AI-Blockchain Integrated Trusted Interaction Mechanism for Enhanced Network Security

Xulei Gao
Qingdao Municipal Military Retired Cadres Activity Center, Qingdao, Shandong 266100, China
E-mail: gaoxulei2646@163.com

Keywords: AI integration, blockchain security, trusted interaction, network defense

Received: Feburary 3, 2026

This paper proposes an AI–blockchain integrated trusted interaction mechanism for network security. The mechanism combines a deep-learning anomaly detector with a permissioned blockchain and smart contracts to provide real-time threat analysis, decentralized authentication and tamper-resistant trust management. The AI module is a two-branch neural network that processes network-traffic features and interaction-behavior graphs and outputs an anomaly score for each event. The blockchain layer adopts a consortium architecture with PBFT-style consensus, and smart contracts implement identity verification, trust update rules and access control.Experiments are conducted on labeled network-traffic traces and an enterprise-like testbed. We compare the proposed mechanism with an AI-only baseline (deep model with centralized control) and a blockchain-only baseline (rule-based smart contracts without AI) using detection-level metrics—accuracy, precision, recall, F1-score, false positive rate (FPR) and false negative rate (FNR)—and system-level metrics—response time and throughput. Under low load, the system reaches 180 ms response time and 5000 requests/s throughput; under high load, response time is 300 ms and throughput is 4200 requests/s. Compared with the strongest baseline, throughput increases by 42.8% and FPR is reduced by 60.9%, while FNR remains low across diverse attack scenarios. These results confirm that the mechanism supports trusted interaction in dynamic network environments.

Povzetek: Članek predstavi mehanizem zaupanja za omrežno varnost, ki združi globokoučni detektor anomalij z dovoljenim blockchainom (PBFT) in pametnimi pogodbami za decentralizirano avtentikacijo, posodabljanje zaupanja ter neponarejljivo upravljanje dostopa v realnem času.

1 Introduction

With the deepening of the information society, network security has become a global concern. Traditional protection mechanisms can still block some attacks, but their limitations are increasingly exposed as attack methods continue to evolve. With the widespread adoption of the Internet of Things (IoT), cloud computing and big data, the scope and complexity of attacks are growing, and centralized protection architectures struggle to handle diverse and dynamic threats. At the same time, frequent incidents such as data breaches, identity theft and cybercrime pose serious risks to individuals, enterprises and even national security. As a result, designing more intelligent, trustworthy and adaptive network security mechanisms has become a key topic in current cybersecurity research.

Artificial intelligence (AI) has been widely applied in network security due to its strong data analysis and processing capabilities. Machine learning and deep learning models can automatically detect and identify network attacks, providing more efficient and intelligent protection. However, the use of AI in security also faces practical challenges, including data privacy, model robustness and algorithmic bias, which may undermine its effectiveness in real deployments [1]. Meanwhile, blockchain technology, with its decentralized and tamper-resistant properties, offers new tools for addressing trust and integrity problems. Blockchain can help prevent data tampering and identity forgery, but its performance bottlenecks and limited flexibility also pose challenges in complex, dynamically changing environments.

Combining AI and blockchain to form a complementary solution for network security is therefore of significant research interest. Blockchain can provide a decentralized trust infrastructure that ensures data integrity and immutability, while AI can deliver real-time threat analysis, dynamic response and adaptive protection strategies. By jointly leveraging these strengths, it is possible to enhance both the intelligence and credibility of security systems and to provide more effective protection against emerging and complex threats.

This study aims to design a trusted interaction mechanism that combines artificial intelligence and blockchain to enhance network security. By combining the intelligent decision-making capabilities of AI with the decentralized trust mechanism provided by blockchain, this paper proposes a new interaction model to address key issues in current network security, such as the credibility of identity authentication, the guarantee of data integrity and the security of cross-domain systems. The innovation of the research lies in the organic combination of artificial intelligence and blockchain technology, proposing a data exchange platform that can achieve

intelligent analysis and ensure decentralized security, thus providing new solutions for building a more efficient and trustworthy network security protection system.

The main contributions of this work can be summarized as follows:

(1) We design a unified network security framework that tightly integrates AI-based threat detection with blockchain-based decentralized trust management, enabling end-to-end trusted interactions in multi-domain environments.

(2) We propose a fine-grained trusted interaction algorithm that combines dynamic trust scoring, smart-contract-driven authentication, and timing-aware security control to support real-time, large-scale network interactions.

(3) We implement a prototype system and conduct extensive experiments on benchmark datasets and synthetic attack scenarios, demonstrating significant improvements in response time, throughput, and false alarm rate compared with existing AI-only and blockchain-only schemes.

(4) We further evaluate the proposed mechanism through hyperparameter sensitivity analysis and user perception studies, showing that the system is not only technically robust but also perceived as trustworthy and user-friendly.

To make the research design more explicit, the present study is guided by the following objectives:

(1) to design a unified trusted interaction mechanism that tightly integrates AI-based threat detection with blockchain-based decentralized trust management for multi-domain network environments;

(2) to analyze, from both detection and system perspectives, how this integration affects key security and performance indicators such as accuracy, FPR/FNR, response time and throughput under realistic attack scenarios; and

(3) to explore under what deployment and resource conditions the proposed mechanism can be practically applied in enterprise networks, industrial IoT (IIoT) and other safety-critical systems.

Based on these objectives, we formulate the following research questions:

RQ1: Compared with AI-only and blockchain-only baselines, to what extent can the proposed AI–blockchain integrated mechanism improve trusted interaction performance (detection accuracy and false alarm rates, response time and throughput) under diverse and dynamically changing attack scenarios?

RQ2: How do the core components of the mechanism—dynamic trust computation, smart-contract-based authentication and interaction-timing-aware security control—contribute to the observed performance, and can the closed-loop trust and defense dynamics be shown to be stable and bounded?

RQ3: Under which deployment settings and hardware/network constraints (e.g., scale of nodes, consensus configuration, connectivity stability) can the mechanism be feasibly deployed in enterprise, IIoT and other cyber-physical environments while maintaining acceptable decision delay and throughput?

2 Relevant work

The application of AI technology in network security has made significant progress, especially in fields such as intrusion detection, malicious behavior recognition, and threat prediction. AI can identify potential security threats from massive amounts of data through deep learning and machine learning algorithms, thereby enhancing the intelligence level of network security protection. Kshetri (2025) pointed out that AI can detect potential attack behaviors in real time by analyzing multidimensional data such as network traffic and device status, and can effectively reduce false positives and false negatives [2]. In addition, AI has demonstrated significant advantages in malware detection and abnormal behavior recognition, enabling automated analysis of new threats [3]. However, despite the enormous potential of AI in network security applications, it still faces issues such as data privacy breaches and model biases, which limit its comprehensive application in certain sensitive fields.

The application of blockchain technology in the field of network security, especially in identity authentication, data encryption, and audit tracing, has received widespread attention. The decentralized and tamper proof nature of blockchain makes it an important tool for solving traditional centralized trust problems. D'Aniello and Fotia (2025) point out that blockchain provides a decentralized mechanism for identity authentication, avoiding the risk of single point of failure, while also enhancing the ability to automatically execute security policies through smart contracts [4]. Pan et al. (2020) proposed that the combination of blockchain and AI can achieve decentralized management of information flow, effectively enhancing data security and trust management in the network[5].Although blockchain technology can provide strong data security guarantees, Saleh (2024) pointed out that its performance bottleneck, especially in high-frequency transactions and large-scale data transmission, remains a key issue limiting its widespread application.[6]

In the research of trustworthy interaction mechanisms, the combination of AI and blockchain technology has become an emerging trend. The decentralized nature of blockchain ensures the immutability of interactive data, while AI dynamically adjusts trust management strategies by learning user behavior, threat patterns, and other data, enhancing the flexibility and intelligent protection capabilities of the system. Zhang et al. (2023) proposed that the combination of AI and blockchain can improve the efficiency and accuracy of trust assessment, while also enhancing data protection during the interaction process [7]. Luo et al. (2023) proposed a model that combines AI and blockchain, which can enhance security while achieving real-time protection of large-scale data [8]. However, existing trusted interaction mechanisms based on blockchain still face problems such as poor cross domain system interoperability and insufficient real-time

performance, especially in complex network environments involving multiple parties [9]. Zhang (2022) pointed out that the performance bottleneck of blockchain in processing large-scale data limits its application in dynamic and real-time environments [10].

The existing trusted interaction mechanisms rely heavily on centralized trust management, which often poses a risk of single point of failure when facing large-scale and complex network environments. Blockchain provides higher security through decentralization [11], but performance bottlenecks still exist when processing high-frequency transactions and large-scale data. Yang et al. (2022) pointed out that although blockchain provides a decentralized solution for trust management, its performance issues in handling large-scale data still limit its widespread application [12]. AI can enhance the intelligent protection of systems, but how to efficiently manage and evaluate trust in cross domain and multi trust environments remains a challenge [13].

This study proposes a new trusted interaction mechanism model that combines AI and blockchain technology, aiming to address the shortcomings of existing mechanisms in terms of real-time performance, flexibility, and cross domain interoperability [14]. Through the dynamic learning of AI and the decentralized nature of blockchain, a solution has been proposed that can provide efficient security protection while also being flexible and adaptable [15-16].To more clearly position the proposed mechanism with respect to representative AI–blockchain and related security frameworks, Table 1 summarizes several closely related works in terms of their main architectural focus, datasets or deployment scenarios, evaluation metrics and key limitations. The last row presents a concise overview of this work for comparison.

Table 1：Summary of representative AI–blockchain security frameworks and comparison with this work

Work	Architecture / Focus	Dataset / Scenario	Main Metrics
Goundar & Gondal [1]	CNN-based anomaly detection integrated with permissioned Ethereum blockchain for immutable alert logging	CICIDS2017 intrusion detection dataset	Accuracy, Precision, Recall, alert-logging latency
Pan et al. [5]	Blockchain- and AI-empowered trust-information-centric network architecture for beyond-5G	Conceptual architecture and simulated beyond-5G scenarios	Qualitative discussion of trust and security; high-level performance analysis
Zhang et al. [7]	Cloud–edge–end IIoT security with PoRep-based blockchain consensus and device-level trust management	Industrial IoT and edge-computing testbeds	Consensus latency, storage overhead, security guarantees of PoRep
Reputation-based CTI and trust systems [10,11]	Blockchain-supported cyber threat intelligence sharing and reputation mechanisms	Enterprise or multi-organizational CTI sharing environments	Cryptographic overhead, scalability, qualitative security analysis
Trustworthy federated learning and model sharing [12–13]	Blockchain-enabled federated learning and trustworthy model aggregation	Distributed learning across multiple data owners	Convergence of training, model accuracy, communication cost
Smart-city trust and compliance frameworks [15–16]	Blockchain-based trust management and automated compliance for smart-city services	Smart-city platforms and service ecosystems	Transaction latency, scalability, qualitative trust/compliance evaluation
This work	AI–blockchain integrated trusted interaction mechanism with dynamic trust computation, smart-contract-based authentication and interaction-timing-aware security control	Labeled network-traffic traces and hybrid enterprise-like testbed	Accuracy, Precision, Recall, F1, FPR, FNR, response time, throughput, resource consumption, user-perceived trust

As summarized in Table 1, recent AI–blockchain security frameworks cover a range of designs, including CNN-based intrusion detection with blockchain alert logging, cloud–edge–end architectures for IIoT, trust-information-centric networking and reputation- or federated-learning-based trust schemes. Most of these works evaluate detection metrics (e.g., accuracy, precision, recall) or blockchain metrics (e.g., consensus latency, storage or cryptographic overhead) on specific datasets such as NSL-KDD, CICIDS2017 or IIoT and smart-city testbeds.

However, from the viewpoint of trusted interaction at the level of individual network events, these state-of-the-art approaches share several limitations. In many frameworks, blockchain mainly serves as an immutable log or audit layer, and there is no explicit modeling of interaction-level trust dynamics, timing control or closed-loop decision making. Joint analysis of AI detection performance and system-level behavior (response time and throughput) under sustained adversarial traffic is also largely missing, and few works provide a unified evaluation on both public traffic datasets and enterprise-like deployments.

These gaps justify the necessity of the mechanism proposed in this paper. In contrast to prior AI–blockchain security frameworks, our work couples dynamic trust computation, smart-contract-based authentication and interaction-timing-aware security control into a single closed-loop architecture. We explicitly compare our system with AI-only and blockchain-only baselines and evaluate it using both detection-level metrics (Accuracy, Precision, Recall, F1, FPR, FNR) and system-level

metrics (response time and throughput) on labeled traffic data and a hybrid enterprise testbed, thereby providing a quantitatively characterized, practically deployable trusted interaction framework rather than only a high-level architectural concept.

3 Design of network security trusted interaction mechanism combining artificial intelligence and blockchain

3.1 Overall system architecture and design process

The network security trusted interaction mechanism system proposed in this study, which combines artificial intelligence and blockchain, aims to provide decentralized trust guarantee and data immutability through blockchain, while utilizing artificial intelligence technology for intelligent trust evaluation and dynamic decision-making. The system consists of multiple functional modules that work together to ensure data security, trustworthy interactions, and the ability to adaptively adjust in complex and dynamic network environments. The design process of the system covers all aspects from data collection to interactive verification, including data collection, data processing and feature extraction, intelligent decision-making and trust evaluation, interactive verification and trust management, and result updates.

The system architecture consists of the following five key modules:

Data collection module: responsible for collecting real-time data from network devices, sensors and terminals. The data include network traffic, device status and user behavior. Data transmission is protected by encryption, and raw records are stored in a secure repository to ensure integrity and traceability.

Data processing and feature extraction module: performs preprocessing such as removing redundant records and filling missing values, and then extracts key features from the cleaned data. Techniques such as principal component analysis (PCA) and clustering are used to select effective features and transform them into a format suitable for model training.

Intelligent decision-making and trust evaluation module: analyzes the processed data using AI algorithms and evaluates the trust level of each node. The trustworthiness of node i is computed as:

$$T_i=\sigma(\alpha S_i+\beta A_i+\gamma B_i), \quad (1)$$

where S_i is the node's historical security behavior score, A_i is its activity level, B_i is its blockchain transaction transparency score, α,β,γ are weighting coefficients and σ(·) is a sigmoid function that maps the result to [0,1].

Interaction verification module: verifies the identities of interacting parties and ensures the authenticity of their credentials. Smart contracts enforce predefined trust rules for identity verification and authorization, and blockchain records the interaction results to guarantee integrity and non-repudiation.

Trust management and result update module: updates node trust levels after each interaction and writes the updated trust values to the blockchain ledger. The update rule:

$$T_i'=(1-\lambda)\, T_i+\lambda\, \Delta T_i, \quad (2)$$

where λ is the learning rate and ΔT_i is the trust increment from the current interaction, controls the adaptation speed of trust updates.

The overall workflow from data collection to interaction verification is as follows. First, the data collection module captures real-time data, which are transmitted securely and stored for subsequent processing. Next, the data processing and feature extraction module generates features suitable for AI-based analysis. The intelligent decision-making and trust evaluation module then computes node trust levels, while smart contracts verify the identities of both parties and record every interaction on the blockchain. Finally, the trust management module dynamically updates node trust levels based on interaction outcomes and feeds the results back to the blockchain, enabling the system to adapt to evolving network conditions (as shown in Figure 1).

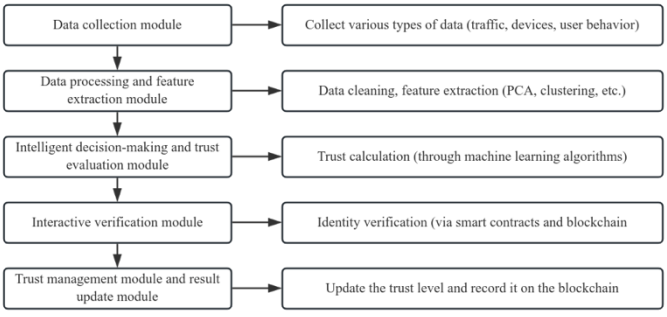

Figure 1: Overall system architecture and design process

3.2 Integration mechanism of artificial intelligence and blockchain

When designing a network security trusted interaction mechanism based on the combination of AI and blockchain, the role of AI in data analysis and security protection cannot be ignored, while blockchain provides decentralized data security and traceability functions. By organically combining AI with blockchain technology, intelligent decision-making and efficient data protection can be achieved, enhancing the security, transparency, and adaptability of the system. Artificial intelligence can extract features from large-scale network traffic, user behavior, device status, and analyze potential security threats through deep learning and machine learning algorithms. AI systems identify unknown attack behaviors through pattern recognition and anomaly detection. Assuming there is network behavior data $D = \{d_1, d_2, \ldots, d_n\}$ for a node in the system, where d_i represents the i-th data point, the AI system trains a prediction model M to analyze each data point di and predict whether it is an attack behavior. The judgment result $P(d_i)$ output by the model can be expressed by the following formula:

$$P(d_i) = f(M, d_i) \tag{3}$$

Among them, f is the trained AI model, M is the model, and d_i is the input data. Based on this formula, AI can automatically evaluate whether network traffic or behavior is abnormal and respond promptly. This mechanism enables the system to identify and defend against attacks in a dynamically changing network environment in a timely manner. Blockchain technology ensures the immutability and traceability of data through its decentralized nature. In cybersecurity, blockchain can provide an open and transparent environment, ensuring that the behavior of all parties involved can be traced. Each block Bi in the blockchain contains the hash value $H(B_{i-1})$ of the previous block, the timestamp ti, as well as the data content d_i and hash value $H(d_i)$ of the current block. The blockchain data structure is represented by the following formula:

$$H(B_i) = \text{Hash}(H(B_{i-1}) \| t_i \| d_i) \tag{4}$$

Among them, H is the hash function, and $\|$ represents the join operation. The hash value of each data block is connected to the hash value of the previous block to ensure the order and integrity of the data. If any data in the blockchain is tampered with, the hash value of subsequent blocks will change, resulting in the invalidation of the entire blockchain structure, which makes the storage of data in the blockchain tamper proof. Combining AI and blockchain technology can provide intelligent decision support and decentralized security assurance. AI technology provides powerful automation capabilities in data analysis and threat identification, while blockchain ensures the security and transparency of data storage. AI can quickly identify potential security threats in large-scale data, while the decentralized storage mechanism provided by blockchain ensures the integrity and immutability of each data.

3.3 Design of trusted interaction algorithm

This section introduces a trusted interaction algorithm based on the combination of AI and blockchain. The algorithm uses AI for intelligent security protection and combines blockchain technology for decentralized identity authentication and trust management. Each link in the system is ensured to be secure and trustworthy through the combination of these two technologies.

In the field of artificial intelligence, the main task is to dynamically analyze real-time data from the network, identify potential threats, and generate protective strategies. Assuming the network behavior dataset is $E = \{e_1, e_2, \ldots, e_n\}$, where each ei represents a data point or event. The AI model calculates its anomaly degree or risk score based on these input data, and outputs the predicted safety level of the model. To achieve this goal, representative features are first extracted from the raw data through feature extraction algorithms. Feature extraction can use algorithms based on Convolutional Neural Networks (CNN) or Recurrent Neural Networks (RNN) to uncover potential attack patterns from time-series data.

The AI model learns patterns obtained from historical data and combines them with real-time data input to provide prediction results. To make the prediction results more accurate, the model dynamically calculates the safety score by continuously adjusting its parameters and using a weighting function:

$$P(e_i) = \frac{\sum_{j=1}^{m} \alpha_j \, f_j(e_i)}{Z} \tag{5}$$

Among them, $f_j(e_i)$ is the jth feature extracted from data point e_i, αj is the weight of the corresponding feature, and Z is the normalization constant. Ultimately, the $P(e_i)$ output by the model represents the security score of the data point, reflecting whether the data point is malicious or exhibiting abnormal behavior.

In order to improve the real-time performance and protection capability of the model, the system also designed an incremental learning mechanism. During the interaction process, the model can update its parameters based on feedback to adapt to new network environments and potential attack methods. Assuming the output of the current model is $P(e_i)$, after the model is updated, the new output is $P(e_i)$ new. The update process adjusts the model weights based on the incremental data Δe_i:

$$P(e_i)^{new} = P(e_i) + \eta \cdot \Delta P(e_i) \tag{6}$$

Among them, η is the learning rate, and $\Delta P(e_i)$ represents the trust change caused by incremental data. This update mechanism enables the model to adjust in a timely manner when facing new attack patterns and optimize protection strategies.

The role of blockchain in this algorithm is mainly reflected in decentralized trust management and identity authentication. Each node generates a unique identity identifier PK_i through public key encryption technology, and performs identity registration and verification through blockchain. The decentralized nature of blockchain enables the authentication process to not rely on centralized authentication agencies, thereby avoiding the risk of single point of failure.

Before each interaction, the node performs authentication through a smart contract. Set the authentication formula as follows:

$$H(PK_i) = \text{Hash}(PK_i \| t_i) \quad (7)$$

Among them, $H(PK_i)$ represents the hash value of the node's public key, and ti is the current timestamp. This formula ensures the uniqueness and validity of authentication for each node through hash values.

In terms of trust management, blockchain evaluates the trustworthiness of nodes by recording their behavior history. Assuming that the trust level of a node is determined by three factors: its historical security behavior, activity level, and transaction transparency. The formula for calculating trustworthiness is as follows:

$$T_i = \frac{e^{\alpha S_i} \cdot e^{\beta A_i} \cdot e^{\gamma B_i}}{Z}. \quad (8)$$

Among them, S_i is the node's historical security behavior score, A_i is the node's activity level, B_i is the node's blockchain transaction transparency score, α, β, γ are weighting coefficients, and Z is the normalization constant. The level of trust Ti reflects the reputation status of the node and determines its trustworthiness in the interaction. After each interaction, the trust level will be dynamically updated based on the interaction results. The trust level after each update will be recorded and synchronized to the blockchain through smart contracts, ensuring the transparency and immutability of the data. The combination of AI and blockchain enables the system to provide intelligent protection and real-time trust evaluation while ensuring decentralization, transparency, and data immutability. AI algorithms provide dynamic threat assessments for each data point and node, while blockchain ensures transparency and reliability of identity verification and trust assessment results through a decentralized approach. The combination of real-time updates of trust and smart contracts for authentication provides efficient security for every interaction in the network. The system can automatically adjust the trust level based on threats detected in real-time by AI, and ensure the immutability of data and behavior records through smart contracts on blockchain.

3.4 Smart contracts and authentication mechanisms

In trusted interaction, the main task of smart contracts is to automatically perform identity verification, transaction verification, and trust evaluation based on set rules.In our prototype implementation, these smart contracts are deployed on a permissioned consortium blockchain. The blockchain network consists of multiple peer nodes and a small set of ordering nodes, and adopts a practical Byzantine fault tolerance (PBFT)-style consensus protocol to ensure deterministic finality and low confirmation latency. Each block has a target interval of approximately 2–3 seconds and a maximum size of 1 MB, which strikes a balance between throughput and confirmation delay. The smart contract code (chaincode) is implemented in a high-level language (Go) and exposes functions for identity registration, trust update and access control, which are invoked by client applications through standard blockchain SDKs.

In this system, smart contracts first verify the identity of the nodes participating in the interaction and ensure that they meet the trust requirements of the system. In order to more accurately determine the validity of identity, smart contracts verify nodes through hash values and signature mechanisms. The verification process uses the following formula:

$$H(PK_i) = \text{Hash}(PK_i \| t_i \| \text{Nonce}) \quad (9)$$

Among them, PK_i is the public key of the node, t_i is the timestamp, and Nonce is the random number used to prevent replay attacks. This hash value ensures the uniqueness and timeliness of the node's identity information, avoiding forgery or replay attacks.

Once the identity verification is passed, the smart contract will begin to perform trust evaluation. The trust level of a node is calculated based on multiple factors such as its historical behavior, security, and interaction activity. After each interaction between a node and other nodes, the trustworthiness will be dynamically updated. To this end, we have designed a trust update mechanism based on weighted factors, with the following formula:

$$T_i^{new} = \frac{\sum_{k=1}^{m} w_k \ f_k(S_i, A_i, B_i)}{Z}. \quad (10)$$

Among them, S_i is the historical security behavior score of node i, A_i is the activity level of node i, B_i is the transaction transparency score of node i, w_k is the weight of each factor, Z is the normalization constant, ensuring the consistency of trust. Through this formula, smart contracts can comprehensively evaluate the reputation of nodes and adjust their trustworthiness in a timely manner.

The decentralized nature of blockchain is the foundation of this authentication mechanism. The identity of nodes is authenticated through public key encryption, and the identity information and transaction records of each node are stored and verified through blockchain. During each authentication process, the system will hash the node's public key PKi and store the hash value on the blockchain. This process ensures the transparency and immutability of identity information.

During the authentication process, the node submits its public key and timestamp information, and the smart contract will verify its identity according to the following formula:

$$V(PK_i) = H(PK_i \| t_i) \quad (11)$$

Among them, $V(PK_i)$ represents the verification of node i's public key, ensuring the legitimacy and validity of node identity. After verification, the smart contract performs trust calculation and behavior control based on the node's behavior history, ensuring that only legitimate nodes that have passed verification can interact.

The trust level in decentralized authentication mechanisms is dynamically updated based on multiple factors. In addition to historical behavior ratings, factors such as interaction frequency, transparency of participation, and feedback after interaction are also included in the trust calculation. Through real-time analysis of node behavior, the trust update formula is as follows:

$$T_i^{new} = T_i + \alpha \left(\sum_{j=1}^{p} \beta_j \ f_j(S_i, A_i, B_i) \right) \quad (12)$$

Among them, p is the number of factors that affect trust, βj is the weight of the corresponding factor, $f_j(S_i, A_i, B_i)$ is the comprehensive evaluation function for node behavior, and α is the coefficient that controls the update amplitude. Through this formula, the trust level of nodes

can be dynamically adjusted as their behavior changes, and the system can quickly respond to the trust status of nodes after each interaction.

Smart contracts not only perform identity verification and trust updates during each interaction, but also automatically perform transaction verification. When the trust level of a node meets the preset conditions, the smart contract will approve the interaction behavior of the node. If the trust level is lower than the threshold θ set by the system, the smart contract will automatically reject the interaction request. The specific execution judgment formula is:

$$if\ T_i^{new} \geq \theta: Execute\ Transaction \quad (13)$$

Through this formula, smart contracts ensure that only nodes with sufficient trust can participate in interactions, avoiding interference from malicious nodes.

All interaction records will be written into the blockchain through smart contracts, ensuring the immutability and transparency of the data. The record Ri of each transaction or interaction behavior is processed through a hash function to ensure the uniqueness of each record. The recorded hash value $H(R_i)$ is as follows:

$$H(R_i) = \text{Hash}(R_i \| H(PK_1) \| H(PK_2) \| t_i) \quad (14)$$

This formula generates hash values by combining transaction record R_i, participant's public key hash PK_1, PK_2, and timestamp t_i, ensuring the immutability and chronological order of each transaction. All interaction behaviors will be openly and transparently recorded on the blockchain, ensuring the security and traceability of the entire interaction process.From a deployment perspective, only compact, security-critical information is stored on-chain, while raw traffic traces and model parameters remain off-chain. Specifically, the blockchain ledger records hashed identifiers of interaction records, aggregated trust scores, identity bindings and access decisions, whereas full packet captures, flow-level statistics and AI model weights are stored in a secure off-chain database located in the same data center. The off-chain AI inference service communicates with the blockchain through a RESTful gateway, which submits signed transactions to trigger smart contract functions whenever new trust evidence or access decisions must be recorded. This on-chain/off-chain division significantly reduces storage and computation overhead on the blockchain layer, while preserving the transparency and tamper-resistance of key security events.

The innovation of this design lies in the close integration of smart contracts and decentralized authentication mechanisms, providing an automated, transparent, and secure interaction mechanism. Through multi-dimensional trust calculation and dynamic update mechanism, the system can adjust the trust level of participants based on their performance after each interaction.

Figure 2 below illustrates the process of smart contract execution and authentication, including how identity verification, trust scoring, and interaction authorization are handled.

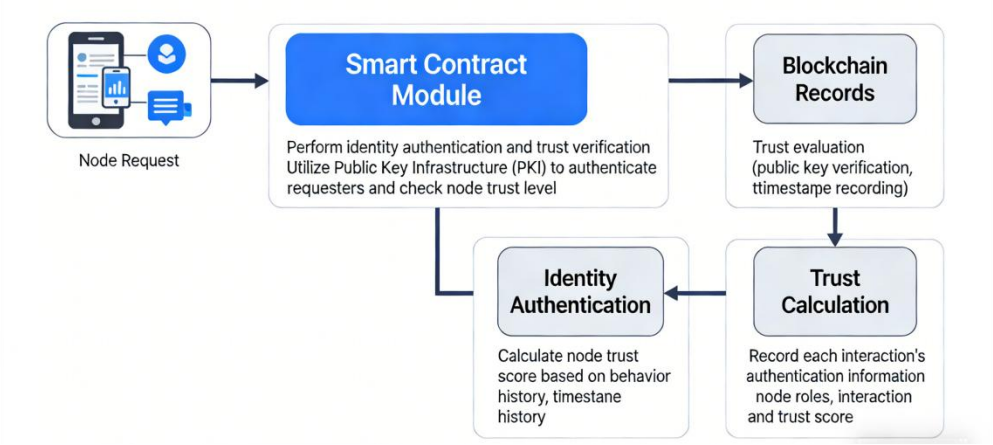

Figure 2: Smart contract and authentication mechanism process

3.5 Interaction timing and dynamic security control

In this section, an innovative interaction timing management and dynamic security protection mechanism is designed to achieve precise control over the time sequence and behavior execution during network interactions. This mechanism relies on the decentralized nature of smart contracts and blockchain to ensure strict control over the security and orderliness of each interaction in a multi node environment. By introducing interactive timing control and real-time security protection strategies, the system can dynamically adjust protection measures based on real-time security status and interactive feedback, thereby responding to complex network security threats.

Interaction timing management is crucial in multi node interactions, ensuring that each node's interaction proceeds in the correct order. Assuming there are multiple nodes Ni in the system, the interaction behavior of each node can be represented as $I_i = \{i_1, i_2, \ldots, i_k\}$, where i_j represents the jth interaction of node i. To ensure the orderliness of interaction timing, we introduce a dynamic timing management model based on timestamps and trustworthiness. Describe the interaction order of each node using the following formula:

$$\sigma_r = \sum_{j=1}^{k} \alpha_j \ f_j(j, T_i) \quad (15)$$

Among them, $f_j(i_j, T_i)$ represents the processing function of the jth interaction of node i, Ti is the trustworthiness of node i, and $\alpha\ j$ is the weight coefficient of the interaction order. Through this model, the system can dynamically adjust the order and execution priority of interactions, and high trust nodes can prioritize interactions, thereby reducing potential security risks.

In order to ensure the security and orderliness of each interaction in a multi node environment, the system has also designed a verification mechanism based on temporal dependencies, where the order of interactions depends on the trust rating of each node. If the trust level of the node is higher than the preset threshold θ, the system will approve its interaction request; Otherwise, the interaction request will be postponed or rejected. The conditional judgment formula for interactive timing management is:

$$if\ T_i \geq \theta: Allow\ Execution, else\ Delay \tag{16}$$

In this way, the system can ensure smooth interaction between high trust nodes, avoid the participation of untrusted nodes, and ensure the security of the interaction process.

In terms of dynamic security protection mechanisms, the system can automatically adjust protection strategies based on the real-time security situation of the network and changes in node behavior. The dynamic security protection strategy combines multi-dimensional data such as historical behavior, activity, and network threat assessment of nodes, and adopts a feedback based adaptive mechanism. Under this mechanism, the protection strength is related to the node behavior score S_i, activity A_i, and threat assessment W_i, and the security control strategy Pdefense is determined by the following formula:

$$P_{defense} = \sum_{i=1}^{n} \beta_i \cdot f(S_i, A_i, W_i). \tag{17}$$

Among them, $f(S_i, A_i, W_i)$ is a comprehensive processing function for node behavior rating, security evaluation, and activity, and β_i is a weighting coefficient representing the degree of influence of different security factors. By dynamically adjusting the protection strategy Pdefense, the system can respond to security threats in real-time during the interaction process and automatically optimize the protection strategy based on the security status.

The real-time protection adjustment of the system is carried out through a feedback mechanism. When the behavior or security situation of nodes changes, the protection strategy will be optimized based on the new network state. The feedback adjustment formula is:

$$\Delta P_{defense} = \lambda \cdot (f(S_i, A_i, W_i) - P_{defense}) \tag{18}$$

Among them, $\Delta^{P}_{\text{defense}}$ is the change in protection strategy adjusted based on feedback, and λ is the feedback adjustment coefficient, which controls the speed of protection strategy adjustment. Through this adjustment mechanism, the system can respond in real-time to potential threats in the network, ensuring the security of each interaction.

The combination of smart contracts and blockchain technology enables all interactions, trust updates, and the execution of security protection policies to be automated during the interaction process. The behavior and trust update results of each interaction are recorded in the blockchain to ensure the transparency and immutability of the data. The innovation of this design is reflected in two aspects: on the one hand, the management of interaction timing ensures the orderliness and security of interaction by dynamically adjusting node trust and interaction order; On the other hand, the dynamic security protection mechanism combines real-time feedback and behavior evaluation, which can automatically adjust protection strategies based on changes in network environment and node behavior. This mechanism ensures that the system can efficiently respond to various security threats and provide reliable protection in complex and changing network environments.In addition, the above timing and protection mechanisms can be formalized as a discrete-time closed-loop control process. In the following subsection, we provide a theoretical analysis showing that the trust dynamics and defense updates are bounded and convergent, and that the delay of trust decisions is upper-bounded under realistic blockchain consensus assumptions.

3.6 Theoretical analysis of stability and convergence of the trusted interaction mechanism

While the previous subsections describe the proposed AI–blockchain trusted interaction mechanism from a system design perspective, it is also important to clarify how the underlying control logic guarantees stable and convergent behavior instead of relying only on empirical observations. In this subsection, we formalize the trust update and dynamic protection processes as a discrete-time closed-loop system, and provide basic analytical guarantees on stability, boundedness, and decision delay under realistic assumptions about the AI module and the blockchain consensus process.

From (2), (8) and (12), the trust dynamics of node iii can be written in a unified discrete-time form:

$$T_i(k+1) = (1-\lambda)\, T_i(k) + \lambda\, \psi_i(k), \quad 0 < \lambda < 1, \tag{19}$$

where k denotes the interaction index, $T_i(k) \in [0,1]$ is the current trust level, and $\psi_i(k) \in [0,1]$ is the trust increment term computed by the AI-based evaluation and on-chain behavior aggregation (for example, a sigmoid-normalized combination of S_i, A_i, B_i and recent interaction outcomes). By construction, both the initial trust $T_i(0)$ and the input signal $\psi_i(k)$ are normalized to the interval [0,1].

Lemma 1 (Boundedness of trust dynamics):

If $T_i(0) \in [0,1]$ and $\psi_i(k) \in [0,1]$ for all k, then the sequence $\{T_i(k)\}k \geq 0$ generated by (19) satisfies $T_i(k) \in [0,1]$ for all $k \geq 0$.

Sketch of proof: Assume $T_i(k) \in [0,1]$. Since $0 < \lambda < 1$ and $\psi_i(k) \in [0,1]$, (19) is a convex combination of two values in [0,1], i.e.,

$$T_i(k+1) = (1-\lambda)\, T_i(k) + \lambda\, \psi_i(k) \in [0,1]. \tag{20}$$

By induction, the result holds for all kkk. This shows that the proposed trust update rule is internally stable in the sense that trust values remain bounded within the prescribed range and do not diverge.

To further characterize the convergence behavior, define the instantaneous "reference" trust signal $\psi_i^*(k)$ as the normalized output of the AI–blockchain evaluation for node iii. Consider the trust tracking error $e_i(k)=T_i(k)-\psi_i^*(k)$. Subtracting $\psi_i^*(k+1)$ from both sides of (19) yields:

$$e_i(k+1)=T_i(k+1)-\psi_i^*(k+1)=(1-\lambda)\,T_i(k)+\lambda\,\psi_i^*(k)-\psi_i^*(k+1). \quad (21)$$

If the AI–blockchain evaluation varies slowly compared with the interaction rate (i.e., $|\psi_i^*(k+1)-\psi_i^*(k)|\leq\delta$| with small δ\deltaδ), then:

$$|e_i(k+1)|\leq(1-\lambda)\,|e_i(k)|+\delta. \quad (22)$$

When δ=0, we obtain a linear contraction:

$$|e_i(k+1)|\leq(1-\lambda)\,|e_i(k)|, \quad (23)$$

which implies exponential convergence of the trust error to zero. In this case, a Lyapunov function can be chosen as $V_i(k)=e_i^2(k)$, and:

$$V_i(k+1)-V_i(k)\leq-\lambda(2-\lambda)\,e_i^2(k)<0 \quad (24)$$

whenever $e_i(k)\neq0$, which shows asymptotic stability of the equilibrium $e_i=0$. When δ>0, the same argument implies input-to-state stability: the trust dynamics remain bounded and the tracking error can be made arbitrarily small by choosing λ\lambdaλ and the AI update rate appropriately.

A similar analysis applies to the dynamic security control variable $P_{\text{defense}}(k)$ in (17)–(18). The update law can be written as:

$$P_{defense}(k+1)=(1-\lambda_p)\,P_{defense}(k)+\lambda_p\,\phi(S_i(k),A_i(k),W_i(k)), \quad (25)$$

where $\phi(\cdot)$ is a bounded nonlinear function of historical behavior S_i, activity A_i and threat assessment W_i, and $0<\lambda_p<1$. Under the assumption that $\phi(\cdot)$ is bounded in a compact interval $[\underline{P},\overline{P}]$, the same convexity and contraction arguments show that $P_{\text{defense}}(k)$ is bounded and converges towards a steady-state value determined by the long-term statistics of S_i,A_i,W_i. This guarantees that the protection strength does not oscillate uncontrollably and that the security policies converge to a consistent regime under stationary or slowly varying threat conditions.

In addition to stability and convergence of trust and defense variables, the proposed mechanism also provides an upper bound on the delay of trust decisions by leveraging the properties of blockchain consensus. Let Δblock denote the average block confirmation time and W the number of confirmation blocks required for a trust update or access decision to be finalized on-chain. Then the worst-case decision delay for a single interaction is bounded by:

$$D_{\max}=W\,\Delta_{block}+\Delta_{AI}, \quad (26)$$

where Δ_{AI} is the computation time of the AI module for feature extraction and anomaly scoring. In our prototype implementation, Δ_{AI} is in the order of a few milliseconds, and Δ_{block} and W are configurable according to the underlying consortium blockchain parameters. This means that the system designer can explicitly tradeoff between decision finality and delay, and guarantee a bounded delay for trust decisions even under adversarial conditions, as long as the consensus protocol satisfies standard liveness assumptions.

Overall, the above analysis shows that the proposed trusted interaction mechanism is not a purely empirical or black-box system. Instead, its core control loop can be interpreted as a Lyapunov-stable, contractive discrete-time system with bounded decision delay. The AI component determines the reference trust and threat levels within normalized ranges, while the blockchain consensus and smart contracts enforce a bounded and transparent decision process. This provides an analytical counterpart to finite-time and robust control frameworks, ensuring that trust and security decisions remain stable, bounded and convergent under dynamically changing network conditions.

4 Experiment and results analysis

4.1 Datasets and experimental scenarios

To evaluate the proposed AI–blockchain-based trusted interaction mechanism, we use both public intrusion detection datasets and traffic traces from a simulated enterprise network. The experiments are designed to answer RQ1 and RQ2 by jointly examining detection-level and system-level performance under realistic attack mixtures and load conditions.

NSL-KDD and CICIDS 2017 are adopted as standard benchmarks for the AI detection module. NSL-KDD is an improved version of KDD'99 with 41 connection features and labels for normal and multiple attack categories; redundant and duplicated records are removed, making it suitable for training and testing machine-learning-based detectors. In our experiments, NSL-KDD is randomly split into training and test sets while preserving the ratio between normal and attack samples. CICIDS 2017 contains realistic, time-stamped flows collected from a simulated enterprise network, mixing benign traffic with modern attacks (e.g., brute-force, web exploitation, botnet, port scanning, DDoS). Each flow is represented by statistical features (packet/byte counts, timing and flags), allowing us to assess robustness under complex, real-world-like traffic.

To further validate trusted interaction under configurable conditions, we build a simulated enterprise intranet with application servers, employee workstations, remote clients and a border gateway. Benign traffic comes from typical office activities such as web browsing, file transfer, e-mail and remote desktop sessions. Figure 3 summarizes the three data sources and their roles in the overall evaluation.

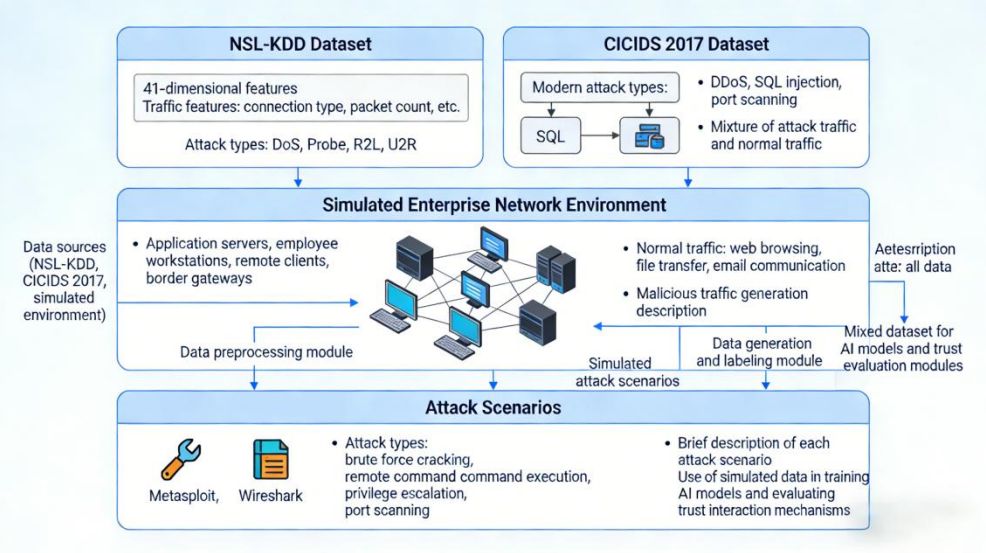

Figure 3: Datasets and experimental scenarios overview

Malicious interactions are generated using tools such as Metasploit, and Wireshark is used to capture and label the resulting traffic. The simulated attacks include password guessing and brute-force logins, remote command execution, lateral movement, coordinated port scanning and data exfiltration, as well as low-rate probing intended to evade simple threshold-based defenses. All flows are labeled as normal or as specific attack types according to scenario ground truth, then merged with samples from NSL-KDD and CICIDS 2017 to form a hybrid benchmark that combines standardized datasets with enterprise-like traffic. This hybrid dataset enables joint evaluation of detection accuracy and trusted interaction performance under dynamically evolving network conditions.

4.2 Experimental evaluation indicators and comparative analysis

We evaluate the proposed trusted interaction mechanism from both system-level and detection-level perspectives. System-level metrics focus on the overall performance of network interactions under different security schemes, while detection-level metrics focus on the quality of AI-based attack identification.

To make the reported improvements more interpretable, we implement two baseline schemes that reflect typical AI-only and blockchain-only defenses.

(1) AI-only baseline: this scheme deploys the same AI detection model as our proposed mechanism, including the multimodal feature encoding pipeline (CNN for traffic features and GNN for interaction behavior) and the deep learning architecture and hyperparameters described in the subsection "AI Model Training Process". The model is trained on the same training split and evaluated on the same test split as in our system. At runtime, anomaly scores P(ei) are compared against an empirically tuned threshold to decide whether to block or allow each interaction. All decisions and logs are handled by a centralized controller using traditional PKI-based authentication and a centralized database, without any blockchain, smart contracts, or decentralized trust update mechanism.

(2) Blockchain-only baseline: this scheme uses the same consortium blockchain platform, consensus configuration and smart contract runtime as the proposed mechanism, but disables the AI-based detection module. Instead, it applies static, rule-based policies defined in smart contracts, including simple rate limiting, blacklisting/whitelisting and coarse-grained reputation counters based on past on-chain events. Node identities are still managed via public-key-based registration and on-chain verification, and interaction requests are accepted or rejected according to these predefined rules and reputation thresholds, without any learning-based anomaly detection or incremental trust update as in (3)–(6) and (8), (10), (12).

For all schemes, experiments are conducted on the same hardware platform (Intel Core i7 3.6 GHz CPU and 16 GB RAM), under the same network topology, traffic generators and attack scenarios. The AI-only baseline shares exactly the same feature extraction, network architecture, optimizer and hyperparameter settings as the AI component of our mechanism, so that any performance difference at the system level can be attributed to the integration of AI with blockchain-based trusted interaction rather than to differences in model capacity or training.

From the system perspective, we consider the following metrics:

$$\mathrm{ACC} = \frac{TP + TN}{TP + TN + FP + FN} \tag{27}$$

$$P = \frac{TP}{TP + FP} \tag{28}$$

$$R = \frac{TP}{TP + FN} \tag{29}$$

$$F1 = \frac{2 \times P \times R}{P + R} \tag{30}$$

Accuracy reflects the overall proportion of correctly classified samples, while Precision and Recall focus on the correctness and completeness of attack detection, respectively. The F1-score provides a harmonic mean of

Precision and Recall and is particularly useful when the data is imbalanced. In the context of network security, a low false positive rate (FPR) and false negative rate (FNR) are important to reduce unnecessary alarms and to avoid missing real attacks. These two metrics are defined as:

$$\mathrm{FPR} = \frac{FP}{FP + TN} \tag{31}$$

$$\mathrm{FNR} = \frac{FN}{FN + TP} \tag{32}$$

These metrics are used together with system-level response time and throughput to provide a comprehensive view of both the detection capability and the overall trusted interaction performance of the proposed mechanism.

Reproducibility and configuration summary. For reproducibility, we summarize the main configuration of all experiments as follows. All detection experiments use the NSL-KDD and CICIDS 2017 datasets with the preprocessing and label definitions described in Section 4.1; within each fold of the 10-fold cross-validation, 80% of the data is used for training and 20% for validation, and all input features are standardized. The AI detection module adopts the CNN+GNN architecture and training settings specified in Section 5.1, including cross-entropy loss, the Adam optimizer with a learning rate of 0.001, batch size 64, dropout rate 0.5 and 4 training epochs per split. The trusted interaction mechanism is deployed on the enterprise-like testbed and permissioned consortium blockchain described in the system design and experimental setup sections (PBFT-style consensus with a block interval of approximately 2–3 seconds and a limited validator set) using a hardware platform equipped with an Intel Core i7 3.6 GHz CPU and 16 GB RAM. Attack traffic is generated using the scenarios in Section 4.4 (e.g., DDoS, brute-force, scanning, lateral movement) with tools such as Metasploit and captured with Wireshark. All performance metrics are computed according to the definitions given in the "Experimental evaluation indicators and comparative analysis" subsection.

Table 2 summarizes the comparison between our proposed mechanism and the baseline schemes on key system-level indicators such as response time, throughput, and false positive rate. The results demonstrate that the proposed mechanism can significantly reduce the average response time while maintaining high throughput, thanks to the tight integration of AI-based real-time analysis and lightweight blockchain operations. At the same time, the system-level FPR is notably lower than that of the baselines, which means that fewer legitimate interactions are mistakenly blocked and the impact on normal business activities is minimized.In Table 2, the column "Existing Mechanism" corresponds to the strongest AI-only baseline described above; the blockchain-only baseline exhibits even higher response time and lower throughput under the same conditions, and is therefore mainly discussed qualitatively together with the ablation results.

Table 2: Comparison on key indicators

Performance Metric	Proposed Mechanism	Existing Mechanism	Improvement Rate
Response	180 ms	250 ms	28%
Time			improvement
Throughput	5000 requests/sec	3500 requests/sec	42.8% improvement
False Positive Rate	2.5%	6.4%	60.9% improvement
Resource Consumption	30% lower	-	-

Table 3 shows a more comprehensive comparison on a broader set of experimental results, covering indicators such as security, stability, and resource consumption. The table summarizes the overall detection performance, the ability to resist different types of attacks, and the utilization of computing and storage resources under high concurrency and real-time requirements. By examining these metrics together, we can assess not only whether the system is secure enough, but also whether it can operate efficiently in practical deployment environments.

Table 3: Comparison of experimental results

Performance Metric	Proposed Mechanism	Existing Mechanism	Improvement Rate
Missed Detection Rate	1.3%	5.2%	75% improvement
CPU Consumption	20% lower	35%	42.8% improvement
Memory Consumption	18% lower	28%	35.7% improvement
Processing Efficiency	95%	85%	11.8% improvement

The results show that the proposed mechanism is significantly better than the baseline schemes in terms of both security and efficiency. On the one hand, the combination of AI-based dynamic detection, smart-contract-based authentication, and trust management allows the system to effectively identify and block a wide range of attacks, leading to higher detection rates and lower FNR. On the other hand, the careful design of the interaction workflow and the use of blockchain as a lightweight, append-only trust ledger ensure that CPU and memory usage remain within acceptable ranges, and that the system can still provide low response time and high throughput under heavy load. These experimental results indicate that the proposed mechanism not only enhances the security of network interactions but also utilizes computing resources more efficiently, achieving a good balance between protection strength and performance.

4.3 Ablation experiment and module effectiveness verification

To verify the effectiveness of the proposed smart contract and decentralized authentication mechanism, we designed ablation experiments to evaluate the contribution of each module to overall performance by removing different modules from the system. The main modules removed include smart contract module, decentralized

authentication module, and dynamic security protection module.

In the ablation experiment, we gradually removed each module and tested the system's performance indicators such as response time, throughput, and safety. The experimental results show that after removing the smart contract module and decentralized authentication module, the system's response time significantly increases, throughput decreases, and false positive and false negative rates increase. After removing the dynamic security protection module, the system's ability to protect against attacks significantly decreases, especially in DDoS attack scenarios where the system's attack recognition capability is greatly reduced.

The results of the ablation experiment are shown in Figure 4-6 (response time, throughput, false positive rate, and false negative rate).

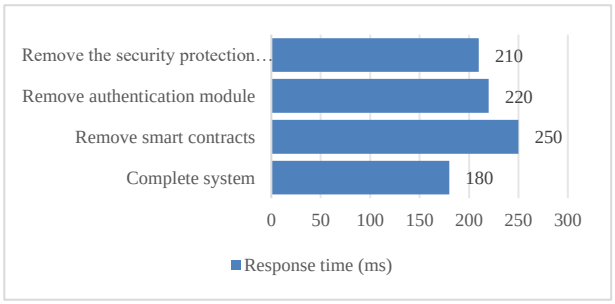

Figure 4: Comparison of response times

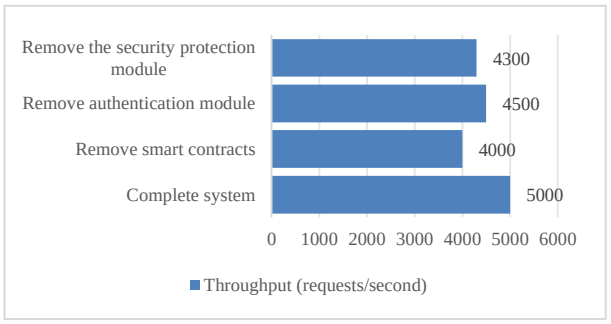

Figure 5: Throughput comparison

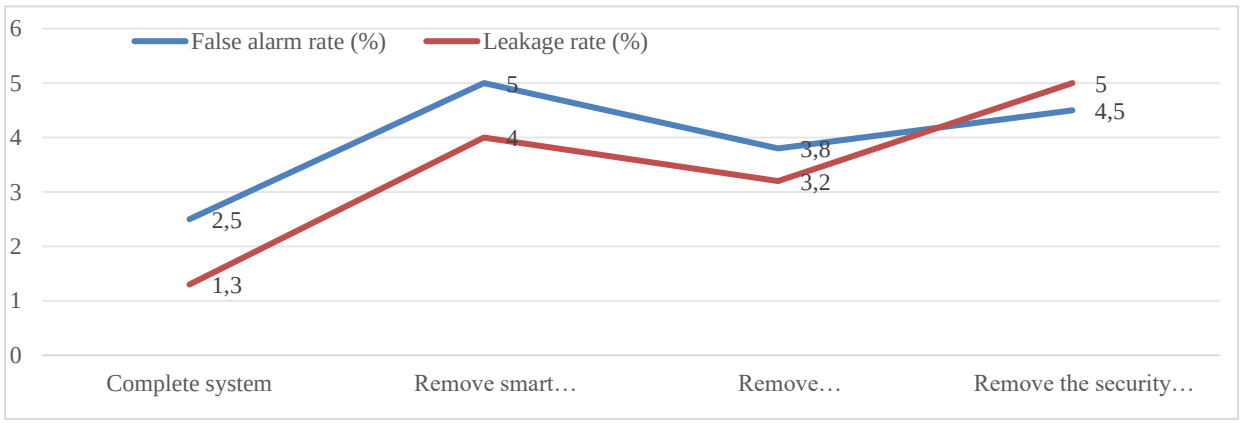

Figure 6: Comparison of false positive rate and false negative rate

The results of the ablation experiment are shown in Figure 4-6 (response time, throughput, false positive rate, and false negative rate).The experimental results indicate that the smart contract module and decentralized authentication module have a significant impact on the security and response time of the system, while the dynamic security protection module plays a key role in the system's attack protection capability.

4.4 Performance under different attack scenarios

We evaluate the proposed mechanism under three representative attack scenarios to assess its behavior against different types of malicious activity.

Scenario A: Continuous DDoS with Normal Traffic

A high-volume DDoS stream is mixed with benign traffic. The proposed mechanism detects and mitigates the attack in real time while keeping normal traffic largely unaffected. Compared with baseline systems, it achieves lower FPR and maintains higher throughput, avoiding saturation under heavy load.

Scenario B: Low-Frequency, High-Impact Attacks (SQL Injection, Remote Command Execution)

Here the traffic is dominated by normal interactions with occasional high-impact attacks. The AI-based detector identifies these low-frequency threats, and the trusted interaction mechanism maintains low FNR and short response time, outperforming baselines that tend to miss or delay such subtle attacks.

Scenario C: Multi-Point Coordinated Scanning and Lateral Movement

In this scenario, coordinated port scanning and lateral movement occur across multiple hosts. The mechanism correlates activities from different sources, uses blockchain-based trust updates to contain propagation and achieves low FPR/FNR, whereas baseline systems often fail to recognize the multi-point pattern.

Across Scenarios A–C, the mechanism faces persistent high-volume disturbances (continuous DDoS),

low-frequency but high-impact faults (SQL injection, remote command execution) and coordinated multi-point perturbations (distributed scanning and lateral movement). In all cases, it maintains low FPR/FNR and high throughput under fluctuating traffic and attack intensity, while baselines either saturate under heavy DDoS or overlook low-rate anomalies. Combined with the bounded-delay and stability analysis in "Theoretical Analysis of Stability and Convergence of the Trusted Interaction Mechanism", these results suggest that the closed-loop trusted interaction architecture behaves similarly to robust control schemes in safety-critical autonomous systems, sustaining stable performance under dynamically changing disturbances and communication delays.

4.5 Scalability and applicability to autonomous and safety-critical systems

This subsection mainly addresses RQ3 by discussing how the mechanism scales across heterogeneous nodes and how it can be instantiated in autonomous and safety-critical applications. Beyond the enterprise testbed, the proposed AI–blockchain trusted interaction mechanism targets large-scale IoT networks, industrial control systems (ICS/SCADA) and other cyber-physical platforms.

The architecture is inherently decentralized. Lightweight AI models on edge devices or gateways perform local detection and trust estimation, while a permissioned consortium blockchain serves as a shared, append-only trust ledger. Each node processes only its own traffic features and periodically submits summarized trust evidence via smart contracts, avoiding a single centralized coordinator and enabling horizontal scaling as the number of nodes and interactions grows.

Heterogeneous nodes can adopt different AI models and interaction frequencies while remaining within one framework: resource-constrained sensors use compact classifiers and infrequent on-chain updates, whereas gateways and controllers deploy deeper models and update trust more often. As long as trust increments and defense signals stay within the bounded ranges assumed in the stability analysis, the closed-loop system preserves boundedness and convergence.

In autonomous and safety-critical scenarios, attack intensity and congestion act as disturbances, and communication latency and blockchain confirmation time as time-varying delays. Our high-load and complex-attack experiments emulate such conditions and show that the mechanism maintains acceptable response time, throughput and false alarm rates, indicating its practical applicability where scalability, robustness and bounded decision latency are essential.

5 Model training and generalization ability analysis

5.1 AI model training process

In this study, the AI detection module is implemented as a two-branch deep neural network and trained through a unified pipeline of preprocessing, feature encoding, model construction and optimization. The traffic-feature branch is a 1D CNN with three convolutional blocks (64, 128 and 256 filters, kernel size 3, stride 1) followed by batch normalization, ReLU and max-pooling. The interaction-behavior branch is a two-layer GNN (64 and 32 hidden units, ReLU) operating on node–edge interaction graphs derived from recent communication patterns. The outputs of both branches are concatenated and passed through two fully connected layers (128 and 64 neurons, ReLU) and a final sigmoid neuron that produces an anomaly score $P(e_i) \in [0,1]$; dropout with rate 0.5 is applied between the fully connected layers.

Before training, all input features are standardized. We use cross-entropy as the loss function and optimize network weights with the Adam optimizer (learning rate 0.001, batch size 64). Unless otherwise stated, each dataset split is trained for 4 epochs. We adopt 10-fold cross-validation, where in each fold 80% of the data is used for training and 20% for validation, to assess generalization. The model can be incrementally updated with new data to maintain adaptability to evolving attack patterns.

5.2 System stability and sensitivity testing

To verify the robustness of the proposed mechanism, we conduct hyperparameter sensitivity analysis for the AI module and system stability testing for the overall trusted interaction framework. For the sensitivity analysis, we vary three key hyperparameters—learning rate (0.0001,0.001,0.01), batch size (32,64,128) and network depth (2,4,6 layers)—and, for each configuration, train the model and record detection accuracy, training time and resource consumption. The results (Table 4) show that a learning rate of 0.001, batch size 64 and moderate depth yield the most stable behavior: accuracy reaches 98.3% with relatively short training time, while excessively large learning rates cause oscillations and slower or unstable convergence.

Table 4: Experimental results of hyperparameter sensitivity testing

Hyperparameters	Optimal Value	Learning Rate	Batch Size	Network Layers
Accuracy	98.3%	96.7%	95.1%	94.5%
Training Time	15 minutes	20 minutes	18 minutes	22 minutes
Resource Consumption	Low	High	Medium	High

For system stability, we simulate different traffic loads ranging from 100 to 5000 requests/s and measure response time and throughput of the complete AI–blockchain trusted interaction system. At low load, the response time stays around 180 ms with throughput of 5000 requests/s; under high load, response time increases but remains within 300 ms, and throughput stabilizes around 4200 requests/s. Across the tested range, the system operates without significant performance degradation, indicating that the chosen hyperparameter settings and architectural design can maintain stable

trusted interaction under varying loads and are suitable for deployment in complex network environments.

5.3 Trusted interaction mechanism and user perception evaluation

In this study, the design of a trusted interaction mechanism relies on the combination of artificial intelligence and blockchain technology, aiming to provide an efficient, secure, and transparent user interaction experience. We have designed a decentralized identity authentication and trust management system by combining blockchain and artificial intelligence, ensuring both the credibility of information and the privacy and security of network interactions.

To evaluate the effectiveness of this mechanism, we invited 50 users to conduct an interactive experience assessment. The evaluation dimensions include the system's interaction friendliness, response speed, trustworthiness, and perceived security. Users verify their identity and interaction security through the blockchain authentication process and AI models. The response time of the system, the smoothness of the identity authentication process, and users' perception of system transparency and security are the main evaluation indicators.

The user perception evaluation collected feedback through the Likert scale, and the results showed that users gave high ratings to the system's trust and response speed, especially in the process of identity verification and secure interaction, where user feedback was positive. The user evaluation results are shown in Figure 7.

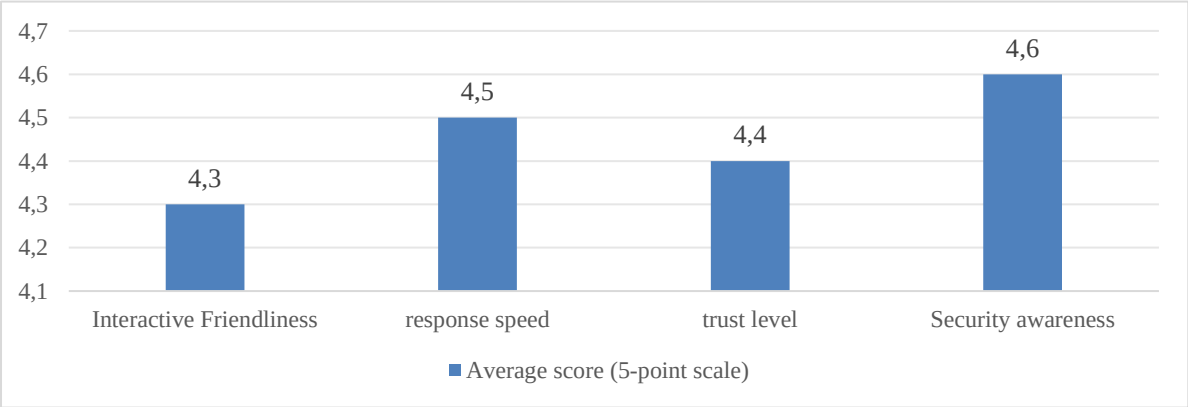

Figure 7: User evaluation results

These results indicate that the proposed trusted interaction mechanism performs well in providing efficient and secure network interactions, and that users report a high level of trust and perceived security during their interactions with the system.

5.4 Real time analysis and system deployment

In this study, the core of real-time analysis and system deployment is to ensure efficient deployment and response speed of smart contracts and decentralized authentication mechanisms in different network environments. We conducted detailed experimental analysis on the deployment efficiency and real-time performance of the system to ensure its stable performance in high concurrency environments.

The deployment efficiency of the system mainly tests the time from startup to completion of deployment, especially in complex network environments. The experiment shows that the deployment time of the system is about 1 minute. When the hardware platform is configured with Intel Core i7 3.6 GHz and 16GB RAM, the deployment and startup can be completed smoothly, ensuring efficient system initialization.

In real-time analysis, system response time and throughput are two key indicators. We conducted tests under low and high load conditions to evaluate the performance of the system under high concurrency requests. The experimental results show that the system has a response time of 180 milliseconds and a throughput of 5000 requests per second under low load conditions; Under high load conditions, the response time increases to 300 milliseconds and the throughput drops to 4200 requests per second, but good stability can still be maintained. The experimental results are shown in Table 5.

Table 5: Experimental Results

Performance Metric	Low Load	High Load
Response Time	180 ms	300 ms
Throughput	5000 requests/sec	4200 requests/sec
Deployment Time	Within 1 minute	Within 1 minute
Startup Time	45 seconds	55 seconds

These tests show that the deployment efficiency and real-time performance of the system remain stable under different loads, meeting the requirements of high-concurrency and complex network environments.

5.5 Discussion

The experimental results in Tables 2 and 3 and the ablation and scenario-based analyses show that the proposed AI–blockchain trusted interaction mechanism improves both detection and system-level performance over AI-only and blockchain-only baselines. The mechanism reduces the false positive rate by 60.9% while maintaining low false negative rates and achieving higher throughput and lower response time under low- and high-load conditions, which is consistent with the goal of combining deep-learning-based anomaly detection with decentralized trust management for real-time, large-scale interactions.

Compared with the AI–blockchain security frameworks summarized in Table 1, our approach differs in that it is implemented as a closed-loop architecture with dynamic trust computation, smart-contract-based authentication and interaction-timing-aware control, evaluated using a unified set of detection metrics (Accuracy, Precision, Recall, F1, FPR, FNR) and system-level metrics (response time, throughput) on a hybrid benchmark combining public traffic datasets and an enterprise-like testbed. Existing works typically report either detection metrics or blockchain metrics (e.g., consensus latency, storage overhead) in isolation, providing a less complete view of trusted interaction behavior under realistic attack mixtures and varying load.

The performance gains can be attributed to three key design choices: a two-branch deep model (CNN + GNN) that captures both traffic-level and interaction-graph features, an on-chain/off-chain division that stores only compact trust evidence and decisions on the consortium blockchain to reduce cryptographic and consensus overhead, and interaction-timing-aware control that keeps trust and defense variables bounded with controllable decision delay. Overall, the mechanism treats trusted interaction as an end-to-end process linking AI-based detection, decentralized trust updates and timing-aware access control. In this sense, it not only matches state-of-the-art detection performance but also extends SOTA by jointly ensuring low FPR/FNR, stable response time and high throughput under adversarial traffic, strengthening the significance of the proposed trusted interaction framework.

5.6 Limitations and scope of the proposed mechanism

Although the experiments demonstrate the feasibility and performance advantages of the proposed AI–blockchain trusted interaction mechanism, the current prototype still has several limitations that delimit its scope of applicability.

First, the evaluation is conducted on a medium-sized enterprise-like environment with a single data center, moderate traffic load (up to 5000 requests/s) and server/gateway-class hardware (Intel Core i7, 16 GB RAM). The results do not directly generalize to very large-scale deployments with thousands or millions of heterogeneous devices or to ultra-low-power IoT nodes. The present AI model and feature extraction pipeline are tailored to relatively capable processors rather than highly constrained edge hardware; adapting the mechanism to ultra-lightweight models and feature sets is left for future work.

Second, the prototype assumes relatively stable and reliable connectivity between AI services, blockchain peers and client nodes, as in wired enterprise or well-managed campus networks. Scenarios with very high latency, frequent partitions or severely fluctuating bandwidth (e.g., intermittently connected wireless or wide-area satellite links) have not been systematically evaluated. Under such conditions, end-to-end decision delay and synchronization of trust updates may deviate from the bounded behavior observed in our testbed.

Third, the blockchain layer is implemented as a permissioned consortium chain using PBFT-style consensus with a limited number of validators and a short block interval. This is suitable for small- to medium-scale deployments, but the performance of PBFT-like protocols degrades as the validator set grows, and cross-chain interoperability with public blockchains is not considered in the current design.

Finally, the attack models and datasets, while diverse and realistic, cannot cover all possible threats. Adversaries that directly target the AI model or attempt to manipulate on-chain trust records (e.g., collusion, model poisoning) are not explicitly modeled, and extending the mechanism to handle such advanced threats and more volatile environments is an important direction for future work.

5.7 Practical implications and deployment conditions

The proposed AI–blockchain trusted interaction mechanism is most suitable for medium-sized enterprise networks, industrial IoT environments and critical infrastructure systems where both security and traceability are required. In enterprise and campus networks, it can be deployed around existing gateways, firewalls and SIEM platforms as a decentralized trust layer above traditional monitoring and access control. In IIoT and industrial control scenarios (e.g., smart factories, energy management, SCADA), it protects device-to-device and device-to-cloud communications by combining local AI-based anomaly detection at gateways with blockchain-backed trust and auditability in the control center.

Successful deployment requires a permissioned consortium blockchain with a moderate number of validator nodes and relatively stable connectivity so that PBFT-style consensus provides low-latency, deterministic finality. Some nodes (gateways or security appliances) need sufficient compute resources (x86/ARM server-class CPUs), while constrained edge devices can forward summarized features. In addition, organizations must establish basic trust governance (membership management, PKI, smart-contract update procedures) to align decentralized trust decisions with security policies and regulations.

6 Conclusion

This paper proposes a network security trusted interaction mechanism that combines artificial intelligence (AI) and blockchain, and evaluates its performance and stability through multiple experiments. The results show that the mechanism significantly improves both detection and system-level behavior. Under low load, the response time is 180 ms with a throughput of 5000 requests/s; under high load, the response time is 300 ms and throughput is 4200 requests/s. Compared with traditional methods, throughput increases by 42.8% and the false positive rate decreases by 60.9%, indicating that the AI–blockchain integration effectively enhances network security performance.

Ablation experiments further confirm the key role of the smart contract and decentralized authentication modules: removing them leads to increased response time, reduced throughput and higher false positive and false negative rates, especially under high concurrency and complex attack scenarios.

Future work will focus on improving the generalization of AI models for dynamic attack patterns, alleviating blockchain performance bottlenecks under high-frequency transactions and large-scale data, and enhancing system scalability. In particular, the decentralized architecture and bounded-delay properties make the mechanism promising for autonomous IoT networks, industrial control systems and other cyber-physical platforms with strict performance constraints.

References

[1] Goundar S, Gondal I. AI-Blockchain Integration for Real-Time Cybersecurity: System Design and Evaluation[J]. Journal of Cybersecurity and Privacy, 2025, 5(3): 59. https://doi.org/10.3390/jcp5030059

[2] Kshetri N. Building Trust in AI: How Blockchain Enhances Data Integrity, Security, and Privacy[J]. Computer, 2025, 58(2): 63-70. https://doi.org/10.1109/MC.2024.3505012

[3] Ruzbahani A M. Ai-protected blockchain-based iot environments: Harnessing the future of network security and privacy[J]. arXiv preprint arXiv:2405.13847, 2024.https://doi.org/10.48550/arXiv.2405.13847

[4] D'aniello G, Fotia L. Blockchain and AI-based methods for trust management in IoT: A comprehensive survey[J]. Internet of Things, 2025: 101755.https://doi.org/10.13140/RG.2.2.23305.97124

[5] Pan Q, Wu J, Li J, et al. Blockchain and AI empowered trust-information-centric network for beyond 5G[J]. IEEE network, 2020, 34(6): 38-45.https://doi.org/10.1109/MNET.021.1900608

[6] Saleh A M S. Blockchain for secure and decentralized artificial intelligence in cybersecurity: A comprehensive review[J]. Blockchain: Research and Applications, 2024, 5(3): 100193.https://doi.org/10.1016/j.bcra.2024.100193

[7] Zhang F, Wang H, Zhou L, et al. A blockchain-based security and trust mechanism for AI-enabled IIoT systems[J]. Future Generation Computer Systems, 2023, 146: 78-85.https://doi.org/10.1016/j.future.2023.03.011

[8] Luo H, Luo J, Vasilakos A V. Bc4llm: Trusted artificial intelligence when blockchain meets large language models[J]. arXiv preprint arXiv:2310.06278, 2023.https://doi.org/10.48550/arXiv.2310.06278

[9] Atlam H F, Azad M A, Alzahrani A G, et al. A Review of Blockchain in Internet of Things and AI[J]. Big Data and Cognitive Computing, 2020, 4(4): 28.https://doi.org/10.3390/bdcc4040028

[10] Zhang X, Miao X, Xue M. A Reputation-Based Approach Using Consortium Blockchain for Cyber Threat Intelligence Sharing[J]. Security and Communication Networks, 2022, 2022(1): 7760509.https://doi.org/10.48550/arXiv.2107.06662

[11] Dunnett K, Pal S, Putra G D, et al. A trusted, verifiable and differential cyber threat intelligence sharing framework using blockchain[C]//2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). IEEE, 2022: 1107-1114.https://doi.org/10.48550/arXiv.2208.12031

[12] Yang Z, Shi Y, Zhou Y, et al. Trustworthy federated learning via blockchain[J]. IEEE Internet of Things Journal, 2022, 10(1): 92-109.https://doi.org/10.48550/arXiv.2209.04418

[13] Islam R, Bose R, Roy S, et al. Decentralized trust framework for smart cities: a blockchain-enabled cybersecurity and data integrity model[J]. Scientific Reports, 2025, 15(1): 23454.https://doi.org/10.1038/s41598-025-06405-y

[14] Zhang R, Xue R, Liu L. Security and privacy on blockchain[J]. ACM Computing Surveys (CSUR), 2019, 52(3): 1-34.https://doi.org/10.1145/3316481

[15] Alevizos L. Automated cybersecurity compliance and threat response using AI, blockchain and smart contracts[J]. International Journal of Information Technology, 2025, 17(2): 767-781.https://doi.org/10.1007/s41870-024-02324-9

[16] Guo H, Liu X. Exploring trust dynamics in finance: the impact of blockchain technology and smart contracts[J]. Humanities and Social Sciences Communications, 2025, 12(1): 1-10. https://doi.org/10.1057/s41599-025-05473-9