actual delivery date. The stock amount is adjusted, and the order is removed from the order receipt list.
- v The replenishment quantity is added to the existing inventory as the lower echelon receives order delivery information from the next upper layer. If the estimated lead time differs from the order lead time, the estimated lead time is then updated.
- vi The bullwhip effect (BWE) ratio is calculated by the difference between the order placed and the demand received and recorded on blockchain.
- vii After every order is received/shipped, inventory analysis is conducted to see if the relative amount of inventory is less than or equal to ROP.
- viii All of the above steps are repeated When any stakeholders receive a demand quantity from their lower echelon.

Notion	meaning
N_i	Participating nodes in blockchain
Val_a	Higher authority validators
Val_s	Subordinate validators
Tn_x	Transaction generated by N_i
Hat_n	total number of nodes in higher authority layer
Sat_n	total number of nodes in subordinate layer
At_k	Malicious nodes

Table I: Frequently used notations

This paper proposes a new consensus method known as RPoC for improving the throughput and scalability of a blockchain-based SCM architecture. Blockchain's consensus algorithm is at its core, and it has a significant influence on its security and efficiency. The essential features required for SCM applications are scalability, security, and efficiency. Taking that into consideration, RPoC utilizes a two-layer design that allows for quick consensus and scalability. By distributing the mining operations to all participating nodes, layering decreases the workload on individual nodes and increases consensus performance.

B. Design of the proposed RPoC algorithm

As shown in Table II, the DPOS method is not actually decentralized, as authority continues to be concentrated in the hands of a small group of users. For scalability, DPOS foregoes decentralization. Therefore, executing an attack is easier because there are fewer individuals in charge of maintaining the network intact. Likewise, Ripple, PoC, and PoI have decentralization issues, therefore they are not viable choices for SCM. On the other hand, the PBFT algorithm's consensus model only works efficiently when the number of nodes in the distributed network is limited. PBFT does not scale efficiently because of the high communication cost that grows exponentially with each extra node in the network. The PoC protocol, from the proof-based consensus category, can be an adequate alternative for SCM because it does not require any resource or coins to invest. However, malware may have the ability to disrupt mining operations.

The algorithm has been designed considering the above evaluation. This section covers the fundamentals of RPoC and the acronyms used in this paper are listed in Table I. The proposed blockchain consensus algorithm has two steps from the creation of a block to its confirmation: consensus nodes selection and transaction confirmation (block confirmation). A rigorous identity verification process must be completed before a node may join the network. If a node wishes to be a Val_a , they must confirm their true identity and agree to share it with the rest of the network. Second, the system generates the node reputation value using a reputation algorithm and then analyses the nodes' credibility. Third, the nodes that choose not to stake their real identification are pushed to the pool of Val_s .

As a result of the fair node selection method, the block addition procedure is optimized, and blocks can be added to blockchain instantaneously after verification. The algorithm's overall structure is depicted in Figure 2. There are two layers

of N_i : Val_a and Val_s . To generate blocks, N_i are operating in parallel. Val_s are in charge of generating micro blocks and sending them to Val_a . These small blocks will be received by Val_a , who will verify them before combining them into a single block.

N_i in our system are equally responsible for confirming Tn_x throughout the entire blockchain and work hand in hand to boost system throughput. The algorithm's fundamental feature is the ability to accurately and efficiently pick consensus nodes to work in parallel. Consensus nodes are chosen at random from a large number of nodes based on their reputation score, their willingness to stake their identity, and the random selection algorithm selects a subset of nodes for each cluster at random.

Figure 2: Layer structure of the proposed mechanism.

1) *Consensus node selection*: Blockchain network is characterized as a peer-to-peer network made up of N_i . In this algorithm, we are classifying validators into two layers: Val_a and Val_s . In order to determine node allocation into each layer, the layering setup requires the use of different methods. Therefore, to establish a consensus node selection mechanism, the algorithm combines a random number generating approach with the node's reputation score system.

Reputation module: Proof-of-Reputation is a permissioned blockchain protocol that preserves the history of transaction outcomes without the involvement of a central third party [39]. In RPoC Val_a are chosen not only on the basis of their reputation (adopted from [39]) but also on the basis of their proved identity. When a stakeholder joins the network for the very first time, it has no previous reputation score. So, an initial reputation score Rep_{min} is assigned to the stakeholder, which is the minimal reputation score to continue operating in the network. The stakeholder's aggregate reputation $R(\Delta_T)$ at time Δ_T is calculated by combining the stakeholder's current and past reputation score.

$$R(\Delta_T) = \sum_{t=0}^{t=i} Rep_{VAL}(T) \quad (1)$$

where $Rep_{VAL}(t_0)$ is the initial reputation score of the stakeholder, and $Rep_{VAL}(t_i)$ is the current reputation score of the stakeholder, happening at Δ_T .

Along with reputation, we also use the value of identities in our algorithm, which implies that Val_a stake their real identities rather than any other resources. For Val_a , we will consider a small number of validators so we can make it a scalable system. The Hat_n layer contains N_i with higher reputation scores and verified identity. Using the random selection technique, the remaining nodes are grouped into clusters of Val_s . Every cluster has a master node, who is chosen based on their reputation score and oversees validating transitions and forwarding them in a small block. The node selection procedure is shown in Algorithm 1.

Algorithm 1 Algorithm of Selecting Consensus Nodes

```

1: procedure CONSNOSELT( $NodesN, AuthSTD$ )
2:    $AuthNodeLst(len(n)) \leftarrow 0$ 
3:    $ClusLst(len(n)) \leftarrow 0$ 
4:    $MasterNodesLst(len(n)) \leftarrow 0$ 
5:   for all  $i, n \in N$  do
6:      $AuthState \leftarrow AuthProcess(n)$ 
7:     if ( $AuthState = TRUE \wedge getRepScore(n) >$ 
 $Rep_{min}$ ) then
8:        $AuthNodesLst(i) \leftarrow n$ 
9:     else
10:       $ClusLst(i) = Random(n \notin AuthNodeLst)$ 
11:    end if
12:  end for
13:  for all  $i, nC \in ClusterLst$  do
14:     $MasterNodesLst(i) = Random(nC)$ 
15:  end for
16: end procedure

```

2) *Block confirmation*: The steps for block confirmation are as follows:

- A node initiates a transaction (Tn_x, Sig_c, T_s) where, Tn_x is transaction, Sig_c is the client's signature and T_s indicates the timestamp.
- The cluster nodes receives and verifies Sig_c and T_s . If the verification is successful, the transaction (Tn_x, Sig_c, T_s)_{cls} is forwarded to the master node in the cluster, where _{cls} is the signature of the cluster node.
- The transaction must be verified by the master node. It verifies that the cluster node's signature is correct and that the transaction has not been registered in blockchain. As soon as the verification is completed, the transaction is signed (Tn_x, Sig_c, T_s)_{cls, m} where _m is the master node signature.
- After signing, the transaction is pushed to the waiting pool. When there are a specific number of transactions in the pool, the master node packs each of them into a small block ($Smallblock_{TX}$)_m and broadcasts it to the same layer.
- After receiving ($Smallblock_{TX}$)_m, other cluster nodes verifies transactions included in the block. Upon successful verification, the master node receives ($CONSENT, Smallblock_{TX}$)_{SL}.

- The master node can send ($CONSENT, Smallblock_{TX}, Sig_{SL}$)_m to the higher authority consensus group. Where Sig_{SL} is all the signatures form subordinate nodes.
- There may still be some Tn_x left in the pool after packing a small block; these Tn_x will be verified first in the new round of consensus.
- After receiving a small block from Val_s , the nodes in the authority layer must validate signatures and transactions of the small block.
- Once its verified, the higher authority nodes send acknowledgment ($ACK_{accepted}, Smallblock_{TX}$)_{auth} to the subordinate nodes. In some case, if verification fails, rejection ($ACK_{rejected}, Smallblock_{TX}$)_{auth} is sent back to subordinates.
- The small blocks are put in chronological order after the verification is successful. A large block will be packaged and added to the blockchain after receiving a minimum of 10 small blocks.

Algorithm 2 presents the module for reaching a consensus on the verification of a block.

Algorithm 2 Algorithm of Reaching a Consensus

```

1: procedure REACHCONS( $MasterNodesML, AuthNodeAN$ )
2:    $SmallLst(len(SL)) \leftarrow 0$ 
3:    $AuthSmallLst(len(SL)) \leftarrow 0$ 
4:    $BlockChain \leftarrow 0$ 
5:    $Block \leftarrow 0$ 
6:   for all  $i, m \in SL$  do
7:      $SmallLst(i) \leftarrow generateSmallBlock(m)$ 
8:      $AuthSmallLst(i) \leftarrow SmallLst(i)$ 
9:     if ( $AuthState$ ) then
10:       $AuthNodeLst(i) \leftarrow m$ 
11:       $Block \leftarrow AuthState$ 
12:       $BlockChain \leftarrow Block$ 
13:    end if
14:  end for
15: end procedure

```

V. SIMULATION AND RESULT ANALYSIS

A. Simulation design

The development of the proposed framework and all the results are obtained using Python 3.9 on a Windows 10 computer with an Intel Core i7 processor running at 2.21 GHz and 16 GB of memory in Visual Studio Code experimental setting. Traditional PoW, DPoS, Ripple, PBFT and PoI consensus algorithms are also simulated, and their experimental results are compared with the RPoC algorithm in order to justify the experimental results.

B. Result analysis

This section compares the experimental results of PoW, DPoS, Ripple, PBFT, PoI, and RPoC consensus algorithms.

The provided results are an average of 10 simulation runs. Throughput efficiency, latency, and scalability are the three primary elements that we consider while evaluating the performance of the RPoC consensus algorithm.

- 1) **Throughput:** Throughput efficiency is expressed by TPS (Transactions Per Second), that can be measured by calculating how many transactions are completed w.r.t.time. It is used to measure how much processing a blockchain network is doing and how much scalability it has.
- 2) **Latency:** This measure is used to calculate the time it takes for a transaction to go from being sent to the network to being written to the ledger. This metric is calculated by comparing the time transactions from when they were submitted to the time they were validated and stored using their timestamps.
- 3) **Scalability:** This metric evaluates the algorithm's capacity to continue to perform properly when its size or volume is modified to fulfill a user request. The re-scaling is usually to a bigger size or volume.

1) *Throughput:* For measuring a system's efficiency, throughput is an important performance indicator. Starting with the initial transaction deployment time, throughput is defined as the number of executed transactions per second where the average throughput is the total throughput divided by the execution time. In this set of experiments, the TPS value of all algorithms was obtained and compared. The graph of average throughput of consensus algorithms for various numbers of transactions is shown in Figure 3. As the number of Tn_x grows from 1 to 100, the average throughput of all algorithms grows. PoI, DPOS, and the RPoC algorithms have the highest throughput under 100 Tn_x , whereas PoW, PoC, PBFT, and Ripple had the lowest. The average throughput of all algorithms drops after 1000 Tn_x as the number of Tn_x grows. Figure 4 presents a chart of the average throughput for DPOS, PoI, and RPoC with varying numbers of Tn_x . As the number of Tn_x grows, RPoC's average throughput always exceeds that of DPOS and PoI. The PoI algorithm's TPS ranged between 6500 and 7000, while the DPOS algorithm's fluctuated between 3000 and 5000 and the RPoC algorithm's oscillated between 10400 and 95000.

Figure 3: Average Throughput with varying number of transactions.

Figure 4: Average Throughput of PoI, DPOS and RPoC with varying number of transactions.

2) *Latency:* Latency is a key metric for assessing a network's performance and determining algorithm delays between nodes. A system with minimal latency is advantageous since it can return transaction processing results more quickly. For instance, the block processing time frame for the PoW algorithm is around 10 minutes, which means that a transaction is successfully written to the blockchain after an average waiting period of 5 minutes [40].

In this test, we investigate the average latency performance of all consensus algorithms with the same amount of Tn_x s, ranging from 10 to 10000. When dealing with a limited number of Tn_x , all algorithms have a low latency. For instance, while there are 100 Tn_x in the system, all algorithms have a low transaction processing latency, but as the number of Tn_x grows, the latency increases as shown in Figure 5. In comparison to all other algorithms, PoW's average latency dramatically increased after 100 Tn_x . PoW's average latency when dealing with 10,000 Tn_x is 1307.56s, which is 900 times higher than RPoC. The RPoC algorithm offers a more consistent transaction processing latency that does not vary significantly as the number of Tn_x grows. We compared the PoI, DOPS, and RPoC algorithms in Figure 6 to gain a clearer understanding. DPOS has a lower latency of 1.7s at 1000 Tn_x , compared to 1.5s at the same Tn_x for RPoC. In terms of latency performance, the DPOS and RPoC algorithms are competitive. In conclusion, the PoW and RPoC algorithms have significant latencies, whereas the DPOS and RPoC algorithms have shorter latency.

Figure 5: Impact of different number of transaction on latency.

Figure 6: Comparison of average latency among PoI, DPOS and RPoC.

3) *Scalability*: Scalability allows a system to respond dynamically depending on the latest settings. The scalability of a distributed system is determined by how the consensus mechanism allows for flexible joining and removal of nodes. The impact of increasing or decreasing the number of nodes during the operation of the consensus algorithm was investigated in the scalability test. Each system's TPS and transaction latency was investigated with various numbers of nodes. This analysis was applied on the PoI, DPOS and RPoC algorithms. Figure 7 shows the transaction processing performance of the system under varying numbers of nodes. It is obvious from the plot that the performance of the DPOS and PoI algorithms degrades with higher numbers of nodes. When there are 50 and 100 nodes in the system, the TPS values of the DPOS algorithm are around 4500 and 3500, respectively. PoI has a little better performance than DPOS. However, with 50 and 100 nodes, our proposed algorithm outperforms with 8500 and 6900 TPS, respectively.

Figure 8 compares the average latency of the DPOS, PoI, and RPoC algorithms with the same number of nodes. It is apparent that as the number of nodes increases, the average latency of each algorithm rapidly increases. The average latency for all three algorithms is identical for a set of 10 nodes. But as the number of nodes grows, the latency starts to increase for both the DOPS and PoI algorithms. On the other hand, when the number of nodes is increased from 50 to 100, the average latency of PoI and DPOS increases about 2 times faster than that of RPoC. In this set of experiments, PoI, DPOS, and the proposed algorithm all perform reasonably well. The performance of the consensus algorithms is not greatly affected by the growth in the number of nodes. The DPOS algorithm is found to have low scalability, but the other two algorithms have relatively high scalability.

Figure 7: Average throughput with varying number of nodes.

Figure 8: Average latency with varying number of nodes.

4) *Security analysis*: In this section we investigate the security of RPoC against a variety of malicious attacks. We suppose that the standard encryption mechanisms are in place and that At_k will not be able to crack them. Below, we look at different types of threats:

Double Spending Attack: When At_k tries to do a second Tn_x with the same data that was already confirmed on the network, this is known as a double-spend attack. In RPoC, storing new blocks does not require solving a challenge or expending resources, it is predicted that a large number of validators will work in parallel. Since RPoC has two consensus layers, the double spending attack will eventually be recognized by the network's large participating nodes. Secondly, the blockchain's distributed nature itself prevents double spending attacks. Due to the fact that all Tn_x are broadcast, validators will eventually receive blocks containing the double spent Tn_x and will be able to detect them during block verification. In this situation, At_k is removed from the validators list, and node details are sent to CA, preventing them from rejoining the network.

Sybil Attack: Sybil Attack is a sort of threat in which a At_k in the network deliberately operates several identities and compromising the legitimacy of reputation systems. As previously stated, RPoC is a two-layer consensus mechanism in which N_i work together with a CA. Every node, who wish to join the network requires a unique id issued by the CA. Furthermore, Val_a are required to provide documents in order to identify themselves, and their true identities are visible to the entire network. As a result, RPoC defends against this attack.

Denial-of-service attacks: In order to disrupt the operations of a targeted network node and make it unavailable, At_k sends a high number of Tn_x to blocks it. It is feasible to protect against this attack using the RPoC mechanism: As block generation rights can only be assigned to nodes that can withstand DoS attacks since network nodes are pre-authenticated. In the case of when a validator is offline for an extended period of time, it can be removed from the validating node list. RPoC safeguards against this attack while also taking advantage of the blockchain's distributed nature.

51% attack The 51% attack demands that At_k gains control of 51% of the network nodes in a network. Getting control of the nodes in our permissioned blockchain network is far more difficult than getting 51% of the network's processing

Table II: Comparative analysis against the existing consensus algorithms.

Algorithms	Scalability	Pros	Cons	Decentralized	Validator selection
Pow	Weak	Decentralization	High consumption	Decentralized	Computational power
DPOS	Strong	High throughput	More centralized	Centralized	wealth
Ripple	Strong	High throughput	Decentralization Issue	Centralized	voting
PBFT	Weak	High throughput	Poor scalability	Decentralized	voting
PoI	Strong	Transaction partnership	Decentralization Issue	semi decentralized	vested coins
PoC	Strong	Easy mining	Redundant disk space	semi decentralized	hard disk space

power with PoW consensus types. In our RPoC, it's far more hard to gain control of the nodes because the Val_a ' identities are at stake; if they engage in any malicious conduct and are exposed, they will be unable to rejoin the network and will lose their reputation in the business community. Furthermore, they are not alone in the consensus process; Val_s are also processing and verifying the Tn_x , making it nearly impossible for an At_k to gain control of a large network. The blockchain network's decisions are unaffected if an At_k has more processing power than the authority node.

VI. CONCLUSION

With the right consensus algorithm, blockchain technology can assist stakeholders in managing a SCM more effectively. Scalability, low latency, high throughput, and decentralization are all desirable characteristics of a successful consensus algorithm and have a direct impact on a blockchain's performance. However, many existing blockchain consensus protocols are incompatible with SCMs. In this paper, a new consensus algorithm, namely Reputation-based proof-of cooperation (RPoC), is proposed for blockchain-based SCM, that does not involve validators to solve any mathematical puzzle before storing a new block. The RPoC algorithm is an efficient and scalable consensus algorithm that selects the consensus node dynamically and permits a large number of nodes to participate in the consensus process. The algorithm decreases the workload on individual nodes while increasing consensus performance through allocating the transaction verification process to specific nodes. Furthermore, this paper highlights some current blockchain consensus algorithms and compares them to the proposed algorithm. Rigorous experiments against those existing consensus algorithms proves the efficacy of the RPoC consensus algorithm in terms of TPS, latency and scalability. To sum up this paper, we want to highlight that the content is not just limited to the RPoC incentive mechanism. In terms of economics, incentive design is a component that builds on the value proposition of a platform and constructs the system for which a platform's token will be built. In the future, we intend to work on the economics of blockchain-based SCM to add a credit incentive mechanism to consensus nodes.

REFERENCES

- [1] S. Nakamoto and A. Bitcoin, "A peer-to-peer electronic cash system," *Bitcoin*.—URL: <https://bitcoin.org/bitcoin.pdf>, vol. 4, 2008.
- [2] F. Longo, L. Nicoletti, A. Padovano, G. d'Atri, and M. Forte, "Blockchain-enabled supply chain: An experimental study," *Computers & Industrial Engineering*, vol. 136, pp. 57–69, 2019.
- [3] S. Saberi, M. Kouhizadeh, J. Sarkis, and L. Shen, "Blockchain technology and its relationships to sustainable supply chain management," *International Journal of Production Research*, vol. 57, no. 7, pp. 2117–2135, 2019.
- [4] X. Xu, D. Zhu, X. Yang, S. Wang, L. Qi, and W. Dou, "Concurrent practical byzantine fault tolerance for integration of blockchain and supply chain," *ACM Transactions on Internet Technology (TOIT)*, vol. 21, no. 1, pp. 1–17, 2021.
- [5] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei, and C. Qijun, "A review on consensus algorithm of blockchain," in *2017 IEEE international conference on systems, man, and cybernetics (SMC)*. IEEE, 2017, pp. 2567–2572.
- [6] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *International Journal of Web and Grid Services*, vol. 14, no. 4, pp. 352–375, 2018.
- [7] L. M. Bach, B. Mihaljevic, and M. Zagar, "Comparative analysis of blockchain consensus algorithms," in *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE, 2018, pp. 1545–1550.
- [8] F. Yang, W. Zhou, Q. Wu, R. Long, N. N. Xiong, and M. Zhou, "Delegated proof of stake with downgrade: A secure and efficient blockchain consensus algorithm with downgrade mechanism," *IEEE Access*, vol. 7, pp. 118 541–118 555, 2019.
- [9] L. S. Sankar, M. Sindhu, and M. Sethumadhavan, "Survey of consensus protocols on blockchain applications," in *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*. IEEE, 2017, pp. 1–5.
- [10] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich *et al.*, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the thirteenth EuroSys conference*, 2018, pp. 1–15.
- [11] P. Barrett, "Zilliqa technical whitepaper," 2017.
- [12] M. Zamani, M. Movahedi, and M. Raykova, "Rapid-chain: Scaling blockchain via full sharding," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer*

- and Communications Security, 2018, pp. 931–948.
- [13] D. Schwartz, N. Youngs, A. Britto *et al.*, “The ripple protocol consensus algorithm,” *Ripple Labs Inc White Paper*, vol. 5, no. 8, p. 151, 2014.
 - [14] Y. Lewenberg, Y. Bachrach, Y. Sompolinsky, A. Zohar, and J. S. Rosenschein, “Bitcoin mining pools: A cooperative game theoretic analysis,” in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, 2015, pp. 919–927.
 - [15] N. Bozic, G. Pujolle, and S. Secci, “A tutorial on blockchain and applications to secure network control-planes,” in *2016 3rd Smart Cloud Networks & Systems (SCNS)*. IEEE, 2016, pp. 1–8.
 - [16] H. L. Lee, V. Padmanabhan, and S. Whang, “Comments on “information distortion in a supply chain: The bullwhip effect”,” *Management science*, vol. 50, no. 12_supplement, pp. 1887–1893, 2004.
 - [17] S. A. Abeyratne and R. P. Monfared, “Blockchain ready manufacturing supply chain using distributed ledger,” *International Journal of Research in Engineering and Technology*, vol. 5, no. 9, pp. 1–10, 2016.
 - [18] Z. Gao, L. Xu, L. Chen, X. Zhao, Y. Lu, and W. Shi, “Coc: A unified distributed ledger based supply chain management system,” *Journal of Computer Science and Technology*, vol. 33, no. 2, pp. 237–248, 2018.
 - [19] J. Nation, “Walmart tests food safety with blockchain traceability,” *Abgerufen unter (10.09. 2018): <https://www.ethnews.com/walmart-testsfood-safety-with-blockchain-traceability>*, 2017.
 - [20] N. Hackius and M. Petersen, “Blockchain in logistics and supply chain: trick or treat?” in *Digitalization in Supply Chain Management and Logistics: Smart and Digital Solutions for an Industry 4.0 Environment. Proceedings of the Hamburg International Conference of Logistics (HICL)*, Vol. 23. Berlin: epubli GmbH, 2017, pp. 3–18.
 - [21] S. Malik, S. S. Kanhere, and R. Jurdak, “Productchain: Scalable blockchain framework to support provenance in supply chains,” in *2018 IEEE 17th International Symposium on Network Computing and Applications (NCA)*. IEEE, 2018, pp. 1–10.
 - [22] S. Wang, D. Li, Y. Zhang, and J. Chen, “Smart contract-based product traceability system in the supply chain scenario,” *IEEE Access*, vol. 7, pp. 115 122–115 133, 2019.
 - [23] Y. Wang, C. H. Chen, and A. Zghari-Sales, “Designing a blockchain enabled supply chain,” *International Journal of Production Research*, pp. 1–26, 2020.
 - [24] S. Rahmzadeh, M. S. Pishvaei, and M. R. Rasouli, “Integrated innovative product design and supply chain tactical planning within a blockchain platform,” *International Journal of Production Research*, vol. 58, no. 7, pp. 2242–2262, 2020.
 - [25] S. van Engelenburg, M. Janssen, and B. Klievink, “A blockchain architecture for reducing the bullwhip effect,” in *International Symposium on Business Modeling and Software Design*. Springer, 2018, pp. 69–82.
 - [26] Q. Lin, H. Wang, X. Pei, and J. Wang, “Food safety traceability system based on blockchain and epcis,” *IEEE Access*, vol. 7, pp. 20 698–20 707, 2019.
 - [27] L. Bader, J. Pennekamp, R. Matzutt, D. Hedderich, M. Kowalski, V. Lücken, and K. Wehrle, “Blockchain-based privacy preservation for supply chains supporting lightweight multi-hop information accountability,” *Information Processing & Management*, vol. 58, no. 3, p. 102529, 2021.
 - [28] G.-T. Nguyen and K. Kim, “A survey about consensus algorithms used in blockchain,” *Journal of Information processing systems*, vol. 14, no. 1, pp. 101–128, 2018.
 - [29] W. Martino, M. Quaintance, and S. Popejoy, “Chainweb: A proof-of-work parallel-chain architecture for massive throughput,” *Chainweb Whitepaper*, vol. 19, 2018.
 - [30] W. Li, C. Feng, L. Zhang, H. Xu, B. Cao, and M. A. Imran, “A scalable multi-layer pbft consensus for blockchain,” *IEEE Transactions on Parallel and Distributed Systems*, 2020.
 - [31] A. Litke, D. Anagnostopoulos, and T. Varvarigou, “Blockchains for supply chain management: Architectural elements and challenges towards a global scale deployment,” *Logistics*, vol. 3, no. 1, p. 5, 2019.
 - [32] J. Zou, B. Ye, L. Qu, Y. Wang, M. A. Orgun, and L. Li, “A proof-of-trust consensus protocol for enhancing accountability in crowdsourcing services,” *IEEE Transactions on Services Computing*, vol. 12, no. 3, pp. 429–445, 2018.
 - [33] D. Huang, X. Ma, and S. Zhang, “Performance analysis of the raft consensus algorithm for private blockchains,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 1, pp. 172–181, 2019.
 - [34] X. Ge, L. Yan, J. Zhu, and W. Shi, “Privacy-preserving distributed association rule mining based on the secret sharing technique,” in *The 2nd International Conference on Software Engineering and Data Mining*. IEEE, 2010, pp. 345–350.
 - [35] X. Zhu, Y. Li, L. Fang, and P. Chen, “An improved proof-of-trust consensus algorithm for credible crowdsourcing blockchain services,” *IEEE Access*, vol. 8, pp. 102 177–102 187, 2020.
 - [36] Y. P. Tsang, K. L. Choy, C. H. Wu, G. T. S. Ho, and H. Y. Lam, “Blockchain-driven iot for food traceability with an integrated consensus mechanism,” *IEEE access*, vol. 7, pp. 129 000–129 017, 2019.
 - [37] H. Yusuf, I. Surjandari, and A. M. M. Rus, “Multiple channel with crash fault tolerant consensus blockchain network: a case study of vegetables supplier supply chain,” in *2019 16th International Conference on Service Systems and Service Management (ICSSSM)*. IEEE, 2019, pp. 1–4.
 - [38] X. Zhang, W. Ruizhen, M. Wang, and L. Wang, “A high-performance parallel computation hardware architecture in asic of sha-256 hash,” in *2019 21st International Conference on Advanced Communication Technology (ICACT)*. IEEE, 2019, pp. 52–55.
 - [39] F. Gai, B. Wang, W. Deng, and W. Peng, “Proof of reputation: A reputation-based consensus protocol for peer-to-peer network,” in *International Conference on*

Database Systems for Advanced Applications. Springer, 2018, pp. 666–681.

- [40] T. Laurence, *Introduction to Blockchain Technology: The many faces of blockchain technology in the 21st century.* Van Haren, 2019.

Aaliya Sarfaraz (Graduate Student Member, IEEE) is currently pursuing the Ph.D. degree in School of engineering and Information Technology with the University of New South Wales, Canberra, ACT, Australia. She received the B.E. degree from the Balochistan University of Engineering and Technology in 2013, and in 2015 M.S. from University of Seoul, South Korea. Her research interests focus on Blockchain, supply chain, big data, including the adoption and use of these technologies.

Ripon K. Chakraborty (Member, IEEE) is the Deputy Director (acting) of the Capability Systems Centre (CSC) and Lecturer on Decision Analytic & System Engineering at the School of Engineering and Information Technology, the University of New South Wales (UNSW Australia), Canberra. As an educator and active researcher in his field, his focus is to deliver authentic teaching by pursuing real-life and practical knowledge to diversified students. Inarguably, his research involvement in the discipline of 'Decision Analytics', 'Applied Operations Research', 'Systems Engineering Project Management', 'Industry 4.0' aids him to expose knowledge gaps in the body of knowledge (e.g., he has got more than 130 internal publications at his credit).

Daryl L. Essam received his B.Sc. degree in computer science from University of New England, Australia in 1990 and his Ph.D. degree from University of New South Wales, Australia, in 2000. Since 1994, he has been with the Canberra campus, UNSW, where he is currently a Senior Lecturer. His research interests include genetic algorithms, with a focus on both evolutionary optimization and large scale problems.