Pertanika J. Sci. & Technol. 34 (2): 671 - 693 (2026)

SCIENCE & TECHNOLOGY

Journal homepage: http://www.pertanika.upm.edu.my/

Measuring the Performance of Blockchain Systems with an Improved Consensus Mechanism using Scalability Metrics

Kolli Lalitha Kumari and Lalitha Surya Kumari*

Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, 500075 Hyderabad, Telangana, India

ABSTRACT

Blockchain is an efficient method to manage and secure data, but scalability remains a limitation. The proposed work concentrates on a novel consensus algorithm, Proof of Useful Work-Authorisation-Storage Availability. The work emphasises the scalability issues and trust-related issues in blockchain-based smart contracts. The primary step in the proposed technique is to verify authorisation by the hash code of the preceding block to generate a digital signature. After information is legitimised, check whether enough storage space is available or not. Transactions are recorded to a block only after they are validated and there is enough space. A transaction is included in a block only after it is mined. To compare how efficient the proposed consensus is, in terms of energy consumption, delay, and the number of transactions it can process, the study uses Python and Solidity. The new method reduced computational energy consumption by 39% and increased transaction capacity by 17%, and decreased network delay by 49% compared to the proof of useful work algorithm. This is an indication that the efficient consensus will help us scale without compromising security and decentralisation. The contribution is beneficial in determining which nodes to apply in the consensus layer. The system can help with banking applications and supply chain systems that are blockchain-based. It highlights three key points: scalability, low delay, and efficient consensus, which are necessary for these systems to function properly.

Keywords: Blockchain, consensus mechanism scalability, computational energy, latency, transaction throughput

ARTICLE INFO

Article history:
Received: 24 May 2025
Accepted: 04 February 2026
Published: 01 April 2026

DOI: https://doi.org/10.47836/pjst.34.2.04

E-mail addresses:
2212030001@kluniversity.in (Kolli Lalitha Kumari)
vlalithanagesh@gmail.com (Lalitha Surya Kumari)
* Corresponding author

INTRODUCTION

Blockchain technology has applications in various fields, such as banking, supply chain management, and the healthcare industry (Andrew et al., 2023).

e-ISSN: 2231-8526

As the network grows in blockchain technology, scalability problems arise. Scalability solutions try to address this difficulty while still ensuring the decentralised nature of the blockchain (Zhang et al., 2021). It is also essential in ensuring interoperability among various blockchain networks. With the increase in the use of blockchain, the ability to exchange information and communicate smoothly across multiple networks is becoming more important. From Figure 1 the working components of blockchain to complete a transaction have been presented. Before adding a transaction to a particular blockchain, it must be verified through mining, consensus, and hashing. These components also enhance the overall performance of a blockchain. Miners validate transactions to be recorded on the global ledger. It should be converted into a fixed-size string of characters when hashing data. Validating blocks of data in a blockchain consensus protocol will be helpful. Based on the review of available literature, scalability (Deng et al., 2024; Fajri et al., 2022; Liu et al., 2021; Ucbas et al., 2023) is a significant issue for handling many nodes that will be added to a blockchain network.

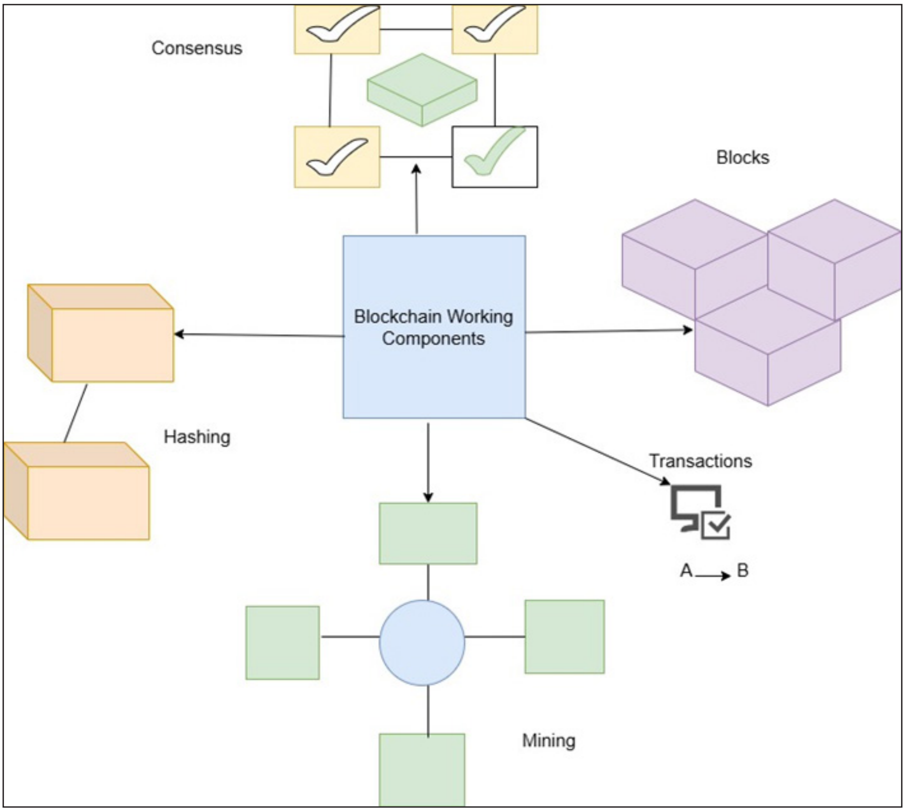


Figure 1. Blockchain working components

Scalability

The ability of a network to perform an increase in the number of transactions without compromising overall efficiency is known as scalability. Because scalability ensures that a network can handle a higher volume of transactions per second. Congestion in a blockchain network can be caused by high demand and limited scalability, resulting in increased transaction fees. Scalability also allows transactions to be more efficient and faster, thus lowering transaction costs. A scalable blockchain provides low latency and fast transaction confirmation, resulting in a more efficient user experience. Strong transaction management will be needed for widespread acceptance of blockchain technology.

Consensus Algorithms

Blockchain networks use the Proof of Work (PoW) consensus protocol to obtain consensus on the state of the distributed ledger as well as protection against nefarious behaviour, such as double spending. In this consensus, the miners race to be the first to solve a computationally difficult puzzle to provide a hash that meets a criterion, such as having a certain number of leading zeros. It is practically not possible to change the blockchain because it requires huge computational resources and energy. After a solution is found, the miner announces the solution or proof to the network, and other nodes can easily verify the proof. The first miner to solve a puzzle correctly is rewarded with a cryptocurrency, and their proposed block is attached to the network.

Proof of Storage (PoS) is a blockchain consensus algorithm that uses storage capacity rather than computational capacity to secure the network. Miners commit idle disk space to hold cryptographic data, e.g., plots or challenges. To produce a block, a miner proves the use of some storage on the network by submitting valid proof computed from their held plots. Proof of Storage is more efficient than Proof of Work because it requires less computational power, instead leveraging the high availability of disk space.

In blockchain technology, the Proof of Authority (PoA) comes to consensus through the delegation of the process to a list of predetermined validators selected according to their verified identity, credibility, and ability to verify transactions. It is based on computational power or token ownership. PoA trusts the validators more, typically verified parties, such as organisations or people with a high reputation and credibility. Validators generate new blocks and validate transactions to maintain high throughput. This approach lowers energy usage and latency when compared to other algorithms specifically meant to apply to consortium and private blockchains. PoUW is an innovative blockchain consensus protocol that addresses the shortcomings of Proof of Work (PoW) by repurposing miner computational work towards meaningful real-world and tangible problems. Miners within a PoW network are performing beneficial activities, such as developing machine learning models, performing scientific simulations, or cracking hard systems, instead of computing random cryptographic functions. In being effort-driven computationally, this process secures the blockchain but also channels the effort towards beneficial outputs. PoUW is a more efficient and sustainable blockchain consensus choice because it keeps the security and decentralisation benefits of PoW but significantly lowers energy expenditure.

In this paper, the proposed work aims to develop an improved consensus algorithm and measures such as computational energy, latency, and transactional throughput to help alleviate the current limitations. This paper's structure elaborates on the limitations of existing work, including scalability. The second part focuses on a proposed methodology, based on the improved consensus algorithm and the PoUWAS architecture, as well as scalability metrics and an algorithm. The third part mainly focuses on measures related to scalability, then compares the existing PoUW and proposed PoUWAS in terms of latency, transaction throughput, and computational energy.

RELATED WORKS

To minimise the transaction cost and improve scalability, a new fault-tolerant method was proposed (Otsuki et al., 2021). An optimised blockchain solution was developed to minimise the computational complexity and improve the performance (Ghamdi, 2022). A brief discussion on PoW, PoS, DPoS, and PoA was given, along with the performance parameters (Kaur et al., 2021). The study analyses how long transactions take to be confirmed in proof-of-work blockchains like Bitcoin (Malakhov et al., 2023).

By enabling fast, low-cost, and accountable decision-making for private or consortium systems like corporate governance, supply chains, or smart cities, a model was proposed based on PoA (Manolache et al., 2022). Improved PBFT consensus reduces communication overhead and enhances scalability while maintaining strong fault tolerance (Yang et al., 2022). In summary, scalability is the main issue for improving blockchain performance. Various techniques, including sharding and consensus protocols, are available to address this issue. Particularly, enhanced consensus algorithms are also essential to maintain the network as scalable. Latency, transactional throughput, and computational energy are a few scalability metrics to maintain low latency, greater transactional throughput, and minimal computational energy to overcome the scalability issue.

Massive IoT Data Collection (MIDC) system was mproposed to improve efficiency, security, mand trust in mlarge-scale heterogeneous WSNs (Zhang et al., 2022). The trade-offs between security, decentralisation, and scalability in Industry 4.0 applications of blockchain technology are examined, as well as how important consensus processes are to preserving security and functionality, particularly in permissioned blockchains (Nasir et al., 2024). ALB-Chain, a scalable blockchain system that uses lightweight Account Aggregation Messages (AAM) and a replica account method to balance shard workloads by optimising shard performance, and the MLPA dynamic partitioning technique outperforms current systems in terms of throughput and latency (Wu & Hua, 2025). Blockchain-based healthcare access control frameworks with an emphasis on smart contract-driven security and privacy-preserving strategies were proposed (Tawfik et al., 2025).

According to Table 1, PoS (Sun et al., 2021) decreased power consumption but created fairness problems. PoUW impacts the blockchain consensus activity by applying its effective scalability and low-energy-consuming technique (Daely et al., 2024). Related work has explored how blockchain technology addresses topics such as scalability. The proposed PoUWAS consists of mining, authorisation, and storage availability aspects. This work limits the scalability issue and highlights the comparison between existing PoUW with the proposed work in terms of latency, transactional throughput, and computational energy. This technique concentrates on enhancing the PoUW algorithm. In the end, it concluded with future enhancements and research directions.

Table 1
Algorithm 1: Findings related to existing system

Author & Year	Consensus Algorithm	Key Features	Limitations	Outcome Performance
(Sun et al., 2021)	Delegated Proof of Stake (DPoS)	Elected validators, fast block creation	Partial centralization centralisation	Up to 1,000 TPS, low latency
(Daely et al., 2024)	Proof of Useful Work (PoUW)	Computational efficiency, sustainability	Work verification complexity	Moderate scalability, reduced energy use

Problem Statement

Blockchain technology has a strong basis in decentralised systems, but still, real-time applications have limitations in terms of scalability and energy consumption, particularly in Proof of Work (PoW). To integrate mining-related activities with scientific solutions and AI model training, PoUW came into existence to address the traditional barriers. Still, these techniques lead to high energy consumption, and issues related to task verification complexity, adaptive scalability, and real-time responsiveness are lacking in current PoUW-related applications. A PoUW consensus that not only assures security but also distributes work across nodes to reduce latency and increase computing energy is required to address these gaps. To provide scalable, energy-efficient, and real-time decentralised applications, the proposed work presents an improved PoUW model that combines blockchain mining with security and authorisation. This method offers the best alternative solution for decentralised applications of the future.

Research Gaps

In healthcare and the Internet of Things, the current limitation is the adoption of blockchain technology and latency-related issues. Because existing consensus protocols either achieve speed at the expense of decentralisation (like PoS and DPoS) or suffer from poor scalability and high energy consumption (like PoW), the main gap is the inability to simultaneously solve the blockchain trilemma (decentralisation, security, and scalability). The lack of adaptability and intelligent node involvement is a crucial second gap. Most protocols employ static rules and do not allow for dynamic node selection or validation process adjustments based on trust metrics or real-time network conditions. To make blockchain a viable solution for high-volume, real-time enterprise systems, there is a general research need for flexible, energy-efficient hybrid consensus algorithms that can intelligently manage node resources, maintain high throughput, and provide low latency.

Novelty

To overcome the limitations, a composite blockchain paradigm should be constituted that synergistically integrates the security strength of the cryptographic consensus with the adaptivity of the improved algorithms. The presented solution is to incorporate a node selection process that uses improved consensus techniques to avoid duplication and improve block generation. It is needed for improved throughput, lower energy use, and quicker transaction confirmation times. This is accomplished by modelling the described methodology in Python and Solidity, deploying the model on a local Ethereum Remix, and testing the performance of the model based on energy cost, latency, and block validation time.

Impact

The proposed consensus technique concentrates on making the blockchain efficient, highly fault-tolerant, and less energy-consuming, which leads to the blockchain being scalable. This consensus technique will be the best alternative to apply to real-time applications to address the traditional barriers. The proposed work concentrates on mining-related activities and authorisation, and storage availability checks also. To maintain data efficiency and security, this method will be the best fit to sustain.

Organisational Chart of the Proposed Methodology

Initially introduced the gaps related to existing consensus algorithms, the need for improvement, and an overview of the proposed consensus algorithm. Then it highlights the steps in the improved consensus algorithm as well as its architecture, including node authorisation, selection, creation of a block, validation, and propagation of data. Then the flowchart describes node-to-node interactions and a validation process for the data and how a block gets added. The proposed consensus algorithm is described in the next step by outlining how node joining, transaction creation, broadcasting, and validation. Then it concludes with a mathematical analysis on energy consumption. Figure 2 presented clearly the organisational chart of the proposed methodology.

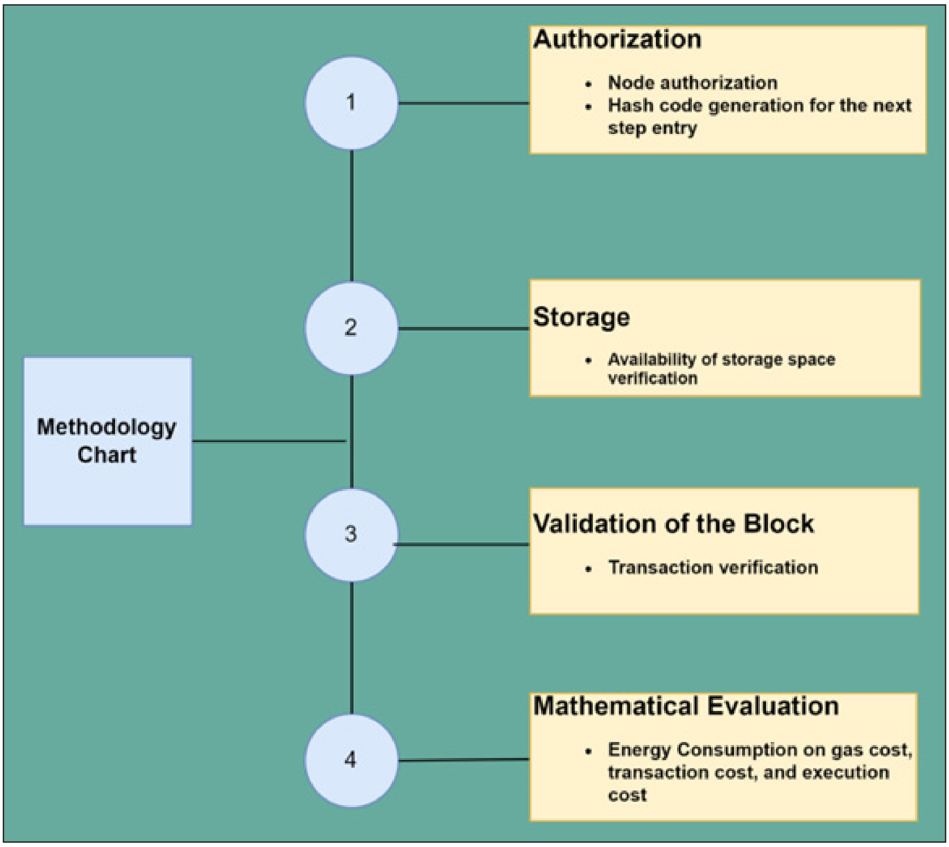


Figure 2. Organisational chart of proposed methodology

PROPOSED METHODOLOGY

Introduction

This approach aimed at designing and developing an improved consensus algorithm to perform a comparative analysis based on the existing algorithms. Based on the parameters, computational energy, latency, and transactional throughput scalability were measured. The motive of this proposed work is to minimise the scalability issues. Apart from this, it is also against fraudulent data on the network, which is lagging in the existing PoUW. Also, the existing algorithm consumes lots of computational power and storage space.

The proposed work will limit the problems and enhance the performance of a network. Figure 3 shows the architecture of the proposed algorithm, PoUWAS. Whenever a block is created, it is added to the blockchain. A specific procedure must be followed to add a block to the blockchain. It checks the transaction's authorisation, storage capacity, and block validation. This proposed architecture combines all three modules: authorisation, storage procedures, and verification.

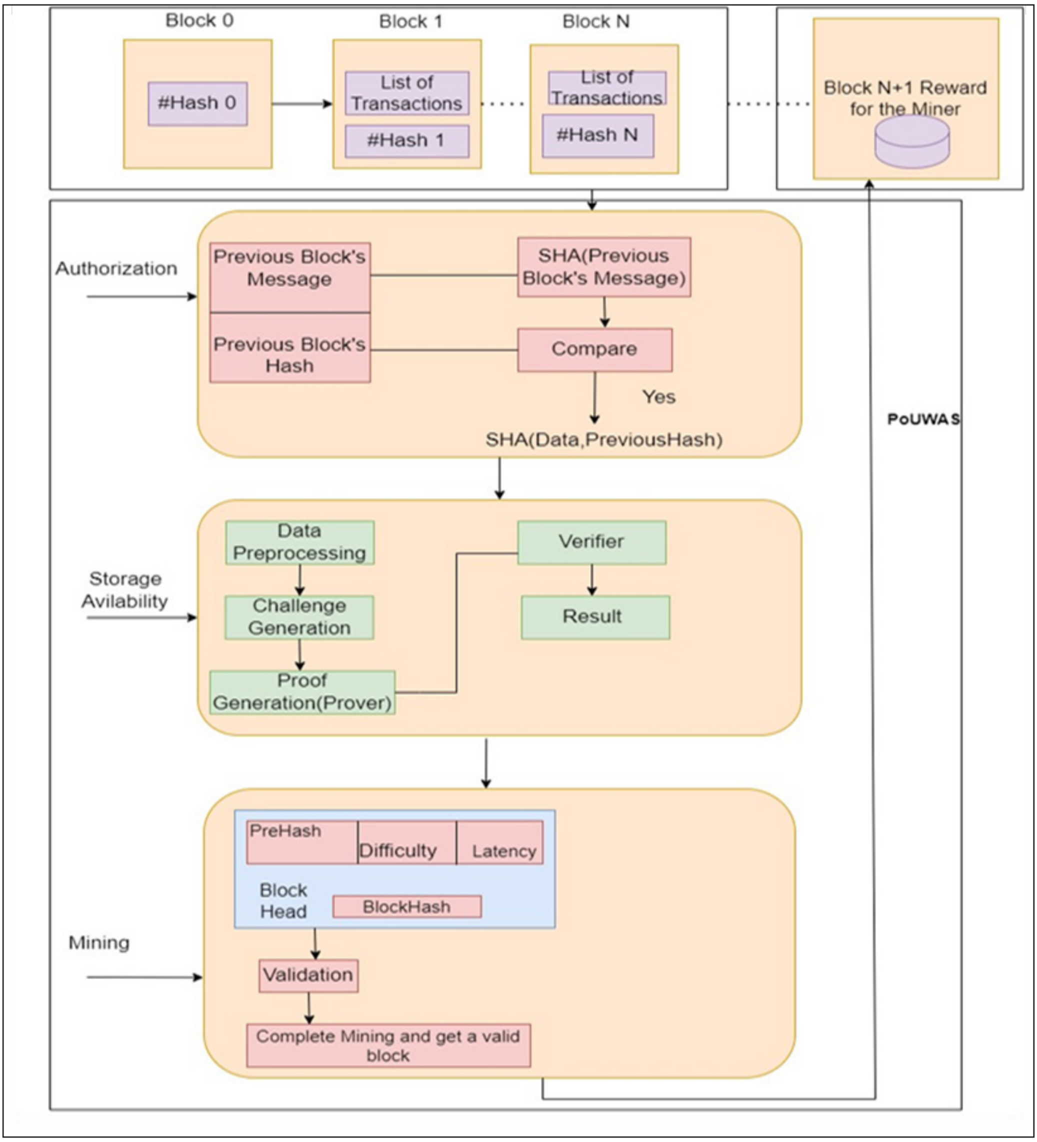


Figure 3. Improved PoUW consensus algorithm architecture

Authorisation

The authorisation method enhances blockchain security by reducing the likelihood of fraudulent activity. Any malicious activity risks harming the real-world reputations of authorisation networks and subjecting them to sanctions, such as revoking their authority, because they depend on a few verified and identifiable validators. Due to this accountability, the validator is discouraged from engaging in dishonest activities, such as altering transaction history or attempting to launch a double-spend attack. The authorisation function takes the following steps. It requires three parameters to proceed, which are the data of the previous block, the hash code of the previous block, and the data of the current block. As a first step, it verifies if the last code hash is indeed the correct one by creating the last code hash and comparing it with the existing previous hash code. Following node authentication, the only hash code that must be relied upon and is needed for the system to work is the one that corresponds to the current block and is generated with the assistance of data and the previous hash code.

Storage

The availability of storage will be verified after block validation. When space is available, the block will be inserted into the blockchain. It also concentrates on reducing scalability problems in a blockchain. Nodes that do not provide on-stored data get penalised. It also maintains efficiency to give robust performance on a blockchain. Plays a vital role in bandwidth-effective usage. If proper storage is not available, then it will follow the rejection process.

Validation of the Block

In this step, the block is validated. When added to the block's transactions, miners compete to find a nonce that meets the required difficulty level to generate a hash. Validation of the block is performed by generating a hash code. It finds latency, which indicates the time taken to validate the block. Difficulty indicates the security level of the block, where zeroes are added as prefixes. It creates a hash code for the current block using the previous hash. By considering these three parameters, the block is validated. This is the concept used by PoUW. To calculate the execution cost, considering not only transaction cost, gas cost, overhead cost, base cost, latency, and computational energy, the proposed work extends the formula to incorporate additional components.

Figure 4 represents how a block is added to a blockchain. Initially, every block-related transaction must be mined, and the verifier must be verified. If the successful execution of every phase is confirmed, the block will be added to the blockchain. Otherwise, it will not be added or included. In this proposed work, verification is done in two different ways after mining. The block is verified based on authorisation and storage availability, allowing scalability issues to be overcome. In Table 2, Algorithm POUWAS available data storage must be examined for the new block generation, and then the miner's authentication must be verified to proceed with block construction activities. The leader is rejected at once if unauthorised. Lastly, the PoUWAS algorithm proves whether a transaction is valid to add a block, according to the availability of storage, authorisation, and correct mining of the transactions. All these three must be integrated to increase the scalability of the blockchain but also keep it functional against unknown attacks.

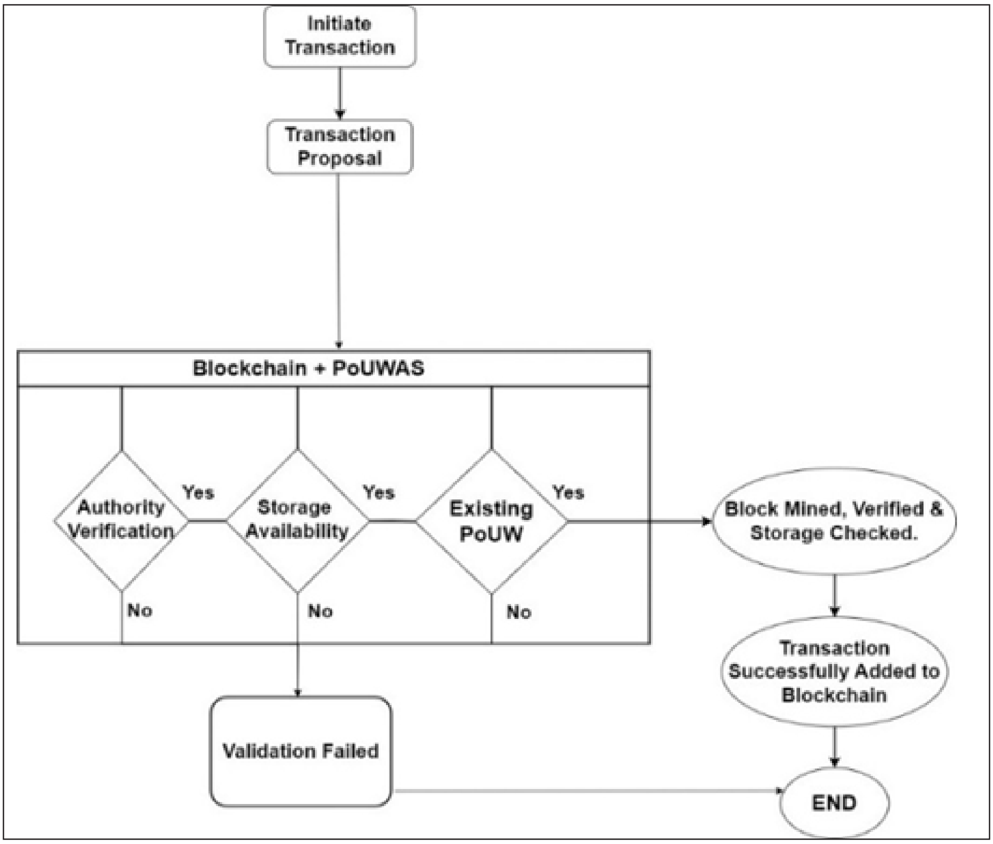


Figure 4. Flowchart for an improved/hybrid consensus algorithm

According to Table 2, the proposed algorithm combines authorisation, storage validation, and mining. It verifies block authorisation by observing whether the hash and current block index accurately follow the previous block. Then compares the requested storage and challenge replies to verify storage capabilities. Lastly, it measures delay as computational effort and executes a Proof of Useful Work (PoUW) by satisfying the specified difficulty level in a limited number of iterations. Finally, a block is added to the blockchain after all verifications are successful otherwise, it is rejected for lack of adequate authority, storage, or work.

Table 2
Proof of useful work-authorisation-storage availability-PoUWAS

```
Inputs:
    prev_idx, curr_blockID, content, prev_content, prev_digest,
    content_digest, node_key, challengeSet [], req_space,
    tx_records, last_digest, validator_key, useful_task
Outputs:
    Confirms block authorisation
    Confirms storage capability
    Adds validated block or rejects insufficient proof
Procedure Verify_BlockAuth (prev_idx, curr_blockID, content, prev_content, prev_digest)
    hash_code ← SHA (content)
    IF prev_idx + 1 = curr_blockID THEN
        IF SHA (prev_content) = prev_digest THEN
            RETURN true
        ELSE
            RETURN false
        END PROCEDURE
Procedure Check_Storage(challengeSet [], req_space, free_space)
    IF free_space ≥ req_space THEN
        IF file_chunk = challengeSet[data_index] THEN
            IF resp_code = SHA (file_chunk) THEN
                IF check_hash = SHA (requested_chunk) THEN
                    RETURN true
                ELSE
                    RETURN false
END PROCEDURE
Procedure Useful_work (level, max_loops = 100000)
    begin_time ← current_time
    probe_val ← random (1500 to 12000)
    loop_counter ← 0
    WHILE probe_val is NOT prime AND loop_counter < max_loops DO
        probe_val ← probe_val + 1
        loop_counter ← loop_counter + 1
    END WHILE
    IF loop_counter = max_loops THEN
        THROW Exception "Computation Limit Reached"
    END IF
    proof_value ← probe_val
    proof_str ← string(proof_value)
    WHILE proof_str does NOT start with '0' repeated level times AND loop_counter < max_loops DO
        proof_value ← proof_value + 1
        proof_str ← string(proof_value)
        loop_counter ← loop_counter + 1
    END WHILE
    delay_time ← current_time - begin_time
    APPEND delay_time TO self.latency_list
    RETURN proof_value, delay_time
END PROCEDURE

Procedure Main_BlockValidation()
    auth_result ← Verify_BlockAuth(prev_idx, curr_blockID, content, prev_content, prev_digest)
    IF auth_result = false THEN
        RETURN "Authorisation Failure"
    END IF

    storage_result ← Check_Storage(challengeSet[], req_space, free_space)
    IF storage_result = false THEN
        RETURN "Storage Verification Failure"
    END IF
 proof, delay ← Useful_Work(difficulty_level, max_loops)
    IF proof is valid THEN
        ADD content To blockchain
        RETURN "Block Verified and Added"
    ELSE
        RETURN "Block Rejected: Insufficient Useful Work"
    END IF
END PROCEDURE
```

Mathematical Evaluation

It is a crucial part of describing whether a blockchain system is scalable, particularly in the case of improved consensus mechanisms. Mathematical evaluation plays an important role in measuring the scalability of a blockchain system. Performance metrics like block validation time, transaction throughput, computational energy, and network latency can highlight how the system performance varies when the number of nodes or transactions increases. This helps to identify potential risks, optimise resource allocation, and ensure that the proposed algorithm can efficiently handle high-end networks without limiting security or performance. So, it serves as a solid base for proof of the applicability and strength of the enhanced consensus mechanism on a practical blockchain. Table 3 summarises the notations used in mathematical evaluation. Based on the equation 1, 2, 3, 4, values related to gas cost, transaction cost and execution costs are measured. Followed by 5,6, 7, 8, 9, 10, and 11 represent the latency and transaction throughput of the proposed consensus algorithm-based blockchain.

$$TC = GC + C_{base}, \qquad [1]$$

$$GC = G_{used} \times G_{price}, \qquad [2]$$

$$EC = E \times C_{energy}, \qquad [3]$$

$$C_{total} = Transactional\ Cost + Gas\ Cost + Execution\ Cost. \qquad [4]$$

$$L1 = ET - ST, \qquad [5]$$

$$Tt = No.of\ transactions\ /L, \qquad [6]$$

$$Overall\ L1 = \sum_{i=0}^{N}(Li), \qquad [7]$$

$$=0.113+0.063+0.015+0.238+0.097+0.045+0.031+0.319+0.02+0.026,$$
$$=0.967$$

$$Overall\ Tt1 = \sum_{i=0}^{N}(Ti), \quad [8]$$

=8.84+31.59+190.37+16.76+51.5+131.56+219.31+25.02+432.7+376.04
=1483.69

$$L2 = ET - S\,T\ , \quad [9]$$

$$Overall\ L2 = \textstyle\sum_{i=0}^{N}(Li), \quad [10]$$

= 0.031 + 0.268 + 0.371 + 0.043 + 0.256 + 0.096 + 0.063 + 0.239 + 0.053 + 0.029,
=1.449

$$Overall\ Tt2 = \textstyle\sum_{i=0}^{N}(Ti) \quad [11]$$

=31.45+7.43+8.08+92.20+19.50+62.44+109.81+33.37+169.27+335.71,
= 869.26

Table 3
Notations related to mathematical evaluation

Notation	Meaning	Description
TC	Transaction Cost	The cost associated with the transaction itself, such as network fees, validator fees, etc.
GC	Gas Cost	The cost of performing computations in a blockchain-based system is calculated as Gas Units × Gas Price.
$C_{latency}$	The cost per unit of latency	(e.g., economic cost or penalty per second of delay).
E	Energy consumed	Measured in Joules or Kilowatt-hours (kWh), depending on the system.
C_{energy}	Cost per unit of energy	cost per kilowatt-hour
C_{total}	Total cost	Measured in Joules or Kilowatt-hours (kWh), depending on the system.
C_{base}	Minimum gas price determined by the network	Measured in currency upon Kilowatt-hours ($/kWh), depending on the system.

Note. Energy consumption interms of Gas cost, Transaction cost and Execution cost with the help of mathematical notations

RESULTS AND DISCUSSION

Blockchain-related activities are conducted with the assistance of Remix IDE and Python Spyder. In Remix IDE, different consensus algorithms like PoUW and PoUWAS were executed, and the computational energy was observed from Figure 5. Then, latency and transactional throughput were observed on PoUW and PoUWAS in Python Spyder from Figure 7 and Figure 8. These platforms are essential to deal with blockchain-related end-to-end transactions. Real-time applications, such as supply chain and healthcare-related frameworks, are also executable. Table 4 represents the recent outcomes related to improved consensus.

In this section, this research mainly explores the scalability metrics regarding computational energy from Table 5, Figure 5, Figure 6, notations, latency, and transaction throughput from Table 3, Table 5, Table 6, Table 7, Figure 7, and Figure 8. The primary motive for this section is to present the PoUW, PoA, PoS, and PoUWAS computational energy details, as well as the comparison between PoUW and PoUWAS latency and transaction throughput details. In Figure 6, this research explores individual algorithm-related gas, transactions, and execution costs and the observed deductions in computational energy.

Table 4
Findings related to related work

Author & Year	Method / Framework	Focus Area	Limitations	Outcome/ Performance
(Pawar, 2025)	Resource-efficient PoUW	Optimising computational resources	Lack of real-time adaptability	Low energy but static scheduling
(Rahman, 2024)	PoUW for scientific computation	Reusing mining power for real computations	Latency in task validation	High utility, moderate delay
(Babaei et al., 2025)	Integrates decision-making techniques	Blockchain on Supply Chain with ML techniques	Scalability	Traceability and security
(Kumari & Kumari, 2025)	integrated PoA+PBFT	Improved Consensus	Limited Scalability metrics	Improved efficiency
Proposed Model (2025)	PoUWAS	Real-time scalable consensus	Under testing for large-scale deployment	Improved scalability, low energy footprint, and real-time DApp suitability

Table 5
Parameters related to scalability

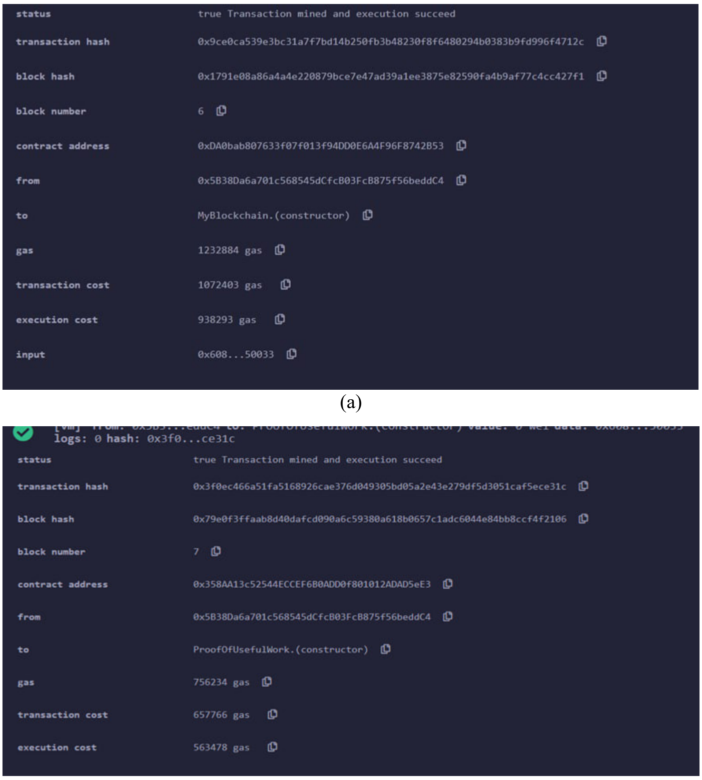
Algorithm	Gas Cost	Transaction Cost	Execution Cost	Total Cost
PoUW (Daely et al., 2024)	1232884	1072403	938293	3243580
PoUWAS	756234	657766	563478	1977478

Table 6
Notations related to the algorithm

Notation	Meaning
L	Latency
Tt	Transactional Throughput
N	Number of Block 0 to Block 9
PoUW	Proof of Useful Work
PoUWAS	Proof of Useful Work Authority Storage
ET	End Time
ST	Start Time

Table 7
Parameters related to the scalability of PoUWAS

Block	Latency	Transaction Throughput
Block 0	0.113	8.84
Block 1	0.063	31.59
Block 2	0.015	190.37
Block 3	0.238	16.76
Block 4	0.097	51.5
Block 5	0.045	131.56
Block 6	0.031	219.31
Block 7	0.319	25.02
Block 8	0.02	432.7
Block 9	0.026	376.04
	Total=0.967	Total=1483.69



```
status                  true Transaction mined and execution succeed

transaction hash        0x9ce0ca539e3bc31a7f7bd14b250fb3b48230f8f6480294b0383b9fd996f4712c

block hash              0x1791e08a86a4a4e220879bce7e47ad39a1ee3875e82590fa4b9af77c4cc427f1

block number            6

contract address        0xDA0bab807633f07f013f94DD0E6A4F96F8742B53

from                    0x5B38Da6a701c568545dCfcB03FcB875f56beddC4

to                      MyBlockchain.(constructor)

gas                     1232884 gas

transaction cost        1072403 gas

execution cost          938293 gas

input                   0x608...50033
```

(a)

```
logs: 0 hash: 0x3f0...ce31c
status                  true Transaction mined and execution succeed

transaction hash        0x3f0ec466a51fa5168926cae376d049305bd05a2e43e279df5d3051caf5ece31c

block hash              0x79e0f3ffaab8d40dafcd090a6c59380a618b0657c1adc6044e84bb8ccf4f2106

block number            7

contract address        0x358AA13c52544ECCEF6B0ADD0f801012ADAD5eE3

from                    0x5B38Da6a701c568545dCfcB03FcB875f56beddC4

to                      ProofOfUsefulWork.(constructor)

gas                     756234 gas

transaction cost        657766 gas

execution cost          563478 gas
```

(b)

Figure 5. Computational energy in terms of gas cost, transactional cost, and execution cost: (a) existing PoUW; (b) Proposed PoUWAS

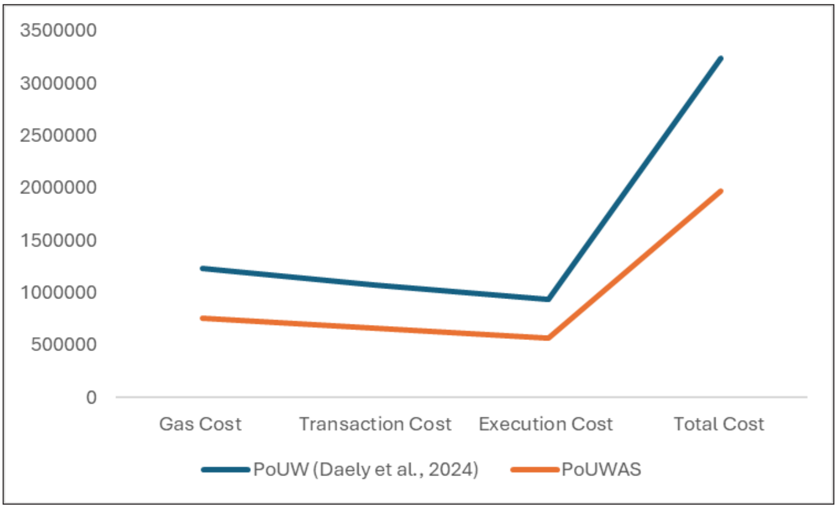


Figure 6. PoUWAS, PoUW computational energy

```
n [23]: runfile('C:/Users/srira/POUWAS Latency and Tt.py', wdir='C:/Users/srira
lock #0 mined in 0.11306428909301758 seconds
ash: 00003ff8139fd370781c9b8225a7381ac724565c24eb795b2ecfdd3afd46ccb5
alidation: Validated
torage: Stored
hroughput: 8.844525605934724 transactions per second

lock #1 mined in 0.06329512596130371 seconds
ash: 00002376808acf4c5da0f235cb3b41530d9a7afbe8bd074ade2ca4b3b857ab11
alidation: Validated
torage: Stored
hroughput: 31.598009635413725 transactions per second

lock #2 mined in 0.015758037567138672 seconds
ash: 00008e027d140ca9e2871c3e67452347b0aafa511f2f1c65eab512fe6ef7d70f
alidation: Validated
torage: Stored
hroughput: 190.37903591854027 transactions per second

lock #3 mined in 0.2386460304260254 seconds
ash: 0000bc9a329556581495699c5422ecc9e76a782b62ea9303a8e365048943debd
```

(a)

```
Hash: 0000bc9a329556581495699c5422ecc9e76a782b62ea9303a8e365048943debd
Validation: Validated
Storage: Stored
Throughput: 16.761225790595773 transactions per second

Block #4 mined in 0.09708523750305176 seconds
Hash: 000095f8a52b4584a15761f809ce562254c1e6b6b344748e93ea840350b8653a
Validation: Validated
Storage: Stored
Throughput: 51.501135791554624 transactions per second

Block #5 mined in 0.045603275299072266 seconds
Hash: 0000339213ab1159c4f772cd0a11d0ddfd34f6ab3cf1ad4a95e28a14ddf71305
Validation: Validated
Storage: Stored
Throughput: 131.56949716114056 transactions per second

Block #6 mined in 0.03191709518432617 seconds
Hash: 0000a34e272a3d4a84258cb13e29ea72c2fca5a5c7a0b7af06c70e5a2639cb5c
Validation: Validated
Storage: Stored
Throughput: 219.31820422798236 transactions per second

Block #7 mined in 0.3197288513183594 seconds
Hash: 00007c02fc13f23b48884333a1c638f55114ce1a036c950a3708504e42f217fa
Validation: Validated
Storage: Stored
```

(b)

```
Block #8 mined in 0.02079916000366211 seconds
Hash: 000075334bb9d9350c0b2be899e0620054203a617d874a5f576fbb4e9
Validation: Validated
Storage: Stored
Throughput: 432.7097824342603 transactions per second

Block #9 mined in 0.026592254638671875 seconds
Hash: 0000c13cfdaa2b48f77e1d0aaa4c96f4254b3ab3a2fa2e1f202407ff2
Validation: Validated
Storage: Stored
Throughput: 376.0493472959403 transactions per second
```

(c)

```
In [25]: runfile('C:/Users/srira/POUW Latency and Tt.py', wdir='C:/Users/srira')
Block #0 mined in 0.031786441802978516 seconds
Hash: 000095891925f52433f95a778718ad2c3c73133b0d09a4ca321af201cc671d7d
Throughput: 31.45995409609817 transactions per second

Block #1 mined in 0.2688877582550049 seconds
Hash: 000025d40ef987276f3804b087d37daa550740e9a284201a4228d522666d94e9
Throughput: 7.438047804702442 transactions per second

Block #2 mined in 0.3712608814239502 seconds
Hash: 000078a40437e1577ff572a45385c42aed924c80e99d779ffd7e98b4f7660fff
Throughput: 8.08057123738345 transactions per second

Block #3 mined in 0.0433804988861084 seconds
Hash: 0000157a0a98b2db838f9774e31822cba17aeb9dda77ac92cb8dcf6f19be126e
Throughput: 92.20733054503685 transactions per second

Block #4 mined in 0.2563595771789551 seconds
Hash: 000015dd6751f5af22b142f68f7db1e1c3082d8ee5550defbc15c6ee295677ed
Throughput: 19.503854917461055 transactions per second

Block #5 mined in 0.09609127044677734 seconds
Hash: 0000ae7e6f2ccb7650b96859f9395014350d39b4a065ab92a60c3ce78f92f2fa
Throughput: 62.44063557597833 transactions per second

Block #6 mined in 0.06374168395996094 seconds
Hash: 0000791f53e3c81a3cdfa6c1537686aa4a553093ffc2d16d7caffe7528b65b09
```

(d)

```
Throughput: 109.8182471049403 transactions per second

Block #7 mined in 0.2396841049194336 seconds
Hash: 0000b4a5d378ac3783227856bc90fd728e1f070256440519d33dbcd3457a2884
Throughput: 33.37726547485945 transactions per second

Block #8 mined in 0.05316877365112305 seconds
Hash: 000029f135436455b00066e5ed16624737405cb1195e862eb8d579ad436be518
Throughput: 169.27228863797387 transactions per second

Block #9 mined in 0.029787302017211914 seconds
Hash: 0000e60d8b43802607d9a0107a588e9887bf3ca10711d965cb73af6badefd191
Throughput: 335.7135196138854 transactions per second
```

(e)

Figure 7. Latency, and transactional throughput: (a), (b), (c) Proposed PoUWAS; (d), (e) existing PoUW

Computational Energy

In blockchain communities, "gas" quantifies the processing needed to perform tasks like the execution of smart contracts. This metric determines the computational resources allocated to transaction verification. On platforms like Ethereum, participants use the network's inherent digital tokens to facilitate operations. The entire amount of cryptocurrency that a user pays for a transaction to be executed on the blockchain is known as the transaction cost. It covers the price of gas as well as any other charges levied by the network or service provider. The costs incurred in carrying out a particular operation or job inside a blockchain-based smart contract are called execution costs. For smart contracts, every individual function or instruction carries its own computational charge. Since these contracts are made up of multiple steps, the total gas required is shaped by the sum of all these execution costs, as noted in 1, 2, 3 and 4. As a result, the combined computational demands will ultimately determine the final amount a user pays to complete the transaction.

Table 5 compares the scalability parameters for individuals and proposed consensus algorithms. The proposed work aims to enhance the improved consensus algorithm compared to the existing PoUW. The improved consensus algorithm combines three consensus algorithms, which are PoUW (Dong et al., 2019), PoA (Hanggoro et al., 2024), and PoS (Heo et al., 2024). In the observation, deductions in terms of gas cost, transaction cost, and execution cost for PoUW and PoUWAS were noted at 38.66%, 38.66%, and 39.94%, respectively. Therefore, deductions in these parameters lead to improvements in the performance of a blockchain. Figure 8 also represents the comparison between all the algorithms mentioned.

From Table 5, Table 8, Figure 7, and Figure 8, Proof-of-Useful-Work (PoUW) is examined using the proposed methodology-based consensus algorithm, specifically PoUWAS, which analyses the gas cost, transaction cost, and execution cost. Ultimately, it was concluded from this observation that combining PoUW with PoA and PoS, i.e., PoUWAS, results in minimal computational energy compared to individual PoUW, and it also focuses on the authorisation and storage availability of a node.

Table 8
Parameters related to the scalability of PoUW

Block	Latency	Transaction Throughput
Block 0	0.031	31.45
Block 1	0.268	7.43
Block 2	0.371	8.08
Block 3	0.043	92.2
Block 4	0.256	19.5
Block 5	0.096	62.44
Block 6	0.063	109.81
Block 7	0.239	33.37
Block 8	0.053	169.27
Block 9	0.029	335.71
	Total=1.449	Total=869.26

Latency

The duration from when a transaction is initiated to fully confirmed and recorded on the blockchain is known as latency in blockchain systems. For instance, blockchains such as Bitcoin that use the Proof of Work (PoW) system have greater latency because it takes approximately ten minutes to include a new block in the chain. Blockchains that use Proof of Stake (PoS) or other newer consensus algorithms can achieve faster confirmation of transactions by lessening the time it takes for a transaction to be included in a block and confirmed.

Transaction Throughput

Transaction throughput in terms of transactions per second on blockchain is the quantity of transactions that the network can process in each time. Scalability and the ability to keep up with the demands of blockchain applications require higher throughput. The throughput of a blockchain is determined by several factors, such as the size of the individual blocks, the frequency of new block generation, and the consensus mechanism employed by the network.

Latency and Transaction Throughput for PoUWAS

To calculate the blockchain latency and transactional throughput, one would have to sum up the latency and throughput of each block. For example, latency (L) and transaction throughput (Tt). Table 6 summarises the different notations and their associated meanings that will be used in the following mathematical equations 5, 6, 7, 8, 9, 10 and 11.

According to the proposed methodology, latency and transactional throughput must be calculated to measure the latency and transactional throughput of the proposed consensus algorithm. Blockwise calculation helps evaluate the overall performance of an algorithm using various metrics, such as latency and transactional throughput, as shown in Table 7. Proposed work research mainly focuses on the latency and transactional throughput for the 10 blocks. Measuring each block's latency and transactional throughput starts from block 0 to block nine and yields values of 0.967 and 1483.69. Here, the latency for 10 blocks is L1, and the overall transaction throughput for 10 blocks is Tt1.

According to the proposed methodology, as observed in Table 7, Figure 7, and the existing consensus algorithm presented in Table 8, Figure 8, latency and transactional throughput were measured. After comparing the proposed methodology with existing methods, a change in improvement was observed in various metrics, including latency, which was reduced by 0.482, and transactional throughput, which improved by 614.43 TPS. The proposed work has reduced latency compared to existing work and increased transactional throughput. The existing consensus algorithm from Table 5 was used to observe the overall latency and transactional throughput for 10 blocks, and its performance was compared with that of the proposed algorithm methodology from Table 8. Overall performance was also compared regarding latency and transaction throughput, and the proposed method achieved better results.

Latency and Transaction Throughput for PoUW

The experimental findings show that, in comparison to conventional Proof of Work (PoW) and Proof of Stake (PoS) processes, there is a notable decrease in block validation time and an increase in throughput. Previous research has shown that hybrid consensus algorithms that combine voting and reputation measures can improve network scalability. (Kumari & Kumari, 2025; Li et al., 2020) demonstrated similar results. These investigations are supported by the current results, which further demonstrate that adaptive or hybrid consensus techniques can maximise resource efficiency and sustain increased transaction throughput without sacrificing security. Financial performance and computational energy were accomplished within the supply chain simultaneously (Goli & Babaee Tirkolaee, 2023). Flexibility, efficiency, and coordination among blockchain applications in supply chains were examined in this study (Haffar & Ozceylan, 2024). A viable supply chain model by integrating a vendor-managed inventory approach and blockchain technology to reduce costs is proposed (Lotfi et al., 2022).

CONCLUSION

This research is unique in that it proposes intelligent decision-making as part of the consensus layer of the blockchain, which has not been explored extensively in the literature. In contrast to previously presented solutions, where merely cryptographic or structural aspects are adjusted, the proposed approach presents a hybrid consensus protocol capable of adapting the participation and validation process at runtime. This solution, besides being more scalable and energy-efficient, is also more trust-adaptable, which is a step forward towards a more sustainable design of a blockchain system that can be used in several fields.

This paper proposes work that mainly concentrates on measuring the improved consensus algorithms with the help of metrics related to scalability in blockchain, as well as comparing the differences in scalability between existing work and the proposed work. Mainly concentrates on transaction throughput, latency, computational energy, and consensus models. After comparing the existing algorithm, proof of useful work with the improved consensus algorithm Proof of useful work, authorisation and storage availability scalability improved regarding latency and transaction throughput in the proposed work. Still, there is considerable scope to improve the overall performance of the blockchain network by applying additional techniques. Future work will focus on this area and integrate with other emerging technologies to utilise blockchain more effectively in real-time applications. Here, the research scope is limited to measuring the proposed consensus algorithm. However, future research will continue with additional metrics related to scalability. After a careful review of the literature on consensus algorithms in blockchain technology, scalability is the primary issue hindering the overall performance of a blockchain. Consensus algorithm improvements are also essential to validate transactions most effectively. Latency and transaction throughput improvements were also crucial for enhancing the overall growth of a blockchain.

ACKNOWLEDGEMENT

The authors thanked Koneru Lakshmaiah Education Foundation, especially the Computer Science and Engineering Department, which provided all the facilities that supported this research. This article is part of the first author's thesis. No funding or grants were provided for this research.

LIST OF ABBREVIATIONS

PoW	:	Proof of Work
PoA	:	Proof of Authority
PoS	:	Proof of Storage
PoUWAS	:	Proof of Useful Work-Authorisation-Storage Availability
DPoS	:	Delegated Proof of Stake
PBFT	:	Practical Byzantine Fault Tolerance

REFERENCES

Andrew, J., Isravel, D. P., Sagayam, K. M., Bhushan, B., Sei, Y., & Eunice, J. (2023). Blockchain for healthcare systems: Architecture, security challenges, trends and future directions. *Journal of Network and Computer Applications, 215*, Article 103633. https://doi.org/10.1016/j.jnca.2023.103633

Babaei, A., Tirkolaee, E. B., Ali, S. S., Roy, S. K., & Weber, G. W. (2025). An efficiency-uncertainty-based consensus optimisation framework with dual-perspective clustering analysis for blockchain adoption in supply chains. *IEEE Transactions on Engineering Management*. https://doi.org/10.1109/TEM.2025.3626659

Daely, P. T., Lee, J. M., & Kim, D. S. (2024). Vehicle routing based mining for proof-of-useful-work blockchain. *Journal of the Korean Institute of Communications and Information Sciences, 49*(1), 124-133. https://doi.org/10.7840/kics.2024.49.1.124

Deng, Z., Li, T., Tang, C., He, D., & Zheng, Z. (2024). PSSC: Practical and secure sidechain construction for heterogeneous blockchains orienting IoT. *IEEE Internet of Things Journal, 11*(3), 4600-4613. https://doi.org/10.1109/JIOT.2023.3302291

Dong, Z., Lee, Y. C., & Zomaya, A. Y. (2019). Proofware: Proof of useful work blockchain consensus protocol for decentralised applications. *arXiv*. https://doi.org/10.48550/arXiv.1903.09276

Fajri, A. I., & Mahananto, F. (2022). Hybrid lightning protocol: An approach for blockchain scalability issue. *Procedia Computer Science, 197*, 437-444. https://doi.org/10.1016/j.procs.2021.12.159

Ghamdi, M. A. A. (2022). An optimised and secure energy-efficient blockchain-based framework in IoT. *IEEE Access, 10*, 133682-133697. https://doi.org/10.1109/ACCESS.2022.3230985

Goli, A., & Babaee Tirkolaee, E. (2023). Designing a portfolio-based closed-loop supply chain network for dairy products with a financial approach: Accelerated Benders decomposition algorithm. *Computers & Operations Research*, Article 106244. https://doi.org/10.1016/j.cor.2023.106244

Haffar, S., & Ozceylan, E. (2024). Blockchain in resilient and leagile supply chains: Research themes and opportunities. *Journal of Industrial and Management Optimisation*. https://doi.org/10.3934/jimo.2024109

Hanggoro, D., Windiatmaja, J. H., Muis, A., Sari, R. F., & Pournaras, E. (2024). Energy-aware proof-of-authority: Blockchain consensus for clustered wireless sensor network. *Blockchain: Research and Applications, 5*(3), Article 100211. https://doi.org/10.1016/j.bcra.2024.100211

Heo, J. W., Ramachandran, G., & Jurdak, R. (2024). nPPoS: Non-interactive practical proof-of-storage for blockchain. *Blockchain: Research and Applications*, Article 100221. https://doi.org/10.1016/j.bcra.2024.100221

Kaur, M., Khan, M. Z., Gupta, S., Noorwali, A., Chakraborty, C., & Pani, S. K. (2021). MBCP: Performance analysis of large-scale mainstream blockchain consensus protocols. *IEEE Access, 9*, 80931-80944. https://doi.org/10.1109/ACCESS.2021.3085187

Kumari, K. L., & Kumari, P. L. S. (2025). Design and analysis of the improved consensus algorithm of the blockchain technology. *International Research Journal of Multidisciplinary Scope, 6*(2), 833-844.https://doi.org/10.47857/irjms.2025.v06i02.03506

Liu, Y., Qian, K., Wang, K., & He, L. (2021). Effective scaling of blockchain beyond consensus innovations and Moore's law: Challenges and opportunities. *IEEE Systems Journal*. https://doi.org/10.1109/JSYST.2021.3087798

Lotfi, R., Rajabzadeh, M., Zamani, A., & Rajabi, M. S. (2022). Viable supply chain with vendor-managed inventory approach by considering blockchain, risk and robustness. *Annals of Operations Research*. https://doi.org/10.1007/s10479-022-05119-y

Malakhov, I., Marin, A., & Rossi, S. (2023). Analysis of the confirmation time in proof-of-work blockchains. *Future Generation Computer Systems, 147*, 275-291. https://doi.org/10.1016/j.future.2023.04.016

Manolache, M. A., Manolache, S., & Tapus, N. (2022). Decision making using the blockchain proof of authority consensus. *Procedia Computer Science, 199*, 580-588. https://doi.org/10.1016/j.procs.2022.01.071

Nasir, N. M., Hassan, S., & Zaini, K. M. (2024). Securing permissioned blockchain-based systems: An analysis on the significance of consensus mechanisms. *IEEE Access*. https://doi.org/10.1109/ACCESS.2024.3465869

Otsuki, K., Nakamura, R., & Shudo, K. (2021). Impact of saving attacks on blockchain consensus. *IEEE Access, 9*, 133011-133022. https://doi.org/10.1109/ACCESS.2021.3115131

Pawar, T. (2025). Cyber-shield machine learning: An intrusion detection system with blockchain-powered secure log storage across network nodes. *Sigma Journal of Engineering and Natural Sciences*, 714-725. https://doi.org/10.14744/sigma.2025.00066

Rahman, R. U. (2024). A novel machine learning-based artificial intelligence approach for log analysis using blockchain technology. *Sigma Journal of Engineering and Natural Sciences*, 1391-1409. https://doi.org/10.14744/sigma.2024.00004

Sun, Y., Yan, B., Yao, Y., & Yu, J. (2021). DT-DPoS: A delegated proof of stake consensus algorithm with dynamic trust. *Procedia Computer Science, 187*, 371-376. https://doi.org/10.1016/j.procs.2021.04.113

Tawfik, A. M., Al-Ahwal, A., Eldien, A. S. T., & Zayed, H. H. (2025). Blockchain-based access control and privacy preservation in healthcare: A comprehensive survey. *Cluster Computing, 28*(8). https://doi.org/10.1007/s10586-025-05308-x

Ucbas, Y., Eleyan, A., Hammoudeh, M., & Alohaly, M. (2023). Performance and scalability analysis of Ethereum and Hyperledger Fabric. *IEEE Access, 11*, 67156-67167. https://doi.org/10.1109/ACCESS.2023.3291618

Wu, A., & Hua, C. (2025). ALB-chain: Achieving scalability and load balance blockchain sharding. *Cluster Computing, 28*(6). https://doi.org/10.1007/s10586-024-04949-8

Yang, J., Jia, Z., Su, R., Wu, X., & Qin, J. (2022). Improved fault-tolerant consensus based on the PBFT algorithm. *IEEE Access, 10*, 30274-30283. https://doi.org/10.1109/ACCESS.2022.3153701

Zhang, J., Tian, R., Cao, Y., Yuan, X., Yu, Z., Yan, X., & Zhang, X. (2021). A hybrid model for central bank digital currency based on blockchain. *IEEE Access, 9*, 53589-53601. https://doi.org/10.1109/ACCESS.2021.3071033

Zhang, L., Li, F., Wang, P., Su, R., & Chi, Z. (2022). A blockchain-assisted massive IoT data collection intelligent framework. *IEEE Internet of Things Journal, 9*(16), 14708-14722. https://doi.org/10.1109/JIOT.2021.3049674