The Scientific Temper (2025) Vol. 16 (2): 3698-3709
E-ISSN: 2231-6396, ISSN: 0976-8653
Doi: 10.58414/SCIENTIFICTEMPER.2025.16.2.02
https://scientifictemper.com/

RESEARCH ARTICLE

Enhancing IOT blockchain scalability through the eepos consensus algorithm

M. Ragul[1], A. Aloysius[2], V. Arul Kumar[3]

Abstract

The integration of blockchain technology with the Internet of Things (IoT) introduces significant scalability, energy efficiency, and security challenges, particularly when using traditional consensus mechanisms like Proof of Work (PoW). IoT networks generate vast amounts of data while operating under resource constraints, necessitating the development of consensus algorithms that balance energy efficiency, transaction throughput, and security. Addressing these challenges is critical for the sustainable adoption of blockchain in IoT ecosystems. This research aims to enhance blockchain scalability and performance in IoT environments through the development of the Enhanced Efficient Proof of Stake (EePoS) consensus algorithm. The objective is to provide a framework that optimizes validator selection, minimizes energy consumption, and ensures robust security against common blockchain threats. The proposed method employs a multi-layered architecture, selective validation, and a behavior-aware penalty-reward system to ensure efficient consensus. Key security metrics, including Probability of Successful Attack (PSA) and Forking Rate (FR), were evaluated to demonstrate the algorithm's resilience. EePoS reduces PSA by dynamically adjusting validator selection based on stake, behavior, and transaction load while decreasing FR through cluster-based voting and hierarchical aggregation. Experimental results demonstrated 20% lower PSA, 30% reduced FR, and 8% faster consensus time compared to ePoS. Throughput improved to 296 TPS while reducing CPU and memory utilization, ensuring robust performance for resource-constrained IoT networks. The novelty of this work lies in the tailored enhancements to the PoS framework, specifically designed for IoT constraints, making EePoS a scalable, energy-efficient, and secure solution for IoT blockchain integration.

Keywords: Blockchain, Consensus Algorithm, EePoS, Energy Efficiency, IoT, Proof of Stake.

Introduction

The integration of blockchain technology with the Internet of Things (IoT) presents significant scalability challenges, particularly when employing traditional consensus algorithms. IoT networks generate vast amounts of data from numerous devices, which can overwhelm conventional blockchain systems. This challenge necessitates a focus on scalability to handle data efficiently without compromising performance (Haque et al., 2024; Gedam and Karmore, 2024). Additionally, IoT devices often have limited computational power and energy resources, making resource-intensive consensus mechanisms, such as Proof of Work (PoW), unsuitable for large-scale IoT deployments (Abbasi et al., 2024; Kaur and Gupta, 2023).

The Proof of Stake (PoS) consensus algorithm and its variants offer promising solutions to these challenges. PoS is inherently more energy-efficient than PoW because it does not require intensive computational power to validate transactions. This characteristic makes PoS particularly suitable for IoT environments, where devices often operate under stringent resource constraints (Maftei et al., 2023). Variants like Delegated Proof of Stake (DPoS) and Proof of Resource (PoR) have been specifically designed to address the unique requirements of IoT systems, such as low latency, high transaction throughput, and resource efficiency (Haque et al., 2024; Abbasi et al., 2024).

The potential of PoS and its derivatives are being used to enhance blockchain scalability in IoT ecosystems. By

[1]Research Scholar, Department of Computer Science, St.Joseph's College(A,) Affiliated to Bharathidasan University, Tiruchirappalli, Tamil Nadu, India.

[2]Assistant Professor, Department of Computer Science, St. Joseph's College(A), Affiliated to Bharathidasan University Tiruchirappalli, Tamil Nadu, India.

[3]Assistant Professor, Department of Data Science, St. Joseph's College(A), Affiliated to Bharathidasan University Tiruchirappalli, Tamil Nadu, India

***Corresponding Author:** M. Ragul, Research Scholar, Department of Computer Science, St.Joseph's College(A,) Affiliated to Bharathidasan University, Tiruchirappalli, Tamil Nadu, India., E-Mail: ragularun12@gmail.com

How to cite this article: Ragul, M., Aloysius, A., Kumar, V.A. (2025). Enhancing iot blockchain scalability through the eepos consensus algorithm. The Scientific Temper, **16**(2):3698-3709.

Doi: 10.58414/SCIENTIFICTEMPER.2025.16.2.02

Source of support: Nil

Conflict of interest: None.

© The Scientific Temper. 2025

Received: 09/01/2025 **Accepted:** 21/02/2025 **Published** : 20/03/2025

examining various innovative consensus mechanisms tailored for resource-constrained devices, many recent studies aimed to highlight how these technologies can facilitate broader adoption of blockchain in IoT networks (Alghamdi et al., 2023; Huang et al., 2022). Further research and development are essential to address the importance of security, scalability, and efficiency in implementing PoS-based systems in IoT. The transition to PoS in IoT applications also encourages the development of innovative mechanisms like Proof of IoT (PIoT) and the Proof of Block and Transaction (PoBTx) protocols, which optimize consensus for IoT environments (Christ et al., 2023; Janani & Ramamoorthy, 2024). This study provides insights into these emerging solutions and their potential impact on the scalability of IoT blockchain systems. Integrating advanced mechanisms like DPoS and PoR into IoT networks can improve performance metrics such as latency, throughput, and resource utilization while maintaining robust security and decentralization (Misic et al., 2023).

IoT systems face unique scalability challenges, primarily due to the sheer volume of data and the large number of devices involved. Blockchain must accommodate these demands without sacrificing performance. Traditional blockchain systems often fall short in this aspect, as they are not optimized for handling the high throughput and low-latency requirements of IoT applications (Haque et al., 2024). Moreover, IoT devices are typically constrained by limited computational and energy resources, making them incompatible with resource-intensive consensus mechanisms like PoW (Abbasi et al., 2024; Kaur & Gupta, 2023).

Another significant challenge is the need for real-time data processing in IoT networks. Many applications, such as smart cities and industrial IoT, require low-latency transactions to maintain operational efficiency. Existing blockchain solutions struggle to meet these requirements, highlighting the need for more scalable and efficient consensus mechanisms tailored to IoT's constraints (Misic et al., 2023).

DPoS involves electing a small number of delegates to validate transactions, significantly reducing the computational burden on IoT devices. This mechanism enhances scalability by improving throughput and reducing latency, making it ideal for large-scale IoT networks (Haque et al., 2024).

PoR leverages the inherent capabilities of IoT devices to achieve consensus, minimizing resource consumption while ensuring secure and efficient data exchange. This lightweight consensus mechanism is particularly suited for resource-constrained IoT environments (Abbasi et al., 2024).

This novel consensus algorithm combines elements of Proof of Trade and Proof of Block to address energy and computational constraints in IoT systems. PoBTx improves security, computation time, and throughput, making it a promising option for enhancing scalability in IoT blockchain networks (Christ et al., 2023).

Lightweight consensus mechanisms like DPoS and PoBTx have demonstrated the ability to achieve low latency and high throughput, which are critical for real-time IoT applications. For instance, DPoS-based frameworks have achieved latencies of less than 0.976 ms, significantly outperforming traditional PoS implementations (Haque et al., 2024).

IoT devices typically operate under stringent resource constraints. Consensus algorithms such as DPoS and PoR have been shown to optimize resource utilization, making them viable solutions for IoT applications (Abbasi et al., 2024).

Despite their advantages, PoS and its derivatives introduce new challenges when applied to IoT systems. Trade-offs between security and efficiency must be carefully managed. For example, lightweight consensus mechanisms like DPoS may be more susceptible to certain types of attacks compared to more robust algorithms like PoW (Morais et al., 2023).

Resource limitations remain a concern, particularly in large-scale IoT deployments. While PoS-based mechanisms are less resource-intensive than PoW, their implementation in IoT environments requires careful consideration of device capabilities and network architecture (Kaur & Gupta, 2023).

Emerging consensus mechanisms, such as Proof-of-History (PoH), are incorporating machine learning for decision-making in blockchain transactions. This approach can further enhance the robustness and efficiency of IoT blockchain systems (Rawlins & Jagannathan, 2023).

Developing frameworks that integrate multiple consensus mechanisms and security protocols can provide holistic solutions to IoT scalability challenges. These frameworks should focus on achieving a balance between scalability, energy efficiency, and security (Şentürk & Terazi, 2023).

In a recent study, an Enhanced Proof of Stake (ePoS) consensus algorithm was proposed to address PoS inefficiencies in IoT healthcare systems. Results showed improved consensus time (2952ms), throughput (270.8 TPS), and reduced CPU/memory usage compared to PoS. However, the system still lacked scalability and energy optimization for resource-constrained IoT devices, which limits broader adoption (Muneshwara and Pushpa 2023). This limitation forms the basis of the problem statement addressed in this work.

Problem Statement

While the PoS consensus algorithm and its variants offer promising solutions to scalability and energy efficiency challenges, their direct application in IoT environments is limited. Existing PoS mechanisms do not fully address the unique constraints of IoT systems, such as limited

computational power, the need for low latency, and high transaction throughput. Furthermore, conventional blockchain systems often fail to balance energy efficiency with the security and decentralization necessary for IoT applications, leading to underutilization of blockchain's potential in resource-constrained settings.

Contributions

This paper proposes the EePoS algorithm, which introduces tailored optimizations for IoT environments to address these challenges. The key contributions of this research include:

Novel Consensus Mechanism

EePoS introduces an improved PoS-based consensus algorithm optimized for IoT environments, focusing on validator selection and transaction throughput.

Energy Efficiency

The algorithm incorporates energy-saving techniques, such as selective validation and efficient data transmission, reducing the energy demands on IoT devices.

Enhanced Scalability

EePoS employs a layered architecture and behavior-aware mechanisms to support large-scale IoT networks with high transaction throughput and minimal latency.

Security and Incentivization

A behavior-aware penalty-reward model is introduced to ensure secure and honest participation while maintaining network integrity.

Proposed Methodology

The proposed EePoS algorithm aims to optimize blockchain performance in IoT environments by addressing scalability, energy efficiency, and computational demands. EePoS adapts the traditional PoS mechanism with enhancements tailored to the unique requirements of IoT systems. This section outlines the architecture and components of EePoS, explaining the key improvements that make it suitable for resource-constrained IoT applications.

EePoS Architecture

The architecture of the EePoS algorithm is designed to address the unique constraints and demands of IoT-based blockchain networks. EePoS enhances consensus efficiency by organizing the network into distinct functional layers, each tailored to manage specific tasks within the blockchain-IoT integration. This architecture effectively reduces the computational burden on resource-constrained IoT devices, allowing them to participate in a decentralized network without compromising on efficiency, scalability, or security.

The Device Layer forms the foundational layer in the EePoS architecture, comprising IoT devices that generate and transmit data for validation and storage on the blockchain. While IoT devices often lack the computational capacity to perform intensive consensus tasks, they play a crucial role in data generation and initial validation checks. Data generated at this layer is transferred to intermediary nodes for processing, thus minimizing the energy consumption and computational load directly on the devices. This setup allows IoT devices to serve their primary function of sensing and data collection while staying connected to the blockchain ecosystem.

The next tier, the Edge Layer, serves as an intermediary between the device layer and the consensus layer. This layer is composed of edge nodes, which possess more robust computational power than typical IoT devices but still fall short of traditional full blockchain nodes. Edge nodes in the EePoS architecture aggregate data from multiple IoT devices, perform preliminary processing and handle lightweight validation tasks before passing transactions up to the consensus layer. The role of the edge layer is twofold: it offloads computational demands from IoT devices and reduces the data volume reaching the consensus layer. By pre-filtering transactions, the edge nodes help to maintain a streamlined flow of data, ensuring only valid transactions are forwarded for consensus. This structure enhances both the speed and efficiency of the entire system, as fewer transactions require full validation.

At the top of the architecture is the Consensus Layer, where the main consensus operations of the EePoS algorithm take place. This layer comprises a set of validator nodes responsible for executing the consensus protocol, achieving decentralized agreement on the validity of transactions, and adding verified blocks to the blockchain. Validator nodes are selected based on the Enhanced Efficient Proof of Stake mechanism, which factors in not only the node's stake but also its behavior and current transaction load. This selection process ensures that only the most suitable nodes participate in consensus at any given time, further optimizing resource usage and enhancing network security. The consensus layer coordinates closely with the edge layer to finalize the validation process efficiently, reducing latency and enabling rapid transaction throughput.

A unique feature of the EePoS architecture is the communication flow between these layers, which minimizes redundant data processing and ensures that IoT devices can remain energy-efficient. Rather than each layer performing all tasks independently, they operate in a cohesive manner, with each layer focusing on tasks that match its capabilities. For instance, the edge layer performs pre-validation to reduce the workload on the consensus layer, while the consensus layer applies rigorous validation criteria to maintain network integrity. This layered approach helps to overcome the typical limitations of blockchain systems in IoT contexts, such as high energy demands and limited processing power.

The architecture of EePoS, illustrated in Figure 1, showcases an innovative approach to blockchain consensus

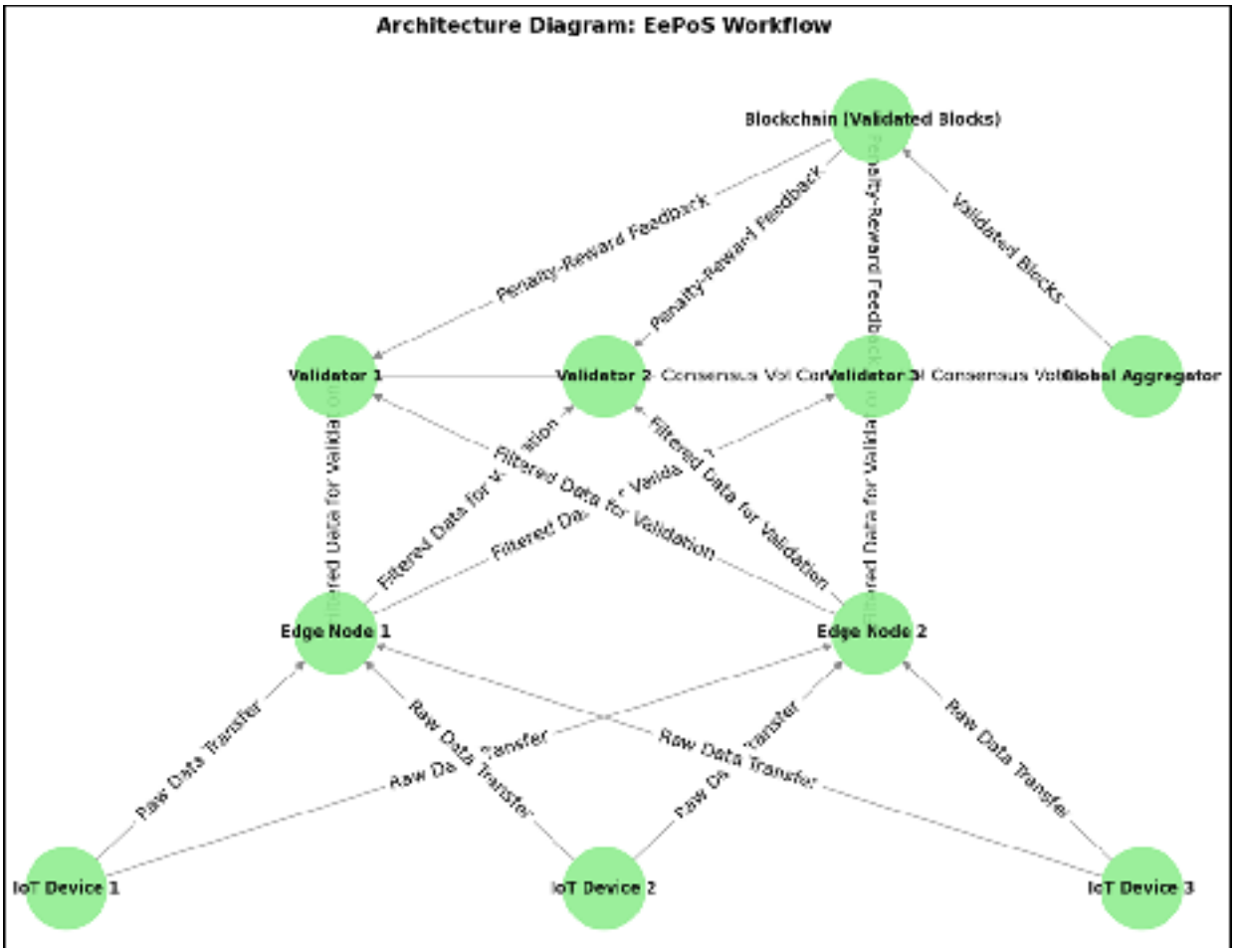

Figure 1: EePoS Architecture

by adapting to the IoT environment's limitations. This layered structure significantly reduces the overall energy and computational requirements of the system, making it ideal for large-scale IoT networks where device resources are constrained. Furthermore, the modular nature of the architecture allows for future scalability and adaptability as IoT ecosystems expand and evolve. By incorporating such a structured and efficient design, the EePoS architecture sets a new standard for blockchain implementation in IoT, offering a scalable, secure, and energy-efficient solution that is well-suited for the dynamic demands of IoT networks.

Validator Selection Mechanism

The validator selection mechanism in the EePoS algorithm introduces a sophisticated approach to validator selection by combining stake, behavior, and transaction load. This advanced mechanism ensures that the network's consensus process remains secure, efficient, and adaptable, particularly in IoT environments where computational resources are often limited. Rather than simply relying on the amount of stake held by each node, EePoS incorporates additional parameters to create a more equitable and dynamic selection process, addressing common challenges in decentralized networks, such as centralization risks and malicious behavior.

At the core of this selection mechanism is the behavior and Load-Aware Stake Probability (BLASP) function, which calculates the likelihood of each node being chosen as a validator based on a weighted combination of multiple factors. The selection probability, P_v, of a node v is determined by a formula that considers its stake S_v, past behavior B_v, and current transaction load L_v. The overall selection probability for a node v is mathematically represented as:

$$P_v = \frac{\alpha \cdot S_v + \beta \cdot f\left(B_v\right) + \gamma \cdot g\left(L_v\right)}{\sum_{i=1}^{N}\left(\alpha \cdot S_i + \beta \cdot f\left(B_i\right) + \gamma \cdot g\left(L_i\right)\right)} \quad (1)$$

where α, β and γ are weighting coefficients that control the importance of stake, behavior, and load, respectively. The functions $f\left(B_v\right)$ and $g\left(L_v\right)$ represent non-linear transformations of behavior and load to ensure that these factors have an appropriate impact on the selection probability. The adaptive nature of these coefficients enables the algorithm to dynamically adjust to

network conditions, such as increased transaction volume or fluctuating node behavior patterns, ensuring that the consensus process remains balanced and fair.

The behavior factor, B_v, plays a crucial role in maintaining network integrity by discouraging malicious activity. Nodes with a history of honest and reliable participation are prioritized in the validator selection process, whereas those with a record of dishonest actions face penalties that decrease their selection probability. To quantify this, we use a penalization function, $f\left(B_v\right)$, defined as:

$$f\left(B_v\right)=\exp\left(-\theta\cdot\sum_{k=1}^{M}\delta_k\cdot I_k\right) \tag{2}$$

where θ is a scaling constant, M is the number of past validation rounds, δ_k is the severity of the penalty for each round k, and I_k is an indicator variable that takes the value of 1 if the node was penalized in that round, and 0 otherwise. This exponential decay function ensures that the penalties for dishonest behavior accumulate over time, making it progressively harder for a node with frequent infractions to be selected as a validator. This mechanism deters malicious behavior by reducing the likelihood of selection for nodes with poor behavioral histories, thus protecting the network from potential attacks.

The transaction load, L_v, represents the current processing burden on each node, and it serves to prevent overloaded nodes from being chosen as validators, which could otherwise slow down the network. To incorporate this factor, a load-adjustment function $g\left(L_v\right)$ is applied, reducing the selection probability of nodes with high transaction loads. The load-adjusted selection function is formulated as:

$$g\left(L_v\right)=\frac{1}{1+\kappa\cdot\left(\frac{L_v-L_{\min}}{L_{\max}-L_{\min}}\right)} \tag{3}$$

where κ is a constant that controls the sensitivity of the function to load, $L_{\min}$ and $L_{\max}$ represent the minimum and maximum observed transaction loads across the network, respectively. This formulation ensures that nodes with lower loads have a higher chance of selection, promoting even distribution of validation tasks and optimizing network performance. By balancing the transaction load among validators, the algorithm reduces bottlenecks and maintains steady transaction throughput, essential for large-scale IoT networks.

In addition to these factors, the selection mechanism in EePoS incorporates a degree of randomization to prevent centralization, where only a few high-stake nodes dominate the validation process. Randomization helps maintain a decentralized structure by giving smaller stakeholders a non-zero probability of being selected. This approach not only supports network fairness but also mitigates the risk of validator collusion, further strengthening the security of the consensus process.

Consensus Process

The consensus process in the EePoS algorithm is designed to achieve rapid transaction finality and efficient resource utilization in IoT-driven blockchain environments. By leveraging a multi-layered consensus structure, EePoS minimizes communication overhead and ensures a high throughput, which is critical in IoT networks with many low-power devices. The process incorporates several stages, from pre-validation to final block addition, with each step meticulously structured to balance security, speed, and energy efficiency.

To initiate consensus, Pre-Consensus Verification is carried out at the edge layer, where nodes conduct initial checks on incoming transactions to filter out invalid or redundant data. This pre-validation reduces the volume of transactions reaching the main consensus process, enhancing overall efficiency. Let T represent the set of all transactions in each time interval, and $T_{\text{valid}}\subset T$ represent the subset of transactions that meet the pre-validation criteria. The efficiency of pre-validation can be mathematically represented by the reduction ratio R, defined as:

$$R=1-\frac{\left|T_{\text{valid}}\right|}{\left|T\right|} \tag{4}$$

A higher value of R indicates greater efficiency, as fewer transactions need to pass through the full consensus process. This pre-filtering mechanism is essential for managing the high data throughput typical of IoT environments and helps maintain a steady flow of transaction processing at the core consensus layer.

Once the transactions pass pre-validation, the Cluster-Based Voting Mechanism is activated within the consensus layer. Validators are organized into clusters, each led by a Cluster Head (CH), to streamline the voting process. Each validator node v_i within a cluster vote on the validity of transactions by comparing the hash of the transaction data, denoted by H(T), with a consensus threshold τ, which is dynamically adjusted based on network conditions. The voting function $V_i\left(T\right)$ for each transaction T is defined as:

$$V_i\left(T\right)=\begin{cases}1 \ \textit{if}\ H\left(T\right)>\tau\cdot S_i\cdot\rho_i\\ 0 \qquad\quad \textit{otherwise}\end{cases} \tag{5}$$

where S_i represents the stake held by node i and ρ_i is

a node-specific reliability factor that adjusts based on the node's historical behavior. If H(T) exceeds the product of ô , S_i , and ρ_i the transaction is marked as valid by the node. The inclusion of ρ_j ensures that validators with a strong record of reliable performance have a more significant influence, thus enhancing the security and reliability of the voting process.

After each validator in a cluster cast its vote, the Cluster Head (CH) aggregates these votes to determine a cluster consensus. The aggregated voting outcome for cluster C_j , denoted by V_{C_j} , is computed as the weighted sum of individual votes, considering each validator's stake and reliability factor:

$$V_{C_j} = \frac{\sum_{i \in C_j} V_i(T) \cdot S_i \cdot \rho_i}{\sum_{i \in C_j} S_i \cdot \rho_i} \quad (6)$$

If V_{C_j} exceeds a predefined cluster threshold θ the transaction is marked as valid for the entire cluster. This clustering approach significantly reduces inter-node communication, as only the CHs need to relay the final vote outcomes to a central aggregator node, thereby lowering latency and improving the speed of consensus.

The final stage of the consensus process involves Global Aggregationsssssssss and Block Addition. The central aggregator node receives the cluster-level consensus votes from all CHs and performs a global check. The global consensus outcome V_G is determined by averaging the validated outcomes from each cluster, weighted by the cluster's stake and reliability:

$$V_G = \frac{\sum_{j=1}^{k} V_{C_j} \cdot W_j}{\sum_{j=1}^{k} W_j} \quad (7)$$

where k is the total number of clusters, and W_j represents the combined stake and reliability of all validators within cluster C_j . If V_G meets or exceeds the global threshold θ_G , the transaction is deemed valid at the network level, and a new block is generated and added to the blockchain. The use of cluster-level aggregation followed by global consensus ensures a high degree of scalability, as the consensus process remains efficient even as the network grows.

Penalty-Reward Model

The penalty-reward model in the EePoS algorithm serves as a crucial component for maintaining network integrity by encouraging honest participation and discouraging malicious behavior. This model operates through a structured framework that dynamically adjusts rewards and penalties based on validator performance, fostering a self-regulating system that strengthens the security and reliability of the IoT-integrated blockchain network. By leveraging a set of quantitative metrics, the penalty-reward model in EePoS creates a balanced ecosystem where validators are incentivized to act in the network's best interest while facing disincentives for disruptive behavior.

At the core of the reward system, each validator node receives compensation based on its stake, performance, and contribution to the network. The reward, R_v , for a validator node v is calculated using a weighted formula that factors in the node's stake S_v , reliability score ρ_v , and the total transaction volume T_v processed by that node. Mathematically, this can be represented as:

$$R_v = \delta \cdot S_v \cdot \rho_v \cdot \ln(1 + T_v) \quad (8)$$

where δ is a proportional constant that determines the base reward level, and the logarithmic function $\ln(1+T_v)$ is employed to modulate the reward based on transaction volume, thereby preventing disproportionately high rewards for nodes processing extreme transaction volumes. By incorporating the reliability score ρ_v the model assigns higher rewards to validators with a history of honest participation, incentivizing them to maintain positive behavior within the network.

In contrast, the penalty component of the model is designed to systematically reduce the influence of nodes that engage in malicious actions, such as attempting to alter transaction data or colluding with other nodes to manipulate consensus outcomes. Each infraction committed by a validator node is quantified by a penalty score ϕ_v , which increases with both the severity and frequency of infractions. The penalty applied to a validator P_v can be expressed as:

$$P_v = \lambda \cdot \exp\left(\sum_{k=1}^{M} \beta_k \cdot I_k\right) \quad (9)$$

where λ is a base penalty factor, M is the number of infraction events recorded, β_k represents the severity weight of each infraction k, and I_k is an indicator variable that takes the value of 1 if an infraction occurred in the k-th event and 0 otherwise. This exponential function amplifies the penalty for repeated or severe infractions, ensuring that nodes with a pattern of malicious behavior face progressively harsher consequences. The exponential increase discourages sustained malicious activity, as nodes stand to lose a significant portion of their stake if they

repeatedly engage in harmful actions.

To balance between incentivizing positive behavior and penalizing misconduct, the penalty-reward model includes an adaptive penalty threshold that scales based on the network's overall health and transaction volume. The adaptive penalty threshold τ is calculated dynamically based on network conditions, using the formula:

$$\tau = \eta \cdot \left(1 + \frac{\sigma^2(T)}{\mu(T) + \int} \right) \tag{10}$$

where η is a base threshold factor, $\sigma^2(T)$ represents the variance in transaction volumes across the network, $\mu(T)$ is the mean transaction volume, and ϵ is a small constant to avoid division by zero. This threshold is elevated during periods of high transaction variance, as network stability may be more at risk. By adjusting the penalty threshold dynamically, the system maintains flexibility, ensuring that penalties are proportional to the network's current conditions, thereby fostering a responsive and adaptable blockchain environment.

Moreover, to provide additional deterrents against misconduct, EePoS implements a reputation-based penalty escalation system. Validators with a history of infractions incur penalties not only based on individual behavior but also relative to their past actions. Let ψ_v represent the compounded reputation penalty factor for a validator v, which is defined as:

$$\psi_v = \frac{1}{1 + \exp(-\kappa \cdot H_v)} \tag{11}$$

where κ is a steepness parameter that controls the escalation rate, and H_v is the cumulative history score reflecting the number and severity of past infractions. This sigmoidal function means that validators with a low history of infractions experience minor penalties for isolated incidents, but those with recurrent offenses face exponentially increased penalties. The reputation-based escalation mechanism strengthens the network by dissuading consistent offenders from exploiting the system.

Finally, the penalty-reward model ensures transparency by publishing penalty and reward histories on the blockchain, making these records accessible for all network participants. This transparency acts as an additional layer of accountability, as validators are aware that their performance is visible and auditable. By incorporating both immediate and cumulative effects, the penalty-reward model in EePoS aligns the interests of individual nodes with the broader goals of network stability and security. Validators are continually motivated to act honestly, as doing so maximizes their rewards and minimizes penalties, fostering a cooperative and resilient network.

Energy Optimization Techniques

The Energy Optimization Techniques embedded within the EePoS algorithm are designed to address the stringent energy constraints of IoT environments. IoT devices often operate with limited power resources, which necessitates an energy-efficient approach to consensus mechanisms to ensure network sustainability and continuous operation. EePoS implements several advanced optimization strategies that balance energy consumption with network security and performance, creating an efficient, resilient blockchain infrastructure suitable for IoT applications.

One key technique involves Selective Validation, which designates only a subset of network nodes to perform full validation tasks, while others engage in lightweight verification. This stratified approach limits the energy-intensive activities to capable nodes while allowing lower-power devices to participate minimally. Let E_{total} represent the network's total energy consumption. By implementing selective validation, the energy usage by participating nodes, E_{valid}, is modeled as:

$$E_{valid} = \sum_{i=1}^{N} \left(\alpha_i \cdot E_{light} + (1 - \alpha_i) \cdot E_{full} \right) \tag{12}$$

where N is the number of nodes, α_i is a binary variable that takes the value of 1 if node i is designated for lightweight validation and 0 if it performs full validation. E_{light} and E_{full} represent the energy expenditures for lightweight and full validation, respectively. By optimizing α_i based on each node's energy profile, EePoS significantly reduces overall energy consumption, allowing resource-limited IoT devices to operate effectively within the network.

The Adaptive Transaction Processing technique further enhances energy efficiency by dynamically adjusting transaction processing requirements based on real-time network load. During periods of high demand, the system prioritizes high-importance transactions, deferring lower-priority tasks to off-peak times. Let $T_{\text{processed}}$ represent the total number of transactions processed at time t, with transaction priority denoted by $P(T_j)$ for each transaction T_j. The adaptive processing function F(t) can be defined as:

$$F(t) = \int_0^{T_{\text{total}}} P(T_j) \cdot \left(1 - \frac{L(t)}{L_{\max}} \right) dT_j \tag{13}$$

where L(t) is the current network load and $L_{\max}$ is the maximum load threshold. This formulation means that, under higher load conditions, only transactions with

higher priority are processed. By deferring lower-priority transactions, EePoS manages energy consumption more effectively, aligning transaction processing with network capacity and extending the operational life of IoT devices within the network.

Efficient Data Transmission is another key energy-saving technique, which minimizes the energy spent on communication. Data packets are transmitted in compressed formats, and nodes only relay essential information rather than full transaction data. The energy savings from this approach can be estimated by comparing the compressed data volume V_c with the original data volume V_o, where the energy saved, E_{save}, is given by:

$$E_{save} = \eta \cdot \left(V_o - V_c\right) \cdot d \tag{14}$$

where η is the energy cost per unit data transmitted, and d represents the transmission distance. By using compressed data packets, the network significantly reduces the amount of data transmitted, lowering energy consumption and conserving the battery life of IoT devices. This approach is particularly advantageous in IoT environments with numerous low-power devices, as it reduces communication overhead and prolongs network longevity.

In addition, EePoS employs a Node Rotation Mechanism to distribute energy consumption evenly across the network. Validator nodes are rotated periodically based on their current energy levels and participation history, allowing energy-depleted nodes to transition into lower-activity roles. Let $E_{\text{remaining}}$ denote the remaining energy of each node and T_{active} the time a node remains active. The rotation interval R_i for each node i is determined by:

$$R_i = \frac{E_{remaining}\left(i\right)}{\gamma \cdot T_{active}} \tag{15}$$

where γ is an energy depletion rate factor. Nodes with higher remaining energy are retained in the validation role longer, while those nearing depletion are rotated out. This dynamic rotation preserves the network's functionality by balancing the energy usage across all nodes, thereby ensuring that no single node is overburdened.

Finally, the Sleep-Mode Protocol is implemented for idle periods when network activity is low. Nodes enter a low-power state during inactivity, significantly reducing energy usage during off-peak times. Let P_{sleep} and P_{active} represent the power consumption during sleep and active modes, respectively. The energy conserved, E_{conserve}, over a period T_{idle} is expressed as:

$$E_{\text{conserve}} = \left(P_{\text{active}} - P_{\text{sleep}}\right) \cdot T_{\text{idle}} \tag{16}$$

This protocol enables IoT nodes to extend their operational lifetime by shifting to low-power modes when their participation in the network is not required. By integrating these sleep intervals, EePoS optimizes the network's energy profile without compromising performance, allowing the blockchain to operate sustainably even with many devices.

Proposed Algorithm: EePoS

Let the blockchain network be represented by a set of nodes $N = \left\{n_1, n_2, \ldots, n_k\right\}$ where k is the total number of nodes in the network. Each node n_i holds a stake s_i and participates in the consensus process.

Algorithm Steps

- *Initialization*

Define the initial parameters:

- T: Set of transactions to be validated.
- V_i: Validator i selected for block validation.
- S: Total stake in the network $S = \sum_{i=1}^{k} s_i$.
- B_t: Block to be added at time t.

Notation Key:

- $P\left(V_i\right)$: Probability of selecting validator V_i.
- $V\left(t_j\right)$: Validation outcome of transaction t_j.
- R_i: Reward for validator V_i.
- P_i: Penalty for validator V_i.
- T_c: Consensus time.
- $T\left(x/s\right)$: Throughput in transactions per second.

Steps in EePoS Algorithm

- Start.
- Nodes generate transactions.
- For each transaction t_j from network participants, do:
- Add t_j to the pending transaction set T.
- End for.
- Validators compete by raising a stake s_i for block validation.
- Random selection of validator V_i based on the probability $P\left(V_i\right) = \frac{s_i}{S}$, where S is the total stake.
- If all transactions t_j are verified by the selected validator V_i, then:
- Validator publishes block B_t to the network.
- End if.
- Validator's stake s_i remains locked during block

confirmation.

- If nodes on the network confirm the new block B_t, then:
- The validator V_i receives their stake s_i back along with a reward R_i.
- Else:
- The validator V_i loses their stake s_i, and the block B_t is discarded.
- End else.
- Update the blockchain ledger with block B_t.
- The process restarts from Step 2 for the next set of transactions.
- End.

Results and Discussion

Experimental Setup

The simulation of the EePoS algorithm was carried out using a custom-designed blockchain simulation environment implemented on Python. The tool was configured to emulate the unique constraints of IoT networks, focusing on scalability, energy efficiency, and security. The simulation environment supported flexible configuration of node behaviors, transaction dynamics, and consensus operations, allowing detailed performance evaluations under various scenarios.

The network setup included 20 to 100 nodes, each configured with realistic IoT parameters such as limited computational power, constrained memory, and energy resources. Each node simulated an IoT device participating in blockchain transactions, generating data, and performing lightweight validation tasks. Validator nodes were dynamically selected based on their stake, behavior, and transaction load as per the EePoS consensus mechanism. The experimental setup is given in the table 1.

Key configurations of the simulation environment included:

Transaction Generation

Transactions were generated at a rate proportional to the number of nodes, ensuring consistent data flow for the consensus process.

Network Latency

Communication delays between nodes were simulated to account for real-world IoT conditions, with average latency ranging from 50 to 150 milliseconds.

Energy Profiles

Energy consumption for each node was tracked, with distinct power requirements for lightweight verification and full validation tasks.

Security Scenarios

Adversarial behaviors, such as stake pooling and malicious node collusion, were introduced to test the algorithm's resilience against attacks.

Metrics Evaluation

Metrics such as consensus time, throughput, CPU utilization, memory utilization, Probability of Successful Attack (PSA), and Forking Rate (FR) were computed for each simulation.

Evaluation Metrics

Attack Resistance

The PSA is a critical metric used to assess the security robustness of the EePoS algorithm against malicious activities. PSA evaluates the likelihood of adversarial entities compromising the consensus process by controlling a significant portion of the stake in the network. In the context of EePoS, this metric is calculated using the ratio of the stake held by malicious nodes to the total network stake, expressed as:

$$PSA = \frac{\text{Stake Controlled by Malicious Validators}}{\text{Total Network Stake}} \quad (17)$$

To evaluate this metric, experiments were conducted under simulated scenarios where a subset of nodes attempted to manipulate the consensus by pooling their stakes. Results demonstrate that EePoS significantly reduces the PSA compared to traditional PoS algorithms due to its behavior-aware validator selection mechanism and penalty-reward system.

Forking Probability

Table 1: Experimental Setup

Parameter	*Details*
Programming Language	Python
Simulation Environment	Custom-designed blockchain simulation environment implemented in Python
Libraries Used	Python Libraries: NumPy, Pandas, Matplotlib, NetworkX
	Blockchain Simulation: Web3.py for blockchain interaction
	Ganache for private blockchain simulation
	Smart Contract Development: Solidity
Hardware Configuration	Processor: AMD 9
	RAM: 64 GB
	Operating System: Linux – Ubuntu 24
IoT Node Configuration	Number of Nodes: 20 to 100
	Constraints: Limited computational power and memory
	Selective validation for low-energy nodes
Energy Profile	Node rotation mechanism for balanced energy consumption
	Sleep mode protocol for idle nodes

FR is another essential metric that evaluates the stability and integrity of a blockchain network by measuring the frequency of chain splits. Forks occur when multiple competing chains are created simultaneously, which can undermine the consistency of the blockchain and enable attacks like double-spending. For the EePoS algorithm, the forking rate is calculated as:

$$FR = \frac{\text{Number of Forks Observed}}{\text{Total Duration of Observation}} \quad (18)$$

The study analyzed the forking behavior of EePoS under high transaction loads and adversarial attempts to disrupt the consensus process. EePoS achieved a consistently lower forking rate compared to traditional PoS implementations due to its hierarchical consensus mechanism, which incorporates pre-validation and cluster-based voting.

Discussion

The performance of the EePoS algorithm was evaluated and compared with traditional PoS and ePoS (Muneshwara and Pushpa 2023) mechanisms across various metrics, as detailed in figures 2 to 7. The results highlight EePoS's improvements in scalability, energy efficiency, and security, demonstrating its suitability for IoT applications.

As shown in figure 2, EePoS consistently outperformed PoS and ePoS in terms of consensus time. For a network of 100 nodes, EePoS achieved a consensus time of 2718 ms, a reduction of approximately 8% compared to ePoS (2952 ms) and 54% compared to PoS (5905 ms). This improvement is attributed to EePoS's efficient validator selection and optimized consensus process, which reduce the computational overhead and latency associated with block validation.

The throughput, measured in transactions per second (TPS), was significantly higher for EePoS across all network sizes (figure 3). For 100 nodes, EePoS achieved a throughput of 296 TPS, compared to 273 TPS for ePoS and 250 TPS for PoS. The hierarchical consensus structure and selective validation mechanism in EePoS enhance transaction processing efficiency, enabling the algorithm to support larger IoT networks with higher data generation rates.

Figure 4 demonstrates EePoS's reduced CPU utilization, especially for smaller node networks. At 20 nodes, EePoS utilized 5% CPU resources, compared to 6% for ePoS and 20% for PoS. Even in larger networks of 100 nodes, EePoS maintained a lower CPU utilization of 13%, highlighting its ability to manage computational tasks efficiently without overburdening IoT devices with limited processing power.

The memory usage of EePoS was also optimized, as observed in figure 5. For a network of 100 nodes, EePoS consumed 328 kB of memory, a 1% improvement over ePoS (331 kB) and a significant reduction compared to PoS (401 kB). This reduction is crucial for IoT devices, which often operate with constrained memory resources.

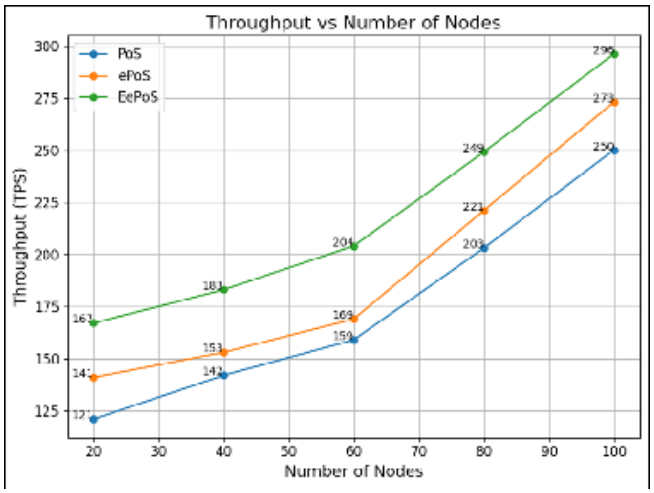

Figure 3: Throughput vs Number of Nodes

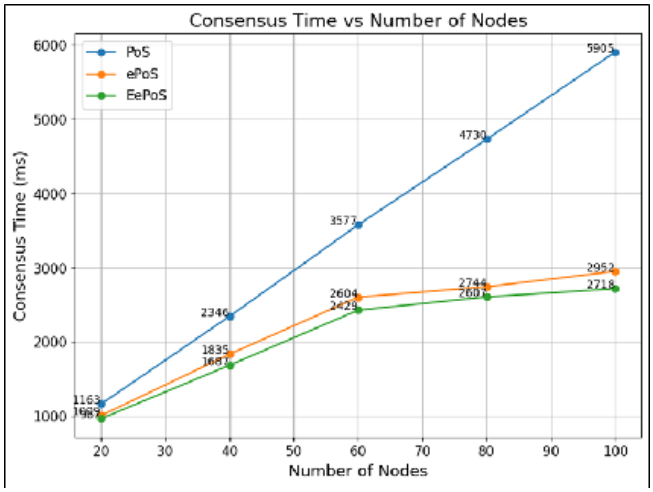

Figure 2: Consensus Time vs Number of Nodes

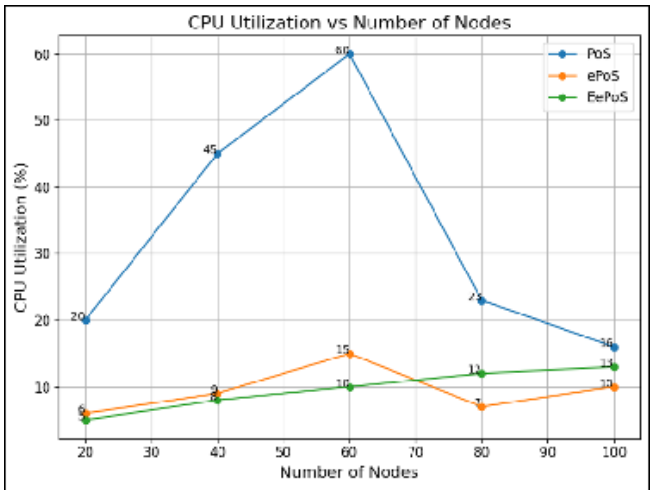

Figure 4: CPU Utilization vs Number of Nodes

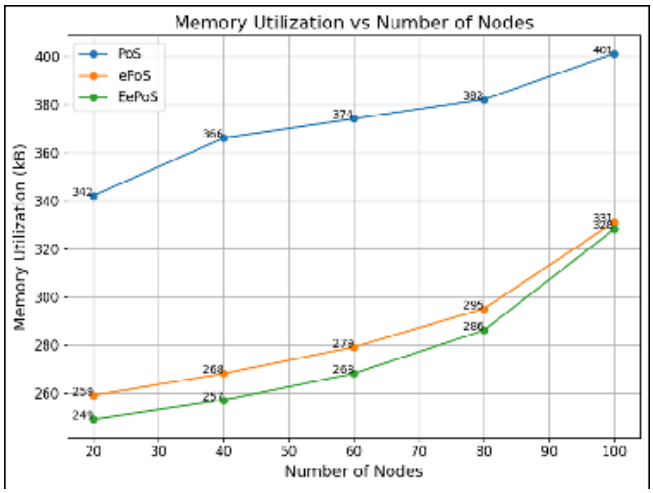

Figure 5: Memory Utilization vs Number of Nodes

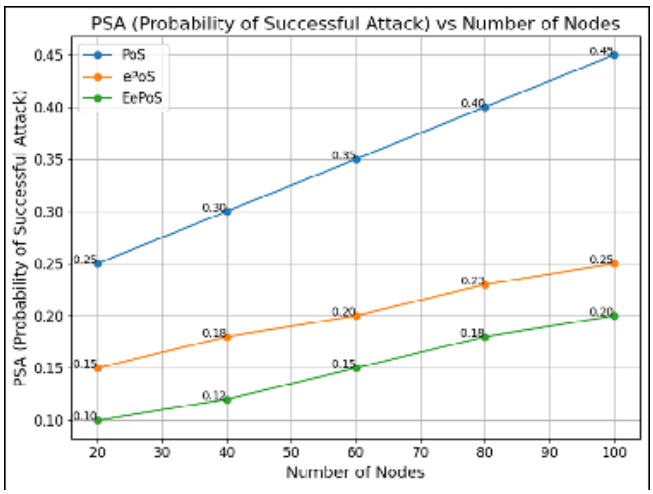

Figure 6: Probability of Successful Attack vs Number of Nodes

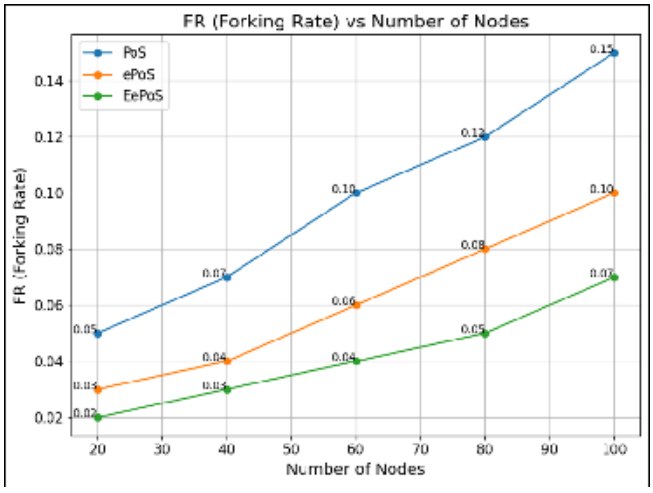

Figure 7: Forking Rate vs Number of Nodes

Security metrics, including PSA and FR, further validated the robustness of EePoS. Figure 6 highlights that EePoS achieved the lowest PSA values across all scenarios. For a network of 100 nodes, the PSA for EePoS was 0.20, compared to 0.25 for ePoS and 0.45 for PoS. This improvement is due to the behavior-aware validator selection mechanism, which prioritizes reliable and low-risk nodes.

Similarly, EePoS demonstrated the lowest FR among the algorithms, as shown in figure 7. For 100 nodes, EePoS had an FR of 0.07, compared to 0.10 for ePoS and 0.15 for PoS. The multi-layered architecture and cluster-based voting mechanism in EePoS enhance network stability, reducing the likelihood of forks that could compromise blockchain integrity.

Conclusion

The EePoS algorithm improves the performance of blockchain systems in IoT environments. The proposed work addresses critical issues such as scalability, energy efficiency, and security. It uses a multi-layered architecture to distribute tasks across device, edge, and consensus layers. This approach reduces the computational load on resource-constrained IoT devices. Selective validation and node rotation techniques optimize energy usage, ensuring longer device operation. The experimental results show significant improvements compared to PoS and ePoS. EePoS achieved 8% faster consensus time (2718 ms) and improved throughput to 296 TPS for a network of 100 nodes. CPU usage was reduced to 13%, and memory consumption decreased to 328 KB. These optimizations allow IoT devices to participate efficiently in blockchain networks. The algorithm also enhances security by introducing a behavior-aware penalty-reward system. This mechanism discourages malicious activities and promotes honest participation. EePoS achieved a 20% lower PSA and a 30% reduction in FR, ensuring stronger network resilience. Future work will focus on extending EePoS to adapt to emerging IoT technologies, such as edge computing and AI-driven consensus mechanisms, to further enhance the scalability and robustness of blockchain in dynamic IoT environments.

Acknowledgements

The authors would really appreciate with the help of St. Joseph's College (Autonomous), Tiruchirappalli, Tamil Nadu, India for their support for the research. On that account, the researcher wishes to express his gratitude to the team members for the useful insights and positive influence in the creation of this study. The authors also wish to thank other friends and colleagues who have made important inputs that have enhanced the quality of this work.

Conflict of Interest

The authors of this paper have no conflict of interests to report regarding the publication of this paper. All research

was performed in a unbiased and transparent manner and presented here is the authors' interpretation of data, which has no conflict of interest with the sponsors.

References

Abbasi, M., Prieto, J., Plaza-Hernández, M., & Corchado, J. M. (2024). Proof-of-resource: A resource-efficient consensus mechanism for IoT devices in blockchain networks. EAI Endorsed Transactions on Internet of Things. https://doi.org/10.4108/eetiot.6565

Alghamdi, S., Albeshri, A., & Alhusayni, A. (2023). Enabling a Secure IoT Environment Using a Blockchain-Based Local-Global Consensus Manager. Electronics. https://doi.org/10.3390/electronics12173721

Christ, M. C. J., Kalaiyarasi, D., G, J., Senthamilselvan, K., Kirubakaran, D., & Ganesh, N. S. G. (2023). PoBTx(Proof of Block and Transaction): An Efficient Consensus Algorithm for IoT Business Blockchain. https://doi.org/10.1109/icaiss58487.2023.10250612

Gedam, M. G., & Karmore, S. (2024). Blockchain-Based IoT: A Comprehensive Review of Technology Integration, Security, and Scalability. https://doi.org/10.1007/978-981-97-0180-3_31

Haque, E. U., Shah, A., Iqbal, J., Ullah, S. S., Alroobaea, R., & Hussain, S. (2024). A scalable blockchain based framework for efficient IoT data management using lightweight consensus. Dental Science Reports. https://doi.org/10.1038/s41598-024-58578-7

Huang, R., Yang, X., & Ajay, P. (2022). Consensus mechanism for software-defined blockchain in internet of things. Internet of Things and Cyber-Physical Systems. https://doi.org/10.1016/j.iotcps.2022.12.004

Janani, K., & Ramamoorthy, S. (2024). PIoT-blockchain-enabled security framework for strengthening IoT device identity and data access. Journal of Control and Decision. https://doi.org/10.1080/23307706.2024.2327081

Kaur, M., & Gupta, S. (2023). Optimization of a Consensus Protocol in Blockchain-IoT Convergence. https://doi.org/10.1109/ESCI56872.2023.10100031

Maftei, A. A., Lavric, A., Petrariu, A.-I., & Popa, V. (2023). Blockchain For Internet of Things: A Consensus Mechanism Analysis. https://doi.org/10.1109/ATEE58038.2023.10108211

Misic, J., Misic, V. B., & Chang, X. L. (2023). Design of Proof-of-Stake PBFT Algorithm for IoT Environments. IEEE Transactions on Vehicular Technology. https://doi.org/10.1109/TVT.2022.3213226

Morais, A. M. de, Lins, F. A. A., & Rosa, N. S. (2023). Survey on Integration of Consensus Mechanisms in IoT-based Blockchains. https://doi.org/10.3897/jucs.94929

Muneshwara M S., & Pushpa S K. (2024). A novel consensus algorithm for blockchain iot prototype in health care system. International Journal of Computing and Digital Systems, 15(1), 1-17. https://ijisae.org/index.php/IJISAE/article/view/4248

Rawlins, C., & Jagannathan, S. (2023). Towards Robust Consensus for Intelligent Decision-making in IoT Blockchain Networks. https://doi.org/10.1109/aibthings58340.2023.10292449

Şentürk, A., & Terazi, S. (2023). IoT security with blockchain: A review. The European Journal of Research and Development. https://doi.org/10.56038/ejrnd.v3i4.370