IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Development Of An Enhanced Proof Of Stake Consensus Mechanism In Blockchain

[1]Onuma Divine, A., [2]Nwiabu Nuka D., [3]Matthias Daniel

[1]Mr., [2]Dr., [3]Dr.

[1]Computer Science Department,

[1]Rivers State University (RSU), Port Harcourt, Nigeria

Abstract: The distributed consensus mechanism is the backbone of a blockchain network, and it plays a key role in ensuring the network's security, integrity, and performance. Most current blockchain networks have been deploying the proof-of-stake consensus mechanisms, in which the consensus is reached based on the number of tokens each validator holdss. However, this mechanism has a number of drawbacks, including the fact that it relies on a single point of failure, which makes it vulnerable to attack. Our Enhanced PoS consensus algorithm contains three phases: consensus process, consensus group formation, and node trust assessment. Applying the EigenTrust concept to the blockchain network lessens the impact of rogue nodes. The data reveals that O((dN)2) (0d1) is the number of messages of our Enhanced proof of stake sent across nodes. Our Enhanced proof of stake has O(N2) communication complexity and a low view change probability. The result of the throughput analysis of our proposed algorithm shows there is a steady increase in the transaction commit rate which was due to the proposed algorithm's minimal amount of communication during the transaction validation process, thus making the time required for transaction confirmation lesser than the traditional PoS. Also, the proposed algorithm discovers and deletes the problematic nodes unlike the traditional PoS algorithm which does not alter the rate of faulty nodes throughout execution. As a result, the rate of problematic nodes will steadily decline and the proposed algorithm can be seen as more secured and reliable in a blockchain network. In terms of node communication, the proposed model uses fewer resources and requires little inter-node communication. The system's efficacy, efficiency, and scalability are also explored, and the constructed method is contrasted with alternative consensus techniques. Finally, a reliable consensus based on the EigenTrust Model can be achieved by reducing the number of consensus nodes, which will increase consensus effectiveness and lower the communication cost. Thus, the experimental results show that the proposed system works better than the traditional PoS method, resulting in a 25% decrease in transaction confirmation time.

Index Terms **– Blockchain, Byzantine Fault Tolerance(BFT), Consensus Algorithm, Mining Processing, Proof-of-Work, Proof-of-Stake.**

I. INTRODUCTION

Blockchain, the technology behind Bitcoin defined by[45], is basically a distributed, immutable record that is updated in real time by untrusted nodes. According to Yuan, it achieves transparency, credibility, integrity, and immutability by combining technologies like distributed ledger, encryption, consensus protocol, and smart contracts[58].

Almost without exception, blockchains are designed to be decentralized, operating as a distributed network of computer nodes that maintains a digital ledger. Consequently, blockchain technology enabled the development of trustless economic systems in which transparent and trustworthy[37] financial transactions could be conducted without the use of intermediaries. Existing banking and payment systems that depend heavily on trust are giving way to cryptocurrencies as a viable alternative.

Public, consortium, and private blockchain[6] deployment options are the three most common types. Vitalik[12] and [59] categorize consortium and private blockchains as permissioned blockchains. As shown in Ethereum [12] and Bitcoin, any node with internet connection may view a public blockchain and take part in

mining, as well as freely perform read, write, and commit operations [45]. Multichain[27] is an example of a private blockchain in which all rights, including read and write, are centralized at a single node. Consortium blockchains, on the other hand, are multi-centralized systems where a predetermined group of nodes decides which nodes have read/write access and which do not.

According to Wang [56], blockchains' scalability and efficiency may be affected by the consensus protocol[7][44][45][58]. The consensus mechanism of a blockchain defines how skeptic nodes may come to terms on a regular basis to add a new block to the chain. The consensus protocol[22][56], as proposed by Bano [7] and Gramoli[26] , has to include the following core features:
(a) uniformity: all regular nodes should agree on the same block;
(b) validity: a consensus node must put forth a determined block;
(c) to maintain liveness (also known as termination), every healthy node must reach a conclusion about a block.

The blockchain consensus's[45][59][22] safety property is defined by the consistency and validity qualities. Existing blockchain consensus algorithms can be loosely categorized as Proof-of-X (PoX) consensus algorithms and Byzantine Fault Tolerant (BFT) consensus algorithms based on the various deployment types of blockchains [54]. The PoX consensus algorithms, such as PoW [45] and PoS [34] consensus algorithms are appropriate for public blockchains with great computing power and poor efficiency. Common BFT consensus algorithms for consortium blockchains include Practical BFT (PBFT)[14], Scalable BFT[8], Zyzzyva[35], and HoneyBadgerBFT[43]. Nevertheless, the majority of BFT-type consensus algorithms currently available have poor scalability; for example, the performance of the PBFT consensus algorithm decreases significantly as the number of nodes increases, and they have a low Byzantine fault-tolerant rate. In addition, the failure of the main node will initiate the view change procedure, which will have an effect on the overall consensus process. This paper is aimed at solving Proof of Stake efficiency and scalability problem using EigenTrust model.

II. RELATED WORKS

In 2017, [33], presented the Ouroboros PoS protocol. A dynamic committee is selected according to the stake allocation, making this system entirely stake-based. Epochs are used in the protocol to denote distinct time intervals. In order to create the seeds used by the FTS algorithm, the committee members engage in a three-stage coin-tossing procedure at the beginning of each epoch.

In Ouroboros[33], the protocol's stakes are used to elect a dynamic committee. To produce seeds for the FTS algorithm, the protocol uses a three-phase coin-tossing approach involving committee members. The algorithm generates coin indices, and the owners of certain coins move up the ranks to become epoch-level leaders and committee members. The protocol is safe against close-range, far-range, and grinding assaults. But it's still susceptible to 51 percent attacks, and bribery attacks aren't even mentioned. When compared to PoW-based networks, Ouroboros has lower transaction confirmation times, higher transaction throughput, and lower energy consumption. It is widely used among digital currencies like Cardano and Sp8de because of the formal definitions and robust theoretical underpinning that back up its security and incentive compatibility.

Similarly to Ouroboros, the Chains-of-Activity (CoA) protocol[21] employs the FTS algorithm to choose leaders, but with a different seed. CoA works by arranging blocks into sets of length l, with the hash of each set being used as a starting point for the FTS procedure. Each period has its own set of leaders, and each new block requires a deposit. If the block is legitimate, the prize is distributed, but the deposit is forfeited. To further stabilize the chain and stop lengthy adversarial forks, the CoA protocol[21] includes checkpoint blocks that add T blocks to the chain. The protocol is safe against assaults such as grinding, range, and no-stake. Checkpoint blocks prevent long-range attacks by ensuring that no block between the first and second most recent checkpoint blocks may change, while the deposit scheme protects against double-spending and bribery attempts.

Since only one block is generated per round in the CoA protocol[21], it requires significantly less power than PoW methods. CoA also offers a high transaction throughput of 40 transactions per second and a short transaction confirmation time of about 6 minutes [9][17]. While important to network security, the study overlooks issues like incentive compatibility and the importance of network synchronization and adversary tolerance thresholds. The cryptocurrency Tezos (https://tezos.com) is partially based on the CoA protocol.

The Ethereum network created the Casper protocol to facilitate the switch from PoW to PoS, and it can run on top of preexisting PoW systems. Casper does not intervene in the selection of a leader as other PoS protocols do. Instead, it utilizes a dynamic committee to generate checkpoint blocks at set intervals, generally every 100 blocks using a Byzantine-Fault-Tolerance (BFT) protocol[14]. The last justified checkpoint is used to determine whether or not a block is complete. Each validator must make a deposit in order to get voting rights that are directly related to the size of their deposit. The amount of the deposit is lowered if fraudulent activity is discovered.

Casper[11] has been shown secure on a partly synchronous network with a presumption of honesty from at least two-thirds of the validators. By instituting a withdrawal delay, it also deals with the problem of dynamic stake allocation and long-range assaults. This delays the period before validators may withdraw their investment, reducing the likelihood that they would engage in fraudulent activity and subsequently quickly cash out their share. The underlying blockchain architecture is designed[54] to deal with other security concerns. Casper's overarching goal is to make the Ethereum network's consensus process more secure, scalable, and efficient.

In conclusion, the Casper protocol[11] provides a means to go from PoW to PoS while also bolstering the safety of the underlying chain. However, the study does not elaborate on the incentive mechanism, and its performance is contingent on the underlying PoW method. The protocol employs a withdrawal delay to handle dynamic stake distribution and long-range assaults, and it has been demonstrated safe in a partly synchronous network with 2/3 honest validators. Casper, Ethereum's forthcoming PoW blockchain protocol, is currently under development.

The Algorand protocol[4][24] uses a committee-based mechanism, similar to Ouroboros, to choose committee members and leaders. However, Algorand employs a cryptographic sortition technique based on stake distribution rather than the FTS algorithm. Each consensus node is given a range of hash values proportional to its stake, and a node is chosen as a leader or committee member if the hash value falls within that node's range. By hiding the identity of the winning node until after it has presented proof, the cryptographic sortition approach keeps it safe from potential threats. The VRF's first seed is generated using a distributed random number generator and reused to generate a new seed for the next iteration. The committee is in charge of adding voting blocks to the chain at the end of each round, therefore the leader selection procedure is not crucial to security.

Algorithand can keep running as long as there is a synchronous phase that follows the asynchronous one. Assuming honest players account for at least 51% of the total stake, this premise demonstrates that Algorand[4] is secure. Fork-related attacks, such as double spending, long-range, nothing-at-stake, and bribe assaults, are mitigated by the committee's decision to finish each block. Using the private key of a node and the seed to choose a leader reduces the risk of a grinder attack since the adversary must interfere with the selection process at the same time.

Energy-efficient and capable of processing up to 875 transactions per second, Algorand employs a cryptographic sortition mechanism to pick leaders and committee members based on stake distribution. In contrast to PoW methods, Algorand[4][50] offers instant finality, drastically cutting confirmation times for transactions to, say, 20 seconds. However, it is concerning that the paper does not provide a clear description of the incentive mechanism. Several digital currencies, such as Algorand and Arcblock, have adopted Algorand as their underlying technology.

In order to confirm blocks, the Tendermint protocol, created by Jae Kwon[36], employs a Byzantine Fault Tolerance (BFT) consensus mechanism. In order to cast a vote, validators must first make a deposit. A deterministic round-robin procedure that takes into account the weight of each validator's vote is used to pick the block proposer. If a proposer is selected, the proposed block might comprise transactions. Similar to Algorand, validators then cast votes to confirm the proposed block, bringing about swift block and transaction finality. Block incentives are given out to validators to encourage them to take part in consensus, whereas malicious action results in the loss of deposits.

As long as at least two-thirds of the voting power is held by honest nodes, the Tendermint protocol—which employs a BFT voting method for block confirmation—has been shown to be safe in the situation of a partly synchronous network. Tendermint[36], like Algorand, avoids forks, which reduces the likelihood of fork-related attacks. The dynamic stake allocation and the round-robin leader selection mechanism are not discussed in the article.

Tendermint is more efficient than PoW protocols because to its single-block-per-round generation mechanism. The protocol, like Algorand, provides high transaction throughput (up to 800 Tx/s) and quick confirmation times (often about 1 second). However, there is no formal definition or theoretical basis for the protocol, and the incentive mechanism has not been thoroughly explored. Tendermint is now utilized to back up the blockchain database BigchainDB[36] and the cryptocurrency network Ethermint[13].

III. Design

Because of its decentralized nature and anonymity, blockchain technology is often used in P2P systems[14][40]. However, due to its special characteristics, it is susceptible to assault from dishonest nodes. Therefore, studies have centered on finding ways to lessen the influence of dishonest nodes. To deal with this, a novel multi-stage consensus approach, Enhanced-PoS, based on the EigenTrust model[10][31], is developed, as shown in Figure 1. There are three steps to this procedure: determining which nodes can be trusted, forming a consensus group, and carrying out the consensus itself. The Eigen-Trust model computes a node's global trust value, which is then utilized to construct the consensus group of nodes with the highest average trust. In a large-scale network, the consensus process runs more smoothly because fewer nodes need to be involved in the consensus group. With each new block added to the blockchain, the global trust value is dynamically adjusted, enabling Enhanced-PoS to start a new round of transactions between nodes.

3.1 EigenTrust Model

The EigenTrust model[31] is a trustworthy model for assessing trust in a P2P environment. It is efficient in evaluating the trustworthiness of each node while maintaining security[51] by monitoring malicious nodes[32][41]. The model tracks the transaction history between nodes to compute a distinct global trust value for each node in the system. In our paradigm, the global trust value Ti can be calculated by taking

$$T_i = C_{1i}T_1 + \ldots + C_{ni}T_n , \quad (1)$$

where T_i is the global trust value of $node_i$ and C_{ij} is the local trust value of $node_i$ to $node_j$.

Node connections may be easily classified into two groups: those that include transactions and those that do not. Three different types of trust values [31][56] will be implemented in the EigenTrust model using node*a* as an example.

1) Indirect trustworthiness Transactions between node*a* and node*b* may be assessed in a *cab*. We denote the number of successful and unsuccessful exchanges between node*a* and node*b* using the formula *Sab* = sat(node*a*, node*b*) unsat(node*a*, node*b*). Then

$$C_{ab} = \frac{max(S_{ab}\,0)}{\sum_x max(S_{zx}\,0)^2} \quad (2)$$

Where x = b and c

2) Suggested reliability rating *Cad*: it may be compared without any transaction between two nodes (a and d). It is founded on transitive trust, and its worth is proportional to that of direct trust. The *Cad* is administered by

$$C_{ad} = \sum_k C_{ak}C_{kd} \quad (3)$$

Where k = b and c

3) Global trust value *Ta k* +1: The algorithm analyzes nodes based on their quantifiable level of trust. The overall trust value of a node, such as node*a*, is determined by combining the trust values of all nodes in the network and incorporating the current global trust value of each node. This results in an evaluation metric that reflects the level of trustworthiness of node*a*.

Assumptions

a) In Enhanced-PoS, behavior consistency is based on the assumption that nodes with higher global trust values are more reliable when it comes to personal gain. This implies that nodes with higher global trust ratings are rational and are more likely to act honestly.

b) Limited transaction time: To ensure the effectiveness of the EigenTrust model, it is crucial to consider time in the transaction analysis. The stability of the transaction set is necessary, and the number of transactions within a fixed time frame, such as an hour or a day is typically analyzed.

3.2 Algorithm of Enhanced Proof-of-Stake

Algorithm 1. DivideNodes

Input: $node_i$, node set $Nodes$
Output: $TxNodes$, $NonTxNodes$

```
TxNodes←∅, NonTxNodes←∅;
for node_j ∈ Nodes do
    if node_j trades with node_i then
        TxNodes←node_j;
    else
        NonTxNodes←node_j;
    end
end
```

Algorithm 2. CalcTxNodeTrust.

Input: $node_i$, $TxNodes$ of $node_i$
Output: Direct trust value C_{ij}

$C_{ij} \leftarrow 0$;
for $node_j \in TxNodes$ **do**
  $S_{ij} = sat(i,j) - unsat(i,j)$;
  $S_{total} = \sum max(S_{ij}, 0)$
end
if $S_{total} = 0$ **then**
  set $C_{ij} = \frac{1}{N}$; ▷N is the size of Nodes
else
  for $node_j \in TxNodes$ **do**
    $C_{ij} = \frac{max(S_{ij}, 0)}{S_{total}}$;
  end
end

Algorithm 3. CalcNonTxNodeTrust.

Input: $node_i$, $TxNodes$, $NonTxNodes$ of $node_i$
Output: Recommended trust value C_{ij}

$C_{ij} \leftarrow 0$;
Find the transaction paths between $node_i$ and $node_j$;
for $node_j \in NonTxNodes$ **do**
  if $node_k \in TxNodes\ of\ node_i$ & $node_k \in TxNodes\ of\ node_j$ **then**
    $C_{ij} = \sum_k C_{ik} C_{kj}$;
  else
    Compute C_{ij} iteratively;
  end
end

Algorithm 4. CalcGlobalTrust.

Input: $node_i$, node set *Nodes*
Output: Global trust T_i of nodei
1 $T_i \leftarrow 0$;
2 **for** $node_j \in Nodes$ **do**
3 $T_i = \sum C_{ji} T_j$;
4 **end**

Algorithm 5. getConsensusGroup.

Input: Node set *Nodes*, Global trust set T, a constant percentage of nodes d $(0<d\leq 1)$
Output: ConsensusGroup
1 ConsensusGroup $\leftarrow \varnothing$;
2 Sort *Nodes* by T;
3 **for** $node_i \in Nodes$ **do**
4 **if** T_i *is in the top* d **then**
5 Add $node_i$ into ConsensusGroup;
6 **else**
7 Exclude $node_i$ from ConsensusGroup;
8 **end**
9 **end**

Algorithm 6. getPrimaryGroup.

Input: ConsensusGroup, fixed proportion $m(0 < m \leq 1)$
Output: PrimaryGroup
1 PrimaryGroup $\leftarrow \varnothing$;
2 **for** $node_i \in ConsensusGroup$ **do**
3 **if** $node_i$ *with the global trust value in top* m **then**
4 Add $node_i$ to PrimaryGroup;
5 **else**
6 Exclude $node_i$ from PrimaryGroup;
7 **end**
8 **end**

3.3 Abstract view of the Enhanced Proof-of-Stake

3.3.1 Node Trust Evaluation

In this step, we will go through how to use the EigenTrust model to assess the trustworthiness of individual nodes. The first step is to have each node in the network have its global trust value equal to 1/N, where N is the total number of nodes[12]. Algorithms 2 and 3 calculate the direct trust value and the suggested trust value between nodes based on their transaction linkages. Nodes with direct transactions are assigned a direct trust value, whereas nodes without direct transactions but with a common transaction partner are assigned a suggested trust value. After that, every node forms a local trust connection, and the trust scores are calculated precisely. Algorithm 4 is used to calculate the product of a node's local trust value and the corresponding global trust value of other nodes in order to achieve the global trust value that appropriately reflects a node's trust level. To facilitate the removal of low-credit nodes for consensus, the global trust value is dynamic and updates with each block.

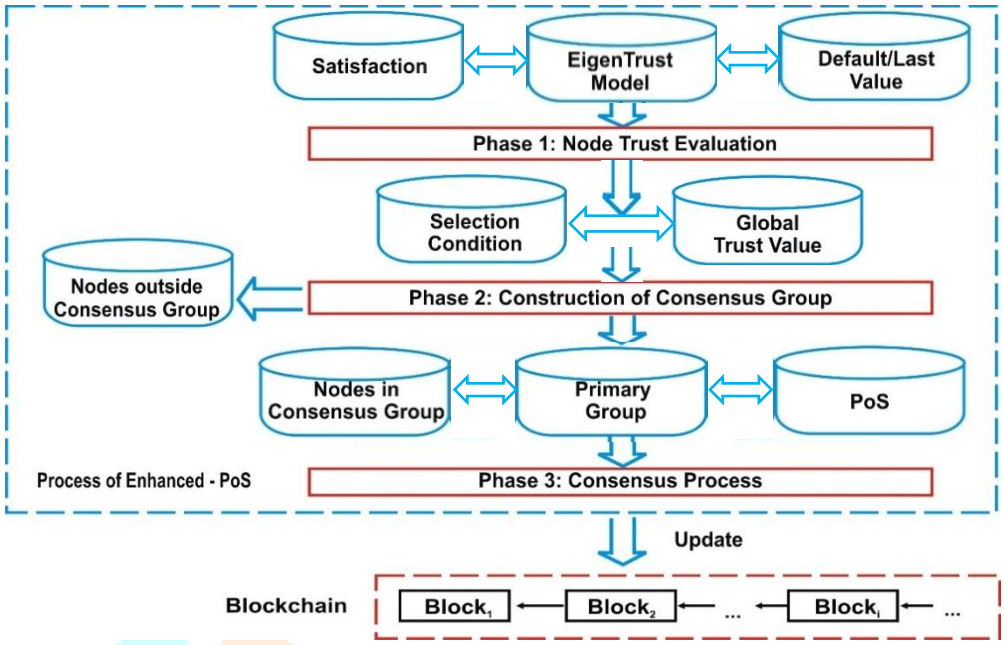

Figure 1: Abstract view of the Enhanced Proof-of-Stake

3.3.2 Construction of Consensus Group

Developing a blockchain's consensus group is the topic at hand. As was said in the introduction, the EigenTrust model[31][32] calculates an overall trustworthiness score for each node in the network. When it comes to making financial decisions, nodes with higher global trust ratings are preferred since they are more likely to consistently exhibit the desired behaviour. As an alternative to incorporating all blockchain nodes in the consensus group, choosing just those with greater trust levels may improve the efficiency and scalability of blockchain consensus algorithms. By removing nodes with less credits, the Byzantine fault-tolerant rate may be increased. The blockchain consensus process may be sped up by lowering the number of nodes in the consensus group, which in turn reduces the quantity of message transmission needed.

We define the requirements for forming a blockchain consensus group to accomplish this goal. A global trust threshold may be established, and once a node's trust value rises beyond that barrier, it will be included to the blockchain's consensus group. However, since nodes' global trust values change over time, this could lead to a widely varying number of blockchain consensus nodes, which could be detrimental to the reliability of blockchain consensus. Our proposed solution includes picking a predetermined percentage of nodes with higher global trust scores to form the consensus group. The method is explained in further depth in Algorithm 5.

In Algorithm 5, we see how to build a trust-based blockchain consensus group. First, a consensus group with no members is formed, and then the nodes are ranked by their global trust ratings. In order to join the consensus group, a node must have a global trust value inside the top d, where d is a constant proportion of the set of all nodes. The consensus group does not include nodes with low global trust levels. By using this strategy, we may expedite and enhance the blockchain's consensus procedure by choosing just the nodes with the highest global trust values. Only the nodes that are part of the consensus group can take part in the consensus process.

3.3.3 Consensus Process

To increase the network's fault tolerance, we propose an Enhanced Proof of Stake (PoS) consensus mechanism for the blockchain. When a ConsensusGroup has been established, its members will vote to create a new block[16]. As we have established, changing someone's worldview is a difficult and time-consuming process that should be avoided wherever feasible. As a result, we choose a small subset of the ConsensusGroup's highest-trust nodes to act as the main group, which will take over for the primary node in the event of Byzantine behaviour or a fail-stop error. This main team will develop the new building block, record its specifications, and check their correctness. Algorithm 6 demonstrates how the total trust value of the nodes is used to choose the main group.

By incorporating the idea of a primary group, our Enhanced-PoS consensus algorithm attempts to mitigate the danger of a view change process brought on by the failure of a single main node or its Byzantine behaviour. There are four steps to this algorithm: group process, pre-preparation, preparation, and response.

1) During the group process phase of Enhanced-PoS, a primary group node will create a block of transactions and share it with the other primary group member nodes for verification. Once the block is validated, it will be temporarily recorded with the same view by each primary group member node. This allows for easy replacement of a failed primary group node without causing a view change process.

2) During the pre-prepare phase, the main group transmits the pre-generated block and group signature to the replica nodes of the consensus group[14][19]. The key group members' privacy is protected and the likelihood of assaults that might cause a shift in perspective is lessened by the usage of the group signature. Any node can verify a group signature without knowing which primary group member generated it.

3) During the prepare phase, replica nodes conduct the transactions in the specified sequence to verify the newly created block and compute the block's hash. The replica nodes will exchange signed preparation messages if the block hash is the same as the current hash. If there are f Byzantine nodes in the consensus group[19], then the consensus nodes won't act until they've received prepare messages from at least 2f of them. The consensus nodes will respond to the client after they have received prepare messages from at least 2f nodes, signalling that the block has been authenticated.

4) When the client receives f+1 identical reply messages, the pre-generated block is added to the blockchain. After that happens, local copies of the blockchain are updated on all participating nodes. Byzantine Fault Tolerance uses a pre-prepare and prepare phase to arrange requests within the same view, and a commit phase to guarantee the full ordering of requests across views. By decreasing the possibility of view change with the main group, our Enhanced-PoS streamlines the Byzantine Fault Tolerance[19] operation method. As new transactions are confirmed and the global trust value of all nodes changes, the primary group's member nodes will morph dynamically throughout the consensus process to produce valid blocks. Each node's global trust value must rise before it may take part in the consensus procedure.

IV. Results and Discussion

In Figure 2, the graph depicts the relationship between time and throughput in the classic PoS algorithm and the suggested technique, with the average throughput gap increasing with time.

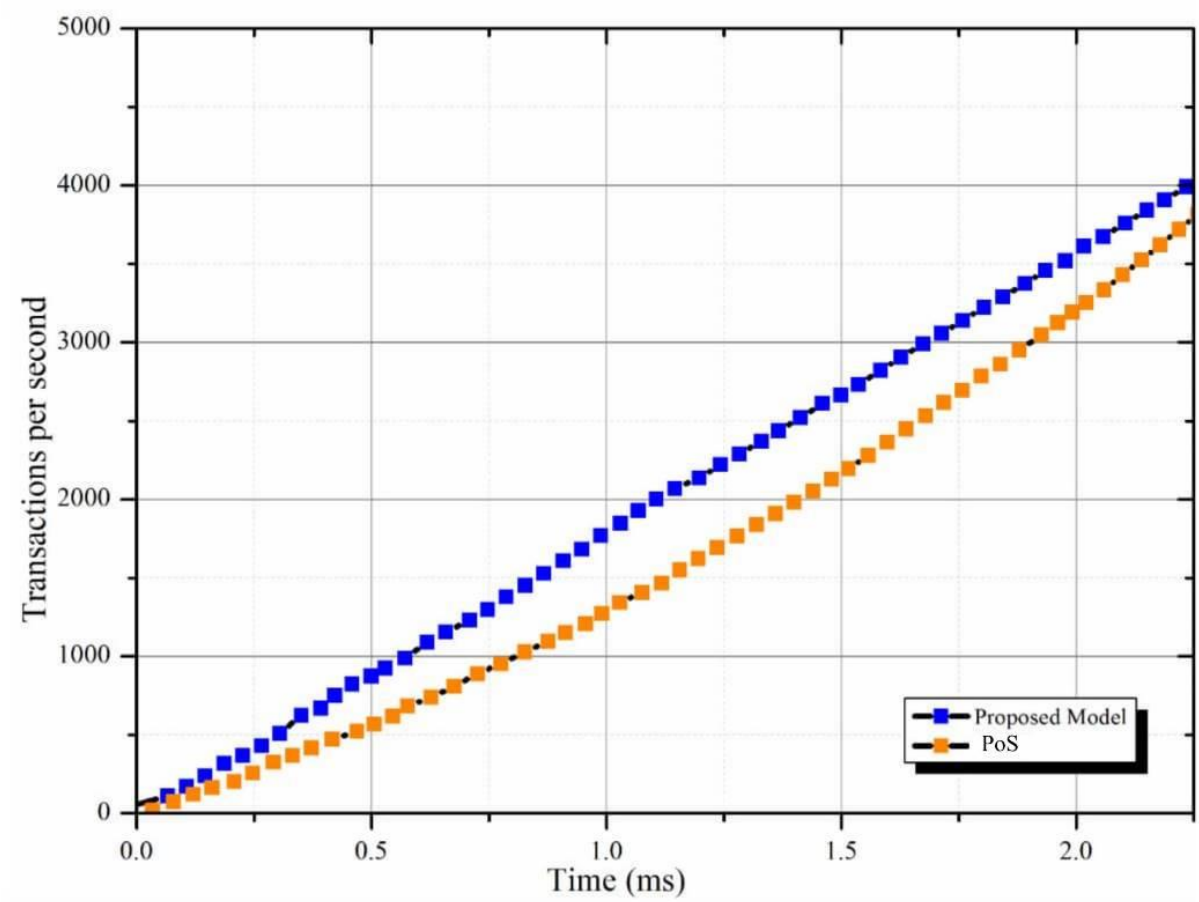

Figure 2: Relationship between TPS and Time

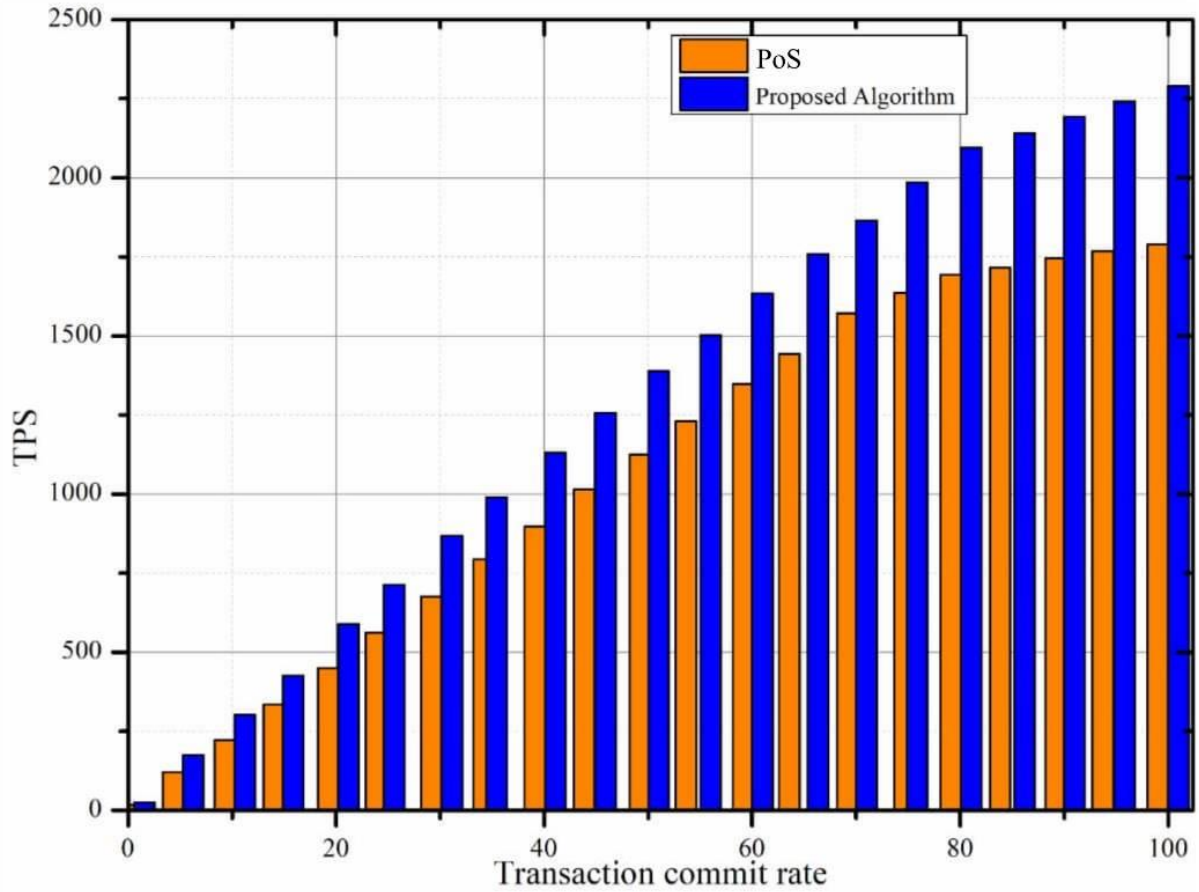

Figure 3: Throughput Analysis

Figure 3 depicts the proposed algorithm's throughput as a function of the nodes' transaction rates, apart from the request rates. The Enhanced-Proof of Stake consistently improves throughput over PoS. The commit rate steadily climbs as the number of transactions approaches 1250. The suggested approach has a shorter transaction confirmation time than PoS does since it requires less communication throughout the validation phase.

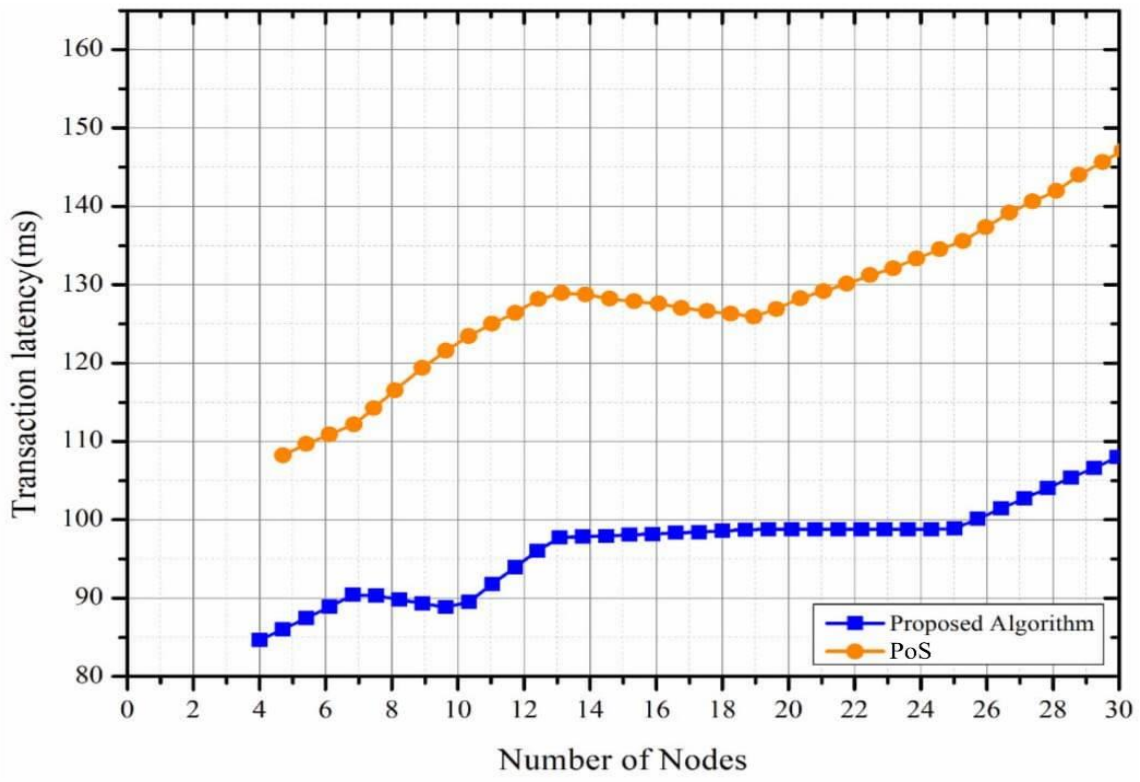

Figure 4: Transaction Latency

See Figure 4 The proposed technique has a lower latency than the traditional PoS algorithm, proving that consensus may be reached amongst nodes more quickly. Separating nodes into consensus and non-consensus nodes is crucial for achieving low latency, since it reduces the total number of nodes involved in the consensus process. The simulations use any number of nodes from two to thirty. As the number of nodes in a network expands, the transaction latency in both PoS and the proposed method also rises. The need of better internal communication throughout the verification process is connected to this. We also put the proposed method to the test by looking at how different block sizes affect latency while accounting for varying quantities of transactions per block. It takes longer for a block to spread since there are more transactions in it, and verification adds even more time to the process.

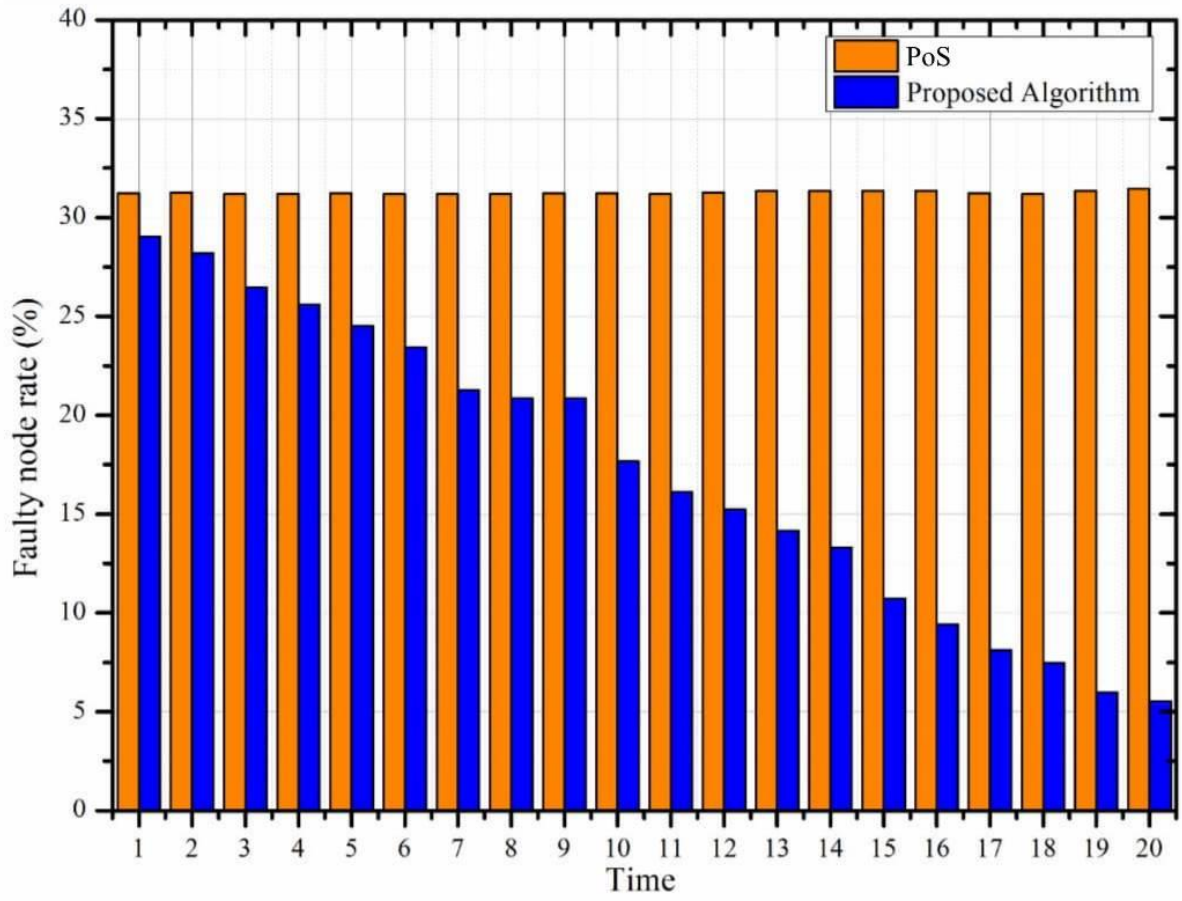

Figure 5: Faulty Node Rate

Figure 5 shows that although the proposed method would identify and remove the defective nodes, the standard PoS algorithm does not affect the rate of faulty nodes during execution. The proposed architecture offers a more secure and dependable blockchain network as a consequence of the decreasing rate of problematic nodes.

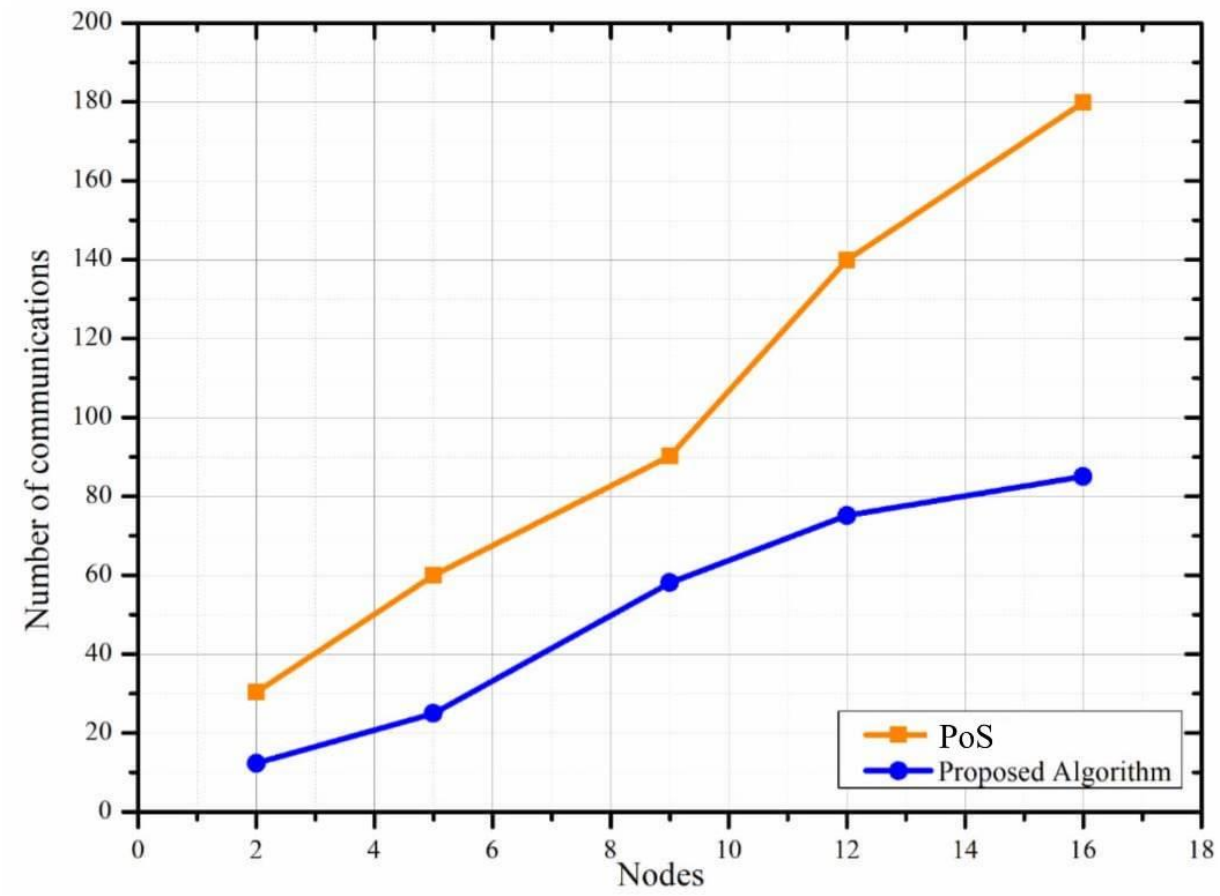

Figure 6: Communication between Nodes

The difference in communication time between PoS and the suggested approach is seen in Figure 6. The suggested method generally has low communication costs. This is mostly because less information may travel between nodes once they have been split into two lists. In addition, only a reliable consensus node is engaged in the transaction validation process, which greatly reduces the need for further communication. The total number of communications for both approaches grows steadily as the number of nodes goes from 2 to 4, then 6 to 12, then 16 to 18, and so on. In general, the suggested method makes less use of resources and calls for less communication between nodes. The communication durations are empirically examined for different node counts and compared with standard PoS approaches, and the results show how the number of messages required to obtain consensus affects the system's performance and deployment resource needs.

4.1 Comparison of Enhanced PoS with other Blockchain Consensus Mechanism

Our Enhanced-PoS is compared to several other popular consensus techniques in Table 4.2. Both PoS and Tendermint employ a three-phase commit consensus, which reduces their communication complexity to O. (N2). Zyzzyva simplifies PoS by using the main requests and the immediate response from the client, which has a communication complexity of O(N). Like other crash-tolerant replication methods, such as Raft and Paxos, XFT has a communication complexity of O(N). Because the HoneyBadgerBFT has a communication complexity of roughly O(N2 + N3logN), we use our Enhanced-PoS to narrow down the set of consensus nodes to a more reliable subset.

Table 1: Analysis of the Enhanced-Proof of Stake algorithm in comparison to other existing algorithms

Consensus Algorithm	Nodes participation in the Consensus Process	Node Reliability Assessment	Network Type	Node Election Fairness	Communication Overhead
PBFT	All nodes	No	Permissioned	No	High
PoW	All nodes	No	Permissionless	No	High
PoS	All nodes	No	Permissionless	No	High
DPoS	All nodes	No	Permissionless	No	High
Proposed Algorithm	All nodes	Yes	Permissionless	Yes	Low

Table 2: Our Enhanced-PoS is compared to other consensus algorithms.

	PoS	Zyzzyva	HoneyBadgerBFT	Tendermint	XFT	Enhanced-PoS
Communication Complexity	$O(N^2)$	$O(N)$	$O(N^2 + N^3 \log N)$	$O(N^2)$	$O(N)$	$O(N^2)$
Byzantine fault tolerance	$\frac{N-1}{3}$	$\frac{N-1}{3}$	$\frac{N-1}{3}$	$\frac{N-1}{3}$	$\frac{N-1}{3}$	$\frac{N-1}{3}$
Primary node	Single	Single	Single	Single	Single	Group
View change probability	High	High	High	High	High	Low

Our Enhanced-PoS has a theoretical communication complexity of O((dN)2) (0 d 1). The communication complexity of a blockchain network is O(N2) when d = 1, indicating that all nodes are members of the consensus group. Our Enhanced-PoS and PoS systems both have varied degrees of fault tolerance and can deal with Byzantine nodes, which may behave arbitrarily. A Byzantine fault-tolerant percentage of PoS extensions with the same Byzantine fault-tolerant design include Zyzzyva, HoneyBadgerBFT, and Tendermint. Tolerating up to N21 Byzantine nodes, XFT suggests that a node cannot control the network and Byzantine nodes at the same time. When we've already established, as d increases, our Enhanced-Byzantine PoS's fault-tolerant rate decreases. Byzantine fault tolerance is quite low at d = 1, much like PoS. Enhanced-PoS uses a primary group to increase resilience and decrease the possibility of the main node showing Byzantine behaviour by means of mutual monitoring, while most consensus algorithms in Table 2 use a single node as the primary node. This method not only minimizes the chance of the view change process, but also protects the privacy of nodes in the main group and mitigates the danger of group signature attacks.

4.2 Discussion

The experimental results show that the suggested PoS consensus algorithm works better than the classic PoS method, resulting in a 25% decrease in transaction confirmation time, and that transaction latency continuously rises with the number of nodes in the network. The proposed method is compared to the most popular consensus algorithms in Table 4.1. This analysis takes into account a wide range of criteria, including the number of nodes participating in the consensus process, the node reliability evaluation, the network type, the fairness of node elections, and the communication overhead.

4.2.1 Effectiveness

Combining Proof of Stake with the EigenTrust model, the Enhanced-PoS consensus method is a multi-step process. There are a number of ways in which the suggested technique diverges from PoS. As a first step, it employs the EigenTrust model to zero in on nodes with high global trust values in an effort to boost the consensus group's credibility. Since these nodes have more to gain financially from being trustworthy, they are given more weight in the network. Second, with Enhanced-PoS, a main group of many nodes with greater trust levels takes the role of PoS's single primary node. This improves the efficiency and scalability of the algorithm while decreasing the risk associated with view change operations. By using a group signature, the most visible members of the group may remain anonymous, making them less of a target. In addition, if a node in the main group fails, it may be quickly replaced by any other node in the primary group without requiring a view update.

4.2.2 Scalability

Byzantine fault-tolerant consensus algorithms have large performance decreases as the number of nodes rises, a problem known as scalability. However, the EigenTrust model and the consensus group in Enhanced-PoS solve this problem by reducing the number of nodes involved in the consensus process without limiting the total number of nodes in the system. Because of this, Enhanced-PoS is better suited for use in very large networks.

4.2.3 Efficiency

1) When using Enhanced-PoS, fewer messages are sent because consensus is reached only between nodes in the Consensus Group. The trust value d and the size of the main group x (1xdN) decide how many nodes will take part in the consensus process. If x=1, the algorithm reduces to a simple Proof-of-Stake scheme, with a message complexity of O((dN)2). The maximum number of messages needed in the consensus process is O(N2), which occurs when all nodes in Nodes are involved (d=1). As a result, the communication complexity is decreased except in the worst situation where it remains the same as in PoS (O) (N2).

2) The PoS algorithm is known to have a Byzantine fault-tolerant rate. We developed Enhanced-PoS to improve upon this percentage. High-trust nodes are chosen to create the consensus group; nodes not included in the group have no bearing on the final conclusion. Valid consensus is achieved when the number of correct messages received is greater than the number of Byzantine nodes in the group. This means that f must be below some threshold number. Unless all nodes are Byzantine, Enhanced-PoS will continue to function regardless of the actions of nodes outside the consensus group. Therefore, the sum of the maximum number of Byzantine nodes within the consensus group and outside the group is the maximum number of Byzantine nodes in the system.

So, (1 d)N + 1 dN = 1 dN. The Byzantine fault-tolerant rate is at its lowest when all nodes participate to the consensus (d=1).

V. CONCLUSION

In this paper, we proposed a new consensus algorithm called Enhanced-PoS to enhance the current PoS consensus procedure. We construct a trustworthy consensus group using the EigenTrust model in our Enhanced-PoS to cut down on the total number of consensus nodes, boost consensus efficiency, and streamline communication. We then substituted a main group with higher trust levels for the PoS's original primary node. By using group signature and mutual supervision, Enhanced-PoS has the potential to boost robustness and decrease the probability of the view change process. Our theoretical analysis showed that, in comparison to other comparable consensus algorithms, our Enhanced-PoS was more efficient and more tolerant to Byzantine faults.

Throughput, latency, and the percentage of faulty nodes were all much higher with the proposed approach compared to the standard PoS consensus. By removing the malicious nodes, security is preserved and system performance is improved.

5.1 Recommendations

Distributed systems on blockchain network, software engineers, and scholars interested in this topic might benefit from reading this paper. This is because blockchain technology has made it possible to create trustless economic systems, which facilitate the execution of transparent and trustworthy financial transactions without the need for intermediaries. Thus, it is clear that a consensus mechanism based on Enhanced-PoS is required in distributed ledger systems.

5.1.1 Future Work

In the future, the proposed technique will need to be tested in a live, transactional environment. Combining sharding methods with a better cluster-head selection mechanism can further enhance communication efficiency and security.

The future of blockchain technology must center on strengthening security. The information is safe and verifiable even though the Blockchain ledger is public and decentralized. Successful data encryption necessitates the development of more robust cryptography algorithms to counter threats like unwanted data tampering.

REFERENCES

[1] Abd-El-Malek, M., Ganger, G., Goodson, G., Reite, M. & Wylie, J. (2005). *SOSP Fault-scalable Byzantine fault-tolerant services*.

[2] Aitzhan, N. Z & Svetinovic, D. (2018). Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams. *IEEE Transactions on Dependable and Secure Computing*, 15(5), 840–852, 2018.

[3] Aiyer, A. S., Alvisi, L., Bazzi, R. A. & Clement, A. (2008). DISC Matrix signatures: From macs to digital signatures in distributed systems.

[4] Algorand (2009). Scaling byzantine agreements for cryptocurrencies. In *Proceedings of the 26th Symposium on Operating Systems Principles*, 51–68.

[5] Amir, Y., Coan, B., Kirsch, J. & Lane, J. (2002). *Byzantine replication under attack.* In DSN

[6.] Androulaki, E., Barger, A. & Bortnikov, V. (2018). Hyper ledger fabric: a distributed operating system for permissioned blockchains. *Proceedings of the 13th European System Conference*, 1–15.
[7] Bano, S., Sonnino, A. & Al-Bassam, M. (2017). Sok: Consensus in the age of blockchains. *arXiv preprint arXiv:*1711.03936
[8] Behl, J., Distler, T. & Kapitza, R. (2014). Scalable bft for multi-cores: Actor based decomposition and consensus-oriented parallelization. *The 10th Workshop on Hot Topics in System Dependability*
[9] Bentov, I., Lee, C., Mizrahi, A. & Rosenfeld, M. (2014). *Proof of Activity: Extending Bitcoin's Proof of Work via Proof of Stake [Extended Abstract]*. SIGMETRICS Perform. *Evaluation Review*, 43(3), 36.
[10] Biryukov, A., Feher, D. & Khovratovich, D. (2017). Guru: Universal reputation module for distributed consensus protocols. *IACR Cryptology ePrint Archive*, 1–20.
[11] Buterin, V. & Grifth V, (2017). *Casper the friendly finality gadget, arXiv:*1710.09437.
[12] Buterin, V. (2014). A next-generation smart contract and decentralized application platform. *White paper*
[13] Buterin, V. (2017). On public and private blockchains. *Ethereum blog*, 7.
[14] Castro, M. & Liskov, B. (1999). Practical byzantine fault tolerance. *Proceedings of the 13rd Symposium on Operating Systems Design and Implementation*.99, 1999, 173–186.
[15] Castro, M. & Liskov, B. (2008). Practical Byzantine fault tolerance and proactive recovery. ACM Trans. *Computer System.*
[16] Chaum D. & Heyst, E. V. (1991). Group signatures. *Advances in Cryptology-EUROCRYPT91*, 257–265.
[17] Clement, A., Wong, E. L. & Alvisi, L. (2009). Making byzantine fault tolerant systems tolerate byzantine faults, *Proceedings of the 6th USENIX Symposium on Networked. Systems Design and Implementation*. 9,153 168.
[18] Clement, A., Wong, E. L. & Alvisi, L. (2009). Making byzantine fault tolerant systems tolerate byzantine faults, *Proceedings of the 6th USENIX Symposium on Networked. Systems Design and Implementation*. 9,153–168.
[19] Dinh, A. (2017). Subtle details in pbft. Available on: https://ug93tad.github. io/pbft/
[20] Dinh, T. T. A., Liu, R. & Zhang, M, (2018). Untangling blockchain: A data processing view of blockchain systems. *IEEE Transactions on Knowledge and Data Engineering*, 30(7), 1366– 1385.
[21] Duong, T., Fan, L. & Zhou, H. S. (2016). 2-hop blockchain: Combining proof of-work and proof-of-stake securely. *IACR Cryptology ePrint Archive*, 1–45.
[22] Dwork, C., Lynch, N. & Stockmeyer L. (1988). Consensus in the presence of partial synchrony. *Journal of the ACM (JACM)*, 35(2), 288-323.
[23] Eyal, I., Gencer, A. E. & Sirer, E. G., (2016). Bitcoin-ng: A scalable blockchain protocol," in *Proceedings of the 13th USENIX Symposium on Networked Systems Design and Implementation*. 45–59.
[24] Gilad, Y., Hemo, R., Micali, S., Vlachos, G. & Zeldovich, N. (2017). *Algorand: Scaling byzantine agreements for cryptocurrencies*. *Proceeding of 26th Symposium Operation System Principles*, 51 - 68.
[25] Golan-Gueta, G., Abraham, I. & Grossman, S. (2018). SBFT: a scalable decentralized trust infrastructure for blockchains. *arXiv preprint arXiv:* 1804.01626.
[26] Gramoli, V. (2017). From blockchain consensus back to byzantine consensus. *Future Generation Computer Systems*.
[27] Greenspan, G. (2015). Multichain private blockchain white paper. Available on: http://www.multichain.com/download/MultiChain-White-Paper.pdf
[28] Hasan, H. R. & Salah, K. (2018). Proof of delivery of digital assets using blockchain and smart contracts," *IEEE Access*. 6, 65439–65448.
[29] Hyperledger fabric v0.6.0, (2007). Available on: https://github.com/ hyperledger/fabric/releases/ tag/v0.6.0-preview
[30]Hyperledger fabric v1.4.0, (2018). Available on: https://hyperledger-fabric. readthedocs.io/en/ latest/whatis.html.
[31] Kalio, Q., & Nuka, N. (2019). A Framework for Sceuring Data Warehouse using Hybrid Approach. *International Journal of Computer Science and Mathematical Theory*. *5(1)*, 45–56.
[32] Kamvar, S. D., Schlosser, M. T. & Garcia-Molina, H. (2003). The eigentrust algorithm for reputation management in p2p networks. *Proceedings of the 12th international conference on World Wide Web*. ACM, 640–651.
[33] Kamvar, S. D., Schlosser, M. T. & Garcia-Molina, H. (2003). The eigentrust algorithm for reputation management in p2p networks. *Proceedings of the 12th international conference on World Wide Web*. ACM. 640–651.
[34] Kiayias, A. & Russell, A. (2018). Ouroboros-bft: A simple byzantine fault tolerant consensus protocol. *IACR Cryptology ePrint Archive*, 1– 21.
[35] King, S. & Nadal. S. (2012). *PPCoin: Peer-to-peer cryptocurrency with proof-of-stake.*
[36] Kotla, R., Alvisi, L. & Dahlin, M. (2007). Zyzzyva: speculative byzantine fault tolerance. *ACM SIGOPS Operating Systems Review.* 41(6), 45–58.
[37] Kwon, J. (2014). Tendermint: Consensus without mining. *Draft v. 0.6, fall*
[38] Lamport, L., Shostak, R. & Pease, M. (2016). The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 4(3), 382-401.
[39] Li M., Zhu, L. & Lin, X. (2018). Efficient and privacy-preserving carpooling using blockchain- assisted vehicular fog computing. *IEEE Internet Things Journal*, 6(3), 4573 – 4584.
[40] Liu, S., Viotti,P., & Cachin, C. (2016). Xft: Practical fault tolerance beyond crashes." *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation*, 485–500.
[41] Lu K., Wang, J. & Li, M. (2016). An eigentrust dynamic evolutionary model in p2p file-sharing systems," *Peer-to-Peer Networking and Applications*, 9(3), 599–612.
[42] Martin J. P. & Alvisi L. (2006). Fast Byzantine consensus. IEEE Transactions on Dependable and Secure Computing, 3(3), 202–215.
[43] Miller, A., Xia, Y., Croman, K. (2016). The honey badger of bft protocols. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 31–42.

[44]Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. [online] Available:https://bitcoin.org/bitcoin.pdf.

[45]Nakamoto, S. (2009). Bitcoin: A peer-to-peer electronic cash system. Available on:https://bitcoin.org/bitcoin.pdf/, *Security*, 38-47.

[46] Nwiabu, N., Allison, I. & Babs, O. (2021). Modelling Case-Based Reasoning in Situation-Aware Disaster Management. *Communication of the IIMA*, 9(2), 41–66.

[47] Nwiabu, N. & Adeyanju, I. (2012). User Centred Design Approach to Situation Awareness. *International Journal of Computer Applications*, 49(17) 75–88.

[48] Oki B. M. & Liskov B. (1988). *Viewstamped replication:* A new primary copy method to Support highly- available distributed systems. *Proceeding 7th ACM Symposium on Principles of Distributed Computing (PODC)*, 817.

[49] Oki, B. M. & Liskov, B. H. (1998). Viewstamped replication: A new primary copy method to support highly-available distributed systems. *Proceedings of the 7th ACM Symposium on Principles of Distributed Computing*, 8–17.

[50] Ongaro, D. & Ousterhout, J. K. (2014). In search of an understandable consensus algorithm. *USENIX Annual Technical Conference*, 305–319.

[51] Paper, N. W. (2016). Delegated byzantine fault tolerance consensus algorithm. Available on: https://docs.neo.org/en-us/basic/consensus/ consensus

[52] Pass, R. & Shi, E. (2018). Thunderella: Blockchains with optimistic instant confirmation. *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 3–33.

[53] Shae, Z. & Tsai, J. J. (1980). On the design of a blockchain platform for clinical trial and precision medicine. *IEEE 37th International Conference on Distributed Computing Systems*, 1972–1980.

[54] Turkanovic, M., Ho¨lbl, M., Kosic, K., Hericko, M. & Kamisalic, A. (2018). Eductx: A blockchain-based higher education credit platform. *IEEE Access*, 6, 5112–5127.

[55] Vukolic, M., (2015). The quest for scalable blockchain fabric: Proof-of-work vs. bft replication. *International Workshop on Open Problems in Network Security*, 112–125.

[56] Wang, W., Hoang, D. T. & Xiong, Z. (2019). A survey on consensus mechanisms and mining management in blockchain networks. *IEEE Access*, 7, 22328 – 22370.

[57] Wang, W., Hoang, D.T., Hu, P., Xiong, Z., Niyato, D., Wang, P., Wen, Y. & Kim, D.I. (2019). Bitcoin and beyond: A technical survey on decentralized digital currencies, *IEEE Communications Surveys & Tutorials*, 18(3), 2084–2123.

[58] Yuan, Y. & Wang, F. Y. (2018). "Blockchain and cryptocurrencies: Model, techniques, and applications. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 48(9), 1421–1428.

[59] Zheng, K., Xie, S. & Dai, (2017). An overview of blockchain technology: Architecture, consensus, and future trends. *IEEE International Congress on Big Data*, 557–564.