Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.066498

ARTICLE

Privacy Preserving Federated Anomaly Detection in IoT Edge Computing Using Bayesian Game Reinforcement Learning

Fatima Asiri[1], Wajdan Al Malwi[1], Fahad Masood[2], Mohammed S. Alshehri[3], Tamara Zhukabayeva[4], Syed Aziz Shah[5] and Jawad Ahmad[6,*]

[1]Department of Informatics and Computer Systems, College of Computer Science, King Khalid University, Abha, 61421, Saudi Arabia
[2]Department of Computing, Abasyn University Peshawar, Peshawar, 25000, Pakistan
[3]Department of Computer Science, College of Computer Science and Information Systems, Najran University, Najran, 61441, Saudi Arabia
[4]Department of Information Systems, L.N. Gumilyov Eurasian National University, Astana, 010000, Kazakhstan
[5]Research Centre for Intelligent Healthcare, Coventry University, Coventry, CV1 5RW, UK
[6]Cybersecurity Center, Prince Mohammad Bin Fahd University, Alkhobar, 31952, Saudi Arabia

*Corresponding Author: Jawad Ahmad. Email: jahmad@pmu.edu.sa

Received: 10 April 2025; Accepted: 10 June 2025; Published: 03 July 2025

ABSTRACT: Edge computing (EC) combined with the Internet of Things (IoT) provides a scalable and efficient solution for smart homes. The rapid proliferation of IoT devices poses real-time data processing and security challenges. EC has become a transformative paradigm for addressing these challenges, particularly in intrusion detection and anomaly mitigation. The widespread connectivity of IoT edge networks has exposed them to various security threats, necessitating robust strategies to detect malicious activities. This research presents a privacy-preserving federated anomaly detection framework combined with Bayesian game theory (BGT) and double deep Q-learning (DDQL). The proposed framework integrates BGT to model attacker and defender interactions for dynamic threat level adaptation and resource availability. It also models a strategic layout between attackers and defenders that takes into account uncertainty. DDQL is incorporated to optimize decision-making and aids in learning optimal defense policies at the edge, thereby ensuring policy and decision optimization. Federated learning (FL) enables decentralized and unshared anomaly detection for sensitive data between devices. Data collection has been performed from various sensors in a real-time EC-IoT network to identify irregularities that occurred due to different attacks. The results reveal that the proposed model achieves high detection accuracy of up to 98% while maintaining low resource consumption. This study demonstrates the synergy between game theory and FL to strengthen anomaly detection in EC-IoT networks.

KEYWORDS: IoT; edge computing; smart homes; anomaly detection; Bayesian game theory; reinforcement learning

1 Introduction

The Internet of Things (IoT) has experienced rapid expansion in various applications, including consumer electronics, agriculture, transportation systems, industry, and healthcare, which aim to enhance comfort and improve the human lifestyle. The IoT systems interconnect smart nodes that exchange data through the internet or a private network. Edge computing (EC) has emerged as a novel paradigm shift in data computation and storage, bringing data near to the end user, leading to the development of IoT edge computing. The distribution of computing nodes across the network minimizes the computational load on the centralized data center and significantly decreases the latency in data exchange [1].

In a smart home environment, EC-IoT can be utilized to control appliances over the Internet remotely. The immense development of IoT devices and the massive data traffic generation in edge networks create additional traffic load due to resource and bandwidth shortages. Although this paradigm offers exceptional features and improved Quality of Service (QoS), it also presents massive privacy and data security risks [2]. Nearly 80% of IoT devices are wide open to possible cyberattacks, emphasizing their noteworthy exposure to security breaches. Hackers may gain unauthorized access to private information and control appliances in smart homes due to security vulnerabilities in IoT devices, highlighting a major issue for IoT expansion [3].

An intrusion detection system (IDS) can monitor network traffic for IoT networks to detect anomalies and intrusions [4,5]. Several simulation-based studies have been conducted to develop IDS frameworks using available datasets [6–8]. The computational resources available to IoT devices are very limited, so hacked IoT devices are undetected in most cases. Malicious devices are only detected when they are not functioning, causing several security and privacy issues in IoT networks [9]. Security methods, including IDS, authentication, and access control, cannot provide complete security systems in EC-IoT networks; therefore, a detailed framework is needed that not only detects anomalies, but also provides the necessary security measures [10,11].

There has been increasing interest in using the latest Artificial Intelligence (AI) and Machine Learning (ML) methods for anomaly detection solutions in IoT edge computing. The deep learning (DL) framework offers an advanced framework for various application scenarios, enabling the discovery of unexpected activities and the design of robust anomaly detection models. The operation of Deep Neural Networks (DNNs) is based on a multilayer hierarchical structure, where each subsequent layer is capable of creating more advanced feature sets than the previous layer. As a result, these frameworks are effective for data visualization and representation.

Existing ML/DL models for anomaly detection are most effective at achieving high accuracy, but they face challenges in complex edge computing environments. Federated learning (FL) and Bayesian game theory are optimistic and supportive technologies that utilize edge resources to allow privacy-aware cooperative learning in edge IoT networks [12]. To address this issue, FL has been utilized to analyse sensor data locally for various anomalies. An anomaly score is computed for each device, and its participation in FL is determined using predefined thresholds. The utilities of attackers and defenders are evaluated using Bayesian game theory, and the double deep Q-learning (DDQL) dynamically adjusts the detection policies to provide an effective response to threats.

In this article, Section 2 provides a brief review of the literature, while a detailed methodology for the proposed framework is presented in Section 3. In Section 4, the results are discussed in detail, and the conclusions and future recommendations are given in Section 5.

2 Literature Review

Anomaly detection is crucial for securing IoT edge networks in smart homes, agriculture, and healthcare. Limited memory size, resource deficiency, and computing power are the major challenges for effective and secure communication in edge networks. Anomaly detection has been performed in IoT systems using machine learning (ML), deep learning (DL), and federated learning (FL) models, which have produced effective performance results [13,14]. These models have been widely used to reduce false alarm rates using local functions and provide easy computational resources [15,16].

Multiple IoT objects are a dynamic strategy used in smart homes with edge processing to deal with adverse situations. A stochastic game network (SGN) is another approach for handling anomaly detection in IoT edge networks, where IoT devices act as players with a set of predefined actions. The IoT edge and

SGN have been integrated to create a complete smart home environment, enhancing scalable and operable solutions. Anomaly detection in the industrial Internet of Things (IIoT) environment is also a challenging task, as IoT devices are exposed to threats to sensitive data [17,18]. Wang et al developed a reliable FL framework for IIoT networks with local model training using deep RL.

Denial of service (DoS), distributed denial of service (DDoS), and Web attacks are various cyberattacks that cause remote intrusion in an edge environment. Attackers can target other systems by entering the system, exploiting, and propagating malicious alerts through the network [19,20]. Intrusion detection systems (IDS) currently using ML/DL models for anomaly detection are best at achieving high accuracies, but faces problems in a complex edge computing system. Recently, a hybrid model has been presented using the chi-square test and Ig-Chi to improve the accuracy of the detection process in complex IoT networks [21]. The architecture was designed in a simpler fashion that can be used and deployed by the car manufacturers. The model was tested on real-time data and performed well in critical situations.

An innovative hierarchical adversarial attack (HAA) method was presented to comprehend the black-box confrontational attack scheme, aiming at the graph neural network (GNN) [22]. An intelligent approach, built on a saliency map method, was considered to produce adversarial samples by efficiently classifying and adjusting features with minimal perturbations. Guo et al. presented an energy-efficient model for time series data [23]. A Subgraph Generation Algorithm (SGA) was utilized to explore the correlations between sensor data. Anomaly detection was performed using a computational light strategy, and dependency graphs were generated. An adaptive ensemble random fuzzy (AERF) algorithm is suggested to detect anomalies in a cloud-based system [24]. The AERF selects samples randomly to enhance the range of base classifiers, enabling efficient handling of disorders caused by irregular sample distribution.

Anomaly detection for real-time surveillance using DNN was performed, focusing on multi-target detection [25]. A-YONet, a combination of YOLO and MTCNN systems, was utilized and deployed in an edge-cloud environment. Two real-time datasets, namely public and home-based, were utilized for the experiments and validation. Anomaly detection for both energy efficiency and privacy preservation was performed in a smart cyber system [26]. Privacy was preserved by covering the abnormal activities of participants while still accomplishing an energy-efficient system for data upload by presenting a suitable amount of additional content. A novel framework for both risk management and anomaly detection was proposed, utilizing edge computing and machine learning models [27]. A practical safety method was employed to confirm the well-being of seafarers, protect vessels, and maintain a secure maritime environment.

Solutions for anomaly detection have been proposed for IoT edge computing environments. This includes utilizing MCUs and TinyML in the Internet of Things, employing various techniques, and utilizing hardware- or software-based edge computing environments [28–32]. A dual defense self-balanced framework with federated learning has been proposed for a lightweight defect procedure in [33]. Various privacy and learning rates are applied during the model aggregation stage to oppose attacks. In [34], the FedShufde framework was proposed to defend user privacy in edge IoT networks. The attackers were prevented from accessing the user's private information, including UAV flight conditions, location, and address. It is worth mentioning that significant improvements have been achieved in solving these complex problems. Data and network security were enhanced, with a focus on practical applications and consistent, decentralized trust procedures. Table 1 presents a comparison of various deep learning (DL) models for anomaly detection in IoT environments. The proposed framework has also been mentioned for reference.

Table 1: Literature review summary

Year	Reference	Algorithm/model	Findings
2021	[24]	Hierarchical federated learning	Anomaly detection strategy for IIoT, achieved high throughput, high anomaly detection accuracy, and low latency
2022	[18]	Federated learning game	Multiaccess edge computing for IIoT, improved accuracy against attacks
2023	[11]	Transformer-based model	IDS for IoT-based smart home, 97.95% accuracy for binary classification and 95.78% for multiple classifications
2023	[13]	Transformer neural network	99% accuracy for intrusion detection in MQTT- enabled IoT networks
2024	[9]	Deep learning approach	IDS for IoT networks, 99.89% detection accuracy in binary classification tasks
2024	[23]	Cuckoo search-optimized deep CNN	Phishing attack detection in IoT ecosystem, 91% detection accuracy
2024	[28]	Deep forest	Implemented layered intrusion detection model in IoT consumer electronics, achieved better accuracy results.
2025	[10]	Five-dimensional gray wolf optimizer for generative adversarial network	Implemented the model on three datasets and achieved accuracy levels, from 94% to 100%
2025	Our work	Bayesian game federated reinforcement learning	Anomaly detection in IoT edge computing, achieved upto 98% detection accuracy

3 Methodology

The proposed framework primarily focuses on privacy-preserving federated anomaly detection (PPFAD) for various attack types, including Denial-of-Service (DoS), Man-in-the-Middle (MitM), IP spoofing, and Brute Force Attacks in the EC-IoT environment. A comprehensive, real-time dataset has been collected using sensor readings from multiple locations. The missing values were handled using preprocessing methods, and data was partitioned for Federated Learning (FL). The proposed framework utilizes Federated Learning (FL), Bayesian Game Theory (BG), and Reinforcement Learning (RL) for privacy-preserving anomaly detection. FL supports the distributed training of device-specific data for local models and aggregation with global models. Participation is controlled in each training session through an adjusted threshold of anomaly values, ensuring efficient resource utilization. BG helps both attackers and defenders achieve optimal defense policies. Double Deep Q-Network (DDQN) optimizes anomaly detection by learning from different attack patterns. This integrated approach is helpful for efficient anomaly detection and addressing the key challenges of consumer electronics in IoT edge computing. This section presents the detailed methodology for the proposed framework.

3.1 Data Collection

The dataset collection for this research was conducted in a residential environment focused on a sensor-based IoT edge network. Real-time data monitoring has been implemented, which includes features for detecting anomalies. The data collected over 24 h captured various operational situations, including both

normal and abnormal activities. The distributed sensor setup confirms that the data from each sensor node remains localized, supporting the FL framework. The selected features are detailed in Table 2, which covers key metrics such as packet size, flow duration, and flow rate, demonstrating the presence of the attack. The selected features capture key network characteristics for attack identification. Features such as packet size, flow duration, and source/destination IP addresses were extracted in real-time using lightweight packet capture tools. Flow rate and connection count were computed using session monitoring. The reputation score was derived from an average of past anomaly scores, while the Trust level was computed using a weighted combination of anomaly and reputation scores. BG and DDQL integration for the privacy-preserving strategy enables the cooperative development of an anomaly detection framework for accurately identifying threats in an IoT edge network. The data that support the findings of this study are openly available in Dataset.

Table 2: Feature list for model implementation

Sr. No.	Features
1	Packet size
2	Flow duration (sec)
3	Flow rate
4	Source IP
5	Destination IP
6	Source port
7	Destination port
8	Connection count
9	Anomaly score
10	Reputation score
11	Trust level

3.2 Federated Learning in PPFAD

Federated Learning (FL) is a decentralized method where multiple local nodes directly train the model and only share model updates with the central server, which is critical for data security. It is important for an IoT edge environment where sensor data is sensitive and cannot be shared openly. The IoT devices, such as sensor nodes in an edge computing environment generate data where FL trains local models without sharing the sensitive data.

Each device node selects its local dataset and participates by training a local model that may comprise attack patterns. The central server receives the model updates from the participating devices, combines them, and creates a global model. The aggregation is performed using Federated Averaging (FedAvg), ensuring that each model update is contributed to the global model. The proposed algorithm for this process has been presented in Algorithm 1. Mathematically, the process can be expressed as

$$\theta^{t+1} = \frac{1}{N} \sum_{i=1}^{N} \theta_i^t \tag{1}$$

where θ_i^t represents the model parameters from device i at time t, N is the total number of participating devices in the federated round, and $\theta_i^{(t+1)}$ shows the global model after aggregation. The weight aggregation can be expressed as

$$W_g = \sum_{i=1}^{N} \frac{n_i}{n} W_i \tag{2}$$

where W_g shows the global model weight, W_i represents the i_{th} edge node weight, and n_i is the number of data samples at the i_{th} node. The model is trained using input data, a loss function, and target data as

$$\min_{W_i} \mathcal{L}(W_i, X_i, Y_i) \tag{3}$$

The adaptive federation is implemented on each device based on the severity of the anomalies. Devices with high anomaly scores or a history of frequent attacks are prioritized for participation in the FL round. This enables the global model to respond quickly to changing attack patterns and enhance anomaly detection accuracy.

$$\text{Participation} = \begin{cases} 1 & \text{if anomaly score} \geq \tau \\ 0 & \text{if anomaly score} < \tau \end{cases} \tag{4}$$

where τ indicates the anomaly score of the threshold, 1 and 0 is the device participation for the next federated round. The framework aims to minimize the false positive (FP) and false negative (FN) rates and the optimization can represented by a loss function $L(\theta)$ in terms of weights λ_{FP} and λ_{FN} as

$$L(\theta) = \lambda_{FP} \cdot FP + \lambda_{FN} \cdot FN \tag{5}$$

Algorithm 1: Federated learning for anomaly detection

Input: Global model parameters θ, number of devices N, dataset D_i for each device i
Output: Updated global model θ
for each federated round $t = 1, 2, \ldots$ **do**
  Randomly select a subset of devices $\mathcal{S} \subseteq \{1, 2, \ldots, N\}$ with sufficient resources
  for each device $i \in \mathcal{S}$ in parallel **do**
    Download global model θ_t
    Train local model using local dataset D_i:
$\theta_i^{t+1} \leftarrow \theta_t - \eta \nabla L(D_i; \theta_t)$
    Send updated local model θ_i^{t+1} to the server
  end for
  Aggregate local models to update global model:
$\theta_{t+1} \leftarrow \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \theta_i^{t+1}$
end for
Return: Final global model θ

Privacy is further enhanced by incorporating differential privacy into the federated learning (FL) framework. Sensitive information can be protected by incorporating noise into the model updates. The differential privacy can be expressed as

$$D_{\text{noisy}} = D_{\text{true}} + N(\mu, \sigma^2) \tag{6}$$

where D_{noisy} shows the noisy model update, and $N(\mu, \sigma^2)$ is the added noise to the model with variance σ^2, and mean μ, respectively.

3.3 Bayesian Game Theory in PPFAD

In a Bayesian Game, players have insufficient information about the actions, types, and strategies. Algorithm 2 shows the detailed scenario for implementing the Bayesian game theory. Let t_D present the defender's type, having the network and the intrusion detection mechanism information, and t_A be the attacker's type, having the information of the attacker's nature. The player i (defender or attacker) has a belief function b_i that shows the probability distribution $b_D(t_A)$ assigned by the defender about the attacker type given as

$$b_D(t_A) = P(t_A|x) \tag{7}$$

where $b_D(t_A)$ represents the attacker's type belief, and x is the observed outcomes regarding attack attempts or network traffic. The defender's expected utility U_D is the payoff's weighted average received under various strategies for taking action a_D in state s is given as

$$U_D(a_D, t_A) = \sum_{t_A \in T_A} P(t_A \mid a_D)\Big[\cdot R_D(a_D, t_A) \cdot C_D(a_D, t_A)\Big] \tag{8}$$

where t_A shows the set of possible attacker types, $P(t_A)$ is the probability of attacker type (t_A), $R_D(a_D, t_A)$ and $C_D(a_D, t_A)$ are the defender's reward and cost for action a_D. Similarly, the attacker's expected utility can be expressed as

$$U_A(a_A, t_D) = \sum_{t_D \in T_D} P(t_D \mid a_A)\Big[R_A(a_A, t_D) - D_A(a_A, t_D)\Big] \tag{9}$$

The temporal utility with a discount factor can be expressed as

$$U_D^t(a_D, t_A) = \sum_{t=1}^{\infty} \beta^t \left(\sum_{t_A \in T_A} P(t_A \mid a_D^t) \cdot R_D(a_D^t, t_A^t) - C_D(a_D^t, t_A^t) \right) \tag{10}$$

The expected utility can be maximized using the Bayesian Nash equilibrium, and the optimized defender's equilibrium condition can be expressed as

$$a_D^*, a_A^* = \arg\max_{a_D, a_A} \sum_{t_A \in T_A} \sum_{t_D \in T_D} P(t_A, t_D) \cdot \Big[R_D(a_D, t_A) + R_A(a_A, t_D)\Big] - \Big[C_D(a_D, t_A) + C_A(a_A, t_D)\Big] \tag{11}$$

The multi-objective Bayesian utility for the defender can be given as

$$U_D(a_D, t_A) = \sum_{i=1}^{n} w_i \cdot \Big(\sum_{t_A \in T_A} P(t_A) \cdot R_{D,i}(a_D, t_A) - C_{D,i}(a_D, t_A)\Big) \tag{12}$$

The Game-theoretic learning dynamics for the reward and action can be expressed as

$$R_D^t(a_D) = U_D(a_D^t, t_A) - \max_{a_D'} U_D(a_D', t_A) \tag{13}$$

$$a_D^{t+1} = a_D^t + \eta \cdot R_D^t(a_D) \tag{14}$$

The probabilistic Bayesian Nash equilibrium for the defender and attacker can be expressed as

$$\pi_D^*(a_D) = \arg\max_{\pi_D} \mathbb{E}_{a_A \sim \pi_A}\Big[U_D(a_D, t_A)\Big] \tag{15}$$

$$\pi_A^*(a_A) = \arg\max_{\pi_A} \mathbb{E}_{a_D \sim \pi_D}\left[U_A(a_A, t_D)\right] \tag{16}$$

Algorithm 2: Bayesian game for intrusion detection

Input: Set of attacker strategies $\boldsymbol{A}_A$, defender strategies $\boldsymbol{A}_D$, type distributions $P(t_A)$ and $P(t_D)$, payoff functions U_A, U_D

Output: Optimal strategies a_A^* and a_D^*

for each defender strategy $a_D \in \boldsymbol{A}_D$ **do**

for each attacker type $t_A \in T_A$ **do**

Compute expected utility for defender:

$U_D(a_D, t_A) = \sum_{t_A} P(t_A) \cdot R_D(a_D, t_A) - C_D(a_D, t_A)$

end for

end for

for each attacker strategy $a_A \in \mathcal{A}_A$ **do**

for each defender type $t_D \in T_D$ **do**

Compute expected utility for attacker:

$U_A(a_A, t_D) = \sum_{t_D} P(t_D) \cdot T_A(a_A, t_D) - D_A(a_A, t_D)$

end for

end for

Find optimal strategies:

$a_D^* = \arg\max_{a_D} E[U_D(a_D, t_A)]$, $a_A^* = \arg\max_{a_A} E[U_A(a_A, t_D)]$

Return: Optimal strategies a_A^* and a_D^*

3.4 Double Deep Q- Learning in PPFAD

The decision-making process is optimized using a Double Deep Q-Network (DDQN) for anomaly detection in an IoT edge environment. The traditional DQN has been improved with DDQN, which is designed to address the overestimation bias issue during action-value estimation by decoupling the Q-value estimation and action selection processes. Each edge node observes network conditions and system metrics and selects actions such as flagging suspicious activity or adjusting prefetching strategies. The state includes real-time traffic features, while actions represent system-level responses to potential anomalies. The reward function is designed to encourage low latency, high detection accuracy, and system stability. DDQL is chosen over standard Q-learning to mitigate overestimation of Q-values. The features extracted from the IoT devices are represented by the state s_t at time t, including the types of attacks and model updates from FL. The action a_t represents the anomaly detection decisions, and r_t defines the reward based on detection accuracy and FP/FN rates.

The evaluation network $Q(s_t, a_t; \theta)$ and the target network $Q'(s_t, a_t; \theta')$ are used in the DDQN. $Q(s_t, a_t; \theta)$ is used for updates during the training process while $Q'(s_t, a_t; \theta')$ is used for stabilizing during the learning process. The update rule for Q-learning is expressed as

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\left[r_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t)\right] \tag{17}$$

where α shows the learning rate, r_{t+1} is the reward after a_t, and γ is the discount factor. The exploration and exploitation are balanced using the epsilon-greedy policy, and the action can be expressed as

$$a_t = \begin{cases} \text{random action}, & \text{with probability } \epsilon \\ \arg\max_a Q(s_t, a), & \text{with probability } 1- \epsilon \end{cases} \tag{18}$$

The positive and negative reward functions for anomaly detection and normal activity can be expressed as

$$r_{t+1} = \begin{cases} +1, & \text{if the anomaly is correctly classified} \\ +1, & \text{if the normal activity is correctly classified} \\ -1, & \text{if the anomaly is incorrectly classified} \\ -1, & \text{if the normal activity is incorrectly classified} \end{cases} \tag{19}$$

DDQL finds the optimal policy $\pi^*(s)$ that maps states s and actions a to maximize the reward. The Bellman equation can be expressed as

$$V(s) = \max_a \left[r(s,a) + \gamma \sum_{s'} P(s'|s,a) V(s') \right] \tag{20}$$

where $P(s'|s,a)$ shows the transition probability from s to s' after a. The loss function for training the evaluation network is expressed as

$$L(\theta) = \mathbb{E}\left[\left(y_t - Q(s_t, a_t; \theta) \right)^2 \right] \tag{21}$$

where y_t represents the target Q-value and is expressed as

$$y_t = r_t + \gamma Q'(s_{t+1}, \arg\max_a Q(s_{t+1}, a; \theta); \theta') \tag{22}$$

The detailed workflow of the DDQL algorithm has been presented in Algorithm 3.

Algorithm 3: Double deep Q-learning for anomaly detection

Input: Replay buffer $\mathcal{D}$, learning rate η, discount factor γ, exploration rate ε
Output: Optimized Q-network $Q(s, a; \theta)$
Initialize primary Q-network $Q(s, a; \theta)$ and target network $Q'(s, a; \theta')$
for each episode $e = 1, 2, \ldots$ **do**
  Initialize state s_0
  for each time step $t = 1, 2, \ldots$ **do**
    Select action a_t using ε-greedy policy:

$$a_t = \begin{cases} \text{random action}, & \text{with probability } \varepsilon \\ \arg\max_a Q(s_t, a; \theta), & \text{with probability } 1 - \varepsilon \end{cases}$$

    Execute action a_t and observe reward r_t and next state s_{t+1}
    Store transition (s_t, a_t, r_t, s_{t+1}) in $\mathcal{D}$
    Sample a mini-batch of transitions from $\mathcal{D}$
    Compute target value:

$$y_t = r_t + \gamma Q'(s_{t+1}, \arg\max_a Q(s_{t+1}, a; \theta); \theta')$$

(Continued)

Algorithm 3 (continued)

Update primary network by minimizing loss:
$L(\theta)=\mathbb{E}\left[(y_t - Q(s_t, a_t; \theta))^2\right]$
Periodically update target network:
$\theta' \leftarrow \theta$
end for
end for
Return: Optimized Q-network $Q(s, a; \theta)$

4 Results and Discussion

This section presents a brief discussion of the network setup and analysis of the results for the proposed PPFAD framework. The network setup has been established for the anomaly detection of different attacks in an edge computing Internet of Things (EC-IoT) environment. The proposed model, based on DDQN and FL, is designed with a lightweight neural network architecture. It ensures compatibility with edge devices that perform localized training, thereby keeping memory and CPU usage to a minimum. Since only model weights are exchanged during federated updates, communication overhead remains low. Our edge nodes operated within acceptable CPU (<40%) and memory (<60%) usage ranges, confirming the model's practicality in residential IoT environments. The system's performance has been analyzed using performance metrics, including False Positive Rate (FPR), False Negative Rate (FNR), accuracy, throughput, and latency. The parameters used for the implementation of federated learning (FL), Bayesian game theory, and double deep Q-learning (DDQL) are presented in Table 3.

Table 3: Parameters for network setup and model implementation

Parameter	Value
Edge devices (Raspberry Pi 4)	5 devices
Communication rounds	10
Local epochs	5 per edge device
Learning rate	0.001
Batch size	32
Neural network architecture	3-layer fully connected
Activation function	ReLU for hidden, linear for output
Discount factor (γ)	0.99
Epsilon decay	Exponential over 100 episodes
Replay buffer size	10,000 samples
Target network update frequency	Every 10 episodes
Loss function	Mean Squared Error (MSE)
Learning rate	0.001
Reward function	+1 (correct), −1 (incorrect)

4.1 Network Setup

A network has been created using 20 sensors at four different locations, each location has five sensors and an edge device. The four edge gateways are used to collect and process data from various sensors. The fifth

device is the central server, which coordinates model aggregation in the federated learning setup. The sensors collect real-time data, including motion, temperature, smoke, and humidity, and send it to the corresponding edge gateway used for pre-processing and anomaly detection. Intrusions such as DoS, MitM, IP Spoofing, and Brute Force attacks have been introduced to identify and respond to abnormal activities.

4.2 Results Analysis

Fig. 1 shows the average learning progress of the model over multiple training episodes. The number of episodes is along the x-axis, and the average cumulative rewards are along the y-axis. The reward progresses as it learns to improve decisions in the presence of anomalies. The increase in rewards signifies the efficiency of the proposed model in anomaly detection while minimizing false detections. The pattern proposes that the method effectively learns from federated data while maintaining strong decision-making through Bayesian game optimization. The reduction of false detection also ensures that the model efficiently accumulates rewards, settling its supremacy in anomaly detection.

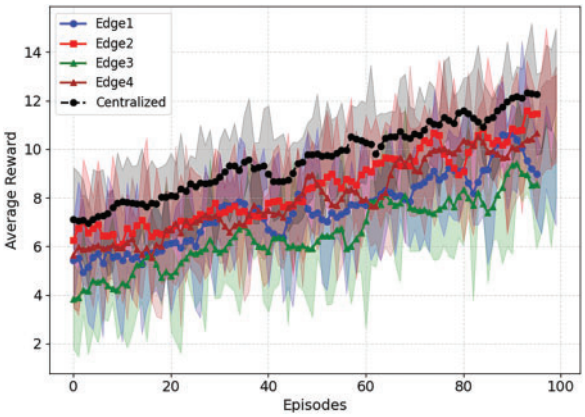

Figure 1: Average rewards over multiple training episodes at various Edge nodes for PPFAD vs centralized model

The average training loss for all the edge devices has been shown in Fig. 2. The decrease in the training loss over the episodes indicates that the model is learning efficiently. It confirms that the model is improving its policy with each iteration. It is worth mentioning that device 1 exhibits the lowest loss, followed by device 3 and device 4, while device 2 shows the highest loss. The loss reduction for the devices is attributed to the balanced state-action selection mechanism. Federated learning also contributes to lower overfitting, while the Bayesian game confirms that the model dynamically adjusts to different anomaly probabilities. This avoids needless penalties and lead to smoother loss convergence.

Fig. 3 shows the average latency results over time steps. It has been noticed that the latency at the four-edge devices using FL is significantly lower compared to centralized learning. The latency is lower for edge device 1, while the remaining devices exhibit slightly higher latencies. Latency at the central server is nearly twice that of the slowest edge device. This is due to the increased communication overhead as all data is transmitted at the central server to be processed.

Fig. 4 shows the average data transmitted per unit time. It is noticeable that at the edge devices, the throughput is expressively higher than that of the central server. Edge device 1 exhibits higher throughput, while the remaining edge devices experience lower throughput values. However, the central server shows lower throughput due to increased waiting time for data aggregation. In the proposed framework, data is

processed locally by each edge device, and only necessary updates are transmitted to the central server. It reduces the overall network load and permits higher throughput at the edge nodes.

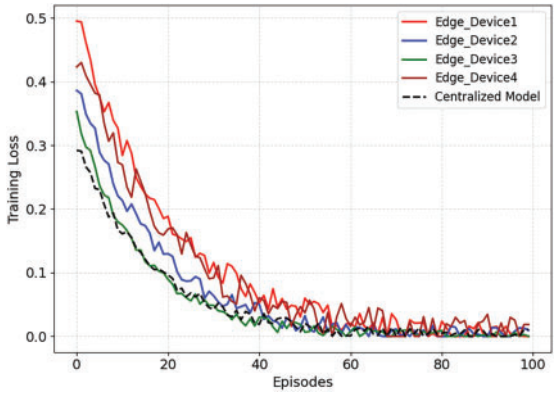

Figure 2: Average training loss over multiple training episodes at various Edge nodes vs centralized model

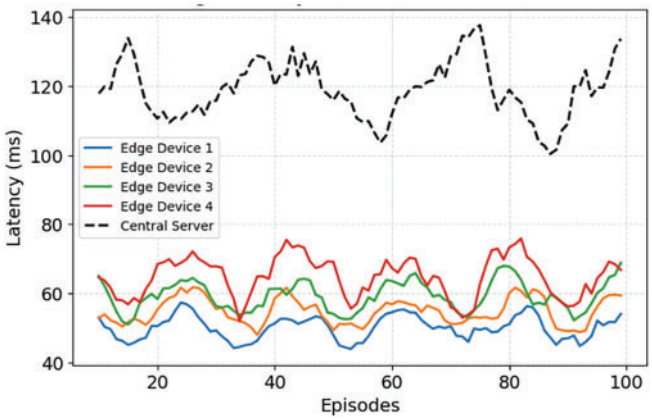

Figure 3: Average latency over multiple training episodes at Edge nodes and central server

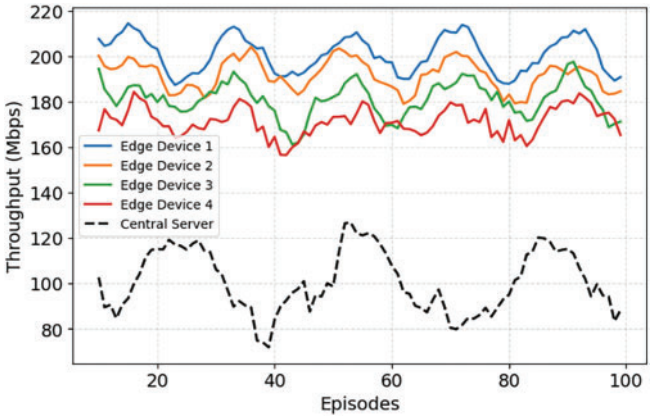

Figure 4: Average throughput over multiple training episodes at Edge nodes and central server

The False Positive Rate (FPR) and False Negative Rate (FNR) are the most important metrics in anomaly detection. FPR measures incorrect classes, while FNR measures undetected threats. A high false positive rate (FPR) leads to unnecessary alerts and increases the security load, while a high false negative rate (FNR) allows undetected attacks to penetrate the network. Table 4 and Fig. 5 show the comparison results of three different models: Isolation Forest (IF), Autoencoder (AE), and Bayesian RL. Results reveal that our proposed approach outclasses the other models and provides an effective and perfect anomaly detection framework.

Table 4: Comparison results of various performance metrics for different models

Model	FPR	FNR	Accuracy	Precision	Recall
Isolation forest	8.3%	9.2%	86%	89%	81%
Autoencoder	6.6%	5.9%	91%	93%	86%
Bayesian RL	2.2%	1.9%	97%	96%	97%

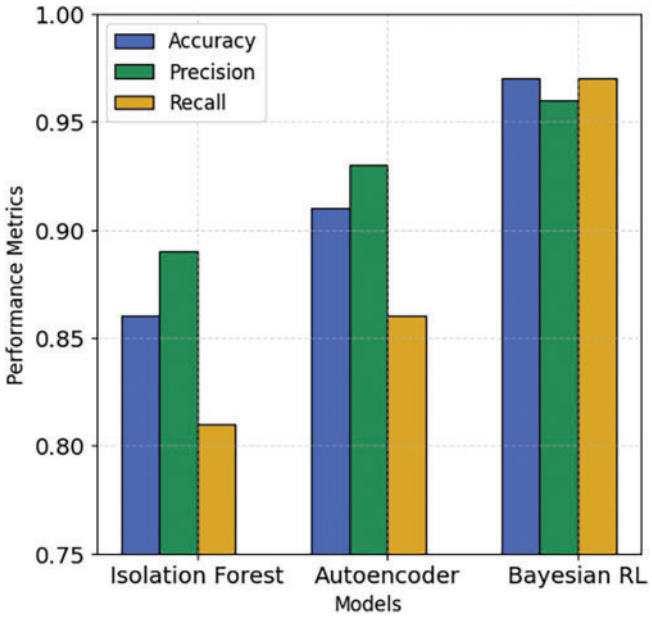

Figure 5: Comparison results of various performance metrics for Isolation Forest, Autoencoder, and Bayesian RL models

Figs. 6 and 7 show the box plots for the three models. It is worth mentioning that the IF possesses the highest FPR and FNR, indicating that it regularly misclassifies attacks and fails to identify actual anomalies. The AE performs better, indicating its ability to differentiate between normal and anomalous traffic, as well as undetected anomalies. The Bayesian RL achieves the lowest FPR and FNR, thereby reducing false alarms and ensuring accurate anomaly identification. It is worth mentioning that the proposed framework also outperforms the other models in terms of accuracy, precision, and recall.

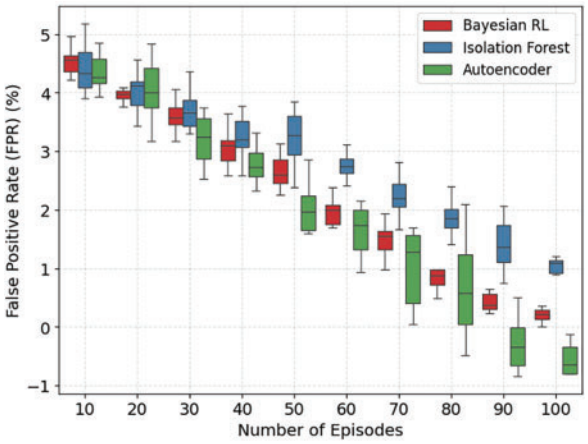

Figure 6: Data distribution of different models for the False Positive Rate

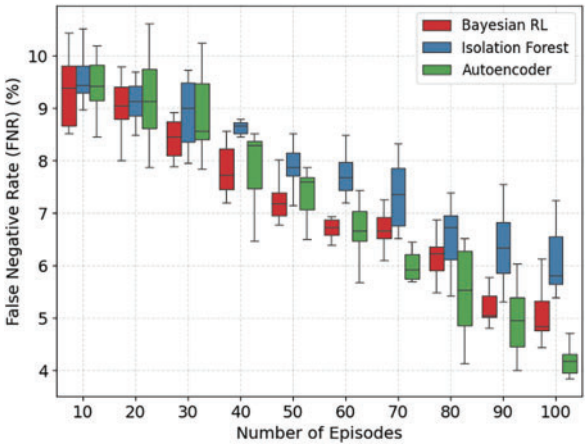

Figure 7: Data distribution of different models for the False Negative Rate

Table 5 shows the comparison results of various performance metrics for various attacks. We observed a measurable tradeoff between precision and recall in evaluating the model's performance across different attack types. The model tends to prioritize higher recall, particularly in detecting IP Spoofing attacks, to minimize false negatives for stealthy threats. It is a critical choice in IoT environments where undetected anomalies can lead to significant security breaches. While this slightly lowers false positives, the tradeoff is acceptable in our edge IoT context, where early detection is prioritized over misclassification. For other attacks, such as DoS and Brute Force attacks, the model maintains a more balanced precision-recall profile, ensuring a more resilient and secure environment.

Table 5: Comparison results of various performance metrics for various attacks

Attack type	Accuracy	Precision	Recall	FPR	FNR
DoS	98.4%	98.1%	98.7%	1.2%	1.3%
MitM	97.9%	97.5%	98.2%	1.5%	1.8%

(Continued)

Table 5 (continued)

Attack type	Accuracy	Precision	Recall	FPR	FNR
IP spoofing	96.8%	96.3%	97.1%	2.1%	2.9%
Brute force	98.1%	97.8%	98.4%	1.6%	1.6%

Fig. 8 shows the results of performance metrics for various attack types. The Denial of Service (DoS), Man-in-the-Middle (MitM), IP Spoofing, and Brute Force (BF) attacks have been analyzed in terms of accuracy, precision, recall, False Positive Rate (FPR), and False Negative Rate (FNR). The model shows a high accuracy of 98.4% in the detection of DoS attacks, while 98.1%, 97.9%, and 96.8% for BF, IP spoofing, and MitM, respectively. The FPR remains below 2.1% for most attacks, which suggests that the model efficiently differentiates between malicious and benign traffic. The model also achieved a low FNR score of 1.3% for DoS attacks and 1.6% for BF attacks. However, the FNR score for MitM and IP spoofing is slightly high at 1.8% and 2.9%, respectively.

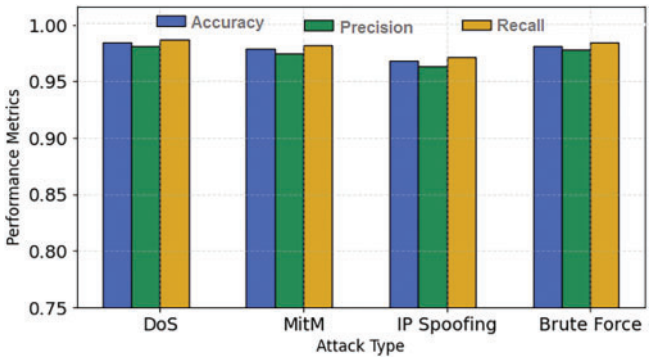

Figure 8: Comparison results of various performance metrics for DoS, MitM, IP Spoofing, and Brute Force attacks

5 Conclusion

This study presents an optimized framework for privacy-preserved federated anomaly detection in edge computing (EC) Internet of Things (IoT) networks. Bayesian game theory, integrated with double deep Q-learning (DDQL), enhances anomaly detection in the EC-IoT environment. Federated learning (FL) inclusion guarantees data privacy locally by handling sensitive information and enabling cooperative anomaly detection. Bayesian game theory ensures strategic design between attackers and defenders, while DDQL optimizes resources. The proposed framework enhances detection accuracy and stabilizes policy management in CE environments. Experimental results show that the proposed framework achieved high detection performance accuracy. This research provides a solid foundation for developing efficient, scalable, and secure anomaly detection systems for EC-IoT. Future work will incorporate a multimodal approach, including unmanned aerial vehicles, in EC-IoT environments.

Acknowledgement: The authors extend their appreciation to the Deanship of Research and Graduate Studies at King Khalid University for funding this work through the Large Group Project under grant number (RGP2/337/46). The research team thanks the Deanship of Graduate Studies and Scientific Research at Najran University for supporting the research project through the Nama'a program, with the project code NU/GP/SERC/13/352-4.

Funding Statement: The authors extend their appreciation to the Deanship of Research and Graduate Studies at King Khalid University for funding this work through the Large Group Project under grant number (RGP2/337/46). The research team thanks the Deanship of Graduate Studies and Scientific Research at Najran University for supporting the research project through the Nama'a program, with the project code NU/GP/SERC/13/352-4.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Fahad Masood and Jawad Ahmad; methodology, Fahad Masood; software, Fahad Masood; validation, Jawad Ahmad, Wajdan Al Malwi and Fatima Asiri; formal analysis, Wajdan Al Malwi and Mohammed S. Alshehri; investigation, Jawad Ahmad and Tamara Zhukabayeva; resources, Wajdan Al Malwi, Fatima Asiri, Syed Aziz Shah and Mohammed S. Alshehri; data curation, Fahad Masood and Tamara Zhukabayeva; writing—original draft preparation, Fahad Masood; writing—review and editing, Jawad Ahmad and Tamara Zhukabayeva; visualization, Fatima Asiri, Syed Aziz Shah and Mohammed S. Alshehri; supervision, Mohammed S. Alshehri; project administration, Jawad Ahmad and Tamara Zhukabayeva; funding acquisition, Wajdan Al Malwi, Fatima Asiri and Mohammed S. Alshehri. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are openly available at https://github.com/researchcsaup/IoT-UAV/blob/75ed7ffaad45f1354658461d3d65ecc80437b472/IoT_Anomaly_Detection.csv (accessed on 09 June 2025).

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Yu W, Liang F, He X, Hatcher WG, Lu C, Lin J, et al. A survey on the edge computing for the Internet of Things. IEEE Access. 2017;6:6900–19. doi:10.1109/ACCESS.2017.2778504.
2. Alwarafy A, Al-Thelaya KA, Abdallah M, Schneider J, Hamdi M. A survey on security and privacy issues in edge-computing-assisted internet of things. IEEE Internet Things J. 2020;8(6):4004–22. doi:10.1109/JIOT.2020.3015432.
3. Gyamfi E, Jurcut A. Intrusion detection in internet of things systems: a review on design approaches leveraging multi-access edge computing, machine learning, and datasets. Sensors. 2022;22(10):3744. doi:10.3390/s22103744.
4. Rani D, Gill NS, Gulia P, Arena F, Pau G. Design of an intrusion detection model for IoT-enabled smart home. IEEE Access. 2023;11:52509–26. doi:10.1109/ACCESS.2023.3276863.
5. Abdusalomov A, Kilichev D, Nasimov R, Rakhmatullayev I, Im Cho Y. Optimizing smart home intrusion detection with harmony-enhanced extra trees. IEEE Access. 2024;12:117761–86. doi:10.1109/ACCESS.2024.3422999.
6. Mtukushe N, Onaolapo AK, Aluko A, Dorrell DG. Review of cyberattack implementation, detection, and mitigation methods in cyber-physical systems. Energies. 2023;16(13):5206. doi:10.3390/en16135206.
7. Islam N, Farhin F, Sultana I, Kaiser MS, Rahman MS, Mahmud M, et al. Towards machine learning based intrusion detection in IoT networks. Comput Mater Contin. 2021;69(2):1801–21. doi:10.32604/cmc.2021.018466.
8. Odeh A, Taleb AA. Robust network security: a deep learning approach to intrusion detection in IoT. Comput Mater Contin. 2024;81(3):4149–69. doi:10.32604/cmc.2024.058052.
9. Khatami SS, Shoeibi M, Oskouei AE, Martín D, Dashliboroun MK. 5DGWO-GAN: a novel five-dimensional gray wolf optimizer for generative adversarial network-enabled intrusion detection in IoT systems. Comput Mater Contin. 2025;82(1):881–911. doi:10.32604/cmc.2024.059999.
10. Javeed D, Saeed MS, Ahmad I, Kumar P, Jolfaei A, Tahir M. An intelligent intrusion detection system for smart consumer electronics network. IEEE Trans Consum Electron. 2023;69(4):906–13. doi:10.1109/TCE.2023.3277856.
11. Ullah S, Ahmad J, Khan MA, Alshehri MS, Boulila W, Koubaa A, et al. TNN-IDS: transformer neural network-based intrusion detection system for MQTT-enabled IoT networks. Comput Netw. 2023;237(5):110072. doi:10.1016/j.comnet.2023.110072.

12. Abou El Houda Z, Brik B, Ksentini A, Khoukhi L, Guizani M. When federated learning meets game theory: a cooperative framework to secure IIoT applications on edge computing. IEEE Trans Ind Inform. 2022 Apr 26;18(11):7988–97. doi:10.1109/TII.2022.3170347.
13. Poornima IG, Paramasivan B. Anomaly detection in wireless sensor network using machine learning algorithm. Comput Commun. 2020;151:331–7. doi:10.1016/j.comcom.2020.01.005.
14. Hasan M, Islam MM, Zarif MI, Hashem MM. Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. Internet Things. 2019;7(20):100059. doi:10.1016/j.iot.2019.100059.
15. Pang G, Shen C, Cao L, Hengel AV. Deep learning for anomaly detection: a review. ACM Comput Surv. 2021;54(2):1–38. doi:10.1145/3439950.
16. Churcher A, Ullah R, Ahmad J, Ur Rehman S, Masood F, Gogate M, et al. An experimental analysis of attack classification using machine learning in IoT networks. Sensors. 2021;21(2):446. doi:10.3390/s21020446.
17. Gupta BB, Gaurav A, Arya V, Attar RW, Bansal S, Alhomoud A, et al. Cuckoo search-optimized deep CNN for enhanced cyber security in IoT networks. Comput Mater Contin. 2024;81(3):4109–24. doi:10.32604/cmc.2024.056476.
18. Wang X, Garg S, Lin H, Hu J, Kaddoum G, Piran MJ, et al. Toward accurate anomaly detection in industrial internet of things using hierarchical federated learning. IEEE Internet Things J. 2021;9(10):7110–9. doi:10.1109/JIOT.2021.3074382.
19. Dhaouadi T, Mrabet H, Alhomoud A, Jemai A. An intrusion detection system based on HiTar-2024 dataset generation from LOG files for smart industrial internet-of-things environment. Comput Mater Contin. 2025;82(3):4535–54. doi:10.32604/cmc.2025.060935.
20. Ullah S, Nasir HM, Kadir K, Khan A, Memon A, Azhar S, et al. End-to-end encryption enabled lightweight mutual authentication scheme for resource constrained IoT network. Comput Mater Contin. 2025;82(2):3223–49. doi:10.32604/cmc.2024.054676.
21. Chen X, Wang P, Yang Y, Liu M. Resource-constraint deep forest based intrusion detection method in internet of things for consumer electronic. IEEE Trans Consum Electron. 2024;70(2):4976–87. doi:10.1109/TCE.2024.3373126.
22. Zhou X, Liang W, Li W, Yan K, Shimizu S, Wang KI. Hierarchical adversarial attacks against graph-neural-network-based IoT network intrusion detection system. IEEE Internet Things J. 2021;9(12):9310–9. doi:10.1109/JIOT.2021.3130434.
23. Guo H, Zhou Z, Zhao D, Gaaloul W. EGNN: energy-efficient anomaly detection for IoT multivariate time series data using graph neural network. Future Gener Comput Syst. 2024;151(12):45–56. doi:10.1016/j.future.2023.09.028.
24. Jiang J, Liu F, Ng WW, Tang Q, Zhong G, Tang X, et al. AERF: adaptive ensemble random fuzzy algorithm for anomaly detection in cloud computing. Comput Commun. 2023;200(3):86–94. doi:10.1016/j.comcom.2023.01.004.
25. Zhou X, Xu X, Liang W, Zeng Z, Yan Z. Deep-learning-enhanced multitarget detection for end-edge–cloud surveillance in smart IoT. IEEE Internet Things J. 2021;8(16):12588–96. doi:10.1109/JIOT.2021.3077449.
26. Cai Z, Zheng X. A private and efficient mechanism for data uploading in smart cyber-physical systems. IEEE Trans Netw Sci Eng. 2018;7(2):766–75. doi:10.1109/TNSE.2018.2830307.
27. Algarni A, Acarer T, Ahmad Z. An edge computing-based preventive framework with machine learning-integration for anomaly detection and risk management in maritime wireless communications. IEEE Access. 2024;12(3):53646–63. doi:10.1109/ACCESS.2024.3387529.
28. Rupanetti D, Kaabouch N. Combining edge computing-assisted internet of things security with artificial intelligence: applications, challenges, and opportunities. Appl Sci. 2024;14(16):7104. doi:10.3390/app14167104.
29. Liu C, Su X, Li C. Edge computing for data anomaly detection of multi-sensors in underground mining. Electronics. 2021;10(3):302. doi:10.3390/electronics10030302.
30. Ngo MV, Luo T, Quek TQ. Adaptive anomaly detection for internet of things in hierarchical edge computing: a contextual-bandit approach. ACM Trans Internet Things. 2021;3(1):1–23. doi:10.1145/348017.
31. Wang X, Wu W, Du Y, Cao J, Chen Q, Xia Y. Wireless IoT monitoring system in Hong Kong–Zhuhai–Macao bridge and edge computing for anomaly detection. IEEE Internet Things J. 2023;11(3):4763–74. doi:10.1109/JIOT.2023.3300073.

32. Mansour RF, Abdel-Khalek S, Hilali-Jaghdam I, Nebhen J, Cho W, Joshi GP. An intelligent outlier detection with machine learning empowered big data analytics for mobile edge computing. Cluster Comput. 2023;26(1):71–83. doi:10.1007/s10586-021-03472-4.
33. Wu X, Yao A, Pal S, Jiang F, Li X, Xu J, et al. A dual defense self-balancing framework against bilateral model attacks in federated learning. In: International Conference on Algorithms and Architectures for Parallel Processing. Singapore: Springer Nature Singapore; 2024. p. 261–70. doi:10.1007/978-981-96-1525-4_14.
34. Yao A, Pal S, Li G, Li X, Zhang Z, Jiang F, et al. FedShufde: a privacy preserving framework of federated learning for edge-based smart UAV delivery system. Future Gener Comput Syst. 2025;166(10):107706. doi:10.1016/j.future.2025.107706.