Future Generation Computer Systems 174 (2026) 108014

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

DETR-BAL: Decentralized mobile sensing intrusion detection via latent mining and Bayesian local optimization ☆

Chen Zhang [a], Zhuotao Lian [b,*], Weiyu Wang [a], Huakun Huang [c], Chunhua Su [a]

[a] The University of Aizu, Jawahar Nagar, Aizuwakamatsu City, 9658580, Fukushima Prefecture, Japan
[b] Hiroshima University, 1-3-2 Kagamiyama, Higashi-Hiroshima City, 7398511 Hiroshima Prefecture, Japan
[c] Guangzhou University, West Outer Ring Road, Guangzhou, 510006, Guangdong, China

ARTICLE INFO

Keywords:
MCS
Decentralized
Committee mechanism
User selection
IDS

ABSTRACT

With the rapid proliferation of mobile sensing in fields such as personal health monitoring in data processing are becoming more prominent. This paper introduces a decentralized DETR framework inspired by blockchain proof-of-work consensus. The framework trains models locally on each device and evaluates the device's reputation based on its historical performance. Only devices meeting predefined criteria are admitted to the update committee, which enhances security. This mechanism reduces reliance on centralized servers and minimizes infrastructure costs. While a supervisory operator ensures the smooth operation of the system. To further enhance trust, we propose a credibility assessment method that integrates risk metrics with data quality scores via a non-cooperative game-theoretic model. By achieving Nash equilibrium, this method not only guarantees local optimality but also prioritizes users who provide high-quality, low-risk data, thereby promoting timely committee updates to achieve global optimality. As a complement to DETR, we propose BAL-IDS, an advanced intrusion detection system (IDS) that extracts latent features using autoencoders and dynamically fine-tunes the hyperparameters of OCSVM using a Bayesian joint local agent optimization strategy. This dual approach enhances the system's resilience to complex threats, especially those that exploit requester feedback mechanisms. Experiments show that our research is superior to traditional schemes.

1. Introduction

Mobile Crowd Sensing (MCS) is a sensing paradigm that utilizes internet platforms to leverage the collective intelligence efforts of a large number of individuals to address complex computational problems [1]. It integrates mobile devices, communication, computing, and artificial intelligence (AI). This form combines the strengths of both humans and machines, relying on advanced AI technologies to optimize collaboration and enhance the efficiency and accuracy of task processing. Essentially, MCS outsources data collection to humans and data aggregation to cloud servers [2]. MCS is characterized by mobile users whose coverage must be controlled, a large pool of mobile devices participating in sensing tasks, users executing perception tasks assigned by cloud servers, and cloud servers aggregating and processing user feedback before publishing results to requesters [3].

These characteristics define the core of MCS technology and emphasize its ability to leverage crowd intelligence and mobile technology for data collection and processing in dynamic environments. There are countless examples of real-world applications of MCS. For instance, in many cities, traffic data from the public is collected through mobile applications [4], which are uploaded to a cloud server, where the cloud service is responsible for processing the noise with further complex computations, and ultimately sending it to the requester. The requester can use MCS to monitor the traffic flow and congestion and thus optimize the route [5].

The sample diagram shown in Fig. 1, illustrates a Conventional MCS system divided into four layers: the requester, the service layer, the network layer, and the sensing layer [6]. The requester, who initiates tasks, is part of the sensing layer but can also assist other requesters in data collection [7]. The requester submits a task to the cloud server, which then assigns the task to a suitable user through the network layer, considering the task's specific constraints [8]. Once the user accepts the task, they collect data according to these constraints and upload it to the cloud server. The server aggregates and processes

☆ This work was supported by JSPS Grant -in-Aid for Scientific Research (C) 23K11103 and the Tertiary Education Scientific Research Project of Guangzhou Municipal Education Bureau under Grant 2024312326, JSPS Grant -in-Aid for JSPS Fellows under Grant 24KF0065 and SCAT Foundation under Grants for Researchers.

* Corresponding author.
E-mail address: zhuotaolian@gmail.com (Z. Lian).

https://doi.org/10.1016/j.future.2025.108014
Received 14 April 2025; Received in revised form 24 June 2025; Accepted 6 July 2025
Available online 19 July 2025

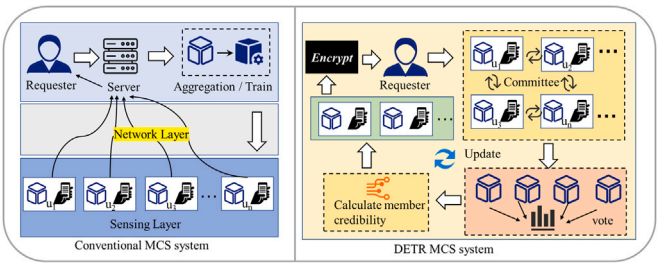


Fig. 1. Difference between Conventional MCS and DETR.

the data. Finally, it returns the results to the requester. Despite its efficiency, conventional MCS systems face several challenges: a single point of failure if the server crashes; limited scalability as the user base grows [9]; network congestion and delays from routing all data through a central server; and vulnerability to attacks that could expose sensitive data [10].

To address the limitations of conventional MCS systems, we propose solutions of voting committees, decentralized federated learning (DFL), and trustworthy models. In this paper, our goal is to protect MCS systems through a DFL committee called **DETR**. DETR allows users to train locally, and then the committee votes and aggregates updates to the user's training model. Federated learning (FL) is feasible in MCS systems [11,12]. The requester submits an encrypted constraint task to the committee, which processes it using a sophisticated algorithm. The task is then assigned to a qualified user to generate a perceptual result. This result is evaluated by the committee using a trusted scoring model. The committee evaluates each result using a trusted scoring model. If a majority agrees, we return it to the requester. Upon receipt, the requester has the opportunity to verify the result. Additionally, the performance of committee members is continuously monitored, and any member receiving a significant number of negative ratings is subject to removal from the committee.

1.1. Motivation and challenges

Our objective is to keep data local while meeting requester requirements, delegating only application updates and access control to service providers, and ensuring honest participants receive fair compensation. To this end, DETR employs decentralized federated learning (DFL) to aggregate model updates under a robust evaluation scheme that limits collusion attacks [13]; supports over-the-air application updates monitored by a trusted third party for rapid adaptation to market changes; integrates a dynamic feedback channel enabling honest users to report issues and drive service optimization [14]; uses automated analysis, anomaly detection, and a transparent audit interface to filter malicious inputs and adapt to evolving threats [15–17]; and implements defenses against both 51% and sybil attacks to safeguard voting integrity and network robustness. To realize these capabilities, we introduce a DFL-based committee framework, DETR, underpinned by a W-type grading credibility algorithm for user evaluation, and an IDS combining autoencoder latent-space mining with Bayesian joint local surrogate optimization (BAL-IDS) to counter adversarial feedback manipulation.

1.2. Our contributions

The contributions of this paper can be summarized in three aspects.

- We introduce DETR, a committee framework built on DFL and inspired by blockchain's Proof of Work (PoW). In DETR, each user's contribution determines a credibility score. Users whose score exceeds a predefined threshold compete for committee seats and gain the right to vote on model updates.
- We design a credibility assessment using non-cooperative game theory, combining users' risk indicators and data contributions. By formulating a game whose Nash equilibrium balances these factors, DETR dynamically adjusts risk-vs-contribution weights to favor high-credibility, low-risk users, ensuring efficient, near-optimal committee selection.
- We introduce BAL-IDS, which uses an autoencoder to learn the latent patterns of normal behavior and filter out noise. We then apply a Bayesian proxy optimization to fine-tune the One-Class SVM's hyperparameters. The resulting SVM more accurately models benign activity and is more robust against attacks that exploit feedback loops.

The paper is organized as follows, with related work in Section 2. Section 3 is the theoretical model and the system framework of DETR. Section 5 is for evaluation experiments and analysis. Section 6 is the conclusion.

2. Related work

This section focuses on decentralized adaptation in mobile-aware scenarios, privacy preservation, communication efficiency optimization, and user selection research.

[18] proposed a hybrid blockchain-based user selection scheme that enhances the security and reliability of MCS systems through smart contracts and ensures user reputation and data quality using semi-Markov models and LSTM algorithms. However, the scheme suffers from high computation and communication overheads, is not suitable for lightweight mobile devices, and the complexity affects scalability. In contrast, our decentralized committee mechanism achieves a high accuracy of 99.37% with local training and historical user reputation, reduces reliance on heavy infrastructure, and provides a more efficient and reliable mobile-aware solution. [9] proposed a blockchain-based mobile crowdsourcing-aware system (MCS-Chain), which aims to achieve fully distributed and decentralized trust management in MCS. In order to improve the efficiency of traditional blockchain technology, MCS-Chain designs a novel consensus mechanism that significantly reduces the computational overhead and solves the fork problem and centralization problem commonly found in existing blockchain systems. The security and efficiency of the system are verified through rigorous security analysis and experimental evaluations. MCS-Chain, despite its improved efficiency, still relies on blockchain technology, which may bring certain computational and communication overheads and is not entirely suitable for resource-limited mobile devices. In addition, although the consensus mechanism of MCS-Chain optimizes the performance, it still needs to be further improved in terms of user reputation evaluation and data availability guarantee.

[19] proposed a blockchain-based mobile crowdsourcing sensing system (BMCS) designed to integrate MCS into industrial systems. This integration aims to enhance the security and reliability of the system through miner-validated data and a dynamic reward mechanism. A prototype was implemented and validated on the Ethereum platform. However, the reliance of BMCS on blockchain technology introduces significant computational and communication overheads. Additionally, the process of miner verification may result in delays, making it unsuitable for resource-constrained mobile devices. [20] proposed a generalized exponential moving average (EMA) model, which is a novel stochastic volatility model featuring time-varying expected returns. The researchers effectively employed particle filtering (PF) for the sequential estimation of states and parameters within financial markets. Additionally, they developed three types of anomaly detectors that can be seamlessly integrated into the PF algorithm to enhance investment decision-making. The results indicate that simple investment strategies based on this framework outperform standard EMA-based strategies, as well as traditional approaches such as equal-weighted, minimum variance, and risk parity portfolios. This research

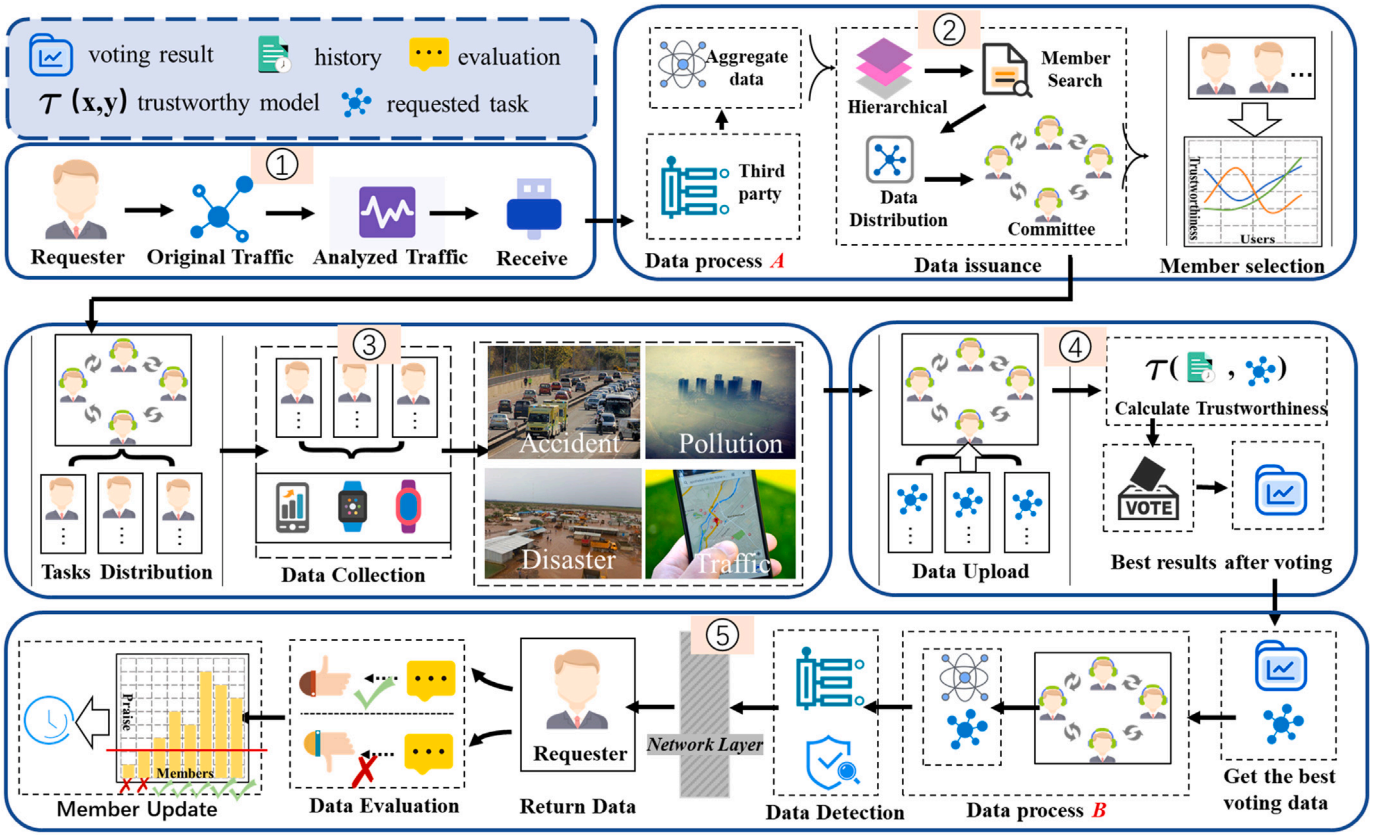


Fig. 2. DETR system framework.

emphasizes investment strategies in financial markets, utilizing particle filtering and sophisticated anomaly detection algorithms, but it does not address the implications of mobile sensing and distributed systems. [21] proposed the problem of coordinated data collection in MCS systems. The study proposes a decentralized approach that combines matching theory and online learning called Collision Avoidance Multi-Arm Slot Machine with Strategic Free Sensing (CA-MAB-SFS). In this approach, the Mobile Crowdsourcing Sensing Platform (MCSP) releases sensing tasks sequentially, the Mobile Units (MUs) indicate their willingness to participate by sending sensing offers, and the MCSP assigns tasks based on the received offers. CA-MAB-SFS, although it improves task assignment and learning efficiency, still relies on matching theory and online learning algorithms, which may entail higher computational complexity and real-time requirements. In addition, CA-MAB-SFS mainly focuses on the stability and efficiency of task allocation, without sufficient consideration of user reputation and data quality. [22] proposed a blockchain-based privacy-preserving scheme for quality-aware worker recruitment reputation (BRPP-QWR). The scheme designs a lightweight privacy-preserving mechanism that combines subaddress retrieval techniques, Pedersen promises, and CLSAG signatures to achieve fast and anonymous verification of the reputation update process. In addition, reputation, Selfishness, and Quality-based Multi-Arm Slot Machine (RSQ-MAB) learning algorithms are proposed to select reliable and high-quality workers.

The existing literature on MCS generally suffers from high computation and communication overhead due to reliance on blockchain or centralized architectures, increased system complexity, and insufficient user reputation and data quality guarantees, limiting their application in resource-constrained and dynamic environments. In contrast, our proposed decentralized committee mechanism framework achieves high accuracy through local model training and selection based on users' historical reputation, significantly reduces reliance on heavy infrastructure, reduces computation and communication overhead, and provides a more efficient, reliable, and adaptable solution.

3. System design

3.1. Application scenario

This paper explores the diverse applications of distributed mobile sensing across several key sectors. In health monitoring, distributed sensors can gather a user's data on steps, heart rate, and sleep patterns, promoting the user's better health and lifestyle choices [23]. For transportation navigation, real-time geolocation data facilitates optimized route planning, effectively reducing commuting times [24]. In environmental monitoring, sensors capturing air quality and noise levels support urban planning initiatives aimed at improving living conditions [25]. Social interaction applications leverage geographic and activity data to suggest relevant social activities, enhancing user interactions [26]. In smart home systems, devices automatically adjust settings such as temperature and lighting based on sensed user location and habits, enhancing both comfort and efficiency [27].

When a distributed mobile sensing task commences, a Trusted Third Party (TTP) broadcasts the initial model to all nodes and selects highly credible users using a credibility algorithm. Next, the requester publishes the task to the TTP, after which all TTP members employ their local datasets for parameter aggregation or data processing. Upon completion of the model training, each member retrieves model parameters from all other nodes via P2P communication, tests these parameters, and uses the resulting accuracy as their evaluation score. Members then exchange their voting scores, collectively identifying and excluding any dishonest participants that cause model deviations according to the trusted model evaluation scheme. Finally, once the model accuracy meets the predefined standard, the model parameters are broadcast to all nodes, and the committee is updated.

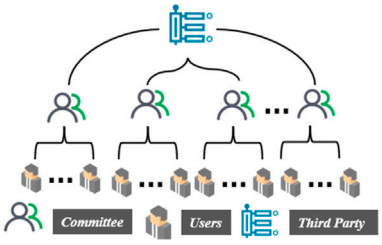


Fig. 3. Trusted third parties act as control and supervision.

Table 1
Formula symbols and meaning.

Symbol	Meaning	Symbol	Meaning
$\mathcal{A}$	Dishonest users.	u_i	a user.
$\mathcal{B}$	Honest users.	p	Packed layered flow.
$\mathcal{Q}$	requester.	N	Sample Size.
C	Committee	U	All users set.
$\mathcal{M}$	Machine Learning.	r	Risk Assessment.
D	Constraint.	v_j	The jth member's vote result
$\mathcal{H}$	History Data.	m	Total Committee Members.
P	System parameters	k	The risk threshold.
$\mathcal{N}$	Credibility.	α	Learning rate parameter.
O	Satisfaction.	β	Loss rate parameter.
$C_j\star$	Nearby users of C_j.	Re_i	Feedback results for user i.

3.2. System framework

As illustrated in Fig. 2, the overall process of our trusted model evaluation scheme begins when the requester sends original traffic containing the task request. This request is parsed and received by the TTP (i.e., the operator). The TTP then aggregates, hierarchically processes, and assigns tasks (process A) based on the data, distributing them to the appropriate committees according to task information. As shown in Fig. 3, the TTP primarily performs supervisory, control, and forwarding functions without engaging in complex computations. Each committee is responsible for verification calculations, data aggregation, and managing users within its region after local users independently form committee organizations. Committee Formation (Member Selection): The committee comprises rigorously verified, reliable users and can exist in multiple organizational forms. Once a task is assigned to a committee, it is distributed to subordinate users who utilize mobile devices (e.g., smartphones, smartwatches, smart bracelets) to complete tasks related to man-made accidents, environmental pollution, natural disasters, or traffic conditions.

After task completion, each subordinate uploads the collected data to their respective committee. The committee then independently verifies each submission; committee members make their decisions without knowledge of others' choices, satisfying the conditions for a Nash equilibrium in non-cooperative game theory. In this setting, each member selects their optimal strategy after considering the actions of others, ensuring that the overall system achieves stable and optimal cooperation. When a consensus is reached among a specified percentage of committee members (process B), the resulting answer is forwarded to the TTP. Subsequently, the TTP applies the BAL-IDS security test to mitigate potential risks, including poisoning attacks, 51% attacks, and Sybil attacks. After successful testing, the securely filtered data is transmitted to the requester via the network layer. The TTP further evaluates the contribution of the task data provided by subordinate users by categorizing it into three levels: normal, risky, and malicious, and refines this assessment using preset weight factors to obtain the final credibility score. Should a committee member's submission or that of its subordinate users deviate from the final consensus, their contribution value is reduced to zero or assigned a negative score.

After receiving the processed data, the requester assesses its accuracy and completeness, providing either positive or zero feedback. This feedback mechanism not only confirms the quality of the answer but also directly influences the benefits distributed among committee members: positive feedback rewards members accordingly, while zero feedback yields no benefit. Such a design encourages all participants to adopt optimal cooperative strategies during decision-making, ultimately enabling the entire system to reach a Nash equilibrium in a non-cooperative game framework, thereby ensuring fair, efficient, and secure data processing and information sharing.

BAL-IDS is a network IDS we developed that first performs autoencoder-based latent feature extraction, then applies Bayesian local proxy optimization alongside a One-Class SVM (OCSVM) to tackle challenging detection scenarios. The system integrates multi-source traffic and signature databases to identify patterns such as abnormal traffic from the same IP or device, bulk false evaluations, and Sybil attacks. Finally, predefined policy rules validate user contribution scores to filter out anomalies, enhancing both detection accuracy and robustness.

4. The proposed DETR

In MCS, ensuring the integrity of data collected from distributed users is critical. In our framework, named DETR, we integrate network traffic modeling, machine learning-based prediction, and a committee voting mechanism for risk assessment. In particular, we model traffic dynamics using a Markov Modulated Poisson Process (MMPP) and implement a multi-stage process to evaluate data credibility, mitigate malicious actions, and update committee members adaptively. Table 1 lists the key symbols and their meanings used throughout this framework.

4.1. MMPP definition

We model network traffic as a sequence of packet arrival processes generated by a continuous-time Markov chain (CTMC) that modulates a Poisson process. Let $S(t)$ for $t \geq 0$ be a CTMC with state space $S = \{s_1, s_2, \ldots, s_N\}$. The transitions between states are governed by the transition rate matrix Q, where for $i \neq j$, Q_{ij} denotes the rate of transitioning from state s_i to state s_j, and the diagonal entries satisfy:

$$Q_{ii} = -\sum_{\substack{j=1 \\ j\neq i}}^{N} Q_{ij} \tag{1}$$

Each state s_i encapsulates network traffic as a four-tuple $s_i = \{a, b, c, d\}$, where a represents packet information, b denotes protocol layer details, c encapsulates control information, and d contains metadata. This multi-dimensional representation allows the model to capture various aspects of network traffic for comprehensive analysis.

4.2. Game formulation

We model our intrusion-detection network as a noncooperative static game. Let the set of players be $N = \{1, 2, \ldots, n\}$, where player i represents the ith sensing node. Each player's strategy space is denoted $S_i \subset \mathbb{R}^m$, a nonempty compact convex set of tunable parameters, and we write the joint strategy profile as $\mathbf{s} = (s_1, \ldots, s_n) \in S \equiv S_1 \times \cdots \times S_n$. Given any profile $\mathbf{s}$, the payoff to player i is specified by

$$u_i(s_i, \mathbf{s}_{-i}) = R_i(s_i, \mathbf{s}_{-i}) - C_i(s_i), \tag{2}$$

in which R_i is the expected benefit from correct intrusion detection and C_i captures the resource cost.

4.2.1. Existence and convergence analysis

Under the standing assumptions that each $S_i \subset \mathbb{R}^m$ is nonempty, compact and convex, and that $u_i(s_i, \mathbf{s}_{-i})$ is continuous in $\mathbf{s}$ and quasi-concave in s_i, one may invoke the classic result of [28] to assert that a Nash equilibrium exists in our game.

Nash Equilibrium Existence: If, for every $i \in N$, the set S_i is nonempty, compact and convex, and $u_i(\cdot, \mathbf{s}_{-i})$ is continuous and quasi-concave on S_i, then there exists at least one Nash equilibrium.

Sketch of Proof: This proof directly applies the Glicksberg fixed point theorem: mapping the joint strategy to its best response set, using the non-empty, convex value and upper semi-continuity of the mapping, it can be seen that there is a fixed point, i.e., a Nash equilibrium. See Appendix A for the full proof.

4.3. User roles and threat model

In the MCS scenario, users are divided into dishonest users or honest users.

4.3.1. Dishonest users ($\mathcal{A}$)

These users may carry out the following malicious activities: *Confidentiality Attacks:* Stealing or disclosing sensitive information from other participants, violating privacy. *Integrity Attacks:* Modifying, fabricating, or omitting sensing data (including model poisoning) to mislead system inference. *Availability Attacks:* Overloading the system through request flooding, delaying data transmission, or launching Denial-of-Service (DoS) attacks to degrade real-time performance. *Collusive Attacks:* Coordinating with other dishonest users to submit consistently false data, evading individual detection mechanisms.

4.3.2. Honest users ($\mathcal{B}$)

The requester $\mathcal{Q}$ (which may have multiple identities) sends traffic information $s_i = \{a, b, c, d\}$ to a TTP.

Task Release: Users define the attributes of task T according to actual application requirements, specify the task objectives, data types, expected outputs, and other necessary requirements and send them to TTP.

Committee Selection: TTP selects a committee according to the task-specific constraint D_1 by solving:

$$C_i = \arg\min_{C_j \in \mathcal{C}} \left\{ \mathrm{dist}(C_j, T) \mid F(C_j) \cap D_1 \neq \emptyset \right\} \tag{3}$$

Here, $F(C_j)$ denotes the feature set of candidate committee member C_j, and $\mathrm{dist}(C_j, T)$ quantifies the "distance" (which may represent geographic distance, response time, or a matching metric) between the candidate and the task T. The condition $F(C_j) \cap D_1 \neq \emptyset$ ensures that only those candidates whose features meet the task-specific requirements D_1 are considered.

User Assignment: Under an additional constraint D_2, the TTP assigns the task to a subset of nearby users by solving:

$$u^* = \arg\min_{u_i \in U} \left\{ \mathrm{dist}(u_i, T) \mid F(u_i) \cap D_2 \neq \emptyset \right\} \tag{4}$$

In this expression, $F(u_i)$ represents the attribute set of user u_i, and $\mathrm{dist}(u_i, T)$ measures the suitability or proximity of user u_i with respect to task T. The constraint $F(u_i) \cap D_2 \neq \emptyset$ guarantees that only users satisfying the necessary conditions D_2 are eligible for assignment.

Local Processing: Once the task is assigned, each selected user u_i performs local processing on the data associated with the task T. Let $D(T)$ denote the raw data relevant to T (e.g., sensor readings, images, or other measurements). Each user applies a local processing function $f_{\mathrm{local}}(\cdot)$ to $D(T)$ to extract or transform the raw data into a more meaningful or compact result. The output of this process for a user u_i is denoted as:

$$Re_i = f_{\mathrm{local}}(D(T)) \tag{5}$$

Committee Voting: After local processing, each generated result Re_i is submitted to the previously selected committee C_i. The committee members then vote on Re_i to validate its authenticity. For each committee member $C_j \in C_i$, we define the voting outcome as:

$$v_{ij} = f_{\mathrm{vote}}(Re_i, C_j) \tag{6}$$

Where $v_{ij} = 1$ indicates that C_j deems Re_i credible, and $v_{ij} = 0$ indicates the opposite. Ultimately, the committee confirms the authenticity of Re_i through a majority voting mechanism:

$$V(Re_i) = \begin{cases} 1, & \text{if } \sum_{C_j \in C_i} v_{ij} \geq \frac{|C_i|}{2} \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

Only when $V(Re_i) = 1$ is the result Re_i regarded as authentic. These local results Re_i are subsequently transmitted to the TTP for aggregation and further evaluation.

4.4. BAL-IDS in DETR

4.4.1. Z-Score

Z-Score normalization is a commonly used data preprocessing method, which aims to transform each feature into a standard normal distribution with a mean of 0 and a standard deviation of 1. Through this transformation, we eliminate the influence of the differences between different features Re_i or T and obtain x. Where x represents the input feature vector obtained by normalization and sequential concatenation.

4.4.2. Autoencoder

An autoencoder is a self-supervised learning model designed to learn a compact, low-dimensional representation z of the input data x, and then reconstruct x as x' via a decoder. we first calculate the mean μ and standard deviation σ of the reconstruction error of normal samples and set the threshold to μ+3σ (corresponding to a confidence interval of about 99%) to ensure a low false alarm rate. The training objective is to minimize the reconstruction error. The autoencoder used in this study is a symmetric three-layer fully connected network (512–256–128), with ReLU as the activation function, Dropout (0.3) and Batch Normalization after the hidden layer, and L2 regularization (1×10^{-4}) applied to the weights; Early Stopping is enabled during training (the validation set loss does not decrease for 10 consecutive epochs) to enhance the robustness and reproducibility of the model. Given an input vector $x \in \mathbb{R}^d$, the encoder maps x to a latent space representation $z \in \mathbb{R}^l$. The pseudocode is shown in Algorithm 1. The forward propagation consists of two stages:

Encoding:

$$h^{(1)} = \mathrm{ReLU}(W^{(1)}x + b^{(1)}) \tag{8}$$

$$z = W^{(2)}h^{(1)} + b^{(2)} \tag{9}$$

Where,

- $W^{(1)} \in \mathbb{R}^{h \times d}$ is the weight matrix from the input to the hidden layer,
- $b^{(1)} \in \mathbb{R}^h$ is the corresponding bias,
- h is the number of hidden units,
- $W^{(2)} \in \mathbb{R}^{l \times h}$ and $b^{(2)} \in \mathbb{R}^l$ are the weight and bias for mapping to the latent space.

Decoding: The decoder reconstructs the input from the latent representation.

$$h^{(2)} = \mathrm{ReLU}(W^{(3)}z + b^{(3)}) \tag{10}$$

$$x' = W^{(4)}h^{(2)} + b^{(4)} \tag{11}$$

Where,

- $W^{(3)} \in \mathbb{R}^{h \times l}$ and $b^{(3)} \in \mathbb{R}^h$ are the weights and biases for the decoder's hidden layer,

- $W^{(4)} \in \mathbb{R}^{d\times h}$ and $b^{(4)} \in \mathbb{R}^{d}$ are the weights and biases for reconstructing the output.

The training objective is to minimize the reconstruction error, commonly defined by the Mean Squared Error (MSE).

$$L(x, x') = \frac{1}{N}\sum_{i=1}^{N} \|x^{(i)} - x'^{(i)}\|^2 \tag{12}$$

Algorithm 1 Autoencoder

Require: Normal data $\mathcal{X}_{\text{norm}}$, all data $\mathcal{X}$, AE parameters, learning rate η, patience P, max epochs E
Ensure: Trained weights θ, threshold τ
Initialize θ randomly, best_val_loss $\leftarrow \infty$, wait $\leftarrow 0$
Compute reconstruction errors e_i on $\mathcal{X}_{\text{norm}}$
Set $\mu \leftarrow \text{mean}(e_i)$, $\sigma \leftarrow \text{std}(e_i)$, $\tau \leftarrow \mu + 3\sigma$
for $t = 1$ **to** E **do**
  Sample minibatch from $\mathcal{X}$
  Forward pass through AE, compute batch MSE L
  Backpropagate and update $\theta \leftarrow \theta - \eta\nabla_\theta L$
  Evaluate validation loss L_{val}
  if $L_{\text{val}} <$ best_val_loss **then**
    best_val_loss $\leftarrow L_{\text{val}}$, wait $\leftarrow 0$
  else
    wait $\leftarrow$ wait + 1
    if wait $\geq P$ **then break**
    end if
  end if
end for
return θ, τ

For a more detailed description, we introduce intermediate variables for backpropagation. Define the output layer error for each sample as (13), Propagating this error back through the decoder hidden layer (14). Then, propagating to the latent layer (15). And similarly, backpropagating through the encoder's hidden layer (16).

$$\delta^{(4)} = \frac{\partial L}{\partial x'} = \frac{2}{N}(x' - x) \tag{13}$$

$$\delta^{(3)} = (W^{(4)})^T\delta^{(4)} \odot \text{ReLU}'(W^{(3)}z + b^{(3)}) \tag{14}$$

$$\delta^{(2)} = (W^{(3)})^T\delta^{(3)} \tag{15}$$

$$\delta^{(1)} = (W^{(2)})^T\delta^{(2)} \odot \text{ReLU}'(W^{(1)}x + b^{(1)}) \tag{16}$$

The gradient for the parameters of each layer is given by (17), with $a^{(0)} = x$, $a^{(1)} = h^{(1)}$, $a^{(2)} = z$, and $a^{(3)} = h^{(2)}$. Parameter updates use gradient descent (18).

$$\frac{\partial L}{\partial W^{(k)}} = \delta^{(k)}(a^{(k-1)})^T, \quad k = 1, 2, 3, 4 \tag{17}$$

$$\theta \leftarrow \theta - \eta\frac{\partial L}{\partial \theta} \tag{18}$$

Where θ represents the parameters of each layer and η is the learning rate.

4.5. One-class SVM for anomaly detection

The OCSVM is designed to build a decision boundary that encloses most of the normal samples, while anomalous samples fall outside this boundary. The pseudocode is shown in Algorithm 2.

4.5.1. Hyperparameter definition

OCSVM requires the selection of two hyperparameters: ν and γ. The hyperparameter $\nu \in (0, 1)$ controls the upper bound on the fraction of training errors and the lower bound on the fraction of support vectors. In theory, the optimal solution satisfies:

$$\nu \geq \frac{1}{N}\sum_{i=1}^{N} I\left(\alpha_i = \frac{1}{\nu N}\right) \tag{19}$$

Where $I(\cdot)$ is the indicator function and α_i are the Lagrange multipliers. The hyperparameter γ is used in the RBF kernel to control the width and decay rate. The RBF kernel is given by

$$K(z, z') = \exp(-\gamma\|z - z'\|_2^2) \tag{20}$$

A larger γ leads to a rapidly decaying kernel (a more complex decision boundary), while a smaller γ results in a smoother boundary.

Algorithm 2 OCSVM

Require: Latent representations $\{z^{(i)}\}_{i=1}^{N}$, ν, γ
Ensure: $\{\alpha_i\}$, ρ, $f(\cdot)$
Compute RBF kernel matrix:
$K_{ij} \leftarrow \exp\left(-\gamma\|z^{(i)} - z^{(j)}\|^2\right)$
Solve the dual:
$\max_{\{\alpha_i\}} \ \rho - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j K_{ij}$
s.t. $0 \leq \alpha_i \leq \frac{1}{\nu N}$, $\sum_i \alpha_i = 1$
Recover ρ from Karush–Kuhn–Tucker conditions
Define decision function:
$f(z) = \sum_{i=1}^{N}\alpha_i \exp\left(-\gamma\|z^{(i)} - z\|^2\right) - \rho$
return $\{\alpha_i\}$, ρ, $f(\cdot)$

4.5.2. Model formulation and optimization

Given the latent representation $z \in \mathbb{R}^l$ from the autoencoder, the primal optimization problem for OCSVM is formulated as (22).

$$\text{s.t. } w^\top\phi(z^{(i)}) \geq \rho - \xi_i, \xi_i \geq 0 \Longrightarrow \tag{21}$$

$$\min_{w,\xi,\rho} \frac{1}{2}\|w\|^2 + \frac{1}{\nu N}\sum_{i=1}^{N}\xi_i - \rho \tag{22}$$

Here, w is the weight vector defining the decision boundary, $\phi(z)$ denotes the kernel mapping to a higher-dimensional feature space, ξ_i are slack variables, and ρ is the bias term. The hyperparameter ν controls the acceptable proportion of outliers.

Using the RBF kernel $K(z, z') = \exp(-\gamma\|z - z'\|^2)$. we transform the problem into its dual form via Lagrange multipliers (24).

$$\text{s.t. } 0 \leq \alpha_i \leq \frac{1}{\nu N}, \sum_{i=1}^{N}\alpha_i = 1 \Longrightarrow \tag{23}$$

$$\max_{\{\alpha_i\}} \rho - \frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j K(z^{(i)}, z^{(j)}) \tag{24}$$

where α_i are the Lagrange multipliers. After training, the decision function is defined as (25):

$$f(z) = \text{sign}\left(\sum_{i=1}^{N}\alpha_i K(z^{(i)}, z) - \rho\right) \tag{25}$$

with $f(z) \geq 0$ indicating a "normal" sample and $f(z) < 0$ indicating an "anomalous" sample. For anomaly scoring, the decision score is defined by (26) and the anomaly score is usually taken as the negative (27):

$$\text{score}(z) = \sum_{i=1}^{N}\alpha_i K(z^{(i)}, z) - \rho \tag{26}$$

$$\text{anomaly\_score}(z) = -\text{score}(z) \tag{27}$$

This approach effectively models the distribution of normal data and tolerates a fraction of outliers by incorporating slack variables.

4.6. Bayesian optimization and local surrogate optimization

Hyperparameter tuning is critical for OCSVM performance. In this context, we seek the optimal values for ν and γ that maximize the F_1 score on a validation set.

4.7. Bayesian optimization

Bayesian optimization uses a Gaussian Process (GP) to model the objective function, allowing efficient exploration of the hyperparameter space with a limited number of evaluations.

First, define the F_1 score on the validation set.

$$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{28}$$

with Precision $= \frac{TP}{TP+FP}$ and Recall $= \frac{TP}{TP+FN}$. The tuning objective is then:

$$\text{Obj}(\nu, \gamma) = 1 - F_1 \tag{29}$$

Assume the objective function $f(\theta)$, where $\theta = [\nu, \gamma]$, follows a Gaussian Process.

$$f(\theta) \sim \mathcal{GP}(m(\theta), k(\theta, \theta')) \tag{30}$$

with $m(\theta)$ as the mean function and $k(\theta, \theta')$ as the covariance function. The Expected Improvement (EI) acquisition function is used to determine the next evaluation point.

$$\text{EI}(\theta) = \mathbb{E}\Big[\max(0, f_{\min} - f(\theta))\Big] \tag{31}$$

where $f_{\min}$ is the best (lowest) objective function value observed so far. This process enables global exploration of the parameter space.

4.8. Local surrogate optimization

After obtaining an initial optimal parameter $\theta_0 = [\nu_0, \gamma_0]$ via Bayesian optimization, local surrogate optimization refines the hyperparameters within the neighborhood of θ_0.

Candidates are generated by uniformly perturbing θ_0.

$$\theta_{\text{candidate}} = \theta_0 + \delta, \quad \delta \sim \mathcal{U}(-r, r) \tag{32}$$

where r is the local search radius. The candidate with the lowest objective value, Obj(θ), is selected as the final hyperparameter configuration.

Algorithm 3 Bayesian Optimization with Local Surrogate Refinement

Require: Objective Obj(θ), initial samples $\{\theta_i\}_{i=1}^n$, GP prior (m, k), BO iterations B, local radius r, local samples M
Ensure: Final hyperparameters θ^*

```
D ← {(θ_i, Obj(θ_i))}_{i=1}^n
for t = 1 to B do
    Fit GP to D with mean m and kernel k
    Select θ_next = arg max_θ EI(θ | D)
    Evaluate y_next = Obj(θ_next)
    D ← D ∪ {(θ_next, y_next)}
end for
θ_0 ← arg min_{(θ,y)∈D} y
θ* ← θ_0, y* ← Obj(θ_0)
for i = 1 to M do
    Sample δ ~ U([−r, r]^2)
    θ_cand ← θ_0 + δ
    y_cand ← Obj(θ_cand)
    if y_cand < y* then
        θ* ← θ_cand, y* ← y_cand
    end if
end for
return θ*
```

4.9. Final model

After hyperparameter tuning via Bayesian and local surrogate optimization, the optimal hyperparameters ν^* and γ^* are obtained. These are used to train the final One-Class SVM on the latent representations produced by the autoencoder.

$$f(z) = \text{sign}\Big(\sum_{i=1}^{N} \alpha_i K(z^{(i)}, z) - \rho\Big) \tag{33}$$

For a new sample z_{new}, the decision score is computed as (34). And the anomaly score is given by

$$\text{score}(z_{\text{new}}) = \sum_{i=1}^{N} \alpha_i K(z^{(i)}, z_{\text{new}}) - \rho \tag{34}$$

$$\text{anomaly\_score}(z_{\text{new}}) = -\,\text{score}(z_{\text{new}}) \tag{35}$$

A predefined threshold is then used to classify z_{new} as "normal" if $f(z_{\text{new}}) \geq 0$ or "anomalous" if $f(z_{\text{new}}) < 0$.

4.10. Prediction and risk evaluation

Incentive Provision and Credibility Calculation: After the committee evaluates and verifies Re_i, it is forwarded to TTP. Once BAL-IDS confirms that Re_i is risk-free, TTP provides incentives to the contributing users. In this process, the incentive is composed of a positive component I^+ and a negative component I^-, which are weighted by parameters θ^+ and θ^-, respectively. The overall incentive $\mathcal{I}$ is calculated as:

$$\mathcal{I}_i = \theta^+ I_i^+ + \theta^- I_i^- \tag{36}$$

Then, by combining this incentive with the cumulative incentive $\mathcal{H}$, the overall credibility $\mathcal{Y}_i$ of all users is updated as:

$$\mathcal{Y}_i = \mathcal{I}_i + \mathcal{H}_i \tag{37}$$

We define additional parameters. Where ψ is the preset value of credibility:

$$\mathcal{Q} = \arccos(\psi) \tag{38}$$

To improve clarity and consistency, we redefine the credibility function $\mathcal{N}(Y_i)$ over normalized intervals. Let P_1 and P_2 be two thresholds satisfying $0 < P_2 < P_1 < 1$. Then, the credibility function is given by:

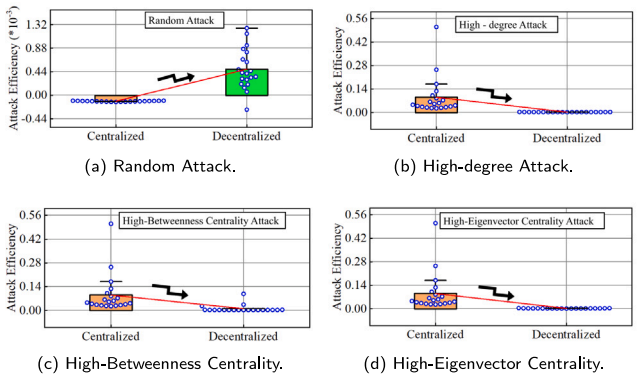
$$\mathcal{N}(\mathcal{Y}_i) = \begin{cases} \cos\left(\mathcal{Q}\frac{1-\mathcal{Y}_i}{1-P_1}\right) & \text{if } \mathcal{Y}_i \in [1, P_1] \\ \cos\left(\mathcal{Q}\frac{\mathcal{Y}_i - P_1}{P_2 - P_1}\right) & \text{if } \mathcal{Y}_i \in (P_1, P_2] \\ 1 - \frac{(1-\psi)\cdot\mathcal{Y}_i}{P_2} & \text{if } \mathcal{Y}_i \in (P_2, 0] \end{cases} \tag{39}$$

TTP decides whether to reupdate the committee based on the $\mathcal{N}(\mathcal{Y}_i)$ of the committee members. When $\mathcal{N}(\mathcal{Y}_i)$ of a member is lower than the preset threshold P_2, the member will be regarded as a risky member and removed. In order to maintain the overall reputation and risk control of the committee, TTP introduces the committee update factor Δ to quantify the proportion of risky members in the current committee:

$$\Delta = \frac{\sum_{i=1}^{N} \mathbb{1}_{\{\mathcal{N}(\mathcal{Y}_i) < P_2\}}}{N} \tag{40}$$

Where, N is the total number of members of the current committee. When Δ exceeds the preset ratio threshold Δ_{crit}, TTP will initiate a comprehensive update of the committee, that is, remove all risky members and introduce new candidates to optimize the committee structure. During the update process, the contribution value of the newly added committee members is determined based on their historical contributions and current risk assessments:

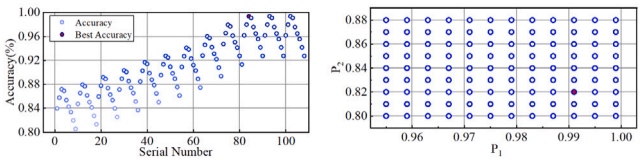
$$\mathcal{Y}_i^{\text{new}} = \lambda \mathcal{Y}_i + (1 - \lambda)\left[1 - \mathcal{N}(\mathcal{Y}_i)\right] \tag{41}$$



(a) Random Attack. (b) High-degree Attack.

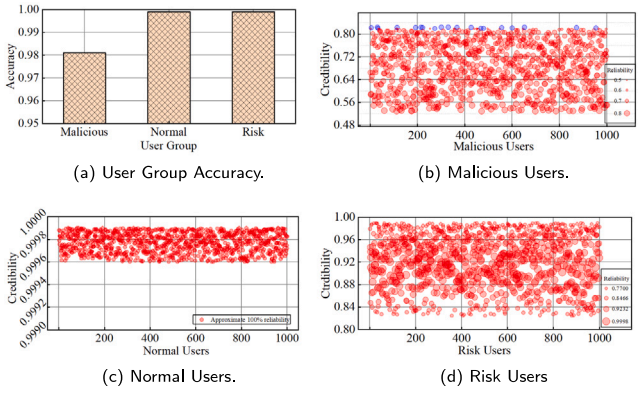
(c) High-Betweenness Centrality. (d) High-Eigenvector Centrality.

Fig. 4. Decentralized vs. centralized security comparison and evaluation.



(a) Accuracy at Different Settings. (b) Parameter details.

Fig. 5. Different parameter settings accuracy and Parameter details.



(a) User Group Accuracy. (b) Malicious Users.

(c) Normal Users. (d) Risk Users

Fig. 6. Accuracy of detecting different user identities.

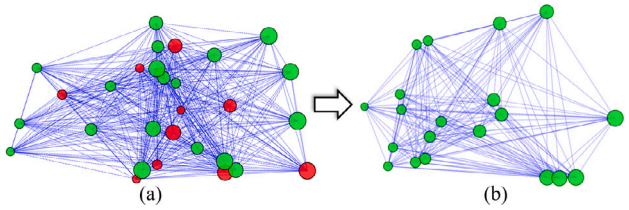


Fig. 7. Committee visualization.

The contribution value is updated by taking into account the historical contribution of the member and the current risk level. Among them, $\mathcal{Y}_i$ represents the historical contribution of the member, and $\mathcal{N}(\mathcal{Y}_i)$ (assessed in combination with BAL-IDS) reflects the degree of credibility. The higher the value, the greater the risk. Therefore, $1 - \mathcal{N}(\mathcal{Y}_i)$ represents the risk compensation effect. The weight parameter λ (value range [0,1]) is used to balance the influence of the two: when λ is large, it focuses on historical contributions, and when it is small, it focuses on risk compensation. In this way, TTP ensures that only members with high contributions and low risks can maintain high credibility, thereby ensuring the efficient operation of the committee.

5. Analysis

This section presents an experimental validation of decentralized DETR with the goal of evaluating the security level and performance metrics.

5.1. Security assessment

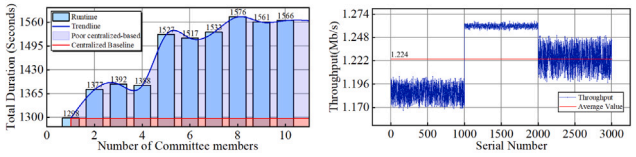
We utilize NetworkX to generate and analyze two primary network topologies: Centralized and Decentralized. To effectively model various attack scenarios, we employ Dask for parallel processing, which allows us to distribute computational tasks across multiple processes. All parameter setting tables 2 and pseudocode examples 4, 5 are organized in Appendix B. The implementation involves defining several attack strategies, including Random Attack, High-Degree Attack, High-Betweenness Centrality Attack, and High-Eigenvector Centrality Attack. These strategies are systematically applied to the network topology at varying levels of attack power (i.e., the number of target nodes) for each network type. To evaluate the effects of these attacks, we employed several assessment metrics. Efficiency Loss: the reduction in the global efficiency of a network following an attack. Attack Efficiency: This is calculated by dividing the efficiency loss by the attack power. The simulation compares how centralized and decentralized networks withstand various attack strategies. The simulation compares how centralized and decentralized networks withstand various attack strategies and intensities. Fig. 4 shows decentralized networks are generally more resilient than centralized networks. In particular, in random attacks, reducing the number of nodes can actually improve network efficiency (by reducing the threat to central nodes), but this improvement sacrifices the integrity of the network structure; under other attacks, the efficiency of centralized networks decreases more significantly due to their centralized structure.

5.2. DETR performance

After confirming that the decentralized architecture is superior to the centralized architecture, we verified the superiority of DETR and the committee mechanism in the mobile sensing scenario. We used the KonIQ-10k dataset to simulate 200 users (10 images per user), covering three types of image quality: normal, risky, and malicious users. CONTRIQUE was used to extract image quality features, and the parameters P_1 and P_2 were fine-tuned based on the user classification accuracy. The results show that among the 108 configurations, No. 82 ($P_1 = 0.991, P_2 = 0.82$) performed best with an accuracy of 99.36% (see Fig. 5(a)).

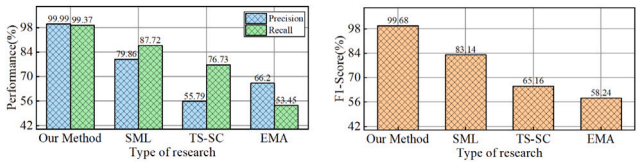
In addition, we evaluated the contribution values of different user groups. The system has a detection accuracy of 98.1% for malicious users, 99.9% (±0.1%) for normal and risky users, and an overall average accuracy of 99.37% (see Fig. 6(a)). Fig. 6(b) shows the malicious user prediction, where the bubble size represents the prediction reliability (larger means higher), and blue and red represent wrong and correct predictions, respectively. The overall results prove that this method has a very low false positive rate in efficiently detecting malicious users.

Figs. 6(c) and 6(d) show normal and risky users, respectively. No prediction errors were observed, but predictions near the decision boundary have lower reliability. As shown in Figs. 7(a) and 7(b), after calculating the contribution values of all users, DETR will remove members with lower contribution values from the committee. When the number of members with low contribution values reaches the preset update threshold, the system will trigger the global update strategy and reselect users with higher contribution values to join the committee to optimize the overall member structure and improve the operating



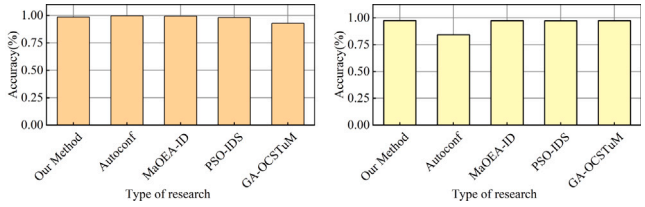
(a) Total Duration. (b) Throughput.

Fig. 8. Total duration and throughput.



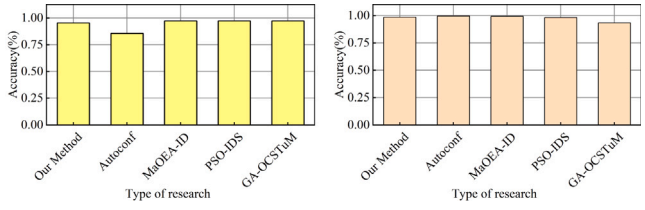
(a) Precision and Recall. (b) F1-Scores.

Fig. 9. Performance comparison of DETR with other works.



(a) Accuracy in NSL-KDD (b) F1-Score in NSL-KDD.

Fig. 10. Performance in NSL-KDD.



(a) Accuracy in ToN-IoT. (b) F1-Score in ToN-IoT.

Fig. 11. Performance in ToN-IoT.

efficiency of the committee. Fig. 8 shows the total running time and throughput of DETR. Fig. 8(a) shows that as the number of committee members increases, the total delay time gradually increases until it stabilizes at around 1550 s. Fig. 8(b) shows that the throughput caused by different types of users is different, with an average throughput of 1.224 Mb/s.

We use precision and recall to evaluate the performance of the proposed method and existing research [18–20]. As shown in Fig. 9(a), our method achieves 99.99% precision and 99.37% recall, far exceeding the comparison method. Further combined with the F1 score (Fig. 9(b)), it proves that our method has higher robustness and reliability in identifying and retaining trusted users.

5.3. BAL-IDS performance

We tested the proposed BAL-IDS using NSL-KDD and ToN-IoT datasets, aiming to effectively resist 51% attacks, sybil attacks, and other network attacks that may occur in DETR. We divide the dataset into a training set and a test set in a 7:3 ratio. The attack types included

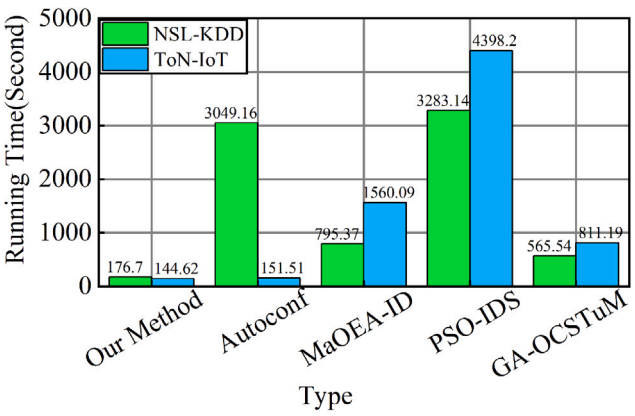


Fig. 12. Comparison of computational cost of BAL-IDS and other works.

in the training set are classified as “known attacks”, the attack types that appear exclusively in the test set are referred to as “unknown attacks”. Based on this classification, we conduct a leave-one-attack-out experiment for each attack type. During the training phase, the specific attack type is excluded, and the model is trained solely with the remaining known attacks. In the testing phase, we specifically evaluate the detection accuracy of BAL-IDS for the “unknown attack”, thereby thoroughly assessing its generalization capability in scenarios involving unknown threats. In the same dataset, the methods we compared with are: Autoconf based on improved Bayesian optimization [29], MaOEA-ID based on bidirectional differential evolution, [30]. PSO-IDS based on particle swarm evolution [31], GA-OCSTuM based on genetic evolution [32]. Under the premise of using only normal traffic for training, Figs. 10 and 11 show the accuracy and F1 scores of BAL-IDS and similar advanced methods on NSL-KDD and ToN-IoT datasets respectively.

As shown in Fig. 10, when trained with normal traffic only, the accuracy and F1 score of BAL-IDS on NSL-KDD are 98.9% and 97.6%, respectively, which is on par with MaOEA-ID (98.9%, 97.6%) and PSO-IDS (98.8%, 97.5%), and significantly better than GA-OCSTuM (91.2%, 97.6%) and Autoconf (99.7%, 85.4%). As shown in Fig. 11, on ToN-IoT, its accuracy and F1 score reach 95.8% and 99.2%, which is on par with MaOEA-ID (97.6%, 99.9%) and PSO-IDS (97.6%, 98.8%), and significantly better than GA-OCSTuM (97.2%, 92.4%) and Autoconf (86.3%, 99.9%), showing excellent stability in the balance between precision and recall, further verifying the applicability and robustness of BAL-IDS for different IoT scenarios. In addition, as shown in Fig. 12, under the same data processing conditions, the computational overhead of BAL-IDS is stably controlled between 144.62 and 176.7 s, while other advanced methods have large fluctuations on different datasets, with the highest computational overhead even reaching 4398.2 s. It can be seen that BAL-IDS significantly reduces the computational overhead while maintaining the same detection performance, further demonstrating its advancement and superiority in practical applications.

6. Conclusion

This study investigates the challenges of assessing cloud service reliability in extreme mobile-sensor environments. Under scenarios such as malicious disconnections, conventional cloud-centric models may compromise sensing integrity and data credibility. To address these issues, we introduce DETR, a decentralized committee framework for robust and flexible trust evaluation. Moreover, we design BAL-IDS, an IDS based on a time-constraint matrix and Markov-chain traffic modeling. BAL-IDS employs autoencoders to extract latent features from high-dimensional data, and integrates Bayesian evolutionary strategies

Table 2
Network topology generation parameters.

Topology Model	Parameters
Erdős–Rényi Random Graph	$N = 100,\ p = 0.05$
Barabási–Albert Scale-Free	$N = 100,\ m = 5$
Watts–Strogatz Small-World	$N = 100,\ k = 4,\ \beta = 0.1$

with local-agent optimization to efficiently solve the OCSVM problem and detect anomalies. For honest-user selection, we propose a game-theoretic credibility metric that combines risk indicators and data contribution. By dynamically adjusting weights, the system prioritizes high–reliability contributors. If a committee member's credibility falls below a threshold, they are removed; vacancies are filled based on updated credibility scores to maintain stability and efficiency. Experimental results demonstrate that DETR and BAL-IDS outperform existing advanced methods. Future work will focus on dynamic node collaboration, lightweight online incremental learning, and integrating federated learning with differential privacy to further enhance system stability, efficiency, and security.

CRediT authorship contribution statement

Chen Zhang: Writing – review & editing, Writing – original draft, Formal analysis, Data curation. **Zhuotao Lian:** Supervision, Methodology. **Weiyu Wang:** Data curation. **Huakun Huang:** Software, Resources. **Chunhua Su:** Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Convergence of the recursive best response dynamics

We further examine the iterative process based on the recursive best response (BR) dynamics:

$$s_i^{(t+1)} \in \arg\max_{s_i \in S_i} u_i\left(s_i, \mathbf{s}_{-i}^{(t)}\right),$$

Where $\mathbf{s}^{(t)} = (s_1^{(t)}, \ldots, s_n^{(t)})$. Under the assumption that each u_i is Lipschitz continuous in $\mathbf{s}$, and that the best response mapping is single-valued and satisfies a contraction condition, one can show that this iteration converges to the equilibrium at an exponential rate. Specifically, if the Lipschitz constant is $L < 1$, then

$$\left\|\mathbf{s}^{(t)} - \mathbf{s}^*\right\| \le L^t \left\|\mathbf{s}^{(0)} - \mathbf{s}^*\right\|.$$

Moreover, the number of iterations required to reach an ε-neighborhood of the equilibrium can be bounded by

$$\left\lceil \frac{\log\left(\varepsilon / \|\mathbf{s}^{(0)} - \mathbf{s}^*\|\right)}{\log L} \right\rceil.$$

Appendix B. Experimental parameters and pseudocode

See Table 2.

Data availability

Data will be made available on request.

Algorithm 4 Generate Topology and Assign Link Attributes

Input: models = {ER, BA, WS}, parameters as in Table C.1
for all model in models **do**
  if model = ER **then**
    $G \leftarrow$ GenerateErdosRenyi($N = 100, p = 0.05$)
  else if model = BA **then**
    $G \leftarrow$ GenerateBarabasiAlbert($N = 100, m = 5$)
  else if model = WS **then**
    $G \leftarrow$ GenerateWattsStrogatz($N = 100, k = 4, \beta = 0.1$)
  end if
  for all edge (u, v) in G **do**
    $\text{bandwidth}_{uv} \leftarrow$ Uniform(10, 100) Mbps
    $\text{delay}_{uv} \leftarrow \max\left(0, \text{Gaussian}(10, 2)\right)$ ms
  end for
end for

Algorithm 5 SimPy Traffic Generation and Attack Injection

Input: $\lambda = 0.2$ flows/s, attack_rates = {1,5,10,20,50}, SIM_TIME
procedure SIMULATE(mode, attack_rate)
  Initialize SimPy environment E
  Instantiate *CentralizedServer* and *DecentralizedNetwork*
  $t \leftarrow 0$
  while $t <$ SIM_TIME **do**
    $\Delta t_{\text{legit}} \leftarrow \text{Exp}(\lambda)$
    Schedule PROCESSREQUEST(mode, service_time=Δt_{legit}) at $t + \Delta t_{\text{legit}}$
    $\Delta t_{\text{attack}} \leftarrow$ Exp(attack_rate)
    Schedule PROCESSREQUEST(mode, service_time=0) at $t + \Delta t_{\text{attack}}$
    Advance t to the next event time
  end while
  Run E until SIM_TIME and record dropped requests
end procedure

References

[1] W. Li, W.j. Wu, H.-m. Wang, X.q. Cheng, H.j. Chen, Z.h. Zhou, R. Ding, Crowd intelligence in AI 2.0 era, Front. Inf. Technol. Electron. Eng. 18 (1) (2017) 15–43, http://dx.doi.org/10.1631/FITEE.1601859.

[2] B. Zhao, X. Liu, W.N. Chen, R.H. Deng, CrowdFL: Privacy-preserving mobile crowdsensing system via federated learning, IEEE Trans. Mob. Comput. 22 (8) (2023) 4607–4619, http://dx.doi.org/10.1109/TMC.2022.3157603.

[3] Y. Liu, L. Kong, G. Chen, Data-oriented mobile crowdsensing: A comprehensive survey, IEEE Commun. Surv. Tutor. 21 (3) (2019) 2849–2885, http://dx.doi.org/10.1109/COMST.2019.2910855.

[4] A.R. Kulkarni, N. Kumar, K.R. Rao, Efficacy of bluetooth-based data collection for road traffic analysis and visualization using big data analytics, Big Data Min. Anal. 6 (2) (2023) 139–153.

[5] X. Yu, C. Wang, L. Xu, C. Wu, Z. Wang, Y. He, W. Wang, When connected and automated vehicles meet mobile crowdsensing: A perception and transmission framework in the metaverse, IEEE Veh. Technol. Mag. (2023).

[6] D. Suhag, V. Jha, A comprehensive survey on mobile crowdsensing systems, J. Syst. Archit. 142 (2023) 102952, http://dx.doi.org/10.1016/j.sysarc.2023.102952.

[7] I. Krontiris, T. Dimitriou, A platform for privacy protection of data requesters and data providers in mobile sensing, Comput. Commun. 65 (2015) 43–54, http://dx.doi.org/10.1016/j.comcom.2015.02.005.

[8] I. Al-hammadi, M. Li, S.M.N. Islam, E. Al-Mosharea, Collaborative computation offloading for scheduling emergency tasks in SDN-based mobile edge computing networks, Comput. Netw. 238 (2024) 110101, http://dx.doi.org/10.1016/j.comnet.2023.110101.

[9] W. Feng, Z. Yan, MCS-Chain: Decentralized and trustworthy mobile crowdsourcing based on blockchain, Future Gener. Comput. Syst. 95 (2019) 649–666, http://dx.doi.org/10.1016/j.future.2019.01.036.

[10] E.T. Martínez Beltrán, M.Q. Pérez, P.M.S. Sánchez, S.L. Bernal, G. Bovet, M.G. Pérez, G.M. Pérez, A.H. Celdrán, Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges, IEEE Commun. Surv. Tutor. 25 (4) (2023) 2983–3013, http://dx.doi.org/10.1109/COMST.2023.3315746.

[11] Y. Liu, H. Li, J. Xiao, H. Jin, Floc: Fingerprint-based indoor localization system under a federated learning updating framework, in: 2019 15th International Conference on Mobile Ad-Hoc and Sensor Networks, MSN, 2019, pp. 113–118, http://dx.doi.org/10.1109/MSN48538.2019.00033.
[12] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, D. Ramage, Federated learning for mobile keyboard prediction, 2018, arXiv preprint arXiv:1811.03604.
[13] M. Zhang, S. Chen, J. Shen, W. Susilo, PrivacyEAFL: Privacy-enhanced aggregation for federated learning in mobile crowdsensing, IEEE Trans. Inf. Forensics Secur. 18 (2023) 5804–5816, http://dx.doi.org/10.1109/TIFS.2023.3315526.
[14] Y. Liu, Z. Yu, B. Guo, Q. Han, J. Su, J. Liao, CrowdOS: A ubiquitous operating system for crowdsourcing and mobile crowd sensing, IEEE Trans. Mob. Comput. 21 (3) (2020) 878–894, http://dx.doi.org/10.1109/TMC.2020.3015750.
[15] H. Jin, L. Su, H. Xiao, K. Nahrstedt, Inception: Incentivizing privacy-preserving data aggregation for mobile crowd sensing systems, in: Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing, 2016, pp. 341–350.
[16] H. Jin, L. Su, D. Chen, K. Nahrstedt, J. Xu, Quality of information aware incentive mechanisms for mobile crowd sensing systems, in: MobiHoc '15: Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing, Association for Computing Machinery, New York, NY, USA, 2015, pp. 167–176, http://dx.doi.org/10.1145/2746285.2746310.
[17] A.I. Middya, S. Roy, Truthful double auction based incentive mechanism for participatory sensing systems, Peer- To- Peer Netw. Appl. (2024) 1–30, http://dx.doi.org/10.1007/s12083-024-01681-3.
[18] S. Zhang, Z. Li, W. Liang, K.C. Li, Z.A. Bhuiyan, Blockchain-based hybrid reliable user selection scheme for task allocation in mobile crowd sensing, IEEE Trans. Netw. Sci. Eng. (2024).
[19] J. Huang, L. Kong, H.N. Dai, W. Ding, L. Cheng, G. Chen, X. Jin, P. Zeng, Blockchain-based mobile crowd sensing in industrial systems, IEEE Trans. Ind. Inform. 16 (10) (2020) 6553–6563.
[20] M. Nakano, A. Takahashi, S. Takahashi, Generalized exponential moving average (EMA) model with particle filtering and anomaly detection, Expert Syst. Appl. 73 (2017) 187–200, http://dx.doi.org/10.1016/j.eswa.2016.12.034.
[21] B. Simon, A. Ortiz, W. Saad, A. Klein, Decentralized online learning in task assignment games for mobile crowdsensing, IEEE Trans. Commun. (2024).
[22] Q. Deng, Q. Zuo, Z. Li, H. Liu, Y. Xie, Blockchain-based reputation privacy preserving for quality-aware worker recruitment scheme in MCS, IEEE/ACM Trans. Netw. (2024).
[23] Z. Wang, H. Xiong, J. Zhang, S. Yang, M. Boukhechba, D. Zhang, L.E. Barnes, D. Dou, From personalized medicine to population health: a survey of mhealth sensing techniques, IEEE Internet Things J. 9 (17) (2022) 15413–15434.
[24] A. Capponi, C. Fiandrino, B. Kantarci, L. Foschini, D. Kliazovich, P. Bouvry, A survey on mobile crowdsensing systems: Challenges, solutions, and opportunities, IEEE Commun. Surv. Tutor. 21 (3) (2019) 2419–2465.
[25] A. Longo, M. Zappatore, M. Bochicchio, S.B. Navathe, Crowd-sourced data collection for urban monitoring via mobile sensors, ACM Trans. Internet Technol. (TOIT) 18 (1) (2017) 1–21.
[26] L. Foschini, G. Martuscelli, R. Montanari, M. Solimando, Edge-enabled mobile crowdsensing to support effective rewarding for data collection in pandemic events, J. Grid Comput. 19 (3) (2021) 28.
[27] F. Zamora-Martinez, P. Romeu, P. Botella-Rocamora, J. Pardo, On-line learning of indoor temperature forecasting models towards energy efficiency, Energy Build. 83 (2014) 162–172.
[28] A. Albarelli, S.R. Bulo, A. Torsello, M. Pelillo, Matching as a non-cooperative game, in: 2009 IEEE 12Th International Conference on Computer Vision, IEEE, 2009, pp. 1319–1326.
[29] L.K. Shar, A. Goknil, E.J. Husom, S. Sen, Y.N. Tun, K. Kim, Autoconf: Automated configuration of unsupervised learning systems using metamorphic testing and bayesian optimization, in: 2023 38th IEEE/ACM International Conference on Automated Software Engineering, ASE, IEEE, 2023, pp. 1326–1338.
[30] J. Zhang, B. Gong, M. Waqas, S. Tu, S. Chen, Many-objective optimization based intrusion detection for in-vehicle network security, IEEE Trans. Intell. Transp. Syst. 24 (12) (2023) 15051–15065.
[31] J. Cui, G. Zhang, Z. Chen, N. Yu, Multi-homed abnormal behavior detection algorithm based on fuzzy particle swarm cluster in user and entity behavior analytics, Sci. Rep. 12 (1) (2022) 22349.
[32] X. Deng, P. Jiang, X. Peng, C. Mi, An intelligent outlier detection method with one class support tucker machine and genetic algorithm toward big sensor data in internet of things, IEEE Trans. Ind. Electron. 66 (6) (2018) 4672–4683.

Zhang Chen received his B.S. degree in Inorganic Non-metallic from Henan University of Technology in 2021. His Master's degree in the Computer Science and Engineering from University of Aizu, Japan, 2023. Currently, he is working towards his Ph.D. degree at the University of Aizu. His research interests mainly focus on Gateway Security, Information Security, and Metaverse, MCS.

Zhuotao Lian received his B.S. in Computer Science from China University of Geosciences, Wuhan, in 2020, and his M.S. and Ph.D. in Computer Science and Engineering from the University of Aizu, Japan, in 2021 and 2024, respectively. He was awarded the NEC C&C Foundation Grants for Researchers in 2023 and the Grant -in-Aid for JSPS Fellows in 2024. Currently, he is a JSPS International Research Fellow at Kyushu University. His research interests include federated learning, differential privacy, blockchain technologies, and AI security.

Weiyu Wang received the B.S. degree in Information Management and Information System from Changzhou University, Changzhou, China, in 2023. She is currently working towards the M.S. degrees with the Department of Computer and Information Systems, the University of Aizu, Aizu-Wakamatsu, Japan. Her research interests include applied cryptography, blockchain technology, and Internet of Things system.

Huakun Huang received the Ph.D. in Computer Science and Engineering from the University of Aizu, Japan, in 2019. He is currently an associate professor with the School of Computer Science and Cyber Engineering, Guangzhou University, China. His current research interests include privacy preserving, machine learning, federated learning and continuous intelligent networks.

Chunhua Su received the B.S. degree for Beijing Electronic and Science Institute in 2003 and received his M.S. and PhD of computer science from Faculty of Engineering, Kyushu University in 2006 and 2009, respectively. He is currently working as a Senior Associate Professor in Division of Computer Science, University of Aizu. He has worked as a postdoctoral fellow in Singapore Management University from 2009–2011 and a research scientist in Cryptography & Security Department of the Institute for Infocomm Research, Singapore from 2011–2013. From 2013–2016, he has worked as an Assistant professor in School of Information Science, Japan Advanced Institute of Science and Technology. From 2016–2017, he worked as Assistant Professor in Graduate School of Engineering, Osaka University. His research interests include cryptanalysis, cryptographic protocols, privacy-preserving technologies in machine learning and IoT security & privacy. He has published more than 100 papers in international journals and conferences.