Fog Node Trust Evaluation Technology Combined with a Consensus Mechanism and its Application

Guanglei Sheng[1,*] and Qingtao Wu[2]

[1]*School of Computer and Artificial Intelligence, Henan Finance University, Zhengzhou, Henan 451464, China*
[2]*School of Computer Science, Zhengzhou University of Aeronautics, Zhengzhou, Henan 450046, China*
E-mail: guangleisheng@163.com; wuqingtao2008@163.com
[*]*Corresponding Author*

Received 14 October 2025; Accepted 02 December 2025

Abstract

To address the issue of node trustworthiness in fog computing networks, this study proposes a fog node trust evaluation technique that integrates consensus mechanisms. This paper adopts the innovative proof of interest algorithm (PoIA) consensus mechanism to achieve lightweight verification. The core methods include formulaic reputation evaluation algorithms to quantify node behavior and economic incentives, as well as deposit mechanisms to increase attack costs. Experimental studies have shown that this technology can still maintain 93.2% availability even under high malicious node rates. After applying this technology, the throughput of the fog computing network is 985TPS, the latency is reduced to 125 ms, and the energy consumption is reduced by 26.8%. The experiment confirmed that this technology significantly improves the accuracy of malicious

Journal of ICT Standardization, Vol. 14_1, 69–100.
doi: 10.13052/jicts2245-800X.1413

behavior detection and resource allocation efficiency through the collaborative innovation of dynamic reputation evaluation and PoIA lightweight consensus, providing a reliable solution for large-scale fog computing deployment.

Keywords: Consensus mechanism, fog nodes, trust, blockchain.

1 Introduction

With the explosive growth of the number of IoT devices and the sharp rise in edge computing demand, fog computing, as an important extension of cloud computing, effectively alleviates the processing pressure of cloud centers and reduces transmission latency by bringing computing, storage and network services closer to data sources [1]. However, the distributed and heterogeneous characteristics of fog computing networks pose significant security challenges. These issues encompass malicious node behavior and resource constraint contradictions [2]. The traditional centralized trust management mechanism is difficult to adapt to the highly dynamic environment of fog computing and has the shortcomings of single point of failure and insufficient transparency [3]. Blockchain technology provides new ideas for trusted governance due to its decentralized, non-tamperable and traceable characteristics. However, directly applying existing blockchain solutions (such as DPoS and PBFT) to fog computing environments faces significant challenge. There is a fundamental contradiction between the computing and communication overhead of the consensus mechanism and the resource constraints of fog nodes, and the static consensus node election mechanism cannot effectively cope with the dynamic changes of the network [4]. The existing research has a single dimension of node behavior evaluation, lacks real-time adaptability, and the economic incentive mechanism is out of touch with trust evaluation, which makes the system vulnerable to Sybil attack and On-Off attack [5]. These limitations highlight the need to build a credible evaluation framework that is lightweight, highly adaptable and economically constrained.

Furthermore, with the explosive growth in the number of Internet of Things (IoT) devices, fog computing networks are increasingly being applied in key areas such as healthcare and smart cities. However, node trust issues may lead to data leakage and service interruptions, making the development of efficient trust evaluation mechanisms an urgent priority.

This paper aims to design and implement a fog node trust evaluation model integrating blockchain technology. The model integrates smart contract-driven resource allocation and dynamic reputation management through a three-tier end-fog-cloud service framework and adopts an innovative PoIA consensus mechanism to achieve lightweight verification. The core innovation of the research lies in the deep integration of node historical behavior, real-time performance and economic incentives, quantifying node credibility through formulaic reputation evaluation algorithms, and introducing deposit and deposit mechanisms to increase attack costs. The contribution of this paper is to build a complete and usable decentralized trust governance scheme. The test results verify that it still maintains 93.2% system availability and a throughput of 985TPS despite a high proportion of malicious nodes, providing a safe and efficient solution for large-scale fog computing deployment.

Common abbreviations used in this article: FSN (fog service node), DSS (data service subscriber), PoIA (proof of interest algorithm).

2 Related Work

As an important extension of cloud computing, fog computing effectively alleviates the processing pressure on cloud centers and reduces transmission delays by bringing computing, storage, and network services closer to data sources. However, its distributed and heterogeneous characteristics introduce severe security challenges, especially the issue of node trustworthiness. Traditional centralized trust management mechanisms are difficult to adapt to the highly dynamic fog environment and suffer from the drawbacks of single point of failure and insufficient transparency. Blockchain technology, with its decentralized, tamper-proof, and traceable characteristics, provides new ideas for trusted governance. However, directly applying existing blockchain solutions (such as DPoS and PBFT) faces the fundamental contradiction of computational and communication overhead and resource constraints. Existing research suffers from a single dimension of node behavior evaluation, a lack of real-time adaptability, and a disconnect between economic incentive mechanisms and trust evaluation, making the system vulnerable to Sybil attacks and On-Off attacks. This section systematically reviews relevant research progress from four aspects: trust management mechanisms, resource allocation optimization, security architecture design, and blockchain integration applications.

(1) Research on trust management mechanisms. Trust management is the core of ensuring security in fog computing environments. Kaur and Auluck (2023) proposed a real-time trust-aware scheduling scheme for fog–cloud systems [6]. They constructed a trust scoring model by dynamically monitoring node behavior indicators (such as task response time and computation accuracy) and employed fuzzy logic reasoning to achieve real-time updates of trust values. This scheme improved task allocation efficiency by 18% in a simulated environment, but it did not consider the dynamics of node resources and lacked economic incentive mechanisms to constrain malicious behavior. Alwakeel (2023) designed a keyword-based trust management system (KBTM) that utilizes natural language processing technology to analyze the semantic consistency between node service descriptions and actual performance, enabling rapid identification of fake service nodes [7]. Experiments showed that KBTM achieved a text matching accuracy of 91.7%, but its high computational overhead made it difficult to deploy widely in resource-constrained fog nodes. Bakhtiari et al. (2024) proposed a trust management scheme based on an improved genetic algorithm [8]. Through multi-objective optimization, it simultaneously balances trust values, resource utilization, and energy consumption indicators. Its adaptive crossover and mutation operators effectively avoid local optima. This scheme outperforms traditional algorithms by 25% in terms of iterative convergence speed, but the training time cost of genetic algorithms limits its application in scenarios with extremely high real-time requirements.

(2) Resource allocation and task scheduling optimization. Resource allocation in fog environments requires a balance between efficiency and trustworthiness. Faraji et al. (2024) modeled the resource allocation problem in fog computing using complex system theory and optimized task offloading strategies using a chaotic particle swarm algorithm, maximizing resource utilization while ensuring delay constraints [9]. Simulation results showed that the scheme maintained 86% resource utilization efficiency under high load conditions, but it did not integrate a trust evaluation mechanism, making it difficult to resist resource exhaustion attacks initiated by malicious nodes. Jain and Kumar (2023) designed a trusted resource allocation scheme (TRAS), which uses historical trust values as weight factors in resource auctions and achieves incentive-compatible resource allocation through a Vickrey auction mechanism [10]. This scheme has been validated on the Indian Telecom dataset for its effectiveness in suppressing false bidding, but its centralized auction architecture poses a single point of failure risk.

Ali et al. (2023) combined disaster genetic algorithms with blockchain technology to optimize cloud–fog–edge task scheduling, using blockchain to store node reputation records and dynamically adjusting scheduling strategies through the adaptive mutation strategy of genetic algorithms [11]. This scheme outperforms traditional methods by 31% in terms of the makespan metric, but the storage overhead of blockchain has not been fully optimized.

(3) Security architecture and privacy protection scheme. With the widespread adoption of fog computing in critical infrastructure, security architecture design has garnered significant attention. Alnaim and Alwakeel (2025) proposed a zero trust mechanism (ZTM-6G) for 6G networks, which achieves security protection for distributed edge and fog environments through micro-isolation technology and continuous identity verification [12]. ZTM-6G employs lightweight attribute-based encryption (ABE) to protect data transmission, with measured encryption latency increasing by only 7 ms on ARM-based processors. However, its dynamic policy update mechanism incurs high control signaling overhead. Fotia et al. (2023) systematically reviewed trust management technologies in edge IoT architectures, summarizing three mainstream approaches: reputation-based, hardware-software proof-based, and blockchain-based [13]. They identified cross-domain trust transfer and privacy protection as future research challenges. Reshi and Sholla (2024) designed privacy enhancement technology for fog computing (PET-Fog), which utilizes homomorphic encryption to process sensitive data and combines zk-SNARKs to enable verifiable computation [14]. This approach reduces the risk of data leakage by 74% in medical IoT scenarios, but computationally intensive encryption operations struggle to meet the demands of real-time tasks.

(4) Blockchain and consensus mechanism innovation. Blockchain technology provides a new paradigm for decentralized trust management. Alvi et al. (2024) developed a secure computing framework for fog-enabled industrial IoT (SCF-IIoT), integrating blockchain storage device identity and access policies, and employing an improved practical byzantine fault tolerant (PBFT) consensus to reduce communication complexity [15]. SCF-IIoT achieved 99.91% availability in factory automation testing, but its fixed election mechanism for consensus nodes struggled to adapt to dynamic changes in network topology. Choppara and Mangalampalli (2024) proposed a reliable task scheduler based on the actor–critic (A2C) algorithm, incorporating blockchain reputation records as reinforcement learning state inputs and optimizing long-term rewards through policy gradient [16]. This

scheme achieved a malicious node identification accuracy of 95.8% on the CloudSim platform, but the training process required a large amount of historical data support. Reshi and Sholla (2025) designed an IBF network combining IoT, blockchain, and fog computing, conducted a comparative study on the trade-offs between energy consumption and throughput for PoW, PoS, and PoA consensus mechanisms, and proposed an adaptive strategy for dynamically switching consensus protocols based on network load [17]. Experiments showed that the dynamic consensus mechanism could reduce energy consumption by 34%, but the network delay fluctuations during protocol switching remained unresolved.

(5) Integrated framework and emerging directions. Burhan et al. (2023) comprehensively reviewed the collaboration mechanisms of IoT applications under the fog computing paradigm, systematically analyzing 142 studies from three dimensions: hierarchical architecture, security issues, and solutions [18]. They pointed out that cross-layer optimization and lightweight cryptography are key future directions. Liu et al. (2023) systematically investigated blockchain-based IoT trust management schemes, classified and compared design choices for consensus mechanisms, reputation models, and storage structures, and noted that off-chain computing and sharding technology are key approaches to addressing scalability issues [19]. Wardana et al. (2024) proposed a lightweight trust management and privacy protection collaborative intrusion detection system (LTP-CIDS), which adopts a collaborative learning mechanism to distribute training and detection models across multiple fog nodes and protects training data through differential privacy [20]. LTP-CIDS achieved a detection rate of 92.3% on the UNSW-NB15 dataset, with a communication overhead reduction of 62% compared to centralized solutions. Agrawal et al. (2024) designed a secure data access control model integrating blockchain and fog computing, protecting distributed storage data with a hybrid encryption algorithm combining AES and RSA, and using smart contracts to automate the execution of access policies [21]. This model reduced data retrieval latency by 41% compared to pure cloud solutions, but the key management mechanism still relies on some centralized components. Murad and Rahimi (2024) innovatively proposed a hierarchical P2P fog architecture (HP2P-Fog), which organizes ordinary fog nodes into autonomous clusters through super nodes and adopts a threshold-based signature scheme to achieve rapid consensus within clusters [22]. HP2P-Fog exhibits good resilience in scenarios with high node joining and leaving frequencies, but its super node election mechanism may have fairness issues.

Table 1 systematically summarizes the core models, advantages, and limitations of relevant research.

In summary, existing research has made significant progress in the field of fog computing trust management, but there are still three main deficiencies: Firstly, most trust evaluation models have a single dimension and lack deep integration of node historical behavior, real-time performance, and economic incentives, making it difficult to effectively respond to On-Off attacks; secondly, there is a fundamental contradiction between the resource overhead of blockchain consensus mechanisms (such as DPoS, PBFT) and the computational constraints of fog nodes, and the static election mechanism cannot adapt to dynamic network changes; finally, the automation level of resource allocation and trust supervision driven by smart contracts is insufficient, limiting system efficiency and security. In summary, existing research exhibits notable deficiencies in the dimensions of trust evaluation and the lightweighting of consensus mechanisms. The integrated blockchain framework proposed in this paper aims to address these limitations through a dynamic reputation model and PoIA consensus. This paper proposes a three-tier service framework integrating blockchain technology, including end–fog–cloud, and an innovative PoIA consensus mechanism. Through a dynamic reputation evaluation algorithm to quantify node credibility, combined with a deposit economic model to increase the cost of attacks, efficient resource allocation and credible governance are ultimately achieved.

Based on the aforementioned analysis, this article will proceed to elaborate on the technical framework of integrated blockchain to address the real-time issue of trust evaluation.

3 Technical Framework

The end–fog–cloud three-layer service framework integrating blockchain technology is adopted to solve the problem of untrustworthy nodes in the integrated network of blockchain and fog computing, as well as the low adaptability between blockchain technology and fog computing network.

3.1 Fog Service Framework

Figure 1 shows the end–fog–cloud three-layer service framework integrating blockchain technology, which consists of user equipment layer, fog service layer and cloud layer from bottom to top.

Table 1 Summary of existing research

Research Model	The Obtained Results	Research Limitations
Real-time trust-aware scheduling	The efficiency of task allocation is increased by 18%	Ignoring node dynamics and lacking economic incentives
Keyword trust management system	Text matching accuracy is 91.7%	High computational cost and difficulty in deploying resource-constrained nodes
Improving trust management in genetic algorithms	The convergence speed is 25% faster than that of traditional methods	High training time cost and insufficient real-time performance
Resource allocation in complex systems	Resource utilization efficiency is 86%	Without integrated trust evaluation, it is vulnerable to resource exhaustion attacks
Trustworthy resource allocation auction	Effectively suppress false bids	The centralized architecture suffers from a single point of failure
Blockchain genetic algorithm scheduling	The makespan metric is improved by 31%	The storage overhead of blockchain has not been optimized
6G zero-trust mechanism	The encryption delay is increased by only 7 ms	The control signaling overhead is relatively high
Privacy-enhanced PET-Fog	Data leakage risk reduced by 74%	The encryption operation is arduous and difficult to meet real-time requirements
Secure computing SCF-IIoT	System availability is 99.91%	The consensus node election mechanism is static and difficult to adapt to topology changes
A2C reliable scheduler	The accuracy rate of malicious node identification is 95.8%	Rely on a large amount of historical data for training
Dynamic consensus IBF network	Energy consumption is reduced by 34%	The fluctuation in protocol switching delay remains unresolved
Collaborative intrusion detection system LTP-CIDS	Detection rate of 92.3%, with communication overhead reduced by 62%	Detection rate is 92.3%, and communication overhead is reduced by 62%
Mixed encryption access control	Data retrieval latency reduced by 41%	Key management remains partially centralized
Hierarchical P2P fog architecture	Good elasticity in high dynamic scenes	The fairness of the super node election mechanism is questionable

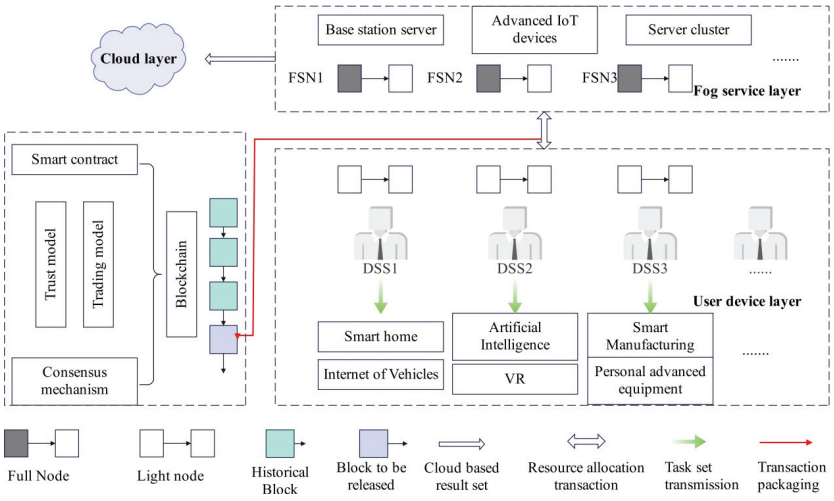


Figure 1 End–fog–cloud three-layer service framework based on blockchain technology.

As shown in Figure 1, the three-tier architecture of terminal–fog–cloud achieves decentralized coordination through blockchain, ensuring the hierarchical and trustworthy nature of data processing.

The user equipment layer is mainly composed of user intelligent devices and data service subscriber (DSS). Smart devices are responsible for generating raw data and raw tasks to be processed. DSSs, acting as subscribers to data services and modeled as user nodes in the blockchain network, are responsible for managing raw data and generating task sets $\phi_i = \{D_i, c_i\}$. Among them, D_i represents the size of the task data and c_i represents the number of CPU cycles required to complete the task. Due to the massive number of smart devices and the easily accessible nature of fog computing networks, the trustworthiness of DSSs cannot be guaranteed. Therefore, evaluating their reputation is the focus of this paper.

The fog service layer is mainly composed of various fog servers, including resource-rich operator base stations and advanced IoT devices with surplus resources. The cloud layer is composed of the cloud computing center server, which is mainly responsible for the cloud storage of large tasks and task processing results that the computing fog service layer cannot handle. Blockchain technology plays the role of a trusted decentralized platform in this framework and is the core component of the end–fog–cloud three-tier service framework. The task offloading and resource allocation process between the user device and the fog server is modeled as a transaction process

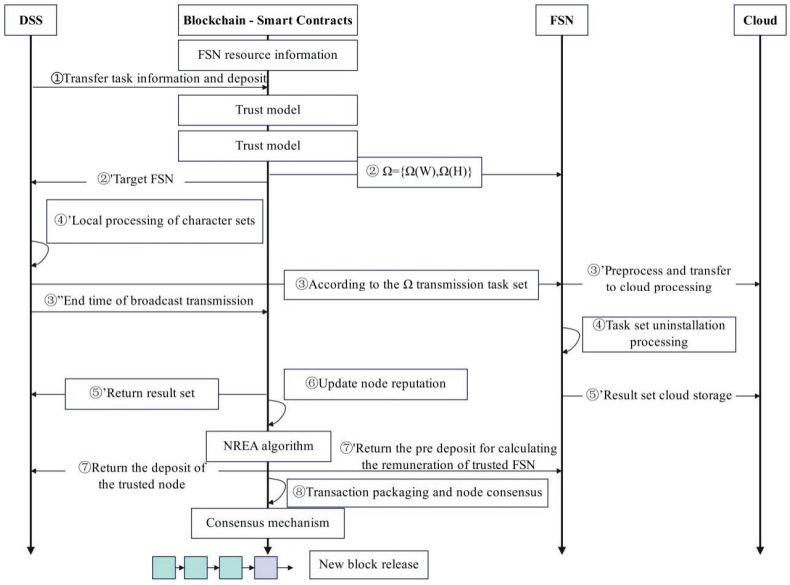


Figure 2 The overall transaction process of the system.

on the blockchain, and the user and the fog server are modeled as nodes in the blockchain.

With the support of blockchain technology, the complete transaction process of DSS from the issuance of task processing requests to the on-chain addition of new blocks is illustrated in Figure 2. Figure 2 illustrates the complete transaction process from task request to blockchain on-chain, where smart contracts drive resource allocation and ensure the credibility of node behavior.

Step ① indicates that the DSS initiates a task processing request, sends the task information $\phi_i = \{D_i, c_i\}$ to the smart contract address, and needs to prepay a certain number of tokens as a deposit based on the trust model. The smart contract deployed in the blockchain collects resource information from fog service nodes in real time and initiates a resource allocation algorithm to generate an allocation strategy based on the current task processing request and idle resource status in each cycle round.

Step ② indicates that the overall allocation strategy generated by the resource allocation algorithm deployed in the smart contract Ω will inform the FSN of the overall allocation strategy, DSS node information, and task information for pre-preparation. In this process, the FSN needs to pay a certain proportion of tokens as a deposit based on the trust model. Step

② also indicates that the DSS is informed of the FSN's node information and transmission channel information. At the same time, the blockchain will record the participating DSS and FSN based on the allocation strategy and will supervise and evaluate the performance of both nodes in the transaction. Any malicious behavior will affect the formulation of subsequent resource allocation strategies.

Step ③ indicates that the DSS transmits the task set to the designated FSN for task computation according to the allocation strategy, which incurs transmission delay and energy consumption. The transmitted task set contains a digital signature encrypted with the DSS private key. With this signature, the FSN can verify the legitimacy of its identity based on the DSS public key after receiving the task set. ③′ indicates that the fog server will process the task according to the allocation strategy or forward it to the cloud for processing. ③″ indicates that after the DSS task transmission is completed, it will broadcast the completion of transmission on the blockchain. At this time, a calculation commission will be paid to the smart contract, and the blockchain will record the time of task transmission completion $t_{i,end}^{k,trans}$.

Step ④ represents the process of task processing at the fog service nodes. During this process, benign fog service nodes will do their utmost to accelerate the progress of task processing, striving to complete the user's task computation and processing in the shortest possible time. Conversely, some fog service nodes may exhibit a chronic slowness in task processing after receiving a task set or even engage in malicious behaviors such as malicious FSNs stealing task data information and not processing task data as required. Step ④′ indicates that, according to the allocation strategy, the computational tasks of the DSS will no longer be unloaded for processing, but will instead be processed locally at the user's end, in order to maximize system benefits.

Step ⑤ indicates that after the task processing is completed, the FSN will broadcast the completion of task processing in the blockchain network, and the smart contract will record the broadcast time of task processing completion $t_{i,end}^{k,comp}$. Step ⑤′ indicates that the FSN will return the task processing result to DSS. In this process, the FSN will use the DSS's public key to encrypt sensitive data, and only the DSS private key can correctly interpret the transmitted content. Additionally, the FSN will append a signature generated by its private key to the result set and provide it to the DSS to verify the FSN's identity. Step ⑤″ indicates that the task set will be sent to the cloud for long-term storage or further data analysis according to the user's preference.

Step ⑥ will be executed after the FSN broadcast calculation is completed. During this process, the node reputation evaluation algorithm will be invoked

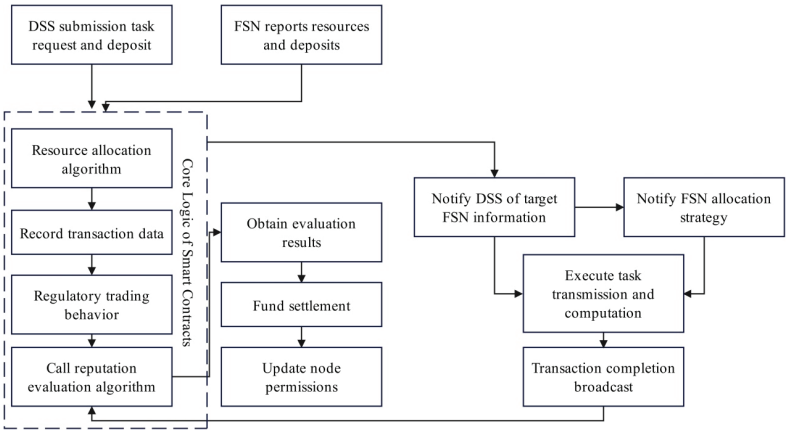


Figure 3 Smart contract-driven resource allocation and supervision process.

to assess the performance of DSS and FSN in the resource allocation transaction process, and the node reputation will be updated based on the reputation evaluation algorithm.

Step ⑦ indicates that this transaction record will be packaged into a block and undergo consensus among nodes based on the consensus mechanism. After verifying the legitimacy of the transactions in the block, the consensus nodes will publish the block to the blockchain. At this point, this transaction record will be permanently stored in the blockchain, along with updates on the reputation of the nodes.

Step ⑧: For benign DSSs, the deposit will be refunded; conversely, for malicious DSSs, the deposit will be deducted and their participation in subsequent online transactions will be restricted. For benign FSNs, the deposit will be refunded and the calculated salary will be paid; for malicious FSNs, the deposit will be deducted and their participation in subsequent online transactions will be restricted.

The resource allocation and supervision process driven by smart contracts is illustrated in Figure 3.

Figure 3 clearly illustrates the working mechanism of smart contracts as decentralized coordination centers: Firstly, they receive task requests from DSS and resource reports from FSN, and generate resource allocation strategies through built-in algorithms. Subsequently, they coordinate the transmission and processing of tasks and monitor node behavior throughout the process. After the transaction is completed, the reputation evaluation algorithm will be automatically triggered to execute fund clearing and node

permission management based on the evaluation results. The entire process is automated through smart contracts to ensure efficient resource allocation and reliable transaction supervision, providing a trusted decentralized governance framework for fog computing environments.

3.2 Implementation of Core Technology of a Fog Service Framework

3.2.1 Node reputation model

Fog computing allows advanced IoT devices with surplus resources to share their surplus resources, greatly enriching the computing and storage resources available to fog service networks, and can serve user nodes more closely than traditional base stations and cloud computing centers. This means that in the system model of this paper, nodes can play the role of fog service nodes and service subscribers in different service cycles. This reflects the flexibility of the end–fog–cloud three-tier service architecture based on blockchain technology.

The blockchain node k has remaining resources, reports the resource information that can provide services to the smart contract, and pays a deposit in advance. At this time, the node k can act as a fog server. According to the trust model, its credibility V^k is divided into four grades, as shown in formula (1)

$$\begin{cases} Untrusted\ node \begin{cases} V_k \in (0, V_{\min}),\ FSN^k\ is\ a\ malicious\ node \\ V_k \in (V_{\min}, 0.6),\ FSN^k\ is\ a\ negligent\ node \end{cases} \\ Trusted\ Node \begin{cases} V_k \in (0.6, 0.8),\ Benign\ nodes,\ trustworthy, \\ low\ efficiency \\ V_k \in (0.8, 1),\ High\ quality\ nodes,\ trustworthy, \\ and\ efficient \end{cases} \end{cases} \tag{1}$$

Among them, the reputation value range of the defined node is [0, 1], the lowest reputation threshold value allowed to participate in network transactions is $V_{\min}$, and the value of the lowest threshold value will be modulated according to different application scenarios. When the reputation value of the node is near $V_{\min}$, the node is at the edge of being defined as a malicious node, and if the node cannot improve the processing task performance in time, it will be recognized as a neglect node. There is a big difference between the time of the neglect node returning to the computing task and

the theoretical computing time, which is considered to have strong total chronicity of the processing task. In order to ensure the interests of trusted user nodes, according to the trust model, the total slow node will be reduced to malicious node after a limited period. Both malicious nodes and total slow nodes are untrusted nodes.

When the DSS broadcasts the task transmission completion time $t_{i,end}^{k,trans}$ in the blockchain network, the FSN starts processing the task, and broadcasts in the network when the task processing is finished and the task processing result is returned, and the end time is $t_{i,end}^{k,comp}$. The transmission completion time and the task processing end time will be recorded and stored in the block body transaction list information, so it can be permanently and correctly read and used in the trust evaluation algorithm and can be used in the subsequent node reputation audit and verification process. From this, the actual processing time ΔT_i^k of the FSN for the user's offload task can be obtained, as shown in formula (2):

$$\Delta T_i^k = t_{i,end}^{k,comp} - t_{i,end}^{k,trans} \tag{2}$$

Compared to the theoretical task processing time $T_i^{k,comp}$, the actual processing time ΔT_i^k is affected by force majeure factors such as unstable operating environment and network fluctuations, and is often longer than the theoretical running time. This model uses the difference between the actual processing time and the theoretical processing time to measure the processing efficiency of the fog server.

After the fog service node returns the settlement result, its credibility iteration is shown in formula (3):

$$V^k(\tau) = \begin{cases} V^k(\tau') + \lambda^{\tau-\tau'} \times \sum_{i\in I} \omega_i^k \times \dfrac{\alpha \times M_i^k \times h_i^k}{\sum_{k\in K}\sum_{i\in I} h_i^k}, & M_i^k \geq 0 \\ V^k(\tau') + \lambda^{-(\tau-\tau')} \times \sum_{i\in I} \omega_i^k \times \dfrac{\alpha \times M_i^k \times h_i^k}{\sum_{k\in K}\sum_{i\in I} h_i^k}, & M_i^k < 0 \end{cases} \tag{3}$$

$$\alpha = \frac{\gamma}{1 + \mu \times e^{\delta \times M_i^k}} \tag{4}$$

$$M_i^k = 1 - \left| \frac{\Delta T_i^k - \phi \times \Delta T_i^{k,comp}}{T_i^{k,comp}} \right| \tag{5}$$

Among them, λ represents the periodic forgetting coefficient, satisfying $\lambda \in (0, 1)$, τ represents the number of transaction cycles in the current round,

and τ' represents the number of cycles in which node k participated in the system transaction last time. The current transaction cycle number is stored in the block after the smart contract formulates the allocation strategy for each round. Therefore, τ and τ' can be obtained from the block transaction list. Setting $\lambda^{\tau-\tau'}$ can reduce the reputation accumulation rate of nodes that have not participated in system transactions for a long time and increase the reputation reduction rate of nodes with untrustworthy behavior. ω_i^k represents the task offloading decision. When fog service node k is designated to serve node i, $\omega_i^k = 1$. When the user processes tasks locally or does not offload tasks according to the allocation strategy, $\omega_i^k = 0$. α represents the computing power discount factor, as shown in formula (4), which is responsible for regulating the FSN computing power ratio and consists of a triplet (γ, μ, δ), where γ regulates its value range, μ represents the exponential scaling factor, and δ represents the time difference scaling factor. α is set to verify the validity of resource information reported by the FSN and its willingness to handle user offload tasks. Using α, the FSN's reputation can be adjusted at varying rates based on its performance in resource allocation transactions. When the FSN completes tasks with minimal time delay, the node's reputation gradually increases. However, if the node exhibits untrustworthy behavior, its reputation rapidly decreases based on α, effectively preventing untrustworthy behavior such as On-Off attacks. M_i^k represents the difference between theoretical processing time and actual processing time, as shown in formula (5). It is the key to measuring the task processing capability of FSN and represents the service quality provided by FSN in this round of transactions. Among them, ϕ represents the maximum acceptable delay coefficient, which can be specifically formulated according to different application scenarios. h_i^k represents the computing power resources allocated to service subscriber i by fog service node k according to the allocation strategy. $\sum_{k\in K}\sum_{i\in I} h_i^k$ represents the total computing power resources participating in the system resource allocation transaction in the current round.

When a blockchain node wants to offload computing tasks in the system, it can act as a service subscriber. For DSS, its credibility is divided into two grades, as shown in formula (6):

$$\begin{cases} V_i \in (0, V_{\min}),\ DSS_i \textit{ is an untrusted node} \\ V_i \in (V_{\min}, 1),\ DSS_i \textit{ is a trusted node} \end{cases} \tag{6}$$

Similar to the FSN reputation model, the reputation of a DSS node is defined in the range [0, 1). The minimum reputation threshold for participating in network transactions is $V_{\min}$, and the value of the minimum threshold will be adjusted according to different application scenarios. When the FSN broadcast ends the task processing, the reputation of the DSS in this round will be updated based on the performance of the user node in this transaction and its proportion of the task set, as shown in formula (7):

$$V_i(\tau) = V_i(\tau') + \lambda^{\tau-\tau'} \times \sum_{k \in K} \omega_i^k \left(\gamma_1 \frac{c_i}{\sum_{i \in I} c_i} + \gamma_2 \frac{D_i}{\sum_{i \in I} D_i} \right) \quad (7)$$

Among them, similar to the fog service node, $\tau - \tau'$ represents the number of cycles since the node last participated in a system transaction. This setting can encourage user nodes to actively participate in task offloading transactions and increase the proportion of active nodes in the blockchain network. γ_1 represents the cycle discount factor, and γ_2 represents the task discount factor.

The dynamic evaluation and update process of node reputation is shown in Figure 4.

Figure 4 systematically illustrates the complete process of dynamic evaluation and updating of node reputation, reflecting a trustworthy evaluation system based on blockchain technology. The flowchart starts with the triggering of a transaction completion event, obtaining key timestamps and performance data through the blockchain, and adopting different evaluation strategies for two types of nodes: FSN and DSS. For FSN nodes, the service quality score M_i^k and computing power discount factor α are calculated using formulas (3)–(5), and the reputation change is comprehensively derived. For DSS nodes, evaluation is based on the proportion of task volume using formula (7). The updated reputation will trigger the automatic response mechanism of the smart contract, executing operations such as fund clearing and permission adjustment, and permanently writing the latest reputation to the blockchain, forming a closed-loop reputation management system.

Unlike the formula for measuring the trustworthiness of service nodes, the DSS, as the initiator of service transactions, primarily evaluates its trustworthiness by assessing the proportion of its unloading task workload in the total workload of all node task sets that apply for task unloading in the current round of system transactions. The larger the task set size of the DSS D_i and the number of CPU cycles consumed to complete tasks, the greater the resource demand in the system. This means that the resource

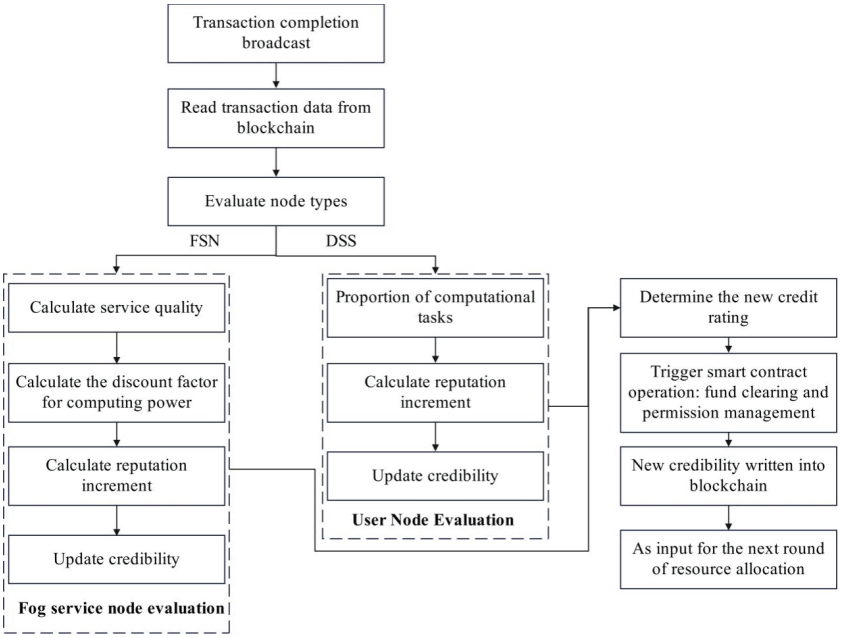


Figure 4 Dynamic evaluation and update process of node reputation.

services provided by fog service nodes in the system can be fully utilized, which can promote the generation of more transactions in the network and the emergence of more fog service nodes. This can effectively increase the number of trusted active nodes in the blockchain network, making the cost of illegal attacks by malicious nodes higher, and better ensuring the interests of secure and trusted nodes in the system.

In order to increase the cost of nodes initiating large-scale attacks, this paper proposes the concepts of deposit d_i and bond d^k. When the DSS issues a computation request to the blockchain network, it is required to store a certain number of tokens in the smart contract as a deposit. The amount of the deposit is proportional to the amount of computation required for the user's task and is related to the reputation of the node. FSNs participating in the computation need to submit a certain number of tokens to the smart contract as a computation bond when announcing their available resource information to the network. The amount of the bond varies according to the different reputation levels of the nodes. When a node initiates a malicious attack, its reputation is severely affected, and the pre-stored tokens will be deducted. The vast majority of the deducted tokens will be returned to the

victim of the transaction, thereby maximizing the protection of the legitimate nodes' rights and interests.

3.2.2 PoIA consensus mechanism

Taking the traditional DPoS consensus mechanism as the breakthrough point, this paper analyzes the incompatibility characteristics exposed by directly applying it to fog service network scenarios, optimizes its performance, and designs a trusted, efficient and lightweight consensus mechanism.

According to the reputation iteration formulas of the FSN and DSS in the trust model, only when the node performs well in the resource allocation transaction and actively participates in the system transaction, will its reputation rise steadily, but its reputation will decline rapidly when there is arbitrary untrustworthy and slow behavior. Therefore, the reputation of a node can measure the importance of the node in the network, and it also reflects that the node has good task processing performance or stable task publishing ability in the network. According to the reputation ranking of nodes, a group of full nodes is selected to form a fog verification node group, in which the nodes, as fog verification nodes (FVNs), will take turns to act as block management nodes (BMs) within a limited time, and actually master the consensus process of blocks. This feature is similar to the DPoS consensus algorithm [86]. However, considering the large number of nodes in the fog computing network scenario and the distrust between devices, the PoIA consensus mechanism omits the step of voting for proxy nodes, but adopts a fog verification node group that far exceeds the number of proxy nodes set by the DPoS consensus mechanism to achieve effective block consensus. A FVN with a standard reputation randomly obtains accounting rights, and the probability definition of the FVN becoming a BM is shown in formula (8):

$$p_{PoIA}(BM) = p_{DPos}(BM) + \frac{\alpha_2}{1 + \alpha_1 \cdot e^{-(V_{FVN} - \bar{V}_{FVN})}} \tag{8}$$

$$p_{DPos}(BM) = \frac{1}{FVNs \cdot length()} \tag{9}$$

$$V_{FVN} \geq V_{Thr} \tag{10}$$

$$\bar{V}_{FVN} = \frac{\sum_{s \in S} V_{FVN}}{S} \tag{11}$$

Among them, $p_{DPos}(BM)$ represents the probability that the delegated representative node in the DPoS consensus mechanism obtains the accounting right. The size of the probability depends on the number of

representatives, $FVNs \cdot length()$, elected by the voting mechanism, which is expressed as formula (9). V_{FVN} represents the node reputation of the fog verification node, which can be obtained from the block. The FVN reputation must meet the constraint of the minimum reputation threshold, as shown in formula (10), where is the minimum reputation threshold for the blockchain node to become a fog verification node. $\bar{V}_{FVN}$ represents the average credibility of the FVN in the fog verification node group, which can be calculated by formula (11), where *S* represents the number of fog verification node groups. α_1 and α_2 represent the probability scaling coefficient and probability shift coefficient, respectively, which are used to adjust the change of the probability curve with the change of credibility.

To achieve lightweight block transaction verification, lightweight block verification is proposed. Firstly, the block manager (BM) collects transaction data within its time slot and packages it into a block to be verified (BLO), which is then broadcasted to all fog verification nodes (FVNs) in the blockchain network for block verification. It is worth noting that, unlike traditional consensus algorithms, the proof-of-interest-algorithm (PoIA) consensus algorithm does not require FVNs to verify all transactions in the BLO. Instead, it only requires one interactive verification with another verification node during the consensus process. Specifically, the BM divides the transaction transactions in the BLO into S transaction groups Tr_s and sends those containing a small number of transactions to the FVN for transaction verification. The FVN will first verify the validity Tr_s locally and then send it to any other FVN in the fog verification node group for dual transaction verification. Under the dual transaction verification mechanism, if the positive feedback ratio of the FVN is greater than 51%, the block verification is considered to be passed. The BM then treats the block as a verified BLO and publishes it to the blockchain network. Otherwise, the BLO will be discarded, the BM's right as an FVN will be deprived, and it will be regarded as a malicious node, restricting its participation in subsequent resource allocation transactions.

Since most applications in fog computing networks are latency-sensitive tasks, and communication and computing resources in the network are extremely valuable, it is necessary to analyze the verification latency incurred by block verification. Firstly, sending transaction groups from the block manager (BM) to the fog virtualization node (FVN) will introduce a certain time delay, which is related to Tr_s the size of the transaction data D_s, and the transmission rate between the BM and the FVN r_{BM-s}. Secondly, local verification at the FVN will also incur a certain time consumption, primarily

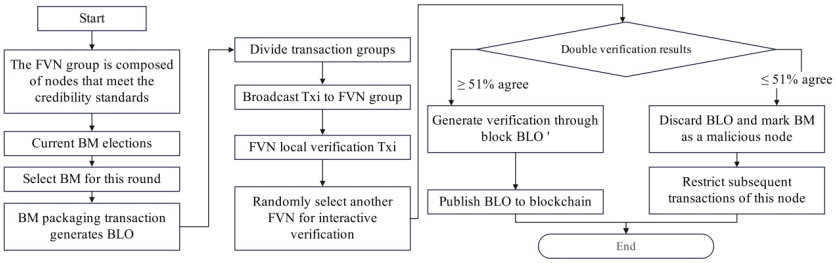


Figure 5 Workflow chart of the PoIA consensus mechanism.

involving the number of CPU cycles required for validity verification c_s and the computational resources utilized by the FVN h_s. Then, the process will move to the FVN interactive verification stage, which mainly involves Tr_s, two influencing factors: the size of the transaction data D_s and the network size function L, as well as the comparison of the results from dual verification. Finally, there is the time delay caused by the FVN returning the verification result to the BM, which is primarily influenced by the size of the verification result and the transmission rate between the FVN and the BM, D_s^{res}. Therefore FVN_s, the total latency incurred by block verification, is as shown in formula (12), T_s^{ver}:

$$T_s^{ver}(PoLA) = \frac{D_s}{r_{BM-s}} + \frac{c_s}{h_s} + D_s|L^2| + \frac{D_s^{res}}{r_{s-BM}}, \quad s \in S \tag{12}$$

To facilitate the analysis and comparison of time delays in solutions based on the traditional DPoS consensus mechanism, each validator is required to verify all transactions in the block BLO. Therefore, the time delay caused by block verification under the DPoS mechanism is as shown in formula (13), FVN_s:

$$T_s^{ver}(DPos) = \frac{D_B}{r_{BM-s}} + \frac{c_{s-B}}{h_s} + D_B|L^S| + \frac{B_s^{res}}{r_{s-BM}} \tag{13}$$

where D_B represents the data size of all transactions in the entire block, c_{s-B} represents the number of CPU cycles required to verify the entire block of transactions, $|L^S|$ represents that all verification nodes need to participate in repeated block verification, B_s^{res} and represents the verification result of the entire block of transactions.

Figure 5 fully presents the workflow of the PoIA consensus mechanism, showcasing its innovative consensus scheme designed for the fog computing

environment. This mechanism begins with the selection of FVN groups based on reputation, randomly selecting block management nodes (BMs) through the probability model of formula (8), breaking through the voting mechanism limitations of traditional DPoS. The core innovation is reflected in the lightweight verification stage: BMs divide transactions into multiple transaction groups and distribute them to different FVNs for parallel verification. Each FVN ensures transaction validity through a dual confirmation mechanism of local verification and random interactive verification, ultimately reaching consensus using the 51% majority rule. This design significantly reduces computational overhead through sharding verification, coupled with a strict punishment mechanism, significantly improving consensus efficiency while ensuring security, perfectly adapting to the resource constraints characteristic of fog computing.

4 Test

4.1 Methods

The purpose of this experiment is to comprehensively verify the performance and practical effect of the proposed fog node trust evaluation model that integrates blockchain technology and consensus mechanism in the actual fog computing environment. Moreover, this paper focuses on evaluating its node reputation assessment accuracy, system throughput, anti-attack capability, resource allocation efficiency and interpretability to ensure that the model can effectively solve the problems of untrusted and low adaptability of nodes in fog computing networks, and provide reliable technical support for decentralized fog services.

The test uses the publicly available fog computing datasets FogBus2 and Google Cluster datasets from worldwide. Among them, FogBus2 covers device resources, task loads and communication records in real fog computing environments in many countries, and Google Cluster provides large-scale cloud computing and task scheduling logs for edge nodes. In the preprocessing stage, data is cleaned first, missing values and abnormal records are eliminated, and continuous features such as task data size, CPU cycle number, and transmission delay are standardized by Z-score. At the same time, categorical variables such as node type and reputation rating are coded uniquely. Finally, the task-resource matching relationship is aggregated according to the spatiotemporal locality to simulate a multi-cycle task offloading scenario in a dynamic fog environment to ensure that the data has both global representativeness and scene diversity.

The test subjects consist of 200 fog nodes and 500 user nodes. The test group uses the trust evaluation model and PoIA consensus mechanism proposed in this paper, while the control group uses the current mainstream baseline models (including DPoS, PBFT and trust management models based on machine learning). The grouping is stratified by node reputation rating and resource capability to ensure comparability. The test covers a performance test (comparing task processing delay, throughput and consensus efficiency), a robustness test (injecting 10–30% malicious nodes to test system stability), a practicality test (evaluating resource utilization and energy consumption), an ablation test (gradually removing components such as reputation model and deposit mechanism to analyze the contribution) and an interpretability test (analyzing the impact of node behavior on reputation score through the SHAP value). In addition, scalability tests (node size increased from 100 to 1000 testing system scalability) and cost-effectiveness tests (comparing attack costs and benefits under the token economy model) were supplemented; all tests were averaged 10 times to ensure statistical significance.

4.2 Test Results

The performance test is conducted in a load environment with a scale of 500 nodes (300 FSN and 200 DSS) and 20% malicious nodes injected. By simulating a multi-cycle task offloading scenario, the performance indicators of the model in this paper and the baseline (DPoS, PBFT and machine learning model) in terms of average task processing latency, system throughput (TPS) and consensus latency are measured. All data are automatically recorded through smart contracts and repeated 10 times on average. The performance test results are shown in Table 2.

The robustness test tests the system availability (successful task completion rate) of the model and baseline under different attack intensities by dynamically injecting 10–30% malicious nodes (simulating false task processing or data theft behavior). The test period is 100 rounds, and the

Table 2 Performance test results

Index	The Model of This Paper	DPoS	PBFT	Machine Learning Model
Average delay (ms)	125	198	312	156
Throughput (TPS)	985	720	580	890
Consensus delay (s)	0.45	0.68	1.2	0.75

Table 3 Robustness test results

Proportion of Malicious Nodes	Model Availability in This Article (%)	DPoS Availability (%)	PBFT Availability (%)
10%	98.5	92.3	95.1
20%	96.8	85.4	88.7

Table 4 Practicability test results

Index	The Model of This Paper	DPoS	PBFT
CPU utilization (%)	88.5	75.2	80.1
Energy consumption (kWh)	12.3	16.8	18.5
Token efficiency (task/token)	8.9	6.2	5.8

Table 5 Ablation test results

Component Removal	Proportion of Throughput Decline (%)
Reputation model	22.3
Deposit mechanism	18.7
PoIA consensus	25.6
Smart contract regulation	20.1

behavior of malicious nodes is monitored and recorded in real time through smart contracts. The robustness test results are shown in Table 3.

The practical test measures the CPU utilization rate, total energy consumption (kWh) and token economic efficiency (the number of tasks completed per unit token) of the model and baseline (DPoS, PBFT) in 100 rounds of task processing cycles, in which the resource utilization rate is calculated in real time through the smart contract resource allocation algorithm, and the energy consumption data is aggregated based on the power consumption model of simulated equipment. The practicality test results are shown in Table 4.

The ablation test gradually removes the core components of the model in this paper (reputation model, deposit mechanism, PoIA consensus, smart contract supervision) and tests the impact of the missing components on system throughput in the same 500-node environment. The baseline is the performance of the complete model. The removal operation is implemented by modifying the smart contract logic. The results of the ablation test are shown in Table 5.

The interpretability test uses the SHAP value analysis framework to extract node behavior characteristics (such as task processing timeliness, resource contribution stability, etc.) from the blockchain transaction log and

Table 6 Interpretability test results

Feature	SHAP Value Weight
Task processing timeliness	0.32
Stability of resource contribution	0.28
Historical reputation trends	0.19
Frequency of participation in transactions	0.12
Token staking ratio	0.09

Table 7 Scalability test results

Node Scale	Model in This Paper (TPS)	DPoS (TPS)
100	1000	850
500	985	720
1000	952	630

Table 8 Cost-effectiveness test results

Models	Attack Cost (Tokens)
The model in this paper	9500
DPoS	4200
PBFT	3800

quantifies the contribution weight of each feature to the reputation score. The test is based on 1000 historical transaction records for model-independent global interpretation. The interpretability test results are shown in Table 6.

The scalability test tests the system throughput changes between the model in this paper and the DPoS baseline under different number of nodes by gradually increasing the network scale (from 100 nodes to 1000 nodes). During the test, the proportion of node types (FSN accounts for 60%, DSS accounts for 40%) and the proportion of malicious nodes (20%) remain unchanged. The scalability test results are shown in Table 7.

The cost-effectiveness test measures the token cost required by the model and baseline (DPoS, PBFT) in this paper by simulating attack behavior (such as malicious node collaboration to reduce system availability by 50%). The attack cost calculation includes the economic loss caused by deposit, deposit deduction and credit penalty. The results of the cost-effectiveness test are shown in Table 8.

In order to further verify the effectiveness of the model in this paper in a real fog computing environment, this paper designs a dynamic environment adaptability test to simulate the dynamic joining and leaving the network of nodes, and tests the stability, reputation evaluation consistency and task

Table 9 Results of dynamic environment adaptability test

Index	The Model in This Paper	DPoS	PBFT
System availability (%)	95.2	88.7	85.3
Reputation evaluation error (MSE)	0.032	0.078	0.095
Number of consensus breaks	2	7	10

processing continuity of the system under node change. The test setup is as follows: based on the initial 500 node scale (300 FSN and 200 DSS), a node dynamic change model is introduced in which 10% of nodes leave the network randomly in each cycle (simulating equipment failure or resource exhaustion), and 10% of new nodes join (simulating new equipment registration), and the initial reputation of new nodes is set to medium level (0.6). The test is run for 100 cycles to compare the performance of the model in this paper with DPoS and PBFT baselines in terms of system availability (successful task completion rate), reputation evaluation error (mean square error of actual and predicted reputation) and consensus stability (number of consensus interruptions). The test data are shown in Table 9.

4.3 Analysis and Discussion

In Table 2, the model proposed in this paper significantly outperforms the baseline in terms of average latency (125 ms), throughput (985TPS), and consensus latency (0.45 s). This is mainly due to the lightweight shard verification feature of the PoIA consensus mechanism (as shown in the shard verification process in Figure 5) and the automated resource allocation driven by smart contracts (as shown in the regulatory process in Figure 3), which effectively reduces verification overhead and task scheduling latency. However, DPoS has performance bottlenecks due to redundant voting mechanisms and PBFT has multiple interactions between nodes. Although the machine learning model is superior to traditional consensus, it still lacks real-time adaptability empowered by blockchain.

In Table 3, when the proportion of malicious nodes increased to 30% our model still maintained an availability of 93.2%, which is much higher than DPoS (76.1%) and PBFT (80.5%). This is due to the fast malicious behavior detection and isolation mechanism of the dynamic reputation model (formulas (3)–(5)), combined with economic penalties for deposits (see Section 3.2.1), effectively curbing On-Off attacks, while the baseline model is more vulnerable to attacks due to fixed consensus nodes or lack of elastic reputation management.

In Table 4, the model presented in this paper leads in CPU utilization (88.5%), energy consumption (12.3kWh), and token efficiency (8.9 tasks/token), indicating that its precise resource matching and token economic incentives achieved through smart contracts (Figure 3) optimize resource utilization and reduce redundant calculations. However, DPoS and PBFT suffer from low efficiency due to consensus node resource competition and high communication overhead.

In Table 5, removing PoIA consensus resulted in a 25.6% decrease in throughput, with the greatest impact, verifying the core contribution of lightweight shard validation (Figure 5). The removal of the reputation model and deposit mechanism resulted in a performance decline of 22.3% and 18.7%, respectively, indicating that node behavior regulation and economic constraints are crucial for system stability. The removal of smart contract regulation led to a 20.1% decrease, highlighting the role of decentralized scheduling in reducing human intervention.

In Table 6, the SHAP values indicate that task processing timeliness (0.32) and resource contribution stability (0.28) are key factors for reputation, consistent with the design goals of the reputation model in this paper (formulas (3) and (5)), enhancing model transparency and user trust. The weights of historical reputation trend (0.19) and participation frequency (0.12) indicate the importance of long-term behavioral records, while the token staking ratio (0.09) verifies the auxiliary influence of economic mechanisms on node credibility.

In Table 7, when the node size increased from 100 to 1000, the throughput of our model only decreased from 1000TPS to 952TPS, with a much lower attenuation rate than DPoS (from 850TPS to 630TPS). This is due to PoIA's shard validation mechanism (formula (12)), which only linearly increases the cost with node growth, while DPoS's global validation leads to an exponential increase in complexity, proving that our model is more suitable for large-scale fog networks.

In Table 8, the attack cost of our model (9500 tokens) far exceeds that of DPoS (4200) and PBFT (3800), due to the combined effect of high deposit and deposit mechanisms, as well as reputation penalties, significantly increasing the economic threshold for malicious behavior. The efficiency advantage of tokens further ensures the returns of benign nodes and forms a positive economic cycle.

In Table 9, our model outperforms the baseline significantly in dynamic environments, mainly due to the lightweight verification features of the smart contract driven resource allocation and supervision process and PoIA

consensus mechanism, which can quickly adapt to node changes and maintain consistency in reputation evaluation. Dynamic node management effectively reduces performance fluctuations caused by node churn through real-time updates of reputation models and token economy incentives, while baseline models are more prone to interruptions and errors due to fixed consensus nodes or complex verification processes. This experiment validates the robustness and practicality of the model in real-world dynamic fog environments, providing support for subsequent deployment.

This article's model achieves advantages in performance, robustness, practicality, and scalability by integrating dynamic reputation evaluation, PoIA lightweight consensus, and smart contract driven resource supervision. The core reason is that the reputation model reflects node behavior in real time, PoIA shard verification reduces computational overhead, and the economic mechanism increases attack costs. The limitation lies in the reliance on global reputation synchronization, which may introduce communication delays, and the token economy relies on initial token allocation. Subsequent research directions include exploring cross chain reputation sharing to reduce synchronization overhead, introducing lightweight encryption to protect privacy, and optimizing consensus parameters to adapt to the dynamic nature of heterogeneous devices.

5 Conclusion

By integrating blockchain technology and the PoIA consensus mechanism, this paper achieves real-time trust evaluation of fog nodes and efficient resource allocation, providing a reliable solution for large-scale fog computing deployment. By building a terminal–fog–cloud three-layer service framework and PoIA consensus mechanism that integrates blockchain technology, this paper successfully achieves efficient trust evaluation and resource optimal allocation of fog nodes. The test results show that the average task latency is reduced to 125 ms, the system throughput is increased to 985TPS, and the system availability of 93.2% is still maintained under 30% malicious node attacks. At the same time, the energy consumption of the model is reduced by about 26.8%, which is significantly better than traditional DPoS and PBFT model. These achievements are mainly due to the accurate quantification of node behavior by the dynamic reputation model, the computational overhead optimization of the lightweight sharding verification mechanism, and the automated supervision process driven by smart contracts. However, this study still relies on the global reputation synchronization mechanism,

which may introduce communication delays, and the token economic model is sensitive to initial allocation. Future work will explore cross-chain reputation sharing mechanisms to reduce synchronization overhead, integrate lightweight encryption techniques to enhance privacy protection, and further adapt to the dynamic nature of heterogeneous devices.

Funding

This work was supported by the by the Supported by the Key Scientific and Technological Project of the Higher Education Institutions of Henan Province in China (24A520052);the Key Scientific and Technological Project of the Higher Education Institutions of Henan Province in China (23A120004).

References

[1] Rehman, A., Awan, K. A., Ud Din, I., Almogren, A., and Alabdulkareem, M. (2023). FogTrust: Fog-integrated multi-leveled trust management mechanism for internet of things. *Technologies*, *11*(1), 27–36.

[2] Ghaleb, M., and Azzedin, F. (2023). Trust-aware fog-based iot environments: Artificial reasoning approach. *Applied Sciences*, *13*(6), 3665–3677.

[3] Alwakeel, A. M., and Alnaim, A. K. (2024). Trust management and resource optimization in edge and fog computing using the cyberguard framework. *Sensors*, *24*(13), 4308–4320.

[4] Ramamurthy, P., and Nandagopal, M. (2023). Enabling trust and security between fog nodes using blockchain technology. *Journal of Intelligent & Fuzzy Systems*, *44*(3), 4605–4612.

[5] Hameed, F. M. H., and Kurnaz, S. (2024). An effective mechanism for FOG computing assisted function based on trustworthy forwarding scheme (IOT). *Electronics*, *13*(14), 2715–2727.

[6] Kaur, A., and Auluck, N. (2023). Real-time trust aware scheduling in fog-cloud systems. *Concurrency and Computation: Practice and Experience*, *35*(10), e7680–e7693.

[7] Alwakeel, A. M. (2023). Performance analysis of a keyword-based trust management system for fog computing. *Applied Sciences*, *13*(15), 8714–8728.

[8] Bakhtiari, N. B., Rafighi, M., and Ahsan, R. (2024). A trust management system for fog computing using improved genetic algorithm. *The Journal of Supercomputing*, *80*(14), 20923–20955.

[9] Faraji, F., Javadpour, A., Sangaiah, A. K., and Zavieh, H. (2024). A solution for resource allocation through complex systems in fog computing for the internet of things. *Computing*, *106*(7), 2107–2131.

[10] Jain, V., and Kumar, B. (2023). A trusted resource allocation scheme in fog environment to satisfy high network demand. *Arabian Journal for Science and Engineering*, *48*(8), 9769–9786.

[11] Ali, N. A. M., and Ali, F. A. M. (2023). Optimizing cloud-fog-edge job scheduling using catastrophic genetic algorithm and block chain-based trust: a collaborative approach. *J Appl Eng Technol Sci*, *5*(1), 569–580.

[12] Alnaim, A. K., and Alwakeel, A. M. (2025). Zero-Trust mechanisms for securing distributed edge and fog computing in 6G networks. *Mathematics*, *13*(8), 32–45.

[13] Fotia, L., Delicato, F., and Fortino, G. (2023). Trust in edge-based internet of things architectures: state of the art and research challenges. *ACM Computing Surveys*, *55*(9), 1–34.

[14] Reshi, I. A., and Sholla, S. (2024). Securing IoT data: Fog computing, blockchain, and tailored privacy-enhancing technologies in action. *Peer-to-Peer Networking and Applications*, *17*(6), 3905–3933.

[15] Alvi, A. N., Ali, B., Saleh, M. S., Alkhathami, M., Alsadie, D., and Alghamdi, B. (2024). Secure computing for fog-enabled industrial IoT. *Sensors*, *24*(7), 2098–2111.

[16] Choppara, P., and Mangalampalli, S. S. (2024). Reliability and trust aware task scheduler for cloud-fog computing using advantage actor critic (A2C) algorithm. *IEEE Access*, *12*, 102126–102145.

[17] Reshi, I. A., and Sholla, S. (2025). IBF network: enhancing network privacy with IoT, blockchain, and fog computing on different consensus mechanisms. *Cluster Computing*, *28*(3), 208–222.

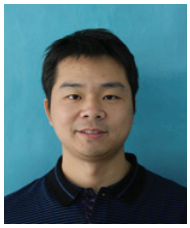
[18] Burhan, M., Alam, H., Arsalan, A., Rehman, R. A., Anwar, M., Faheem, M., and Ashraf, M. W. (2023). A comprehensive survey on the cooperation of fog computing paradigm-based IoT applications: layered architecture, real-time security issues, and solutions. *IEEE Access*, *11*, 73303–73329.

[19] Liu, Y., Wang, J., Yan, Z., and Wan, Z. (2023). A survey on blockchain-based trust management for Internet of Things. *IEEE Internet of Things Journal*, *10*(7), 5898–5922.

[20] Wardana, A. A., Kolaczek, G., and Sukarno, P. (2024). Lightweight, trust-managing, and privacy-preserving collaborative intrusion detection for internet of things. *Applied Sciences*, *14*(10), 4109–4121.
[21] Agrawal, R., Singhal, S., and Sharma, A. (2024). Blockchain and fog computing model for secure data access control mechanisms for distributed data storage and authentication using hybrid encryption algorithm. *Cluster Computing*, *27*(6), 8015–8030.
[22] Murad, S. A., and Rahimi, N. (2024). Secure and efficient hierarchical P2P fog architecture: a novel approach for IoT. *IEEE Internet of Things Journal*, *11*(10), 18796–18807.

Biographies

Guanglei Sheng was born in Henan, China, in 1982. From 2000 to 2004, he studied at Zhengzhou University and received his bachelor's degree in 2004. From 2004 to 2007, he studied at Zhengzhou University and received his master's degree in 2007. Currently, he works at Henan Finance University. He has published 17 papers, two of which have been indexed by EI. His research interests include machine learning and embedded systems.



Qingtao Wu was born in Henan, China, in 1981. From 2000 to 2004, he studied at Zhengzhou University and received his bachelor's degree in 2004. From 2005 to 2008, He studied at Chongqing University and received his master's degree in 2008. He has published 5 papers. His research interests include image processing and big data.