Federated Learning-Based Network Threat Detection System with Digital Twins for IoT Security

Feixian Sun
School of Modern Information Technology, Henan Polytechnic, Zhengzhou 450046, China
Zhengzhou Key Laboratory of Electronic Intelligent Sensor Application Technology, Zhengzhou 450046, China
E-mail: 18538771228@126.com

Keywords: internet of things, digital twin, blockchain technology, federated learning support, network threat detection system

Received: May 23, 2025

Abstract: With the widespread use of Internet of Things technology, network threats are increasing, posing a serious challenge to the security of Internet of Things systems. To address this challenge, an efficient network threat detection system is designed by combining digital twin technology and federated learning algorithms. The research first uses the decentralized and immutable characteristics of blockchain technology to securely store and verify the data in the Internet of Things network, while combining the digital twin technology to carry out virtual mapping of the Internet of Things entity, real-time monitoring of its status, and timely detection of potential threats. Subsequently, a comprehensive simulation of the Internet of Things system is conducted via the digital twin network to generate data samples. Following this, the federated learning algorithm is employed, enabling multiple participants to collaboratively train the model. This approach enhances the model's detection capabilities while safeguarding the privacy of local data. Additionally, a distributed architecture is adopted to facilitate the efficient processing and analysis of large-scale Internet of Things data. Finally, the proposed system is tested. The test results show that in terms of registration time, when the number of attributes is 10, the registration time of the research system is about 0.57 seconds, the registration time of the micro-step online threat intelligence platform is about 0.59 seconds, and the registration time of the Check Point Infinity platform is about 0.62 seconds. When the number of access policies is 10, the shortest encryption time of the research system is 0.52 seconds, the second is 0.54 seconds of the micro-step online threat intelligence platform, and the longest is 0.58 seconds of the Check Point Infinity platform. In comparison to the benchmark system, the proposed research system demonstrates superior efficiency during the registration and encryption phases. This is primarily attributed to the precise modeling of Internet of Things devices using digital twin technology and the efficient data processing capabilities inherent in the federated learning algorithm. Consequently, the research system offers a swifter and more effective solution for detecting network threats within the Internet of Things environment.

Povzetek: Združuje digitalne dvojčke, verigo blokov ter federativno učenje za decentralizirano zaznavanje groženj v IoT. Poudarjen je zasebnostno ohranjen trening in razlaga modela, je preizkušeno na simulacijah in resničnih scenarijih.

1 Introduction

In recent times, with the swift advancement of Internet of Things (IoT) technology, more and more devices are connected to the network through information sensing devices and protocol agreements, achieving intelligent recognition, positioning, tracking, and monitoring functions [1, 2]. With the increasing severity of network threats, it poses a serious threat to the security and reliability of IoT systems [3]. Therefore, developing an efficient and reliable network threat detection system is extremely important for ensuring the normal operation of the IoT. Federated Machine Learning (FML), as a highly promising technological approach, has received widespread attention [4]. FML, also known as Federated Learning (FL), united learning, or alliance learning, is based on deep learning and can combine data from multiple devices to improve model accuracy while maintaining device data privacy [5]. Ma applied FL to the advancement of a smart tourism service system based on the IoT and machine learning, which not only improved the system's performance but also ensured data privacy [6]. Digital Twin (DT) technology greatly enhances the data processing capabilities of physical entities by constructing virtual images in the digital space, providing powerful technical support for deeper analysis and monitoring of the status of IoT devices [7, 8]. For example, Zhao et al. raised an efficient communication FL method for industrial IoT DT systems, further optimizing the efficiency of data processing and model training [9]. In addition, blockchain technology also serves as a crucial component in ensuring the security of IoT data. By offering tamper-resistant data storage and transmission capabilities, blockchain can effectively safeguard data

against tampering and theft, thereby establishing secure and dependable connections among IoT nodes. Currently, there is a wealth of research focused on the application of blockchain technology in the IoT domain. For example, Sasikumar et al. [10] proposed a blockchain-based trust mechanism for DT industrial IoT, while Zheng et al. [11] further explored the integration of blockchain and DT technology in the context of trusted DT for IoT, aiming to balance data privacy protection and the secure and reliable operation of the system. The detailed work summary table is shown in Table 1.

Table 1: Summary of related work

References	Model/method	Data set	Detection accuracy	Calculate the cost	Safety feature	Shortcoming	Contribution
[6]	IoT + Machine learning + FL	Smart travel service dataset	High	Intermediate	Basic encryption	Lack of dynamic device modeling and high computational latency	A preliminary framework for the combination of iot and FL is proposed, which lays a foundation for the subsequent research
[9]	Industrial IoT DT + FL	Industrial IoT data set	High	Intermediate	No blockchain support	Inadequate privacy protection	The introduction of DT technology improves the adaptability and accuracy of the model
[10]	Blockchain + DT	Industrial IoT data set	Intermediate	Intermediate	Tamper-resistant	No FL is integrated, and the detection accuracy is not quantified	The combination of blockchain technology enhances the security and immutability of data
[11]	Blockchain + DT	IoT data set	Intermediate	Intermediate	Trusted authentication	Poor scalability	A scheme combining blockchain and DT is proposed to enhance the credibility of the system
Proposed system	FL + DT + Blockchain	IoT cyber threat dataset	High	Low	High	No obvious deficiency	FL, DTs, and blockchain technologies are integrated for efficient and secure cyber threat detection

From Table 1, the existing DT research relies on centralized data aggregation and fails to make full use of FL's distributed privacy protection capability. The proposed system realizes local data non-sharing through FL and combines blockchain encryption storage to double guarantee privacy. The existing FL model lacks the real-time virtual mapping of DT and cannot respond to threats dynamically. The proposed system uses DT to provide real-time device status image and FL model to update dynamically to improve the timeliness of threat detection. The blockchain scheme introduces high latency, which is difficult to apply to large-scale IoT, and the proposed system optimizes FL participation node selection, reduces the frequency of blockchain transactions, and balances security and efficiency. The proposed system innovatively integrates FL, DTs, and blockchain technologies to enable efficient and secure cyber threat detection. The system demonstrates excellent performance on the IoT network

threat dataset, achieving high detection accuracy with low computing costs, and boasts a high level of security. These attributes compensate for some of the limitations in previous literature, offering a more comprehensive and advanced solution for the IoT network security domain. It is anticipated to foster technical advancements and practical applications in this field.

2 Methods and materials

2.1 IoT network threat detection based on blockchain technology and DT technology

With the increase in the number of IoT devices, the amount of data and requests that central nodes need to process has significantly increased, seriously affecting the system's response speed [12]. In the traditional centralized model of IoT, data processing and management are highly dependent on the central node, and once the node fails or suffers a cyber attack, the entire system may be paralyzed. This issue underscores the vulnerability of existing IoT systems when faced with large-scale device access, and it also reveals the core challenge of the research: How to construct an efficient and reliable network threat detection system under a decentralized architecture using DT and FL technologies, in order to enhance the security and stability of IoT systems. The distributed ledger of blockchain technology can store data on multiple nodes instead of relying on a single central node [13]. The incorporation of blockchain technology in the IoT significantly mitigates the risk of a single point of failure, thereby enhancing the system's robustness and resilience to faults [14]. The flowchart illustration of the IoT blockchain architecture is in Figure 1.

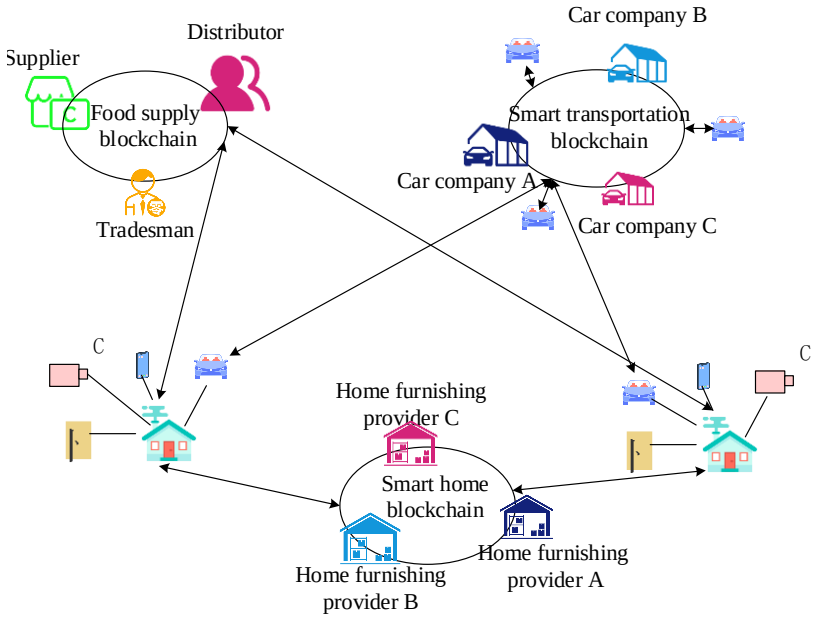


Figure 1: Flowchart illustration of the IoT blockchain architecture

From Figure 1, in the intelligent transportation system, IoT devices such as traffic signal controllers and vehicle sensors are faced with network threats such as distributed denial of service attacks and false data injection. DT technology can build a virtual image of the traffic network, simulate the attack propagation path, and rehearse defense strategies. In a smart home environment, where devices such as home gateways and smart appliances are vulnerable to man-in-the-middle attacks or new types of malware, the blockchain alliance chain can restrict the data access rights of manufacturers and user nodes, ensuring the privacy and security of device data. In the food supply system, the cold chain sensors in the supply chain nodes may be physically tampered with, leading to the falsification of temperature records. Moreover, blockchain technology can store the detection results of each link on the chain for regulators to trace, thus effectively preventing data falsification and leakage. In order to accurately detect threats to the IoT network, a blockchain regulatory framework that supports multi-party participation is studied. The framework combines symmetric encryption algorithms to quickly encrypt data generated by IoT devices. The key generation center will query the local database to see if the regulatory center has completed registration. After registration, the key generation center generates a registration record expression, as shown in equation (1).

$$R \leftarrow (U \| iw_s \| N \| t_{\mathrm{Re}g})^k \tag{1}$$

In equation (1), the customer set is represented by symbol U, the central signature is represented by symbol N, and the regulatory center is represented by symbol iw_s. $t_{\mathrm{Re}g}$ is time. The data information expression of the regulatory center is shown in equation (2).

$$M_1 \leftarrow (m \| C \| J_1 \| t_{\mathrm{R}A} \| id_{RA}) \tag{2}$$

In equation (2), the data to be regulated is represented by symbol m, customer information data is represented by symbol C, the time required for regulation is represented by symbol t_{RA}, and the regulatory code is represented by symbol id_{RA}. The calculation formula for regulatory results information by regulatory agencies is shown in equation (3).

$$J_1 \leftarrow (id_m \| C \| id_{BC} \| Q \| \mu_{RA}) \tag{3}$$

In equation (3), RA stands for supervisory center, the code to be regulated is represented by symbol id_m, and the audit judgment result is represented by symbol μ_{RA}. The business chain code is represented by symbol id_{BC}. The ciphertext b_1 expression corresponding to the symmetric key k in the regulatory center is shown in equation (4).

$$b_1 = A.Enc(PK, k, T) \tag{4}$$

In equation (4), PK represents the common parameters in the attribute based encryption algorithm and T represents the access structure in the attribute based encryption algorithm. The ciphertext b_2 expression is shown in equation (5).

$$b_2 = F.Enc(k, M_1) \tag{5}$$

The encrypted data (b_1, b_2) will be handed over to the regulatory authority for another regulatory judgment, and the judgment formula is shown in equation (6).

$$M_2 \leftarrow (m \| C \| J_2 \| t_S \| id_S) \tag{6}$$

In equation (6), t_S represents the time during which the regulatory authority conducted supervision, and J_2 represents the regulatory outcome information. A smart contract is a self-executing protocol with terms that are encoded in the blockchain as code, which automatically triggers an action when a predefined condition is satisfied. The smart contract will compare the supervision results before and after the two times, and the supervision results of the regulator and the regulator node are consistent, and the supervision information expression is shown in equation (7).

$$L \leftarrow (M \| t_{pro}) \tag{7}$$

In equation (7), t_{pro} represents the data processing time. A DT Network (DTN) is a virtual digital model of a physical network facility created by the DT technology. It can create a virtual image of a physical network facility to simulate the propagation path and impact scope of a network attack. This enables security teams to simulate potential cyber threats in a controlled environment, evaluate the effects of different defense strategies, and select the optimal response [15, 16]. In the IoT blockchain architecture, DTNs can leverage the distributed ledger, smart contracts, and consensus mechanisms of blockchain to enhance data security, privacy protection, and trust building. The schematic diagram of the DTN is in Figure 2.

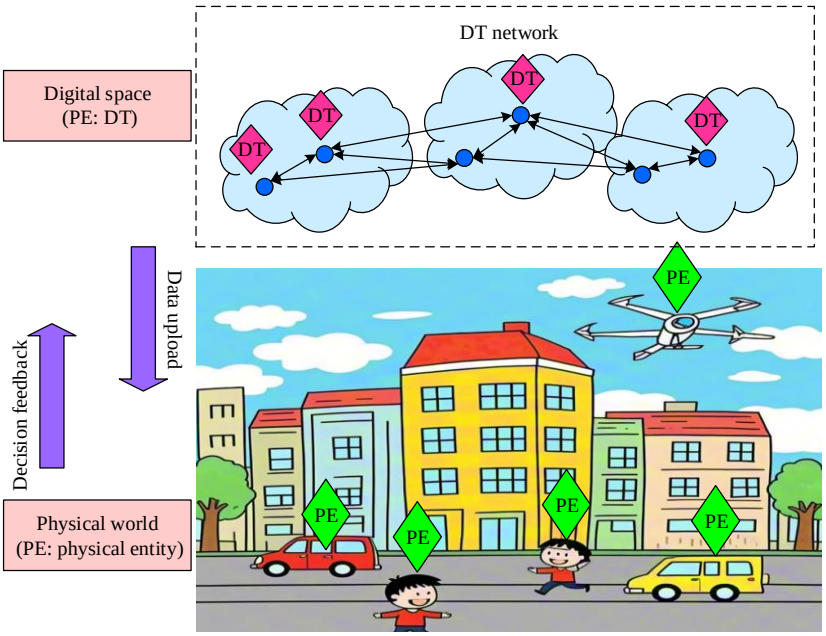


Figure 2: Schematic diagram of DTN

As shown in Figure 2, the DTN includes physical space, data space, and inter entity communication. The DTN collects real-time data of physical entities through IoT technology and transmits it to the digital space for processing and analysis. Data space is the core of DTNs, used to store and process multi-source data from physical space. The bidirectional data flow between physical entities and DTs is used for real-time synchronization of status and feedback optimization. The information sharing and collaboration between DTs break the limitations of

physical space, enabling federated simulation and optimization of complex tasks. Bloom Filter (BF) is an efficient space saving data structure used to determine whether an element belongs to a set. In the cache penetration problem, the BF can store the key existing in the database in advance. When the query request arrives, the BF determines whether the key exists first, so as to avoid invalid database queries. Therefore, the study adopts BF to quickly determine whether data already exists, avoiding duplicate storage and processing of the same data, thereby saving storage space and improving system efficiency. The flowchart illustration of BF is in Figure 3.

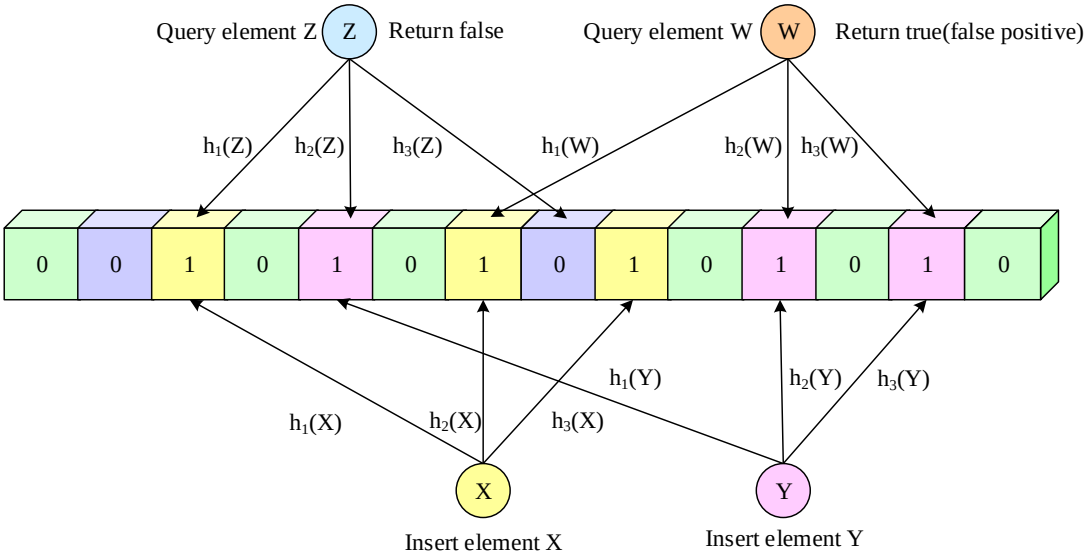


Figure 3: Flowchart illustration of the BF

In Figure 3, when inserting an element x into the BF, each hash function $h_i(x)$ is used to hash element x, and the corresponding bit for each position calculated by the hash function is set to 1. When querying whether an element Z exists in BF, the same hash function $h_i(Z)$ is used to hash element Z, and it is necessary to check whether the bits corresponding to each position calculated by the hash function are all 1. The false positive of BF refers to BF incorrectly determining that an element exists in the set, but in reality, the element is not in the set. False positives are an important characteristic of BF, and their probability of occurrence is closely related to the parameters of BF. Assuming the false positive rate is f, the calculation formula is shown in equation (8).

$$f = (1-1(1-\frac{1}{m})^{n\times k})^k \approx (1-e^{\frac{n\times k}{m}})^k \quad (8)$$

In equation (8), n is the number of elements contained in the set, k is the number of hash functions in the filter, and m is the number of bits. In the IoT environment, BF can be combined with distributed intrusion detection systems to distribute detection functions on edge nodes of the IoT. Each node can independently run BF to quickly detect abnormal behavior in the local network.

2.2 Design of IoT network threat detection system integrating DTN and FL support

The previous section studied the use of blockchain technology and DT technology to optimize the IoT network threat detection performance. However, in the complex IoT ecosystem, device heterogeneity results in significant variations in data formats, processing capabilities, and communication protocols among different devices. This heterogeneity seriously hinders the unified and efficient modeling and management of IoT devices using DT technology. FL not only protects data privacy, but also allows different devices to participate in model training according to their own computing power and data characteristics. Distributed learning methods enable IoT devices to jointly optimize detection models without sharing original data, thus overcoming the obstacles caused by device heterogeneity. Therefore, the introduction of FL provides a more flexible and efficient solution for IoT network threat detection. Unlike traditional centralized learning methods, FL typically requires consolidating all data into a central location for model training. Among them, the expression of the local model parameter α_a is in equation (9).

$$\alpha_a = \arg\min_{\alpha} \beta(\alpha, D_a) \quad (9)$$

In equation (9), D_a represents the local dataset. The expression of the global model parameter α_{new} is shown in equation (10).

$$\alpha_{new} = \frac{\sum_{a=1}^{K} n_a \cdot \alpha_a}{\sum_{a=1}^{K} n_a} \tag{10}$$

In equation (10), n indicates the number of clients participating in FL. FL can be divided into horizontal FL, vertical FL, and federated transfer learning. Among them, horizontal FL is suitable for scenarios where the participating datasets have the same feature space but different sample spaces. Horizontal FL treats data as horizontally partitioned in a tabular view, where each participant has different samples but the feature dimensions of the samples are the same. The schematic diagram of the training process for horizontal FL is shown in Figure 4.

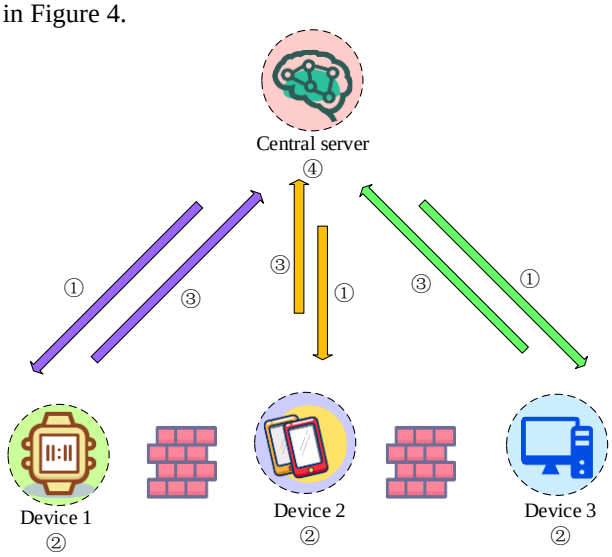


Figure 4: Horizontal federation learning training process diagram

In Figure 4, the central server first initializes a global model parameter, which is sent to all participating parties. After receiving the global model parameters, each participant uses their local dataset to train the model locally. The training process may involve multiple iterations until a predetermined local training standard is reached. After completing local training, each participant will receive a locally trained model parameter. After receiving the local model parameters uploaded by all participants, the central server aggregates these parameters using the federated averaging algorithm. After completing the model aggregation, the central server sends the updated global model parameters back to all participants. After receiving the new global model parameters, each participant updates the parameters based on them. The entire process is repeated until a certain global convergence standard is reached or the preset maximum number of iterations is reached. Support Vector Machines (SVMs) can effectively solve regression and classification problems in high-dimensional feature spaces. The linear SVM model expression is shown in equation (11).

$$\min_{w,b} \frac{1}{2}\|\omega\|^2 \; s.t. y_i(\omega^T x_i + b) \geq 1 \tag{11}$$

In equation (11), ω represents the normal vector of the hyperplane and b represents the intercept. The formula for solving hyperplanes is in equation (12).

$$g(x) = \omega^T x + b \tag{12}$$

The Karush-Kuhn-Tucker condition is a set of necessary conditions in optimization problems used to solve nonlinear programming problems with constraints. The study uses the Karush-Kuhn-Tucker condition b, and the result of the Karush-Kuhn-Tucker conditional function $f(x)$ represents the expression of the classification prediction result, as shown in equation (13).

$$f(x) = sign(\sum_{i=1}^{g} \kappa_i y_i x_i^T x + b) \tag{13}$$

In equation (13), sign is the sign function and x is the input quantity vector. This research adds SVMs on the basis of horizontal FL to construct a horizontal FL support system framework, as shown in Figure 5.

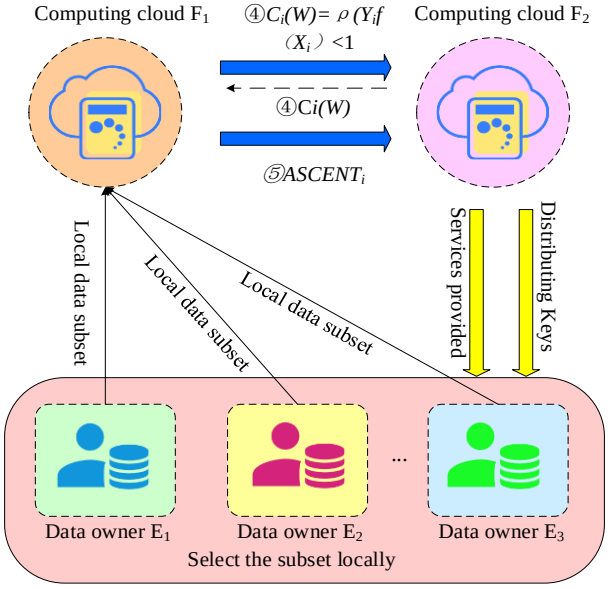


Figure 5: Horizontal FL support system framework

In Figure 5, during the preprocessing stage, the data owner is responsible for preprocessing the local data, collaborating with the computing cloud to execute the subset selection algorithm, encrypting the local training data using the encryption key supplied by the service cloud, and generating a ciphertext dataset. The computing cloud collaborates with data owners and service clouds to complete standardized operations and subset selection algorithms. Encrypted datasets uploaded by data owners are received and stored in a secure environment. The service cloud distributes encryption keys to data owners and collaborates with the computing cloud to complete standardized operations. During the model training and interactive computation phase, the computing cloud conducts model training on a ciphertext dataset, and all computation processes are completed within the ciphertext to ensure data privacy. Meanwhile, it interacts with the service cloud for interactive computation to update and aggregate model parameters. The service cloud interacts with the computing cloud for computation,

providing decryption keys to support ciphertext calculation. The data owner remains offline and does not directly participate in the calculation process. When needed, the data is obtained from the service cloud to obtain the final trained model for local inference or other applications. In the model distribution and service phase, the service cloud obtains the final global model after the model training is completed. The data owner obtains the final model from the service cloud and uses it locally for inference or other applications. In other applications, data owners can utilize the model to analyze the energy consumption of IoT devices. By evaluating the energy utilization efficiency through the trends in energy consumption data output by the model, they can then formulate energy-saving optimization strategies. These strategies may include adjusting equipment operating parameters and scheduling equipment operating times reasonably, thereby achieving the goal of energy conservation and emission reduction. The design diagram of an IoT network threat detection system that integrates DTN and FL support is shown in Figure 6.

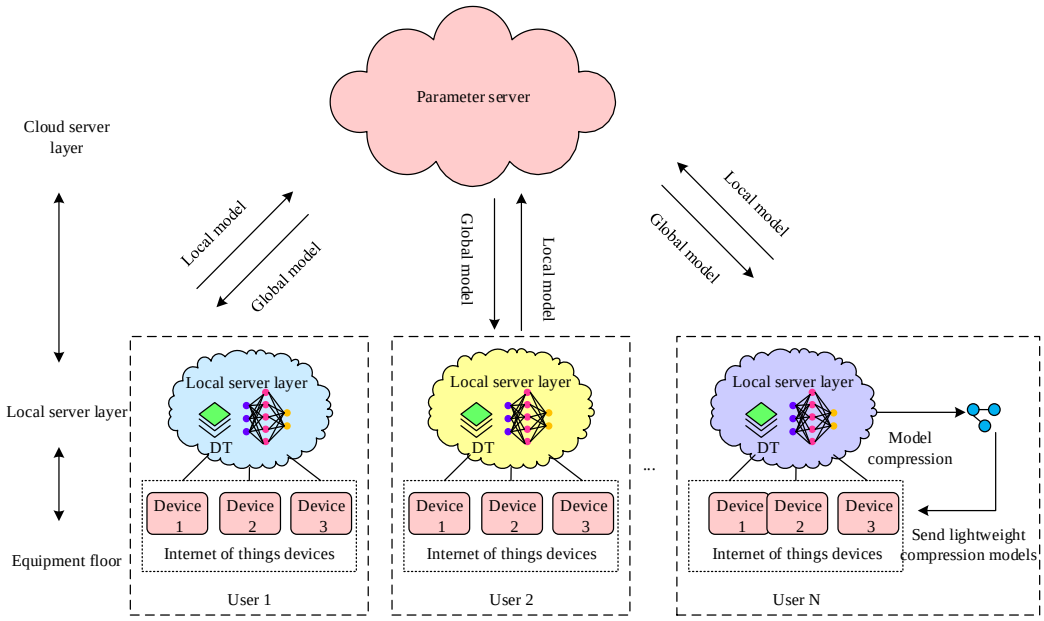


Figure 6: Design of IoT network threat detection system supported by DTN and FL

As shown in Figure 6, intelligent IoT devices collect real-time data and transmit it to local servers through encrypted communication protocols. Then, through the DT system, real-time monitoring and anomaly detection of intelligent IoT devices can be achieved. In each round of FL, the local server utilizes local data and computing resources to train the local model. The local server sends the updated model parameters to the cloud server for aggregation, generating a new global model. The cloud server sends the new global model parameters back to the local server for the next round of training. The main role of the cloud server in the system is the model aggregation center, which receives updated model parameters uploaded from various local servers, performs aggregate calculation based on these parameters, fuses multi-party data features, and generates new global model parameters. The updated global model parameters can comprehensively capture the overall characteristics of the local data involved in FL, enhance the model's generalization ability, and subsequently provide a more representative model foundation for the next round of training on the local server. This, in turn, improves the threat detection performance of the entire system. The local server uses model compression technology to compress personalized models into lightweight models, and deploys the compressed models to intelligent IoT devices to achieve offline threat detection. The DT system and local model jointly monitor the status of intelligent IoT devices and trigger alerts when abnormal behavior is detected. Assuming there are a total of N customers, set as F, each user's IoT device is represented by a symbol D_j, and the corresponding DT generated by the local server is represented by symbol DT_j. The expression for DT_j at time t is shown in equation (14).

$$DT_j(t) = \left\{ Model_j(t), DATA_j(t), Comp_j(t) \right\} \tag{14}$$

In equation (14), $Model_j(t)$ represents device, $Comp_j(t)$ represents available computing resources, and $DATA_j(t)$ represents historical data. To achieve load balancing, training tasks can be decomposed into multiple subtasks and then assigned to different devices. The expression for the computing task $Task_v$ assigned to device v is shown in equation (15).

$$Task_v = \frac{Comp_v(t)}{\sum_{v=1}^{M} Comp_v(t)} Task \tag{15}$$

In equation (15), $Comp_v(t)$ represents the available computing resources of the device v at time t. To

improve the interpretability of threat detection model, XAI module is embedded in FL multi-level model. Each client uses SHapley Additive exPlanations (SHAP) to calculate input features (contribution to local threat classification results), generating feature importance heat maps. The server identifies cross-domain consistency threat features by aggregating the SHAP value distribution of each client through federation. For complex admixture samples, Local Interpretable Model-agnostic Explanations (LIME) was used to reconstruct interpretable alternative models in DT virtual environments. Based on Ubuntu16.04 operating system and using Python and Solidity programming languages, the research used Eclipse Hono to connect IoT devices and collect relevant information, and gathers data to a single AMQP 1.0 endpoint. The Python library that interacts with the Ethereum blockchain through Web3.py enables the deployment and invocation of smart contracts. Solana.py was used to interact with the Python library of Solana blockchain, Paho-MQTT was used to achieve MQTT protocol communication between IoT devices and servers, and InfluxDB was used to store and query time series data. Finally, a network threat detection system with DT and FL support for IoT was constructed.

3 Results

3.1 Algorithm performance testing

To confirm the accuracy and scientificity of the research outcomes, a simulation experimental environment was set up, as shown in Table 2.

Table 2: Simulation experiment environment setting table

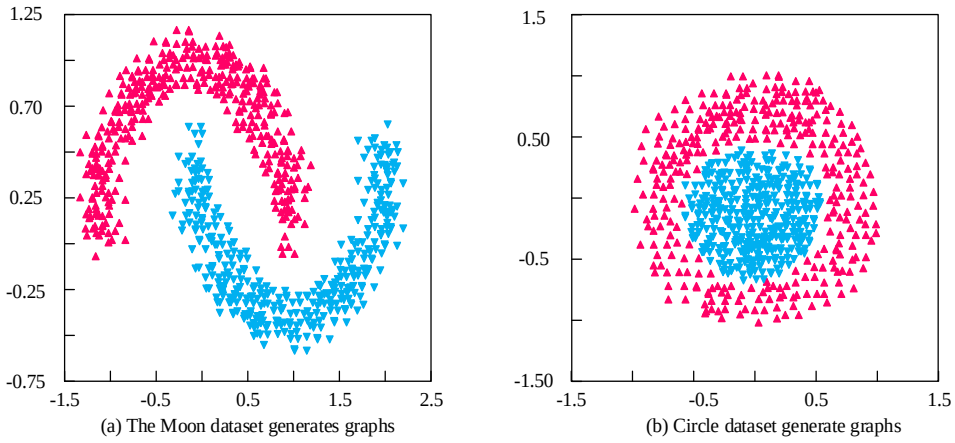
Name	Disposition
Processor	Intel(R) Core(TM) i7 6700H @ 3.40 GHz
Operating system	Windows 10 64
Graphics card	NVIDIA Geforce RTX 3060
Internal memory	16 GB
Hard disk	1 TB

The simulation experiment used two datasets, Moon and Circle. The Moon dataset was usually a two-dimensional synthetic dataset used to demonstrate and test classification algorithms. The data points in this dataset were sampled from two noisy crescent shape data distributions, each containing two features. The Circle dataset referred to a dataset that generates circular distributions, consisting of two sets of points with circular distributions, used to test the algorithm's ability to process complex shaped data. Ring dataset is a multi-modal, multi-version annotated dataset mainly used for target detection tasks, including ring images in various scenes, which is suitable for the research in the field of computer vision and target detection. The Pumpkin Seeds dataset is an agricultural classification dataset that contains 2,500 grayscale and binary images of pumpkin seeds for the classification task. The hyper-parameter settings of the four types of data sets are shown in Table 3.

Table 3: Hyper-parameter setting table

Data set	Lot size	Penalty factor	Object mapping dimension	Learning rate	Gamma value
Moon	16	1.0	100	0.001	2.0
Circle	16	1.0	100	0.001	2.0
Ring	16	1.0	100	0.002	0.1
Pumpkin Seeds	16	1.0	100	0.01	0.1

The generated graphs of the research algorithm on two datasets are shown in Figure 7.

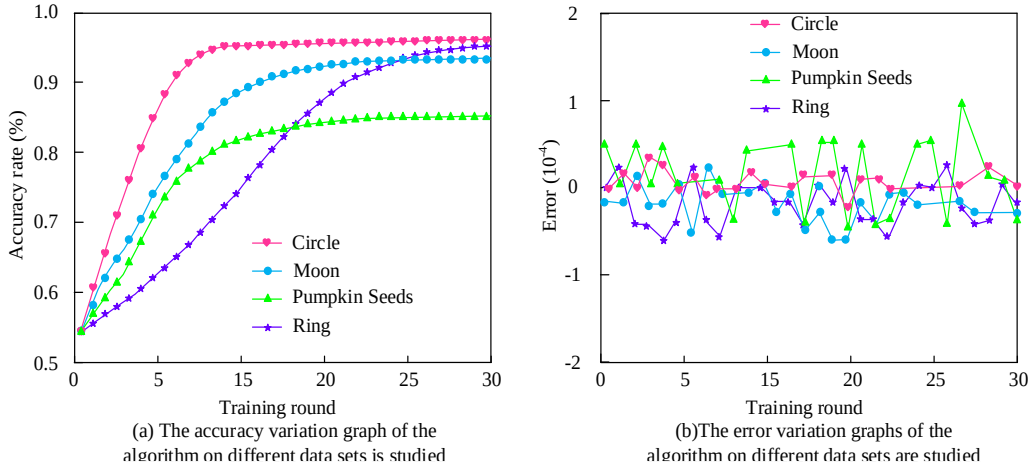


(a) The Moon dataset generates graphs (b) Circle dataset generate graphs

Figure 7: Generated graphs of the research algorithm on two datasets

Figure 7 (a) shows the generated image of the Moon dataset. From Figure 7 (a), on the real dataset Moon, the research algorithm could clearly distinguish the data in the Moon dataset, and ultimately present it as two crescent shapes. Figure 7 (b) shows the generated image of the Circle dataset. From Figure 7 (b), on the real dataset Circle, the research algorithm accurately classified the data points into two circular distributions, presenting clear

circular contours. In the simulation experiment, the data set was divided into training set and test set. Among them, the training set accounted for 80%, which was used for the training of the model. The test set accounted for 20% and was used to evaluate the performance of the model after training. The maximum number of iterations for algorithm training was set to 30. In each iteration, participants updated the locally trained algorithm parameters and interacted with other participants via secure communication protocols. The variation trend of accuracy and error of the algorithm on different data sets is shown in Figure 8.



(a) The accuracy variation graph of the algorithm on different data sets is studied

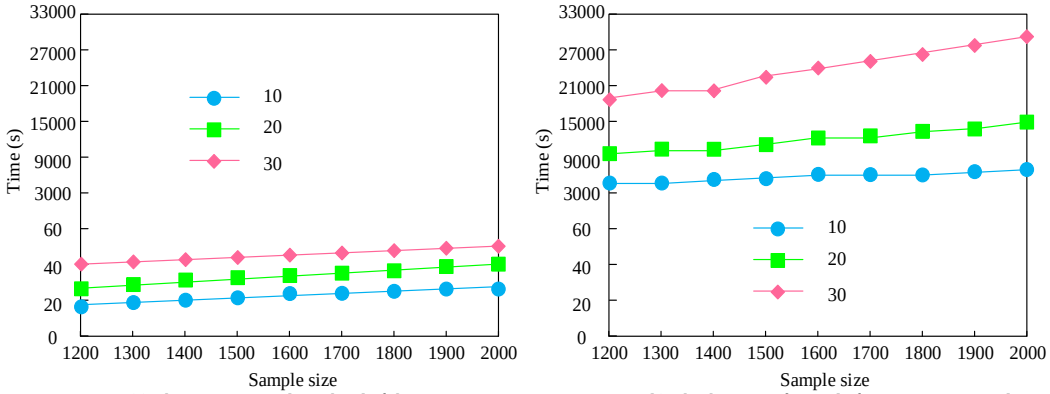
(b)The error variation graphs of the algorithm on different data sets are studied

Figure 8: The trend of accuracy and error of the algorithm on different datasets

Figure 8 (a) indicates the accuracy variation of the research algorithm on various datasets. From Figure 8 (a), the research algorithm had the highest accuracy on Circle dataset, reaching 97.3%, and the lowest accuracy on dataset Pumpkin Seeds, reaching 83.6%. This indicated that even in the face of more complex datasets, the research algorithm could still maintain good data monitoring performance. Figure 8 (b) shows the error variation of the research algorithm on different datasets. From Figure 8 (b), the error of the research algorithm fluctuated up and down within the range of -1×10^{-4} to 1×10^{-4} on the four types of datasets, with a relatively small fluctuation range.

3.2 System performance analysis

To verify the performance advantages of the proposed system in network threat detection, comparative experiments were carried out in the Distributed Denial of Service attack (DDoS) scenario based on the real object networking dataset ToN-IoT. Check Point Infinity platform and Weibu online threat intelligence platform were selected as the baseline system to conduct quantitative analysis from two dimensions of computing cost and response time. The experimental results are shown in Figure 9.



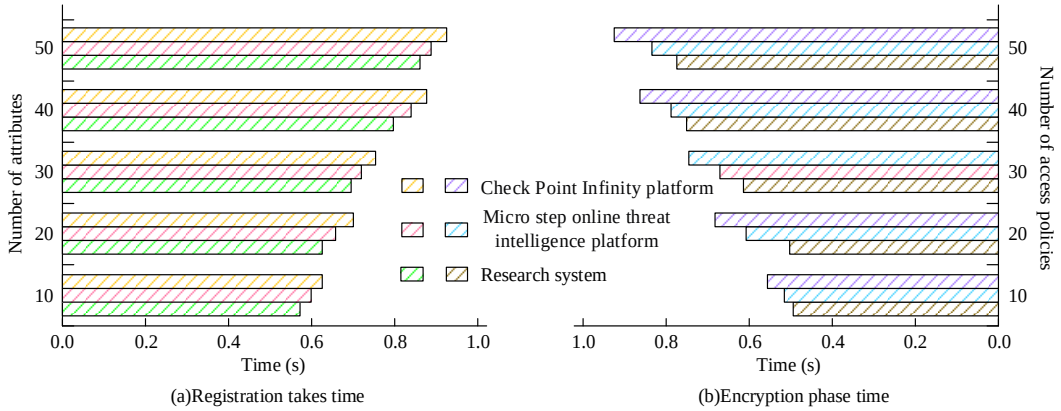
(a) The computational overhead of the system in different dimensions of the data set is studied

(b) Check Point Infinity platform's computational overhead in different dimensions of the dataset

Figure 9: The total computational cost of the system proposed by this study and the Check Point Infinity platform

Figure 9 (a) shows the computational cost of the research system in different dimensions of the dataset. From Figure 9 (a), as the sample size increased, the computational cost time of the research system increased linearly. When the sample size reached 2000, the computational cost time of the research system was 23 seconds in 10 dimensions, 38 seconds in 20 dimensions, and 42 seconds in 30 dimensions. This was because the real-time attack traffic simulation and node isolation capability of the DT module significantly reduced redundant computation. The dynamic client scheduling mechanism of FL module avoided the resource bottleneck

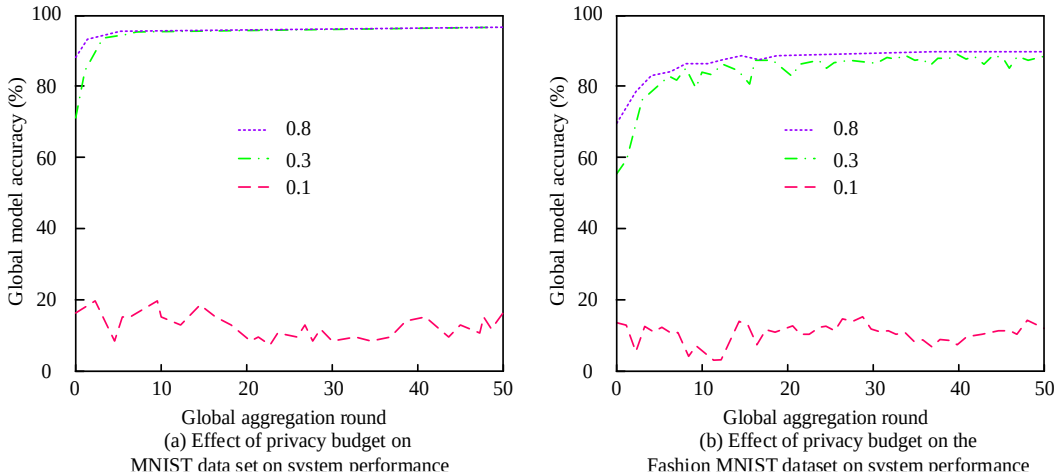
caused by single point dependence. Figure 9 (b) shows the computational cost of the Check Point Infinity platform in different dimensions of the dataset. As shown in Figure 9 (b), when the sample size reached 2000, the computational cost time of the Check Point Infinity platform in the three dimensions was 3280 s, 12580 s, and 30060 s, respectively. This gap stemmed from the baseline platform's lack of virtualized attack rehearsal and distributed collaborative learning capabilities, resulting in inefficient processing of high-dimensional attack data. In the system running architecture, the registration process allowed the system to verify the identity of users or devices and assign corresponding permissions to them. Authorized users or devices could access system resources. This built a solid defense line at the access control level, greatly enhancing the security and stability of the system, and effectively resisting potential illegal access and malicious attacks. The study also introduced a micro step online threat intelligence platform for comparative experiments. The experiment outcomes are in Figure 10.



(a)Registration takes time (b)Encryption phase time

Figure 10: Comparison of the efficiency of different IoT network threat detection systems

Figure 10 (a) shows the time spent on registration. According to Figure 10 (a), when the number of attributes was 10, the registration time of the research system was about 0.57 seconds. When the number of attributes increased to 50, the time taken to research system registration increased by only about 0.31 seconds. Figure 10 (b) shows the time diagram of the encryption phase. From Figure 10 (b), the increase of policies would lead to the increase of system encryption time. When the number of access policies was 10, the shortest encryption time of the research system was 0.52 seconds. To analyze the impact of privacy budget on the performance of the network threat detection system model supported by DT and FL for the IoT, two datasets of MNIST and Fashion MNIST were respectively used for analysis. The experimental results are shown in Figure 11.



(a) Effect of privacy budget on MNIST data set on system performance

(b) Effect of privacy budget on the Fashion MNIST dataset on system performance

Figure 11. Impact of privacy budget on system performance on different data sets

Figure 11 (a) shows the impact of privacy budget on system performance on the MNIST dataset. From Figure 11 (a), as the privacy budget rose, the accuracy of the research system also improved. When the privacy data was 0.8, the accuracy of the research system was 92%, indicating that the system could still maintain excellent detection performance at higher levels of privacy protection. Figure 11 (b) shows the impact of privacy budget on system performance on the Fashion MNIST dataset. From Figure 11 (b), although the accuracy of the research system decreased under different privacy budgets in this dataset, overall, the accuracy of the research system

was around 88%, indicating that even with an increase in privacy protection requirements, the network threat detection system still exhibited good detection ability and stability, verifying its practicality and effectiveness in the IoT environment. To comprehensively verify the effectiveness of the proposed method, ablation experiments were conducted on a large data set CIFAR-10, and the experimental results were shown in Table 4.

Table 4 Ablation experiment table

Model setup	Accuracy rate (%)	Error rate (%)	Computational overhead (seconds)	Privacy budget
Basic model	65.7	30.0	20	/
DT module	78.5	25.6	22	/
FL module	72.2	27.1	21	1.2
Research system	82.3	20.2	24	0.8

From Table 4, after the introduction of DT module, the accuracy rate increased from 65.7% to 78.5%, indicating that DT technology could significantly improve the classification performance of the model. After the introduction of FL module, the accuracy rate increased from 65.7% to 72.2%, indicating that FL technology could improve the classification performance of the model. The accuracy of introducing DT and FL module was 82.3%, which was significantly higher than other settings, indicating that the combination of DT and FL could further improve the classification performance of the model.

4 Discussion and conclusion

4.1 Discussion

Given the swift advancement of the IoT, network threat detection becomes a key task in ensuring the security of the IoT. Traditional network threat detection systems often face problems such as high computational overhead, insufficient privacy protection, and limited ability to process complex data. To address these issues, the research proposes an IoT network threat detection system that integrates DTN and FL support.

The research first tested the IoT network threat detection algorithm that integrated DTN and FL support. The results showed that the research algorithm could clearly distinguish sub-nodes and present clear shape contours on the Moon and Circle datasets. On the Circle dataset, the accuracy of the system was as high as 97.3%, and on the more complex Pumpkin Seeds dataset, the accuracy reached 83.6%. This indicated that the research algorithm could maintain good classification performance when facing data of different complexities, which is consistent with the results obtained by Choi W et al. in their DT research in the power generation industry [17]. Then, the performance of the IoT network threat detection system that integrated DTN and FL support was evaluated. The experiment outcome indicated that under the same testing environment and dataset dimensions, the computational cost time of the research system was much lower than that of the Check Point Infinity platform. This was because the research system adopted DT technology, which reduced redundant computing and significantly lowers computational overhead [18]. The study introduced a micro step online threat intelligence platform for comparative experiments, and the experimental outcomes indicated that the study could cope with the dynamic changes in the number of attributes in the IoT environment. This was because the research system adopted a lightweight registration mechanism, which significantly reduced registration time by optimizing the registration process and reducing unnecessary calculation steps. This is consistent with the results of Landrum et al. in a study of a lightweight chemical registration and data storage system [19]. Finally, experiments were conducted based on two datasets, MNIST and Fashion MNIST, respectively. The experimental results showed that under the condition of privacy protection, the detection performance of the research system was somewhat reduced, but it could still maintain excellent detection performance. It is consistent with the research of Yuan et al. on knowledge sharing of vehicle Internet privacy protection based on low-cost federal generalized learning [20]. This was mainly due to the synergy of horizontal FL architecture and DT technology, which effectively guaranteed data privacy by allowing data to be directly processed and model trained on local devices, while significantly reducing data transfer costs. DT technology further improved the detection accuracy and response speed of the system by modeling physical devices and synchronizing real-time data. This combination not only ensured high performance of the model, but also provided a more efficient and secure solution for cyber threat detection in IoT environments.

On the large data set CIFAR-10, the computational overhead of the research system was 24 seconds, which was slightly increased compared with the basic model, but still within the acceptable range, indicating that the system had good scalability when processing large-scale data sets. Although the introduction of DTs and FL modules increased the computational overhead, this increase was linear and did not lead to an exponential increase in computational time, indicating that the system could adapt to the increase in data volume. The research system showed the highest accuracy and the lowest error rate on large data sets, indicating that the combination of DT and FL could effectively improve the classification performance of the model. Even in the case of a large amount of data, the performance improvement indicated that the system could maintain a high detection accuracy and had good scalability when processing large-scale data sets. The privacy budget of the research system was 0.8, indicating that a good balance was achieved between privacy protection and performance. Even on large data sets, the system could provide effective privacy protection

while maintaining high detection performance. The system had good scalability in privacy protection and could adapt to scenarios with different privacy requirements.

In summary, the research system adopted an FL architecture, DT technology, privacy budget mechanism, and efficient encryption technology. These technologies and strategies protected privacy while minimizing the impact on detection performance, ensuring that the system had efficient, secure, and reliable threat detection capabilities in the IoT environment.

4.2 Conclusion

In response to the challenges posed by IoT network threats, an innovative IoT network threat detection system that integrated DTNs and FL support was proposed. Performance testing was conducted on the research algorithm, and in regard to classification performance, the accuracy of the research algorithm on the Circle dataset was as high as 97.3%. Even on the more challenging Pumpkin Seeds dataset, the accuracy reached 83.6%, indicating that the research algorithm could still maintain good monitoring performance when processing complex datasets. In addition, the error fluctuation range of the research algorithm on the four types of datasets was extremely small, ranging from -1×10^{-4} to 1×10^{-4}, further demonstrating the stability and reliability of its performance. In terms of privacy protection, as the privacy budget increased, the accuracy of the system substantially improved. When the privacy budget was 0.8, the system accuracy reached 92%, indicating that at a higher level of privacy protection, the system could not only effectively protect data privacy but also maintain excellent detection performance. In summary, the system proposed by the research showed excellent performance in classification performance, computational efficiency, and privacy protection, and could effectively respond to complex and changing network threats in the IoT environment, which had important practical application value. In the IoT environment, real-time data processing is very important. In the future, edge computing and real-time data analysis technology can be combined to further improve the real-time performance and response speed of the system.

Funding

This study was supported by the Science and Technology Planning Project of Henan Province, China (Grant No. 242102320167), and the research project “Design and Implementation of Real Time State Monitoring System for Distribution BoxBased on AIoT” of Henan Polytechnic in 2024 (2024ZK29).

References

[1] Sonthitham, P., Sonthitham, A., & Khuantham, C. (2021, May). Development of internet of things technology for control system mulberry smart farm: A case study at Surin Province. In 2021 18th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 989-992. DOI:10.1109/ECTI-CON51831.2021.9454873.

[2] Sun J, Xu Q H, Yu F C, Zhu J, Zhang RX. (2020). Fault detection of sewage treatment hydraulic system based on dynamic GRNN model with internet of things technology. Chinese Hydraulics & Pneumatics, (11), 120-126. DOI:10.11832/j.issn.1000-4858.2020.11.019.

[3] Li, J., Zhi, J., Hu, W., Wang, L., & Yang, A. (2020). Research on the improvement of vision target tracking algorithm for Internet of things technology and Simple extended application in pellet ore phase. Future Generation Computer Systems, 110, 233-242. DOI:10.1016/j.future.2020.04.014.

[4] Sattar, K. A., & Baroudi, U. (2024). Performance evaluation of cluster-based federated machine learning. Neural Computing and Applications, 36(14), 7657-7668. DOI:10.1007/s00521-024-09487-3.

[5] Wen, J., Zhang, Z., Lan, Y., Cui, Z., Cai, J., & Zhang, W. (2023). A survey on federated learning: challenges and applications. International Journal of Machine Learning and Cybernetics, 14(2), 513-535. DOI:10.1007/s13042-022-01647-y.

[6] Ma, H. (2024). Development of a smart tourism service system based on the Internet of Things and machine learning. The Journal of Supercomputing, 80(5), 6725-6745. DOI:10.1007/s11227-023-05719-w.

[7] Zhang, Z. (2024). SD-WSN Network Security Detection Methods for Online Network Education. Informatica, 48(21), 51-66. DOI:10.31449/inf.v48i21.6257.

[8] Alcaraz, C., Meskini, I. H., & Lopez, J. (2025). Digital twin communities: an approach for secure DT data sharing. International Journal of Information Security, 24(1), 1-19. DOI:10.1007/s10207-024-00912-1.

[9] Zhao, Y., Li, L., Liu, Y., Fan, Y., & Lin, K. Y. (2022). Communication-efficient federated learning for digital twin systems of industrial Internet of Things. IFAC-PapersOnLine, 55(2), 433-438. DOI:10.1016/j.ifacol.2022.04.232.

[10] Sasikumar, A., Vairavasundaram, S., Kotecha, K., Indragandhi, V., Ravi, L., Selvachandran, G., & Abraham, A. (2023). Blockchain-based trust mechanism for digital twin empowered Industrial Internet of Things. Future Generation Computer Systems, 141, 16-27. DOI:10.1016/j.future.2022.11.002.

[11] Zheng, Q., Wang, J., Shen, Y., Ding, P., & Cheriet, M. (2022). Blockchain based trustworthy digital twin in the Internet of Things. In 2022 International Conference on Information Processing and Network Provisioning, 152-155. DOI:10.1109/ICIPNP57450.2022.00040.

[12] Aryavalli, S. N. G., & Kumar, G. H. (2024). Futuristic vigilance: Empowering chipko movement with cyber-savvy IoT to safeguard forests. Archives of

Advanced Engineering Science, 2(4), 215-223. DOI:10.47852/bonviewAAES32021480.

[13] Zhao, W., Zhang, S., Xue, L., Chang, T., & Wang, L. (2024). Research on model of micro-grid green power transaction based on blockchain technology and double auction mechanism. Journal of Electrical Engineering & Technology, 19(1), 133-145. DOI:10.1007/s42835-023-01541-9

[14] Alturki, N., Alharthi, R., Umer, M., Saidani, O., Alshardan, A., Alhebshi, R. M., et al. (2024). Efficient and Secure IoT Based Smart Home Automation UsingMulti-Model Learning and Blockchain Technology. CMES-Computer Modeling in Engineering & Sciences, 139(3). 3387-3415. DOI:10.32604/cmes.2023.044700.

[15] Ariansyah, D., Isnan, M., Rahutomo, R., & Pardamean, B. (2023). Digital Twin (DT) smart city for air quality management. Procedia Computer Science, 227, 524-533. DOI: 10.1016/j.procs.2023.10.554.

[16] Hakimi, O., Liu, H., & Abudayyeh, O. (2024). Digital twin-enabled smart facility management: A bibliometric review. Frontiers of Engineering Management, 11(1), 32-49. DOI:10.1007/s42524-023-0254-4.

[17] Choi, W., Hudachek, K., Koskey, S., Perullo, C., & Noble, D. (2024). Digital twin in the power generation industry. JMST Advances, 6(1), 103-119. DOI:10.1007/s42791-024-00065-1.

[18] Tellechea-Luzardo, J., Winterhalter, C., Widera, P., Kozyra, J., de Lorenzo, V., & Krasnogor, N. (2020). Linking engineered cells to their digital twins: a version control system for strain engineering. ACS Synthetic Biology, 9(3), 536-545. DOI:10.1101/786111.

[19] Harahsheh, K. M., & Chen, C. H. (2023). A survey of using machine learning in IoT security and the challenges faced by researchers. Informatica, 47(6), 1-54. DOI:10.31449/inf.v47i6.4635.

[20] Yuan, X., Chen, J., Zhang, N., Ye, Q. J., Li, C., Zhu, C., & Shen, X. S. (2024). Low-cost federated broad learning for privacy-preserved knowledge sharing in the RIS-aided internet of vehicles. Engineering, 33, 178-189. DOI:10.1016/j.eng.2023.04.015.