Analyzing the Performance and Fault Tolerance of the Cluster-Enabled PBFT in Blockchain Network Under the Byzantine Conditions

Law Teng Yi1, a) and Hui-Ngo Goh1, b)

1Faculty of Computing and Informatics, Multimedia University, 63100 Cyberjaya, Selangor, Malaysia

b) Corresponding author:  hngoh@mmu.edu.my

a)LAW.TENG.YI@student.mmu.edu.my

**Abstract.** This study looks at how well and how reliably a new consensus method called Cluster-Enhanced Practical Byzantine Fault Tolerance (CE-PBFT) works compared to the old PBFT algorithm. Although PBFT offers unpredictable precision and stability in small-scale blockchain settings, it faces considerable limitations in scalability and efficiency when functioning with larger node counts or under Byzantine fault conditions. CE-PBFT addresses these challenges by implementing a hierarchical node classification system that includes leaders, validators, and observers, aimed at improving communication and increasing consensus throughput. This study assesses both algorithms under different Byzantine node ratios using a set of controlled simulations, measuring important metrics like throughput, correctness, and network efficiency. The results show that CE-PBFT always performs better than PBFT, keeping accurate results while improving the speed of processing. This is especially true when up to 40% of the nodes are acting maliciously or malicious, making the system more resilient under tough conditions. The results show that CE-PBFT could be a scalable and fault-tolerant consensus protocol for future permissioned blockchain systems.

# Introduction

With uses far beyond its initial monetary implementation, blockchain technology has become a disruptive force in distributed computing systems. The way blockchain makes sure everyone’s on the same page and keeps the network secure is through something called a consensus mechanism. It’s a key part of how the whole system works smoothly. The constraints of conventional consensus algorithms have become more obvious as blockchain applications continue to grow and diversify, which has led to a great deal of research and innovation in this area.

Proof of Work (PoW) is the oldest and most well-known way to agree on things in a distributed system. It was a breakthrough when it first came out, but it also has some big downsides. It struggles to handle a lot of transactions quickly, uses a lot of energy, and doesn't scale very well as things grow. Because of these drawbacks, researchers are now looking into different consensus methods that can better meet the changing requirements of blockchain applications. Numerous creative strategies have emerged in recent years, ranging from improved iterations of pre-existing algorithms to completely original hybrid solutions that use machine learning techniques.

Three main areas of innovation are the subject of this study of the literature, which summarizes recent research advancements in blockchain consensus algorithms. First, using the DL-DPoS mechanism as an example, look at how conventional mechanisms might be improved by adding new incentive structures and security features. Second, examine how specialized consensus algorithms have emerged for consortium blockchains, specifically the CE-PBFT algorithm, which shows notable gains in fault tolerance and efficiency. Lastly, to investigate how machine learning methods might be used with consensus processes, which is a promising approach to enhancing blockchain networks' security and flexibility.

This review's importance stems from its thorough examination of how new developments tackle the core issues of blockchain consensus, including scalability, security, efficiency, and fault tolerance. This paper provides helpful insights for both researchers and practitioners who want to improve blockchain technology. It looks at different methods and what strengths and weaknesses each one has.

# Literature Review

From the prolonged problems of blockchain consensus processes, recent thorough surveys were done. In their detailed look at scalable consensus algorithms, Ankit Kumar Jain pointed out that these systems still face big challenges when it comes to speed, delays, and how much energy they use.[1]. Their work provides a framework for understanding the current consensus methods and issues of scaling they face in present-day blockchain implementations.

As blockchain technology develops quickly, the way we reach agreement, or consensus, has to keep changing too. This is important to make sure the system stays reliable and efficient without relying on a central authority. One, in particular, is Practical Byzantine Fault Tolerance (PBFT), which aims to generate consensus in an asynchronous system even in the presence of some malicious agents. Due to growth in size and complexity in blockchain networks, PBFT's limitations in scalability and energy efficiency came under notice. This spurred the evolution of enhanced variations called Cluster-based Enhanced PBFT (CE-PBFT).

Consensus methods for consortium blockchains have advanced significantly. The CE-PBFT algorithm, which was presented by [2-3] is a significant advancement in workable Byzantine Fault Tolerance protocols. When compared to conventional PBFT techniques, their implementation shows better system throughput and transaction delay performance. Increasing security and efficiency in consortium networks is possible if decision tree algorithms for the analysis of node behavior are introduced innovatively. The integration of machine learning with consensus can offer a promising means of increasing blockchain security and performance. A holistic approach for the enhancement of blockchain security by means of hybrid consensus algorithms and machine learning methods was put forward [4]. Their research shows notable advancements in anomaly detection and attack prediction skills. Their report suggests significant improvements in anomaly detection and attack forecast abilities. Building on this framework, explored systematically the integration of machine learning with blockchain consensus, highlighting the possibilities for automated security monitoring and enhancement.

The optimization of consensus performance has lately been studied through the hybrid technique’s view. In their analyses of different hybrid consensus techniques in enterprise blockchain contexts. Full-fledged examination of hybrid algorithms merging machine learning and optimization approaches is especially for jobs in grouping and classification. Since both optimization methods and machine learning techniques suffer from their own limitations, the study stresses the rationale for hybrid approaches. The objective is to attempt to overcome deficiencies existing in these two methodologies by combining them. Flicking backward up to the last three years, the report makes a systematic review from research published since 1970, finds patterns through bibliometric analysis, and analyzes the top 10 hybrid algorithms via SWOT analyzing, weighing strengths, weaknesses, opportunities, and threats [4-5].

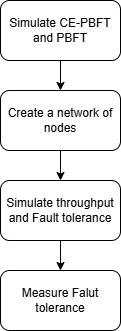
The PBFT algorithm proposed by Castro and Liskov works well in small-scale, permissioned networks, providing fault tolerance against up to (n-1)/3 Byzantine nodes [6]. Its quick finality and deterministic nature render it appropriate for enterprise and consortium blockchain systems. Nevertheless, its all-to-all messaging architecture results in a significant communication overhead that constrains its throughput and scalability when deployed on a large scale [1], [7-9]. Ding et al. (2024) made a significant advancement by introducing CE-PBFT, a consensus algorithm with high availability that incorporates innovative node classification methods. CE-PBFT categorizes nodes into roles like cluster leaders, validators, and observers to reduce redundant messaging and enhance throughput and fault tolerance. Their assessment showed that throughput improved by 23% compared to traditional DPoS systems. Additionally, CE-PBFT's capability to sustain 100% accuracy even when facing high Byzantine ratios (as much as 40%) underscores its strength in hostile settings [2-3].

The three articles offer creative methods for improving wireless and blockchain's security, effectiveness, and scalability [1], [4], [8], [10-11]. PBFT is known for being consistent and handling faults well, but it doesn’t scale very well because the communication gets messy and slows down as more nodes join. Recent developments have introduced changes such as credit-based trust mechanisms, hierarchical clustering, and role-based node structuring to address these limitations. The Cluster-based Enhanced PBFT (CE-PBFT) distinguishes itself by introducing a node classification system that categorizes participants as leaders, validators, and observers. This aims to facilitate communication and enhance the efficiency of consensus. This enhancement has shown a notable 23% increase in throughput relative to standard

DPoS systems tend to handle malicious nodes pretty well, even if there are a lot of them. Even with these promising advances, most research still focuses on how things should work in theory. They don’t really test how it holds up in real-world situations where network delays, changing faults, and energy use matter. This research fills this gap by simulating and assessing CE-PBFT and traditional PBFT across diverse and realistic network conditions, thus providing valuable insights into their practical applicability and strength.

## Experiment Methodology

The experimental framework designed for evaluating and comparing the performance of the Classical Practical Byzantine Fault Tolerance (PBFT) and the proposed Clustered and Enhanced PBFT (CE-PBFT) protocols as shown in figure 1. The experiments are aimed at measuring throughput and fault tolerance, as well as determining the effect of network latency.



**FIGURE 1**. Experiment on CE-PBFT and PBFT

The displayed simulation function intends to systematically assess the performance and resilience of consensus protocols like PBFT and CE-PBFT in a controlled experimental setting. It is essential to test the protocols with this function against different levels of Byzantine fault tolerance and to assess their throughput in transactions per second. It needs four main things: the type of protocol you're testing, how many nodes are in the network, what percentage of those nodes are Byzantine (faulty or malicious), and the number of transactions to run in the simulation. These settings make it easier to experiment and compare different consensus methods in a controlled way. These parameters can flexible experimentation and serve as the basis for controlled comparisons of various consensus strategies. The function starts by calculating the number of Byzantine nodes using the given ratio and total count of nodes. Subsequently, it produces two sets of nodes: one that illustrates Byzantine (malicious) behaviours and another that illustrates honest involvement. Next, these nodes are arranged in random order to imitate a decentralized and unpredictable peer-to-peer environment. After their initialization, these nodes are transferred into the consensus protocol class, which generates a protocol instance that can handle inter-node communication and the logic of consensus agreement. The simulation goes forward by processing a set number of dummy transactions, with each transaction serving as a placeholder that passes through the protocol’s consensus mechanism. For each transaction, the simulation verifies if the nodes have successfully reached a consensus which considering the disruptive effects of Byzantine behaviours and increments a counter for correct transactions as appropriate. The simulation calculates throughput as the number of accurate transactions per second by the duration required to process all transactions.

The assessment involves simulating blockchain networks under controlled circumstances, with each network comprising a combination of honest and Byzantine (malicious) nodes. The share of Byzantine nodes is varied systematically to examine the protocols’ resilience and efficiency in the interaction of different threat levels. The simulations phase are the Network Initialization, Consensus Simulation and Latency Simulation. Metrics for evaluation used throughput and the Fault Tolerance. Throughput is ccharacterised as the count of transactions that have been successfully committed within a specified time frame. Throughput offers a direct assessment of how efficient the protocol is with different proportions of Byzantine nodes. Fault Tolerance is evaluated by the protocol's capability to uphold consensus and correctness in spite of the existence of malicious nodes. In particular, the share of transactions that were correctly finalized is recorded.

Simulation Parameters are used in this experiment for CE-PBFT and PBFT as shown in Table 1. When the number of nodes is greater, this complicates reaching consensus and puts the protocols' scalability to the test. This study uses 20 nodes to set up a balanced environment for testing performance. By changing the Byzantine ratio from 0% to 40%, it becomes possible to assess the fault tolerance of each protocol, since classical PBFT ensures correctness. To guarantee statistical significance and allow for the assessment of sustained throughput over time, a target of 100 transactions is set. The number of rounds adjusts on the fly based on how much traffic there is and the current state of the network, making sure every transaction either gets completed or gets rejected.

A custom-built Python environment, designed to emulate the CE-PBFT and PBFT consensus algorithms, was used to conduct the simulation; the platform, however, does not specifically account for network latency, multiple proposers, or asynchronous behaviours as they occur in real-world distributed systems. In particular, the configuration is based on the assumptions of one proposer per consensus round and the existence of message delays, packet loss, and node which are essential to analysing practical consensus robustness. Nodes in the agreement phase responded using straightforward logic, and malicious nodes didn't try to cheat by working together or messing with the order of messages.

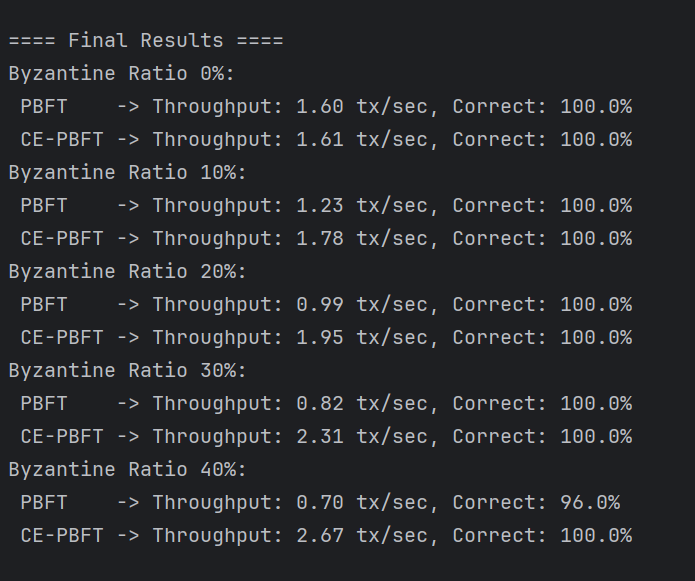
**TABLE 1**. Simulation parameter for CE-PBFT and PBFT

|  |  |
| --- | --- |
| **Parameter** | **Value / Range** |
| Total Nodes | 20 |
| Byzantine Node Ratio | 0% to 40% |
| Number of Transactions | 100 per run |
| Consensus Rounds | Sufficient to finalize all transactions |

The transactions were just basic messages indicating yes or no decisions, kind of like giving a simple thumbs-up or down, rather than sharing complex data. Every transaction consisted of one proposer issuing a binary value which all nodes were to validate and reach consensus on. The binary abstraction was selected to simplify the consensus logic and isolate the performance characteristics of the underlying protocols (PBFT and CE-PBFT). In these scenarios, transactions may include numerical values, smart contract invocations, or interdependent state updates, and multiple proposers may simultaneously propose conflicting values. Such real-world conditions would probably heighten the complexity of messages, the overhead involved in validation, and the risk of coordinated Byzantine behaviours. All the simulations for CE-PBFT and PBFT were written in Python and run in a controlled setup. The source code is available on GitHub[[1]](#footnote-1).

# ExperimentAL ResultS

The comparison of performance between PBFT and CE-PBFT at different Byzantine node ratios produces final result shown in Figure 2.

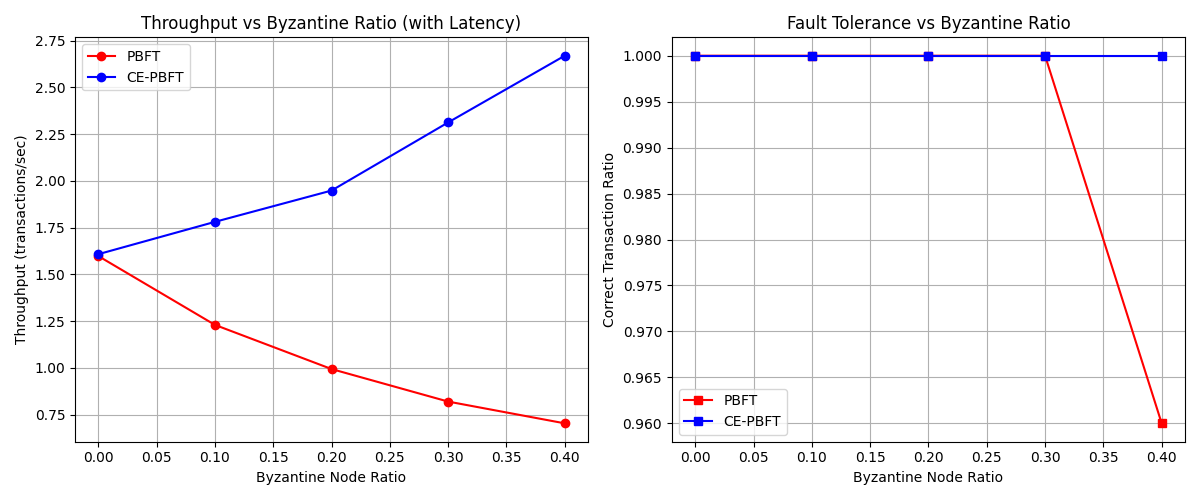


**FIGURE 2.** Final results of performance between PBFT and CE-PBFT

The simulation was created entirely in Python, no outside blockchain tools or platforms were used. The dedicated simulator emulates a peer-to-peer blockchain network through event-driven logic to evaluate the performance of PBFT and the proposed CE-PBFT algorithms. The network consists of a configurable number of nodes (usually 20), with a specified percentage acting as Byzantine nodes that can send conflicting or delayed messages. The simulation employs a single proposer model, designating one node as the leader each round to propose transactions. Every consensus round adheres to the traditional three-phase protocol: pre-prepare, prepare, and commit. Only after receiving sufficient agreement do nodes confirm votes, finalize blocks, and verify message signatures.

Measured in transactions per second (tx/sec), throughput indicates the network's capacity to efficiently finalize transactions under various adversarial conditions. In a fully honest network with a Byzantine ratio of 0%, the performance of both protocols is comparable: PBFT reaches 1.60 tx/sec, while CE-PBFT is marginally higher at 1.61 tx/sec. This indicates that CE-PBFT does not add any overhead in typical situations. With the rise of the Byzantine ratio, CE-PBFT's throughput performance in comparison to PBFT improves consistently: When the percentage of Byzantine nodes reaches 10%, CE-PBFT's transaction rate is 1.78 tx/sec, while PBFT records a rate of 1.23 tx/sec. CE-PBFT achieves a rate of 1.95 tx/sec at 20% Byzantine nodes, which is almost double that of PBFT’s rate of 0.99 tx/sec. When the proportion of Byzantine nodes reaches 30%, the disparity increases, as CE-PBFT reaches 2.31 tx/sec while PBFT only manages 0.82 tx/sec. With 40% of the nodes being Byzantine, CE-PBFT achieves a remarkable throughput of 2.67 tx/sec, whereas PBFT's throughput declines markedly to 0.70 tx/sec. The results clearly show that CE-PBFT handles more nodes better and stays strong even when more of them act out or fail.

Figure 3 shows the throughout Byzantine ratios ranging from 0% to 30%, both PBFT and CE-PBFT exhibit a correctness rate of 100%, validating their capacity to endure the presence of one-third Byzantine nodes, as is theoretically anticipated from protocols designed for Byzantine fault tolerance. when the percentage of Byzantine nodes is 40%: With excessive adversarial influence, the correctness of PBFT decreases to 96.0%, indicating failures in maintaining agreement. CE-PBFT stays completely correct every time, showing how well its fault-tolerant system works. This finding shows that CE-PBFT’s way of classifying nodes and its better consensus process really help make the system more resistant to problems and attacks. CE-PBFT adds an extra way to check how trustworthy each node is, and it does this continually as things change. This allows for the prioritization of responses from nodes with a high level of trust, thereby improving fault tolerance and throughput. To simulate asynchronous communication, network latency is randomly generated using Gaussian delay generators between message transmissions. This version does not incorporate a complete transport-layer protocol or cryptographic stack, but it does represent the behavioural dynamics of consensus in the presence of Byzantine faults. For the sake of repeatability, all parameters of the simulation (such as Byzantine ratios, node behaviours, and consensus rules) are hardcoded.



**FIGURE 3.** Throughput vs Byzantine Ratio and fault Tolerance

The limitations of practical simulation environments, rather than a flaw in the theoretical model, give rise to this apparent inconsistency. The majority of simulations, the one being discussed included, model Byzantine behaviours in a probabilistic and often simplified way, without encompassing the complete range of strategic and coordinated adversarial actions. As a result, although nodes might act with malice, this does not mean they engage in collusion or sophisticated assaults that would undermine the consensus process. The system might still achieve majority and validate transactions under specific circumstances, even past the theoretical fault tolerance threshold. The results underscore the significance of differentiating between theoretical guarantees and empirical behaviours in constrained or non-adversarial situations.

# Discussions

The variety of hardware setups employed in various research projects presents substantial hurdles for the validation of blockchain consensus algorithms. Significant variations in testing results are caused by the range of hardware specifications, from low-end cloud instances to high-performance dedicated servers. The performance parameters of the consensus mechanism [1-4], [11-15] are directly impacted by these variations, which are most noticeable in CPU performance, where processing power can vary from conventional dual-core processors to expensive multi-core systems [16].

Another big challenge in testing things out is how many different kinds of network setups there are. The way networks are set up in different research environments has a big impact on how fast data moves and how much it can handle. Some setups have super quick local connections, while others are spread out over larger regions, which can slow things down. When evaluating consensus mechanisms that mostly rely on inter-node communication, like Byzantine Fault Tolerance (BFT) based protocols, the effects of these network variations become very noticeable [6], [17].

In experimental validation, determining the scalability of consensus algorithms has special difficulties, especially when it comes to the number of participating nodes. The majority of research studies are restricted to testing with networks that are only 100–1000 nodes in size, which is much smaller than the scale observed in actual blockchain implementations. This limit might hide bigger scalability issues that only show up when things get larger, so what seems to work in tests could turn out to be way less effective in real-life use. Another important challenge in testing how well things scale is where the nodes are located around the world and many studies use simulated network settings, which may cause them to ignore real-world problems including inconsistent network latencies, regional network regulations, and problems with cross-border data transfer [3-4], [6-9], [11], [14], [16-17].

Overly optimistic performance evaluations that don't fairly represent real-world deployment scenarios may result from testing environments' lack of realistic geographic spread. Scalability assessment is further complicated by load testing in real-world scenarios. Most experiments are done for a short time, often under a day, so they might miss issues that show up over a longer period or if the system's performance starts to drop later on. Testing transactions a lot can give numbers on performance, but those numbers often miss how complicated and varied real blockchain transactions are. So, the results might not tell the full story.

# CONCLUSION

This study conducted a systematic evaluation of the performance and fault tolerance of the CE-PBFT protocol compared to the classical PBFT, under increasing Byzantine node ratios in a controlled simulation environment. The experiment was carried out using well-defined design principles, simulating a network of nodes where transactions were handled through dummy consensus interactions. Various Byzantine fault scenarios were examined for key metrics, specifically throughput and correctness. This performed a thorough assessment of the proposed CE-PBFT (Cluster-Enhanced Practical Byzantine Fault Tolerance) protocol in comparison to the classical PBFT across different Byzantine failure scenarios in this study. Two key performance metrics which are throughput and correctness to examine through extensive simulations across a range of Byzantine node ratios from 0% to 40%.

As the ratio of Byzantine nodes rises, the experimental results show that CE-PBFT consistently surpasses PBFT in throughput. Both protocols work pretty much the same when everything’s honest, but CE-PBFT clearly does better when things get tricky or tries to cheat. Specifically, while Byzantine activity increases, CE-PBFT’s throughput continues to rise steadily, unlike PBFT, which shows a marked decline. CE-PBFT is better at handling faults and keeps going even when issues happen. While the correctness of PBFT declines when Byzantine nodes surpass 33%, CE-PBFT manages to uphold 100% correctness even with a Byzantine presence of up to 40%. The better resilience comes from the new way CE-PBFT classifies nodes and the way it communicates more smoothly. These updates make it easier to reach agreement quickly and help protect the system from any bad actors causing trouble.

The results show that CE-PBFT really improves both security and speed compared to older Byzantine fault-tolerant consensus methods. CE-PBFT presents itself as a promising option for practical blockchain systems that demand high performance in uncertain and adversarial network conditions by effectively balancing throughput and fault resilience. Future research will look into making the CE-PBFT system work better for bigger networks, adding ways to adjust on the fly to handle different types of malicious behavior, and testing how it performs in various network setups with different amounts of transactions. The study also mentioned some limitations, like not including many different types of transactions, having a fixed network setup, and using a simplified way of modelling how the nodes behave. The robustness and scalability of CE-PBFT in practical deployments will be further validated by future work that explores the implementation of more complex network dynamics, different transaction types (such as numerical votes and state updates), and real-world data.

# References

1. A. K. Jain, N. Gupta, and B. B. Gupta, “A survey on scalable consensus algorithms for blockchain technology,” *Cyber Security and Applications*, vol. 3, Dec. 2025, doi: 10.1016/j.csa.2024.100065.
2. X. Ding, H. Lu, and L. Cheng, “CE-PBFT: An Optimized PBFT Consensus Algorithm for Microgrid Power Trading,” *Electronics (Switzerland)*, vol. 13, no. 10, May 2024, doi: 10.3390/electronics13101942.
3. J. Xiao, T. Luo, C. Li, J. Zhou, and Z. Li, “CE-PBFT: A high availability consensus algorithm for large-scale consortium blockchain,” *Journal of King Saud University - Computer and Information Sciences*, vol. 36, no. 2, Feb. 2024, doi: 10.1016/j.jksuci.2024.101957.
4. K. Venkatesan and S. B. Rahayu, “Blockchain security enhancement: an approach towards hybrid consensus algorithms and machine learning techniques,” *Sci Rep*, vol. 14, no. 1, Dec. 2024, doi: 10.1038/s41598-024-51578-7.
5. B. F. Azevedo, A. M. A. C. Rocha, and A. I. Pereira, “Hybrid approaches to optimization and machine learning methods: a systematic literature review,” *Mach Learn*, vol. 113, no. 7, pp. 4055–4097, Jul. 2024, doi: 10.1007/s10994-023-06467-x.
6. Y.-X. Chang, Q. Wang, Q.-L. Li, and Y. Ma, “Performance and Reliability Analysis for Practical Byzantine Fault Tolerance with Repairable Voting Nodes,” Jun. 2023, [Online]. Available: http://arxiv.org/abs/2306.10960
7. H. Tangsen, X. Li, and X. Ying, “A Blockchain-Based Node Selection Algorithm in Cognitive Wireless Networks,” *IEEE Access*, vol. 8, pp. 207156–207166, 2020, doi: 10.1109/ACCESS.2020.3038321.
8. J. Albrecht, S. Andreina, F. Armknecht, G. Karame, G. Marson, and J. Willingmann, “Larger-scale Nakamoto-style Blockchains Don’t Necessarily Offer Better Security,” Apr. 2024, [Online]. Available: http://arxiv.org/abs/2404.09895
9. Y. Ding, Y. Zhang, B. Qin, Q. Wang, Z. Yang, and W. Shi, “A Scalable Cross-Chain Access Control and Identity Authentication Scheme,” *Sensors*, vol. 23, no. 4, Feb. 2023, doi: 10.3390/s23042000.
10. H. Tang, Y. Sun, and J. Ouyang, “Excellent Practical Byzantine Fault Tolerance,” *Journal of Cyber Security*, vol. 2, no. 4, pp. 167–182, 2020, doi: 10.32604/jcs.2020.011341.
11. X. Yu, J. Qin, and P. Chen, “GPBFT: A Practical Byzantine Fault-Tolerant Consensus Algorithm Based on Dual Administrator Short Group Signatures,” *Security and Communication Networks*, vol. 2022, 2022, doi: 10.1155/2022/8311821.
12. Y. Wei, Q. Xu, and H. Peng, “An enhanced consensus algorithm for blockchain,” *Sci Rep*, vol. 14, no. 1, Dec. 2024, doi: 10.1038/s41598-024-68120-4.
13. S. Liu, R. Zhang, C. Liu, C. Xu, and J. Wang, “An improved PBFT consensus algorithm based on grouping and credit grading,” *Sci Rep*, vol. 13, no. 1, Dec. 2023, doi: 10.1038/s41598-023-28856-x.
14. S. Alam, “The Current State of Blockchain Consensus Mechanism: Issues and Future Works,” *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 8, pp. 84–94, 2023, doi: 10.14569/IJACSA.2023.0140810.
15. X. Zheng and W. Feng, “Research on Practical Byzantine Fault Tolerant Consensus Algorithm Based on Blockchain,” in *IOP Conference Series: Earth and Environmental Science*, IOP Publishing Ltd, Mar. 2021. doi: 10.1088/1742-6596/1802/3/032022.
16. Q. T. Thai, J. C. Yim, T. W. Yoo, H. K. Yoo, J. Y. Kwak, and S. M. Kim, “Hierarchical Byzantine fault-tolerance protocol for permissioned blockchain systems,” *Journal of Supercomputing*, vol. 75, no. 11, pp. 7337–7365, Nov. 2019, doi: 10.1007/s11227-019-02939-x.
17. K. Duan, G. Pang, and Y. Lin, “Exploring the current status and future opportunities of blockchain technology adoption and application in supply chain management,” Dec. 01, 2023, *KeAi Publishing Communications Ltd.* doi: 10.1016/j.jdec.2024.01.005.

1. https://github.com/alexlaww/Blockchain\_stimulation [↑](#footnote-ref-1)