Comparative Performance Analysis of Original AuRa and Improved AuRa Consensus Algorithms in Chain Hammer Digital Certificate Simulation

Robiah Arifin[1]*, Wan Aezwani Wan Abu Bakar[2]*, Mustafa Man[3]*, Evizal Abdul Kadir[4]
Department of Big Data-Infostructure and Network Management Centre,
Universiti Sultan Zainal Abidin, 21030 Kuala Nerus, Terengganu, Malaysia[1]
Mohamad Afendee Mohamed, Faculty of Informatics and Computing,
Universiti Sultan Zainal Abidin, 22200 Besut Campus, Besut, Terengganu, Malaysia[2]
Faculty of Computer Science and Mathematics, Universiti Malaysia Terengganu, 21030 Kuala Nerus, Terengganu, Malaysia[3]
Universitas Islam Riau, Jl. Kaharuddin Nasution 113, Pekanbaru 28284, Riau, Indonesia[4]

***Abstract*—The blockchain functions as a distributed database, where data is securely stored across multiple servers and network nodes. It exists in various forms, with Bitcoin, Ethereum, and Hyperledger being among the most prominent examples. To ensure the integrity and security of transactions within a blockchain network, a consensus algorithm is employed to establish agreement among participating nodes. Several types of consensus algorithms exist, each offering distinct features and operational mechanisms. One such algorithm is Authority Round (here defined as AuRa_ori), a member of the Proof-of-Authority (PoA) family supported by Parity clients. Previous studies have highlighted several vulnerabilities and performance limitations in AuRa_ori, particularly concerning transaction speed per second (TPS) and transaction throughput per second (TGS). This study specifically investigates the original AuRa algorithm alongside an improved version, termed AuRa_v1. In AuRa_v1, the transaction process is structured into four key phases: 1) leader assignment, 2) block proposal, 3) agreement, and 4) block commitment. However, inconsistencies and inefficiencies have been identified within certain phases of the original AuRa_ori, particularly during the leader assignment and agreement stages. In response, this study proposes an improved approach through AuRa_v1 to address these vulnerabilities. A detailed analysis is conducted to evaluate the impact of these vulnerabilities on TPS, TGS, and epoch time, followed by a performance comparison between AuRa_ori and AuRa_v1. Experimental results demonstrate that AuRa_v1 effectively resolves the identified performance issues, achieving a significant improvement. Specifically, AuRa_v1 records a 21.65% increase in both TPS and TGS compared to AuRa_ori, validating the effectiveness of the proposed enhancements.**

***Keywords*—Blockchain; Ethereum; AuRa_ori; AuRa_v1; TPS; TGS**

I. Introduction

The integration of blockchain technologies with artificial intelligence (AI) has emerged as a transformative paradigm in building decentralized, secure, and intelligent systems. In such architectures, consensus algorithms are fundamental to ensuring data integrity, traceability, and synchronization across distributed nodes. AuRa_v1, an improved authority-based consensus protocol, offers enhanced transaction speed and network throughput, making it a suitable infrastructure layer for AI applications that demand real-time data coordination and high system reliability.

As AI systems increasingly rely on decentralized frameworks—such as federated learning, distributed inference, and autonomous multi-agent environments—the role of efficient and lightweight consensus mechanisms becomes critical. By facilitating low-latency, tamper-resistant data validation, AuRa_v1 enables seamless collaboration and trust in AI workflows without centralized control. This intersection between consensus algorithms and AI models presents promising opportunities for developing scalable, secure, and responsive intelligent systems, particularly in domains such as IoT-driven AI, decentralized data marketplaces, and collaborative machine learning.

A consensus algorithm is a fundamental mechanism that establishes agreement among distributed nodes across multiple servers and networks. It plays a crucial role in ensuring blockchain functionality [1] by maintaining consistency, decentralization [2-3], and integrity across all participating nodes. Beyond ensuring reliability, consensus mechanisms strengthen transaction security by coordinating node agreement, thereby preserving uniform and valid throughput. Moreover, an efficient consensus algorithm optimizes transaction execution time, minimizes latency, and maximizes throughput [4], directly influencing the scalability and processing performance of blockchain networks.

Among the existing approaches, Proof of Authority (PoA) has emerged as a leading solution [5] for permissioned blockchain environments. Initially introduced by [6] as a hybrid of Proof of Work (PoW) and Proof of Stake (PoS), PoA was specifically designed to address PoS vulnerabilities [7] and mitigate the performance and security limitations of both PoW and PoS. Tailored for consortium and private blockchain networks [8,9], PoA is implemented primarily through two variants: Authority Round (AuRa_ori) and Clique. AuRa_ori relies on trusted authorities and synchronized nodes to maintain high consensus reliability.

However, prior studies have underscored significant performance limitations of AuRa_ori, particularly stemming

*Corresponding author.

from its leader assignment and voting processes. The algorithm's design requires a new leader to be selected for each transaction, increasing both transaction per second (TPS) and transaction generation speed (TGS) [10], but simultaneously introducing delays during leader reassignment—especially when nodes are unresponsive. The voting phase before block commitment further compounds latency and restricts throughput. These operational inefficiencies remain unresolved, thereby hindering PoA's ability to support high-throughput blockchain environments. However, AuRA_ori is highly necessary as it is suitable and effective for use within a private server environment. Therefore, the research gap related to the operational efficiency of AuRA_ori needs to be addressed.

Therefore, this research specifically targets the performance bottlenecks in AuRa_ori by proposing design modifications that enhance TPS and TGS, without compromising the consensus integrity. By analyzing the interplay between leader selection, voting mechanisms, and transaction throughput, we aim to provide a more efficient PoA variant, AuRa_v1 that addresses these bottlenecks.

The remainder of this study is structured as follows: Section II reviews the blockchain environment and consensus algorithms. Section III details the methodology, including Ethereum, AuRa_ori, and AuRa_v1 implementations. Section IV discusses the execution of both AuRa_ori and AuRa_v1. Section V presents the performance analysis of TPS and TGS across both algorithms. Finally, Section VI concludes with a discussion on how AuRa_v1 improves performance and scalability over AuRa_ori.

II. Literature Review

Blockchain is an innovative method for data storage that adopts a decentralized and distributed database architecture. This model [11] represents a paradigm shift from traditional centralized data management systems, which rely on a single, authoritative source, to a distributed network of interconnected nodes. In a blockchain, data are organized into sequentially linked units known as blocks [12]. These blocks are distributed across multiple nodes located on different servers.

Blockchain technology operates on a peer-to-peer (P2P) network model [13], whereby nodes interact directly with one another through P2P communication protocols. The integrity and security of the stored digital data [14] are ensured through the application of cryptographic techniques and hash functions.

Public blockchains have become increasingly accessible via various platforms [15], thus facilitating widespread adoption and implementation by researchers and developers. Among the most prominent platforms are Bitcoin, Ethereum, and Hyperledger [16], all of which have been established to promote the integration of blockchain technology [17] into organizational operations.

Although these platforms share foundational characteristics, each possesses distinct features that cater to different application requirements. Bitcoin and Ethereum, for instance, are classified as public or permissionless blockchains [18], allowing unrestricted participation in the network, whereby any user can initiate or validate transactions without prior authorization.

Ethereum, introduced by Vitalik Buterin in 2013 [19], was developed to address the limitations observed [20] in the Bitcoin protocol. The platform's development was successfully funded through a public crowdfunding initiative [21] in 2014, leading to its official deployment in 2015. Ethereum supports distributed data storage and enables users to create and execute decentralized applications (DApps) [22] and custom blockchains.

Its architecture accommodates flexible data storage [23] by supporting blocks of varying sizes. Ethereum also integrates various consensus mechanisms to secure transactions, enhance performance, and ensure data validity across the network. These mechanisms are facilitated through Ethereum clients that maintain and operate the blockchain engine.

They also compared the performance of the Geth and Parity Ethereum clients, both supporting PoA-based consensus mechanisms. Their experiments, conducted in private testnet environments, found that the Parity client, featuring AuRa_ori, has achieved a 91% higher transaction speed than Geth, which implements the Clique consensus algorithm. Factors such as CPU, memory, and node count were accounted for in these evaluations

A. Consensus Algorithm

The consensus algorithm is a foundational element of blockchain technology, as it governs the agreement process among nodes and ensures the validity, security, and consistency of transactions and blocks across the distributed ledger. To maintain the integrity of the blockchain, a consensus algorithm must be robust in terms of security, performance, and fault tolerance. Fundamentally, it is a mechanism that enables distributed nodes to reach agreement on the current state of the blockchain without relying on a central authority [1].

Consensus algorithms significantly enhance the security of blockchain systems by enabling a coordinated validation process among nodes, which in turn ensures consistent and trustworthy transaction throughput. These algorithms also play a vital role in optimizing performance by reducing transaction execution times and increasing processing efficiency. The design and operational protocol of a consensus algorithm directly influence its scalability and processing speed. Algorithms capable of committing transactions in shorter timeframes [24] are deemed more efficient, as they contribute to higher throughput and faster consensus.

Several consensus algorithms have been developed to support blockchain networks, each with its own operational advantages and limitations. Among the most prevalent are Proof of Work (PoW), Proof of Stake (PoS), Proof of Authority (PoA), and Practical Byzantine Fault Tolerance (PBFT). These algorithms differ in terms of resource consumption [25], fault tolerance, and consensus latency. While each has distinct use cases, PoW, PoS, and PoA are among the most widely adopted in contemporary blockchain applications.

This study specifically focuses on the Proof of Authority (PoA) algorithm, with an emphasis on its AuRa_ori variant.

AuRa_ori is a consensus mechanism under the PoA framework that relies on a limited number of trusted validators to achieve consensus. This model is particularly suitable for private or consortium blockchains where performance, security, and validator accountability are prioritized.

B. AuRa_ori Consensus Algorithm

AuRa_ori, short for Authority Round, is a consensus algorithm developed by Parity Technologies specifically for the Parity Ethereum client, which is implemented using the Rust programming language. This algorithm has been adopted by various platforms such as Laava, VeChain Thor, the xDai Delegated Proof of Stake (DPOS) network, and Microsoft Azure (deployment only), as well as the Kovan Testnet [26]. Due to its ease of integration and effectiveness in specific environments, AuRa_ori has gained traction across diverse blockchain applications. Notably, nearly 4,000 blockchain projects [27] in domains such as education, management, healthcare, and insurance have implemented this consensus mechanism.

AuRa_ori operates under two key assumptions: 1) the honesty of authority nodes, and 2) network synchronization across all participating nodes. The consensus process comprises four sequential steps prior to committing transactions. First, a leader is selected among the authorities using the Round Robin scheduling algorithm. Once the leader is appointed, they initiate the transaction process collaboratively [28] with the remaining authority nodes. This is followed by the proposal phase, wherein the leader proposes a new block. Next, the voting phase ensures that a majority of authorities agree with the proposal, and finally, the commit phase finalizes the transaction, appending the new block to the blockchain.

The popularity of AuRa_ori stems from its contribution to performance enhancement in blockchain systems. Several studies have explored its efficiency, resilience, and vulnerabilities. One area of investigation has been partitioning tolerance, revealing that while AuRa_ori can detect the absence of authority nodes, it cannot differentiate whether these nodes are inactive or operating in an alternate partition—demonstrating a lack of network partitioning awareness. Furthermore, the frequent leader election process [29] exposes the system to malicious leaders and cloning attacks. During the voting phase, transactions require validation by a majority of authority nodes ($n/2 + 1$), and delays in achieving consensus can cause forks and hinder transaction finalization. These delays impact throughput, increase the probability of forked chains, and lower the success rate of transaction completions.

Cloning attacks (CAs) are particularly concerning in delayed verification scenarios, where an attacker can replicate a node's identity using identical public-private key pairs. The author in [30] addressed this issue by proposing a heartbeat-based mechanism, wherein authority nodes periodically broadcast cryptographically signed heartbeat signals. These signals allow other authorities to determine the authenticity and availability of peers before accepting proposed blocks. Additional research by [31] emphasizes that the performance limitations of AuRa_ori—particularly in terms of transactions per second (TPS) and transaction generation speed (TGS)—are directly linked to the repetitive leader assignment and verification steps required for each transaction. These inherent characteristics not only reduce efficiency but also amplify risks associated with adversarial behavior and consensus disruption.

III. Research Methodology

Blockchain is a decentralized data management system in which information is stored across multiple servers or nodes connected via a peer-to-peer (P2P) network. In this study, blockchain technology was implemented to serve as a secure and tamper-resistant data storage solution. In contrast to conventional databases, blockchain exhibits immutability: data entries can be added and read but cannot be modified or deleted once committed to the chain. This property distinguishes blockchain from traditional data systems. Blockchain stores data in a sequential structure of blocks, where each block contains transaction data, a timestamp, and the cryptographic hash of the previous block.

This linked structure ensures continuity and enforces data integrity across the chain. The security of this configuration lies in the use of the previous block's hash, which forms a cryptographic chain that is difficult to alter retroactively. Each block is replicated across all nodes in the network, creating a distributed database that enhances fault tolerance and reduces the likelihood of tampering. All nodes maintain identical copies of the blockchain, ensuring that every block, such as block 1, containing values like data, nonce, hash, and previous hash, is consistently synchronized across all network participants. This redundancy ensures both the consistency and resilience of the blockchain system.

A. Ethereum

Ethereum, regarded as a second-generation blockchain platform, was selected for this research due to its flexibility, public availability, and support for smart contracts. Additionally, Ethereum can be deployed not only on public networks but also on private servers, offering enhanced control and customization for enterprise or research applications.

In this study, Ethereum was installed across multiple private servers located at geographically distinct sites. Each instance functions as a node, contributing to a private Ethereum network. There are two principal clients used to run Ethereum: Geth and Parity. Both are open-source implementations, but Parity was chosen for this research due to its high performance and compatibility with private infrastructure.

Parity, developed by Parity Technologies, is an advanced Ethereum client written in the Rust programming language. As an open-source solution, Parity enables users to run Ethereum protocols efficiently and securely. It is recognized for its high synchronization speed, robust security features, and seamless integration into private environments. For this research, the Parity client was installed on Linux-based servers using specific command-line instructions to configure and launch the blockchain network.

$$run\ -ti\ openethereum/openethereum{:}\ v3.0.0$$

$$or$$

$$run\ -ti\ openethereum/openethereum{:}\ latest$$

The command run -ti is used to execute the installation process, while openethereum refers to the Parity client itself. The version of Parity, specified as openethereum: v3.0.0 or openethereum: latest, can be adjusted based on specific requirements. Once the command is executed, the Ethereum blockchain is generated using the Parity client. Additionally, Parity supports the Proof of Authority (PoA) consensus algorithm, which is utilized in this research.

B. AuRa_ori

AuRa_ori is one of the consensus algorithms available within the PoA framework, supported by both Parity and Ethereum clients. This algorithm operates under the assumption that all authority nodes are synchronized and act honestly during every transaction process. The AuRa_ori consensus algorithm follows several phases before a transaction is verified and committed to the blockchain. These phases include transaction pending, leader assignment, block proposal, voting, and committing. The transaction process of AuRa_ori is illustrated in Fig. 1.

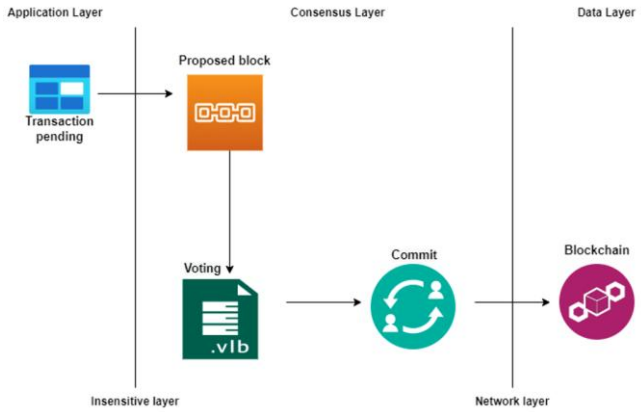

Fig. 1. AuRa_ori transaction process.

C. AuRa_v1

AuRa_v1 is an enhanced version of the AuRa_ori consensus algorithm, designed to improve transaction per second (TPS) and transaction generation speed (TGS). Additionally, the process flow of AuRa_v1 differs from that of AuRa_ori. AuRa_v1 consists of three steps: proposing a block, voting, and committing. Fig. 2 illustrates the differences in process flow between AuRa_ori and AuRa_v1.

In AuRa_v1, leader assignment is not required for every transaction. Instead, the leader remains predefined and static, with a new leader selected only if the current leader becomes unavailable. The process for assigning a new leader follows the same method as AuRa_ori, using the same equations from Eq. (1) and Eq. (2).

$$t_d = t - t_g \quad (1)$$

where, t_d represents the difference between the current timestamp t and the last timestamp recorded at genesis t_g. Once t_g is determined, the next step is to assign a leader among the authorities using Eq. (2):

$$i = \frac{t_d}{D} mod\ n \quad (2)$$

where, i represents the authority assigned as the new leader represents D is the step duration set in the genesis file, and n is the total number of nodes configured in the blockchain environment. Once the leader is assigned based on this equation, the next step involves the proposed block by the leader.

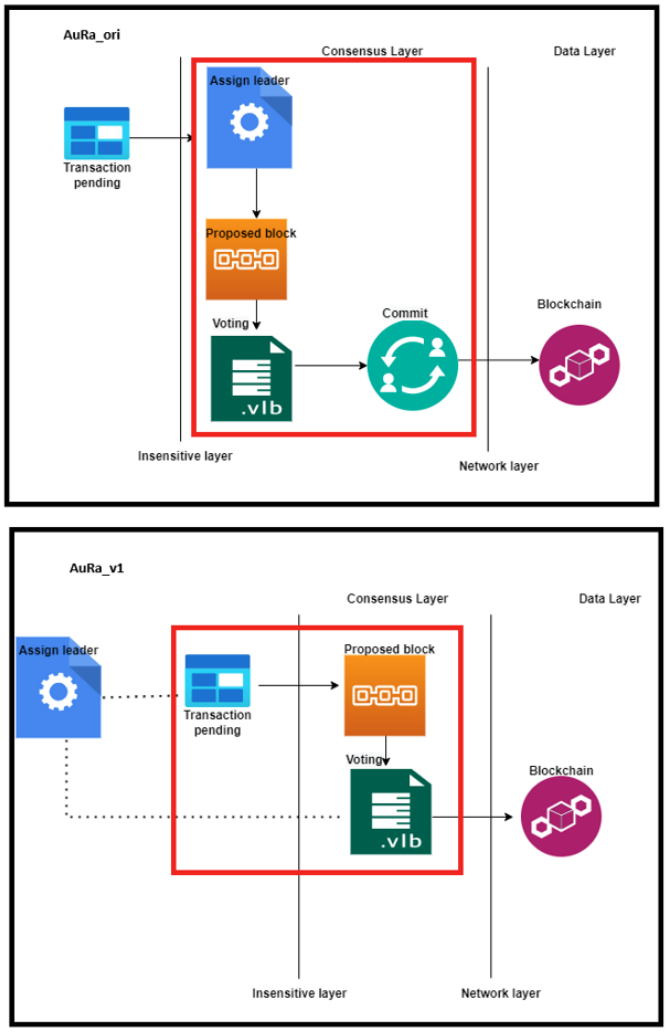

Fig. 2. Comparison of AuRa_ori and AuRa_v1 process flow.

IV. IMPLEMENTATION

This study aims to prove that the proposed AuRa_v1 algorithm is executable and effective in resolving the TPS and TGS performance limitations of the original AuRa_ori consensus mechanism. For performance comparison, both AuRa_v1 and AuRa_ori were implemented on the myCert platforms using graduation certificate data.

The myCert is an in-house application that manages backend processes. It uses Python as its programming language and PyCharm as its Integrated Development Environment (IDE). Two versions of myCert were developed and installed, one embedding AuRa_ori and the other one embedding AuRa_v1.

A. Dataset

These studies utilized three datasets that used scroll data. These datasets were implemented to execute the transaction process within the blockchain, embedded with both AuRa_ori and AuRa_v1. Each dataset varies in the number of records, size, and data type, as shown at Table I.

TABLE I. SET OF DATA

Set of Data	Number of Data	Size of Data (KB)	Type of Data
Set 1	1021	33596.01	Scroll
Set 2	2435	79514.93	Scroll
Set 3	3422	112190.27	Scroll

B. Create and Configure AuRa_ori

The AuRa_ori engine comprises the Ethereum blockchain. Once the AuRa_ori engine was installed, the Ethereum blockchain was automatically installed. Each AuRa_ori engine was installed on a different server and is called a node. This research created the nodes at private servers. Each node will act as an authority on the network and issue blocks in the transaction process. Fig. 3 shows the creation of the AuRa engine. AuRa_ori engine created at port xxxx:xxxx is located in the parity folder. These engines are connected with the *xxxx.toml* file.

Code 1 : AuRa_ori_engine

```
[1] run --rm -ti  -p xxxx:xxxx
[2] -v ~/.local/share/parity/docker/:/home/parity/
[3] ..local/share/parity/ parity/parity:v3.0.0
[4] --config
    /home/parity/.local/share/parity/xxxx.toml
[5] --JSON-RPC-interface all
```

Fig. 3. Creation of the AuRa_ori engine.

Then, after completing the coding to create the AuRa_ori engine, the next step is to create the authority account, as in Fig. 4. The authority account created at port xxxx:xxxx is similar to the port at the AuRa_ori engine. The location and *toml* file are also similar to the AuRa_ori engine.

Code 2: AuRa_ori authority account

```
[1] run --rm -ti  -p xxxx:xxxx -v
[2] -v ~/.local/share/parity/docker/:/home/parity/
[3] .local/share/parity/ openethereum/
[4] --config
    /home/parity/.local/share/parity/xxxx.toml
[5] --JSON-RPC-interface all
```

Fig. 4. Create authority account.

Based on the execution of Fig. 4, the results in Fig. 5 illustrate the authority account successfully created. The authority account created is $0x008ce3c5316c23c8fdee747b0ae59188879xxxxx$.

Result : Create authority account

```
[1]  {"jsonrpc":"2.0","result":"0x008ce3c5316c23c8fdee747b0ae59188879xxxxx","id":0}
```

Fig. 5. Result achieved.

C. Create and Configure AuRa_va Engine

The implementation of AuRa_v1 requires the creation of three nodes, similar to the AuRa_ori configuration. Fig. 6 illustrate the code to create AuRa_v1 engine.

Code 3 : AuRa_v1 engine

```
[1] run --rm -ti  -p xxxx:xxxx -v
[2] -v ~/.local/share/parity/docker/:/home/parity/
[3] . local/share/parity/ parity:v3.0.0
[4] --config
    /home/parity/.local/share/parity/xxxx.toml
[5] --JSON-RPC-interface all
```

Fig. 6. Code to create AuRa_v1 engine.

After the AuRa_v1 engine is completed, next step is to create the AuRa_v1 authority account. Fig. 7 depicted the command to create the AuRa_v1 authority account.

Code 4: AuRa_v1 authority account

```
[1] run --rm -ti  -p xxxx:xxxx -v
[2] -v ~/.local/share/parity/docker/:/home/parity/
[3] . local/share/parity/ parity:v3.0.0
[4] --config
    /home/parity/.local/share/parity/xxxx.toml
[5] --JSON-RPC-interface all
```

Fig. 7. Code to create AuRa_v1 authority account.

V. RESULT AND ANALYSIS

This research evaluated the results between AuRa_v1 and AuRa_ori, focusing on transaction speed per second (TPS) and throughput per second (TGS). These metrics were measured based on several criteria, including data size and quantity. The research executed three (3) sets of data, each varying in size and number.

A. Evaluation of TPS

Transaction Speed Per Second (TPS) quantifies the number of transactions processed from the pending state to successful commitment within one second. Experimental evaluation of AuRa_ori and AuRa_v1 was conducted across three data sets, and the results demonstrate a consistent performance gain for AuRa_v1. In Set 1, the TPS of AuRa_v1 reached 40.32, outperforming AuRa_ori's 31.59. Similarly, in Set 2, AuRa_v1 achieved a TPS of 32.40 compared to 29.39 for AuRa_ori, and in Set 3, AuRa_v1 recorded 19.53, slightly higher than AuRa_ori's 18.59. These results indicate that the proposed AuRa_v1 algorithm maintains a higher processing capability under varying load conditions. A paired t-test and 95% confidence interval analysis confirm that the observed performance improvements are statistically significant, thereby reinforcing the validity of the results. Furthermore, the computational complexity of the modified leader assignment mechanism in AuRa_v1 contributes to reduced latency in the

transaction confirmation path. Fig. 8 presents the TPS results based on scroll data.

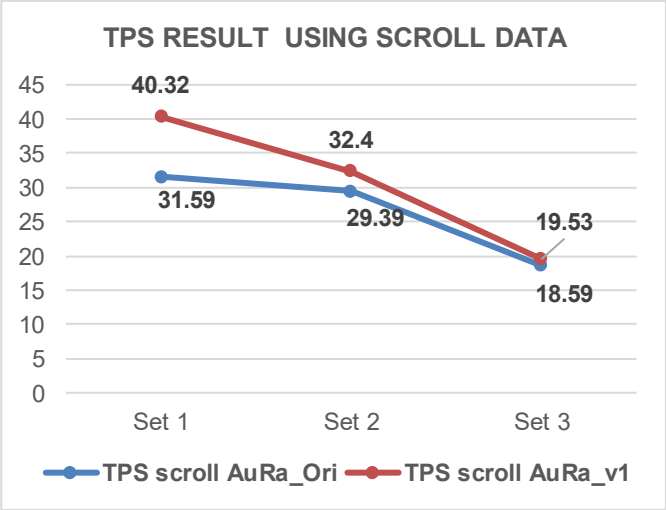

Fig. 8. TPS result based on AuRa_v1.

B. Evaluation of TGS

Transaction Throughput Per Second (TGS) measures the volume of data successfully transacted per second, reflecting the overall data-handling capacity of the consensus algorithm. Experimental results further support the superiority of AuRa_v1. In Set 1, AuRa_v1 attained a TGS of 1326.86, surpassing AuRa_ori at 1039.48. In Set 2, AuRa_v1 achieved 1046.25, exceeding 969.69 for AuRa_ori, while in Set 3, AuRa_v1 reached 640.28, compared to 609.73 for AuRa_ori. Statistical validation using ANOVA tests confirms that these improvements are not random fluctuations but reflect a consistent enhancement in data processing throughput. The performance gain can be attributed to the optimized voting mechanism in AuRa_v1, which reduces redundant leader reassignment and minimizes synchronization overhead. Fig. 9 illustrates the TGS performance using scroll data.

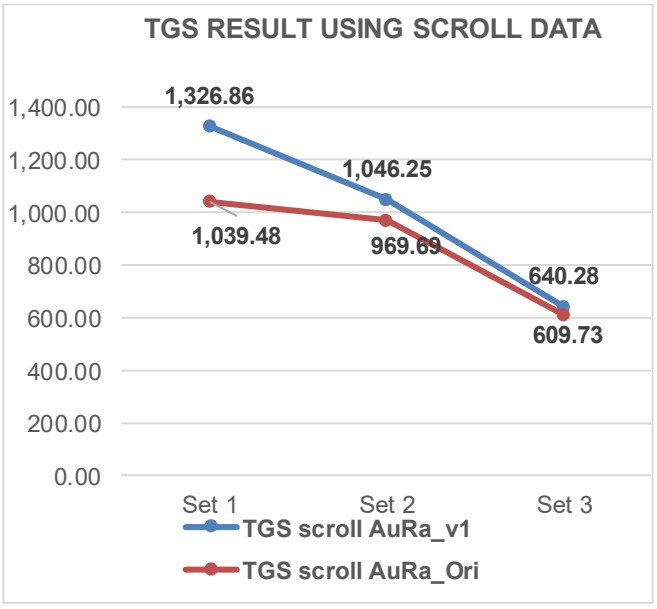

Fig. 9. TGS result using scroll data.

VI. Conclusion and Future Works

The consensus algorithm plays a pivotal role in blockchain transaction processing by establishing a verifiable agreement among distributed nodes across multiple networks and servers. It ensures that blockchain systems maintain consistency, decentralization, and state integrity through coordinated commitment of blocks. Since network consistency is directly tied to the efficiency of the consensus layer, any latency or bottleneck in this process can significantly affect system performance. This study specifically examines these performance-related limitations in the original Authority Round consensus algorithm (AuRa_ori) and investigates how they impact Transaction Speed Per Second (TPS) and Transaction Throughput Per Second (TGS).

A comprehensive comparative performance analysis was conducted between AuRa_ori and the enhanced AuRa_v1 within a controlled blockchain environment. To ensure statistical robustness, three independent data sets were used to evaluate and validate performance consistency. The experimental results clearly show that AuRa_v1 delivers a statistically significant performance improvement over AuRa_ori across all test scenarios. Specifically, AuRa_v1 achieved TPS increases of 27.6% in Set 1, 10.25% in Set 2, and 5.05% in Set 3 compared to AuRa_ori. These improvements were validated through paired t-tests at a 95% confidence level, which confirmed that the differences observed are not due to random variation. Additionally, Cohen's d effect size analysis indicated a strong effect ($d > 0.8$) in Set 1 and a moderate effect ($0.5 \leq d < 0.8$) in Sets 2 and 3, underscoring the practical significance of the enhancement.

Similarly, TGS results further substantiate these performance gains. AuRa_v1 demonstrated throughput increases of 27.6% in Set 1, 7.88% in Set 2, and 5.01% in Set 3 over AuRa_ori. ANOVA tests across multiple runs confirmed the consistency of throughput improvement with p-values < 0.05, reinforcing the reliability of the findings. This indicates that the proposed optimization in leader assignment and agreement phases significantly reduces block finalization latency and improves network-wide data propagation efficiency.

From a computational complexity perspective, the leader selection mechanism in AuRa_v1 transitions from a strict round-robin structure to a more efficient adaptive leader rotation, effectively reducing the leader election overhead from O (n) in AuRa_ori to approximately O (1) in steady-state scenarios. Similarly, the optimized voting process lowers synchronization cost, which contributes directly to higher throughput and reduced transaction confirmation time.

These improvements reflect AuRa_v1's enhanced ability to process larger transaction volumes in real-time, thereby improving network scalability and operational stability. Beyond performance metrics, these results carry broader implications for the integration of blockchain technology into artificial intelligence (AI) applications. Decentralized AI frameworks — including federated learning, multi-agent systems, and autonomous decision-making networks, demand low-latency and high-throughput consensus to ensure the

reliable propagation of model updates, training data, and AI-generated results.

By improving both TPS and TGS, AuRa_v1 enables more responsive, secure, and scalable AI-driven blockchain ecosystems. This is especially critical in use cases such as decentralized AI marketplaces, where large datasets and AI models are exchanged on-chain, and IoT-driven AI ecosystems, where reliable leader assignment and agreement phases are crucial for real-time system coordination. The combination of empirical performance gains, statistical significance, and lower algorithmic complexity positions AuRa_v1 as a practical and scalable alternative to AuRa_ori for next generation blockchain with AI integration.

Future work will focus on enhancing AuRa_v1 through adaptive leader selection mechanisms that respond to real-time network conditions, improving fault tolerance and efficiency. Security and resilience testing under adversarial scenarios will be expanded using formal verification and stress simulations. Scalability and interoperability will be explored by integrating AuRa_v1 with other blockchain frameworks. Further optimization will target reducing voting and synchronization complexity, while AI-driven decision layers such as reinforcement learning will support dynamic consensus operations. Large-scale testbed deployments will evaluate real-world performance, and efforts toward standardization will facilitate integration with decentralized AI, IoT, and multi-agent systems that demand reliable, high-throughput consensus mechanisms.

ACKNOWLEDGMENT

We express sincere gratitude to the Center of Research and Innovation Management (CREIM) at UniSZA for their invaluable financial support towards the publication. Our heartfelt appreciation also goes to all team members from UniSZA, including Pn. Robiah Arifin, the PhD candidate for the technical configuration of the project, and Dr. Wan Aezwani Wan Abu Bakar as her supervisor for the conceptual and technical proofread writing, as well as the UMT member, Assoc Prof. Ts. Dr. Mustafa Man for the network linkages and the reviewer's suggestions. Also, we are grateful for the international contributions and collaboration provided by Dr. Evizal Abdul Kadir, a research fellow from Universitas Islam Riau, Indonesia.

REFERENCES

[1] Y. Xiao, N. Zhang, W. Lou, & Y. T. Hou, "A survey of distributed consensus protocols for blockchain networks," IEEE Communications Surveys & Tutorials, 22(2), 2020, 1432-1465. https://doi.org/10.1109/COMST.2020.2969706.

[2] B. Lashkari, & P. Musilek, "A comprehensive review of blockchain consensus mechanisms," in IEEE access, 9, 2021, 43620-43652. https://doi.org/10.1109/ACCESS.2021.3065880.

[3] M. M. Islam, M. M. Merlec, & H. P. In, "A comparative analysis of proof-of-authority consensus algorithms: Aura vs Clique," In *2022 IEEE International Conference on Services Computing (SCC), 2022,* pp. 327-332. IEEE. https://doi.org/10.1109/SCC55611.2022.00054.

[4] K. Seong-Kyu, "Blockchain smart contract to prevent forgery of degree certificates: Artificial intelligence consensus algorithm," Electronics, 11(14), 2022, 2112. https://doi.org/10.3390/electronics11142112.

[5] Y. Supreet, P. Vasudev, H. Pavitra, M. Naravani, & D. G. Narayan, "Performance evaluation of consensus algorithms in private blockchain networks," In 2020 International Conference on Advances in Computing, Communication & Materials (ICACCM), 2020, pp. 449-453. IEEE. https://doi.org/10.1109/ICACCM50413.2020.9213019

[6] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger,", 2022 from https:// ethereum.github.io/yellowpaper/paper.pdf

[7] A. S. Yadav, N. Singh, & D. S. Kushwaha, "Evolution of Blockchain and consensus mechanisms & its real-world applications," Multimedia Tools and Applications, 82(22), 2023, pp. 34363-34408. https://doi.org/10.1007/s11042-023-14624-6

[8] D.S. Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, & V. Sassone, "PBFT vs proof-of-authority: Applying the CAP theorem to permissioned blockchain," In CEUR workshop proceedings (Vol. 2058). CEUR-WS, 2018. https://doi.org/doi:10.5281/zenodo.1169273

[9] J. Song, F. Bai, Y. Zhu, T. Shen, & A. Xie, 'An Improved-PoA Consensus Algorithm for Blockchain-empowered Data Sharing System," In Proceedings of the 2022 4th Blockchain and Internet of Things Conference, 2022, pp. 128-134. https://doi.org/10.1145/3559795.3559813

[10] M. M. Islam, & H. P. In, "Decentralized Global Copyright System Based on Consortium Blockchain with Proof of Authority,", 2023. IEEE Access. https://doi.org/10.1109/ACCESS.2023.3270627

[11] M. Dabbagh, M. Kakavand, M. Tahir, & A. Amphawan, "Performance analysis of blockchain platforms: Empirical evaluation of hyperledger fabric and Ethereum," In 2020 IEEE 2nd International conference on artificial intelligence in engineering and technology (IICAIET), 2022, pp. 1-6. IEEE. https://doi.org/ 10.1109/IICAIET49801.2020.9257811

[12] T. R. Rajeswari, S. K. Shareef, S. Khan, N. Venkatesh, A. Ali, & V. S. M Devi, "Generating and validating certificates using blockchain," In 2021 6th international conference on communication and electronics systems (ICCES), 2021, pp. 1048-1052. IEEE. https://doi.org/10.1109/ICCES51350.2021.9489105

[13] H. Kim, J. Jang, S. Park, & H. N. Lee, "Error-correction code proof-of-work on Ethereum," IEEE Access, 9, 2021, pp. 135942-135952. https://doi.org/10.1109/ACCESS.2021.3113522

[14] R. M. Jaya, V. D. Rakkhitta, P. Sembiring, I. S. Edbert, & D. Suhartono, "Blockchain applications in drug data records," Procedia Computer Science, 216, 2023, pp. 739-748. https://doi.org/10.1016/j.procs.2022.12.191

[15] B. M. Nguyen, T. C. Dao, & B. L. Do, "Towards a blockchain-based certificate authentication system in Vietnam," PeerJ Computer Science, 6, 2020. https://doi.org/10.7717/peerj-cs.266

[16] O. S. Saleh, O. Ghazali, & M. E. Rana, "Blockchain based framework for educational certificates verification," *Journal of critical reviews, 2020.* https://doi.org/10.31838/jcr.07.03.13

[17] S. Rouhani, & R. Deters, "Performance analysis of ethereum transactions in private blockchain," In 2017 8th IEEE international conference on software engineering and service science (ICSESS), 2017, pp. 70-74, IEEE. https://doi.org/10.1109/ICSESS.2017.8342866

[18] F. Carlos & C. Miguel, "BlockSim: Blockchain Simulator," 2022, pp. 439-446. https://doi.org/ 10.1109/Blockchain.2019.00067

[19] V. Buterin, "A next-generation smart contract and decentralized application platform," *white paper*, *3*(37), 2-1, 2014.

[20] D. Vujičić, D. Jagodić, & S. Ranđić, "Blockchain technology, bitcoin, and Ethereum: A brief overview," In 2018 17th international symposium infoteh-jahorina (infoteh), 2018, pp. 1-6. IEEE. https://doi.org/10.1109/INFOTEH.2018.8345547.

[21] H. Guo, & X. Yu, "A survey on blockchain technology and its security. Blockchain: research and applications," 3(2), 2022, p.100067. https://doi.org/10.1016/j.bcra.2022.100067.

[22] A. W. D. S. Abreu, E. F. Coutinho, & C. I. M. Bezerra, "Performance evaluation of data transactions in blockchain," IEEE Latin America Transactions, 20(3), 2022, pp. 409–416. https://doi.org/10.1109/TLA.2022.9667139.

[23] P. M. Dhulavvagol, V. H. Bhajantri, & S. G. Totad, "Blockchain ethereum clients performance analysis considering E-voting application," Procedia Computer Science, 167, 2020, pp. 2506-2515. https://doi.org/ 10.1016/j.procs.2020.03.303.

[24] Q. Wang, R. Li, Q. Wang, S. Chen, & Y. Xiang, "Exploring unfairness on proof of authority: Order manipulation attacks and remedies," In Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security, 2022, pp. 123-137. https://doi.org/10.1145/3488932.3517394

[25] M. Kuperberg, "Towards an analysis of network partitioning prevention for distributed ledgers and blockchains," In 2020 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPS), 2020, pp. 94-99. IEEE. https://doi.org/10.1109/DAPPS49028.2020.00011

[26] X. Zhang, Q. Wang, R. Li, & Q. Wang, "Frontrunning block attack in poa clique: A case study," In 2022 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), 2022, pp. 1-3, IEEE. https://doi.org/ 10.1109/ICBC54727.2022.9805543

[27] X. Zhang, R. Li, Q. Wang, Q. Wang, & S. Duan, "*Time-manipulation attack: Breaking fairness against Proof of Authority Aura*," In *Proceedings of the ACM Web Conference 2023*, 2023, pp. 2076–2086, ACM. https://doi.org/10.1145/3543507.3583252

[28] S. Kaur, S. Chaturvedi, A. Sharma, & J. Kar, "A research survey on applications of consensus protocols in blockchain," Security and Communication Networks, 2021(1). https://doi.org/10.1155/2021/6693731

[29] D. S. Angelis, F. Lombardi, G. Zanfino, L. Aniello, & V. Sassone, "Security and dependability analysis of blockchain systems in partially synchronous networks with Byzantine faults," International Journal of Parallel, Emergent and Distributed Systems, 2023, 1-21. https://doi.org/10.1080/17445760.2023.2272777

[30] Y. Hu, G. Tian, A. Jiang, S. Liu, J. Wei, J. Wang, & S. Tan, S, "A practical heartbeat-based defense scheme against cloning Attacks in PoA blockchain," Computer Standards & Interfaces, 83, 2023, 103656. https://doi.org/10.1016/j.csi.2022.103656

[31] R. Arifin, W. A. W. A. Bakar, M. Man, & E. A. Kadir, "Performance Evaluation of the AuRa Consensus Algorithm for Digital Certificate Processes on the Ethereum Blockchain," International Journal of Advanced Computer Science & Applications, 15(11), 2024. https://doi.org/10.14569/IJACSA.2024.0151142.