Federated Learning With Adaptive Regularization for Efficient Edge Data Corruption Detection in Edge Intelligence

Md Rashedul Islam, *Member, IEEE*, Md Palash Uddin, *Member, IEEE*, Yong Xiang, *Senior Member, IEEE*, Kuo-Hui Yeh, *Senior Member, IEEE*, Lu Liu, *Member, IEEE*, and Jonathan Kua, *Member, IEEE*

***Abstract*—Edge intelligence is an emerging distributed computing paradigm driven by the rapid proliferation of Internet of Things (IoT) devices and advancements in edge computing and artificial intelligence. With latency-sensitive data cached across multiple Edge Servers (ESs), efficient Edge Data Integrity Verification (EDIV) has become increasingly critical. Traditional 'challenge-response' EDIV methods incur high computation and communication costs by verifying all ESs indiscriminately, even when only a few may be corrupted. A recent Federated Learning (FL)-based framework partially mitigated this by identifying potentially corrupted ESs early using homogeneous activity data. However, under heterogeneous ES data, this approach suffers from reduced detection accuracy, slower convergence, and no clear guidance for subsequent verification rounds, thereby limiting overall cost reduction. To address these limitations, we propose Federated learning with Adaptive Regularizer-based Edge Data Integrity Verification (FedAR-EDIV), an FL-based framework equipped with an adaptive objective regularization strategy specifically designed to handle heterogeneous data distributions. FedAR-EDIV accurately identifies suspicious ESs during FL training, accelerates convergence, and significantly reduces computation and communication costs in the final EDIV stage. It achieves up to 16× faster communication and 9.1× lower computation cost compared to baseline EDIV methods, while maintaining detection accuracy above 99.78% on heterogeneous KDD99 activity data. Additionally, a dynamic reputation mechanism is applied after each EDIV round to reduce unnecessary audits on trustworthy ESs and allocate more scrutiny to potentially corrupted ones. Theoretical analysis verifies convergence, correctness, and security, while extensive experiments on heterogeneous datasets demonstrate superior accuracy, efficiency, and cost-effectiveness compared to existing methods.**

Received 2 May 2025; revised 9 September 2025; accepted 14 October 2025. Date of publication 22 October 2025; date of current version 8 December 2025. This work was supported in part by the Australian Research Council under Grant LP190100594. Recommended for acceptance by B. Ravindran. *(Corresponding author: Yong Xiang.)*

Md Rashedul Islam, Md Palash Uddin, Yong Xiang, and Jonathan Kua are with the School of Information Technology, Deakin University, Geelong, VIC 3220, Australia (e-mail: md.islam@deakin.edu.au; m.uddin@deakin.edu.au; yong.xiang@deakin.edu.au; jonathan.kua@deakin.edu.au).

Kuo-Hui Yeh is with the School of Industry Academia Innovation, Institute of Artificial Intelligence Innovation, National Yang Ming Chiao Tung University, Hsinchu 300093, Taiwan, and also with the Department of Information Management, College of Management, National Dong Hwa University, Hualien 97401, Taiwan (e-mail: khyeh@nycu.edu.tw).

Lu Liu is with the Department of Computer Science, University of Exeter, EX4 4QJ Exeter, U.K. (e-mail: l.liu3@exeter.ac.uk).

This article has supplementary downloadable material available at https://doi.org/10.1109/TCC.2025.3623572, provided by the authors.

Digital Object Identifier 10.1109/TCC.2025.3623572

***Index Terms*—Edge data integrity, edge computing, adaptive regularization, dynamic reputation, data integrity verification.**

I. Introduction

THE rapid proliferation of edge devices, from smartphones and Internet of Things (IoT) sensors to autonomous vehicles, has outpaced the evolution of traditional computing infrastructures. The number of edge devices is projected to exceed 20 billion globally by 2025 [1], creating substantial challenges in achieving efficient and reliable data processing on a scale. Although cloud computing has provided centralized and scalable solutions over the past decade [2], its ability to ensure low latency and efficient communication diminishes as edge deployments expand [3].Mobile Edge Computing (MEC) has emerged as a compelling alternative that addresses cloud limitations by decentralizing computation and bringing data processing closer to end users [4]. This architectural change is essential for latency-sensitive applications, such as augmented reality, autonomous vehicles, and industrial automation, which require stringent service level agreements (SLAs). As illustrated in Fig. 1, edge caching involves multiple Edge Servers (ESs), such as ES_1, ES_2, and ES_3, that cache data replica from an App Vendor (AV) and deliver them to proximate data users eu_i (where $i \in 1, \ldots, 13$), leveraging edge infrastructure providers like IBM, AWS, or GCP to minimize access latency. In addition to enhancing responsiveness, MEC facilitates edge intelligence, where AI-based analysis is performed directly at the edge. This reduces data transmission overhead, improves resource utilization, and supports real-time, autonomous decision-making. Such paradigms are instrumental in enabling applications such as real-time object detection, predictive maintenance, and adaptive caching strategies without relying heavily on cloud backends.

While MEC enhances computational efficiency and enables intelligent processing at the network edge, it introduces new challenges in decentralized data management. Unlike traditional cloud systems managed by a single AV, MEC consists of distributed ESs that independently store and process data replicas [5]. Ensuring the integrity of these replicas, referred to as Edge Data Integrity (EDI) [6], is critical for maintaining system security and reliability. Most existing EDIV methods rely on the conventional 'challenge-response' mechanism [6], [7], where the AV or a TPA periodically sends challenges to ESs, requiring cryptographic proofs of data integrity. Although widely adopted, this mechanism incurs high computation and communication costs, especially under frequent verifications in large-scale deployments [8]. As the number of ESs increases,

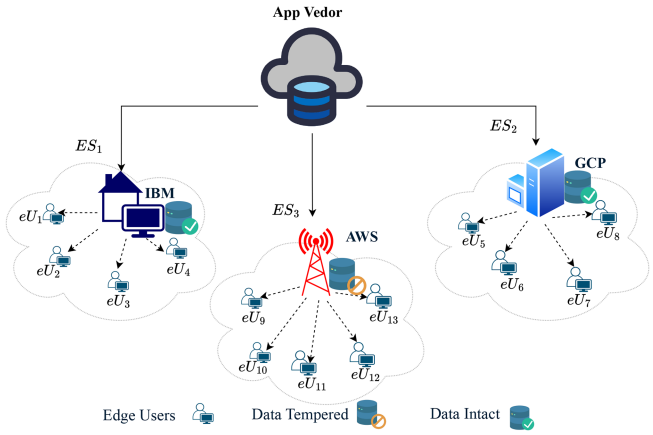


Fig. 1. An illustration of edge caching, where ES1 and ES2 are trusted ESs and ES3 is a potentially corrupted ES.

each ES must individually generate and transmit proofs, escalating the verification load. Moreover, due to the risk of ES tampering, frequent checks are often essential. These interactions, conducted over backbone networks, further strain system resources [8]. Consequently, the repeated communication and processing overhead underscore the need for a scalable, resource-efficient EDI solution that can maintain data trustworthiness without overwhelming the network or its stakeholders.

A promising alternative to traditional verification is machine learning-based anomaly detection, which uses ES activity logs to identify suspicious behavior by comparing it to a learned model of normal activity. However, centralized ML approaches are impractical due to privacy concerns, as aggregating raw log data exposes sensitive information, and are also resource-intensive, limiting their scalability for EDIV. Federated Learning (FL) addresses these concerns by enabling a shared global model training in individual ESs without requiring sensitive activity data transfers to a central repository [9]. Recently, Zhao et al. [10] demonstrated that potentially corrupted ESs can be flagged early by using a global model developed through the FL process, thus reducing the communication cost before the final verification of the EDI. Specifically, individual ESs train a shared global model on their own activity data separately and then send the model updates to the AV for aggregation. The new aggregated global model is sent back to the ESs for retraining purposes. This process of training and aggregation continues until convergence. Then, the final global model is used to identify potentially corrupted ESs that undergo a 'challenge-response' procedure to confirm the integrity of their cached data. This mechanism eliminates the need to verify every ES and thus significantly lowers overall communication costs.

However, in this FL-based benchmark EDIV approach [10], two key issues remain unaddressed. First, it assumes homogeneous activity data across ESs for FL model training, whereas real-world MEC deployments frequently exhibit heterogeneous activity data due to diverse usage patterns, regional variations, or application-specific demands. Such heterogeneous data distributions often degrade the accuracy of corrupted ES detection and slow down the convergence of the FL model, ultimately delaying the benefits of early detection. While standard FL algorithms such as FedAvg perform poorly in these non-IID settings due to uncontrolled model drift, FedProx attempts to address this challenge by introducing a proximal regularization term. However, FedProx applies a fixed regularization strength uniformly across all ESs, which fails to account for the varying degrees of divergence that can exist among them. As a result, both approaches face difficulty in achieving reliable convergence and maintaining high detection accuracy when operating in heterogeneous environments. This, in turn, negatively impacts the overall effectiveness and efficiency of FL-based EDIV. Second, this existing method performs one-off FL-based verification and lacks a strategy for subsequent EDIV rounds. For example, if the FL model flags an ES as corrupted but the 'challenge–response' procedure later deems it not corrupted, the system does not distinguish between honest ESs and those previously flagged as corrupted when performing the next EDIV rounds. As a result, all ESs continue to be re-evaluated equally, diminishing the long-term benefits of early FL-based filtering. A more efficient system should prioritize ESs dynamically, thus intensifying scrutiny for suspicious nodes while reducing verification overhead for trusted ones.

To address these challenges, this paper introduces Federated learning with Adaptive Regularizer-based Edge Data Integrity Verification (FedAR-EDIV), which is specifically designed for efficient detection of data corruption in MEC.[1] In particular, FedAR-EDIV is a two-phase approach that integrates blockchain-enabled FL with adaptive regularization and a multi-round verification strategy. Two smart contracts are deployed: one coordinates the aggregation of locally trained models in Phase 1, and the other manages subsequent EDIV verifications in Phase 2. Phase 1 focuses on collaboratively training a deep neural network model across different ESs' private activity data for their probable corruption detection. Specifically, each ES participates in the FL process by training the model locally on its own activity data. A critical innovation in this phase is our adaptive objective regularization strategy, which aligns local model updates to mitigate the impacts of heterogeneous activity data distributions. By preventing excessive divergence among ESs' local models, this strategy enables faster convergence without sacrificing privacy. As a result, the global model is trained in fewer global training rounds, reducing overall training time and enabling the correct identification of potentially corrupted ESs. Therefore, the framework significantly reduces the number of ESs requiring the more expensive 'challenge-response' procedure in Phase 2, substantially lowering the total computation and communication costs in the final EDIV. The integration of blockchain ensures that no single node in this decentralized environment serves as the central FL coordinator, preserving both the integrity and fairness of the training process [11]. Once corrupted ESs have been flagged in Phase 1, FedAR-EDIV in Phase 2 initiates a blockchain-based 'challenge-response' procedure to externally verify the cached data of each ES. This process stores cryptographic proofs on a tamper-resistant ledger, eliminating the need for an untrusted third-party auditor (TPA). In addition, an effective reputation mechanism updates ES reputation scores based on Phase 2 verification outcomes. This mechanism distinguishes between ESs that require deeper scrutiny (ESs with lower reputation) and those that can be assigned a lower verification priority (ESs with higher reputation)

[1]The proposed learning based approach detects potentially corrupted ESs by identifying behavioral anomalies in activity logs. Rather than learning specific attack signatures, it captures deviations from normal patterns such as irregular access or abnormal response behavior, allowing it to detect a broad range of corruption sources, both adversarial and benign.

in subsequent verification rounds, ensuring that the benefits of early detection are maintained over time while minimizing overall computation and communication costs in the EDIV. Our key contributions are summarized below.

- *Adaptive regularization to address heterogeneous activity data in FL for improved corruption detection:* We introduce an adaptive objective regularization strategy specifically designed to manage heterogeneous distributions of activity log data across ESs in the FL process. This strategy ensures that local model updates remain consistent with the evolving global model, thereby accelerating model convergence and enhancing detection accuracy. Consequently, our approach effectively reduces computation and communication overhead during EDIV.
- *An effective reputation mechanism for efficient multi-round EDIV:* We propose an effective reputation mechanism that strategically updates the reputation scores of ESs based on their verification outcomes in each EDIV round. By prioritizing fewer verifications for trustworthy ESs and allocating deeper scrutiny to those previously flagged as potentially corrupted, this mechanism streamlines the multi-round verification process. It significantly improves the long-term efficiency of EDIV by minimizing unnecessary verifications, effectively reducing overall computation and communication costs.

The remainder of the paper is organized as follows. Section II reviews existing EDIV techniques and their limitations. Section III presents the proposed FedAR-EDIV framework, including its system architecture, adaptive regularization strategy, and verification workflow. Section IV provides theoretical analyses, while Section V details the experimental evaluation across both framework phases. Section VI concludes the paper and outlines future research directions.

II. Related Work

Early research on Cloud Data Integrity (CDI) [12], [13] established foundational schemes such as Proof of Retrievability (PoR) [14], [15] and Provable Data Possession (PDP) [16]. These methods enable an auditor/verifier (AV) to verify the integrity of remotely stored files through probabilistic proofs. Despite their significant influence, many PoR- and PDP-based solutions rely on a TPA [17], adding complexity regarding security and trust delegation. As edge computing expands to encompass numerous geographically dispersed nodes, directly applying CDI methods becomes inefficient, prompting further exploration of EDIV solutions [5].

EDIV-specific approaches typically follow a 'challenge-response' paradigm, in which an AV or TPA periodically challenges ESs to collect integrity proofs. Tong et al. [6] were among the first to investigate EDIV, adapting PDP from the cloud domain and extending its theoretical guarantees [18], although encountering limitations in multi-replica scenarios [19]. Merkle tree-based schemes, such as EDI-V [20], improved verification efficiency but still incurred significant communication overhead and relied on probabilistic checks. Conversely, Li et al. [21] introduced EDI-S, employing elliptic curve cryptography to achieve deterministic guarantees, albeit at increased computational complexity. Other approaches, such as CooperEDI [19], delegate verification tasks directly to ESs, consequently increasing bandwidth usage. Meanwhile, EdgeWatch [22] adopts blockchain-based incentives to eliminate the need for TPAs but introduces additional on-chain transactions, potentially diminishing verification efficiency. Additionally, Cui et al. [7] introduced ICL-EDI, leveraging homomorphic signatures to reduce computational complexity. Ding et al. [23] proposed EDI-DA, supporting dynamic updates of edge data replicas using an Index-Single Linked Table (I-SLT) and dynamic arrays; however, these methods still rely heavily on frequent 'challenge-response' interactions. Zhao et al. proposed a Smart Inspection Algorithm (SIA) to selectively challenge "unreliable" replicas, later extending it to dynamic verification assignment [24] for improved resource allocation. Nonetheless, most existing methods lack a comprehensive strategy for data replica filtration and verification scheduling, potentially leading to excessive overhead in large-scale deployments. While batch verification [18] accelerates proof processing, it typically requires a TPA, raising privacy concerns [25] and potential attack vectors [26]. An FL-based approach has recently been explored for data filtering [10], but standard aggregation algorithms (e.g., FedAvg) may degrade accuracy in the presence of heterogeneous data distributions.

Classical 'challenge-response' EDIV methods incur substantial computational and communication overhead, especially as the number of ESs increases and frequent verifications become necessary. Recent machine learning-based anomaly detection approaches that rely on centralized user activity data pose significant privacy risks and require resource-intensive data aggregation, thus limiting scalability. FL has emerged as a promising alternative, enabling global model training without centralizing sensitive data. Zhao et al. [10] recently proposed an FL-based framework for early detection of potentially corrupted ESs; however, their approach assumes homogeneous data distributions and lacks clear guidance for subsequent verification rounds, motivating the development of enhanced EDIV methodologies.

III. Proposed FedAR-EDIV Approach

This section introduces the proposed FedAR-EDIV scheme, beginning with an overview of its architecture. Subsequently, the FedAR-EDIV algorithms and each entity's detailed operational procedures are thoroughly explained. For convenience, we provide the system model, threat model, and design goals in Appendix A, available online, and the notations and symbols used throughout the discussion in Table S1 of the supplementary document.

A. Overview

The FedAR-EDIV process is structured into two primary phases. (i) FL with adaptive regularizer-based potentially corrupted ES detection and (ii) a blockchain-supported 'challenge-response' mechanism in which EDIV confirms and reputation updates for participating ESs. The initialization occurs only once at the start of the system's operation. Meanwhile, the federated model training in Phase 1 and the integrity verification in Phase 2, managed by the AV, are executed iteratively over rounds $\mathbf{q}$. The overview of the proposed FedAR-EDIV scheme is depicted in Fig. 2, while the primary steps are outlined in Algorithm 1. A concise overview of these phases is provided below. The process begins with initialization, during which key parameters are negotiated. Two distinct smart contracts, called AgSC and VfSC, are deployed on the blockchain. The AgSC contract facilitates the orchestration of FL training and aggregation in

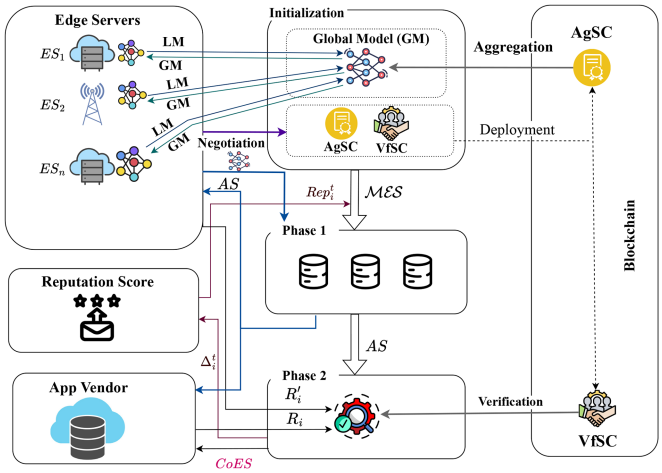


Fig. 2. The proposed FedAR-EDIV architecture.

Phase 1, while the VfSC contract is responsible for EDIV in Phase 2.

Phase 1: The FL framework enabled by blockchain incorporates the AgSC smart contract to develop a corruption detection model, termed $Global_{Model}$. This model is trained using activity data extracted from local logs in the ESs, enabling it to identify potentially corrupted ESs effectively. Once $Global_{Model}$ is trained, each ES independently runs the model to detect the corrupted ES (referred to as AS). To address heterogeneous data distributions among participating ESs, adaptive regularization is applied during the FL process, resulting in faster convergence and enhanced detection accuracy, consistent with the overall objective of minimizing computation and communication costs in the final EDIV procedure.

Phase 2: Moving into Phase 2, the VfSC smart contract performs the final EDIV. For each corrupted ES ($ES_i \in AS$), VfSC requests integrity proofs from both the ES that holds the corrupted data replica (R'_i) and the AV (R_i). These proofs are then compared to confirm whether the data remains intact or has been corrupted. Upon concluding the verification process, an effective reputation mechanism updates each ES's reputation score, rewarding honest ESs and penalizing corrupted ones. This approach minimizes unnecessary overhead in subsequent EDIV rounds by prioritizing future checks on those ESs with lower reputations. Note that FedAR-EDIV supports dynamic edge environments where ESs may join or leave over time. New ESs can register via the AgSC contract and participate in subsequent FL rounds with a neutral initial reputation, while departed ESs are excluded from future rounds. Its round-based, stateless design enables seamless adaptation without reinitialization, ensuring robustness and scalability in time-varying deployments. The complete execution flow of the proposed FedAR-EDIV framework is formalized in Algorithm 1. The specific operations of each procedure are further detailed as follows.

Algorithm 1: FedAR-EDIV.

Input: ES set $\mathcal{MES}$, global model w^1, initial reputation scores $\{Rep_i^1\}$, regularization parameters $\{\mu_i^1\}$, learning rate η, meta-learning rate γ, penalty factor α, reward factor β, local datasets $\{D_i\}$
Output: Corrupted data replica set $CoES$

procedure FEDAR-EDIV($\mathcal{MES}, w^1, Rep_i^1, \mu_i^1, D_i, \eta, \gamma, \alpha, \beta$)
  for each EDIV round $r = 1$ **to** q **do**
    $AS \leftarrow \varnothing$, $CoES \leftarrow \varnothing$
    Deploy two smart contracts on the blockchain:
      • $AgSC$: Manages FL coordination
      • $VfSC$: Conducts integrity proof verification
    // Phase 1: FL-Based Corruption Detection
    for $t = 1$ to R **do**
      for each ES $ES_i \in \mathcal{MES}$ **in parallel do**
        $(w_i^{t+1}, \mu_i^{t+1}) \leftarrow$ CLIENTUPDATE($ES_i, w^t, \mu_i^t, D_i, \eta, \gamma$)
      end for
      Update global model: $w^{R+1} \leftarrow \sum_{i=1}^{|\mathcal{MES}|} \frac{n_i}{n} w_i^{t+1}$
    end for
    $Global_{Model} \leftarrow$ ANOMALYDETECTION(w^{R+1})
    for each $ES_j \in \mathcal{MES}$ **do**
      $AS \leftarrow AS \cup Global_{Model}(ES_j, Data_j)$
    end for
    // Phase 2: External Verification and Reputation Update
    $CoES \leftarrow$ FINALVERIFICATION(AS)
    $\{Rep_i^{q+1}\} \leftarrow$ UPDATEREPUTATION($\{Rep_i^q\}, CoES, \alpha, \beta$)
    return $CoES, \{Rep_i^{q+1}\}$
  end for
end procedure

B. Phase 1: FL With Adaptive Regularization Driven Corrupted ESs Detection

1) Process Initialization: This section outlines how corruption detection models are built using historical user activity data. In the initial system stage, where there is no log data yet, a traditional 'challenge-response' mechanism identifies corrupted ESs: the AV dispatches cryptographic challenges to ESs, prompting them to return integrity proofs (for instance, hash values or digital signatures). Any discrepancies between these proofs and the references of the AV signal indicate potential corruption. Although this method effectively ensures basic EDIV, it incurs high computation and communication costs, particularly in networks with a large number of ESs. Nonetheless, it establishes essential initial trust, creating the foundation for more efficient FL–based detection once adequate operational log data becomes available. Importantly, the traditional challenge–response mechanism is used only during the system's bootstrapping phase, when activity log data is insufficient to support FL-based anomaly detection. In most cases, 1–2 EDIV rounds are sufficient to accumulate the necessary operational data for initializing federated model training. Once adequate logs are collected, the system transitions from challenge–response to the FL-based detection phase. This shift allows the system to leverage learned behavior patterns for scalable and adaptive detection of corruption, significantly reducing verification overhead while maintaining detection accuracy.

To facilitate low-cost verification in subsequent rounds, the corruption detection model is trained once in Phase 1 and then repeatedly applied to multiple ES verifications without incurring excessive computation and communication costs. Since most ESs $ES_i \in \mathcal{MES}$ generally behave correctly – corrupted data

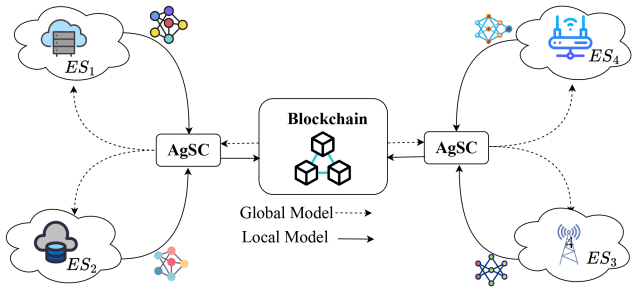


Fig. 3. A blockchain-based FL setup, multiple ESs (ES_1 through ES_n) that collaborate to build a unified global model, orchestrated by a specialized smart contract ($AgSC$).

Algorithm 2: CLIENTUPDATE With Adaptive Regularization and Meta-Learning.

procedure CLIENTUPDATE($ES_i, w^t, \mu_i^t, D_i, \eta, \gamma$)
  Input: Global model w^t, local dataset D_i, adaptive parameter μ_i^t, learning rates η, γ
  Initialize local model: $w_i^t \leftarrow w^t$
// Step 1: Local Update with Adaptive Regularization
  Optimize local model w_i^{t+1} by minimizing the objective in (2)
  ▷ This includes a client-specific regularization term scaled by μ_i^t
// Step 2: Meta-Learning Update of Regularization Parameter
  Update μ_i^{t+1} via gradient descent as in (3)
  ▷ Adjust μ_i^{t+1} based on the divergence between w_i^{t+1} and w^t
  return w_i^{t+1}, μ_i^{t+1}
end procedure

replicas represent only a small subset for AV [19] – the detection logic focuses on operational log anomalies, that is, significant deviations from the typical behavior of an ES [27]. The system minimizes the overall verification workload by rapidly isolating these connected ESs.

Given MEC's stringent privacy and security requirements, our framework combines FL with blockchain technology, ensuring secure and efficient model training, as shown in Fig. 3. Before training, various key parameters must be defined: number of communication rounds (CRs), loss functions, initial gradients, convergence criteria, learning rates, and aggregation methods. According to [28], the AV can delegate these parameters to connected ESs, facilitating scalability and reducing central coordination overhead.

2) Model Training and Update With Adaptive Regularizer: The $ES_i \in MES$ is treated as an FL client. Suppose that each node ES_i maintains a local dataset $D_i = \{(x_h, y_h)\}_{h=1}^{|D_i|}$, where $|D_i|$ is the number of data samples, x_h is the hth feature vector, and y_h is its corresponding label. These data typically originate from *operational logs*, capturing user activities over a given time frame. Without loss of generality, the FL training objective can be formulated as:

$$\min_{w_i^{(t)} \in \mathbb{R}^d} \left[\frac{1}{|D_i|} \sum_{(x_h, y_h) \in D_i} f\left(w_i^{(t)}; x_h, y_h\right) \right], \tag{1}$$

where $w_i^{(t)}$ denotes node i's local model at round t, w^t is the *global model* broadcast at the start of round t, and $f(\cdot;\cdot)$ is the local loss function. Because each D_i is extracted from operational logs, its distribution may differ significantly from those at other servers, leading to data heterogeneity (*non-IID conditions*). In such cases, the standard FL objective appends a proximal term $\mu \frac{1}{2} \| w_i^{(t)} - w^t \|^2$ to anchor local updates around the global model w^t:

$$\min_{w_i^{(t)} \in \mathbb{R}^d} \left[\frac{1}{|D_i|} \sum_{(x_h, y_h) \in D_i} f\left(w_i^{(t)}; x_h, y_h\right) + \frac{\mu}{2} \left\| w_i^{(t)} - w^t \right\|^2 \right], \tag{2}$$

where μ determines how tightly $w_i^{(t)}$ is pulled toward w^t. This static-μ approach (e.g., FedProx [29]) partially addresses heterogeneity but cannot fully accommodate the wide range of divergence across individual ESs.

To better handle the static μ, we propose an *adaptive* regularizer that adjusts μ *in real time*. Specifically, each client's regularization strength μ scales with its current level of divergence from w^t. As such, our framework extends the traditional *regularization-based* FL objective by introducing a *client-specific* parameter μ_i. After each local update produces $w_i^{(t+1)}$, we optimize μ_i^t by performing gradient descent on $\left\| w_i^{(t+1)} - w^t \right\|^2$. This *meta-learning* step enables μ_i adaptation in real-time to the node's drift, assigning stronger regularization to larger deviations. Formally,

$$\mu_i^{(t+1)} = \mu_i^{(t)} - \gamma \nabla_{\mu_i} \left\| w_i^{(t+1)} - w^t \right\|^2, \tag{3}$$

where γ denotes the meta-learning rate. It typically falls within a small positive range (e.g., $[10^{-5}, 10^{-2}]$) and is selected through empirical tuning to ensure balanced and controlled updates of μ_i across clients. Although γ remains fixed throughout the training process, it directly influences the rate at which each client's regularization parameter μ_i adjusts in response to local model divergence. A smaller γ leads to more gradual changes in μ_i, resulting in conservative adaptation, while a larger γ enables faster adjustments but may introduce instability if not properly tuned. Importantly, even with a fixed γ, the regularization parameter μ_i evolves dynamically during training. This allows each client to autonomously regulate its regularization strength based on its deviation from the global model, thereby enhancing the ability to personalize learning under heterogeneous data distributions. Thus, each client's μ_i^t reflects how far $w_i^{(t+1)}$ diverges from w^t: nodes with greater drift receive correspondingly higher regularization, improving resilience under data heterogeneity. Once each node finalizes $w_i^{(t+1)}$, these local models are *aggregated:*

$$w^{(t+1)} = \sum_{i=1}^{N} \frac{n_i}{\sum_{j=1}^{N} n_j} w_i^{(t+1)}, \tag{4}$$

where $n_i = |D_i|$ and $\sum_{j=1}^{N} n_j$ is the overall sample count. This procedure repeats until w^t converges. We present the convergence analysis in Section IV-D. Numerous BFL frameworks (e.g., [30], [31]) enable local models to be processed in parallel, mitigating performance limitations tied to sequential model aggregation. However, since different real-world scenarios demand unique ways of integrating blockchain and FL, we do not provide

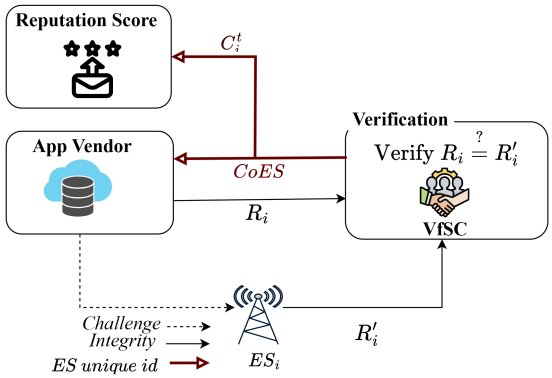


Fig. 4. Phase 2 final verification process, featuring decentralized verification supported by VfSC.

Algorithm 3: FINALVERIFICATION.

Input: Corrupted ES set AS from Phase 1
Output: Confirmed corrupted ES set $CoES$

procedure FINALVERIFICATION(AS)
  Initialize $CoES \leftarrow \emptyset$
  for each node $ES_i \in AS$ **do**
    $R_i \leftarrow$ GENERATEPROOF(original_data$_i$)
    $R_i' \leftarrow$ GENERATEPROOF(cached_data$_i$)
    if $R_i \neq R_i'$ **then** ▷Mismatch implies corruption
      $CoES \leftarrow CoES \cup \{ES_i\}$
    end if
  end for
  return $CoES$
end procedure

an exhaustive implementation guide for every possible context. For a more extensive discussion on BFL, we refer interested readers to [32]. Notably, the study in [33] indicates that only a small fraction of gradients (approximately 0.1% with the highest magnitudes) proves critical for gradient aggregation. Consequently, gradient compression techniques—such as those in [34], [35]—can be employed to reduce communication costs during federated training.

Once the global model is established, each ES adopts it as a corruption detection tool, thus performing an initial filtration step for EDIV. Specifically, each ES $ES_i \in \mathcal{MES}$ runs the corruption detection model (trained during phase 1) to identify potentially corrupted ESs with minimal interaction between the AV and the ES. When corrupted data replicas are detected, the ES proactively alerts the AV. Utilizing a pretrained model for corruption identification is typically more cost-effective than generating a full integrity proof and can be scheduled periodically or triggered on demand. Frequent checks lead to faster detection of corrupted data but increase the ES's computation cost, whereas lower frequencies reduce overhead at the risk of delayed detection. To realize the adaptive regularization and meta-learning process discussed above, we define the procedure CLIENTUPDATE, provided in Algorithm 2. This procedure is called iteratively for each client in Phase 1 and receives all necessary parameters, including the current global model w^t, local dataset D_i, adaptive regularization parameter μ_i, learning rate η, and meta-learning rate γ.

C. Phase 2: External Verification

1) Final Verification: This phase is initiated by the AV upon receiving a corruption report from Phase 1. The procedure FINALVERIFICATION (Algorithm 3) is responsible for confirming the integrity of potentially corrupted ESs using a blockchain-enabled *'challenge–response'* protocol. Specifically, the output of Phase 1 (a set of potentially corrupted ESs denoted by AS) is passed into this procedure, which systematically verifies each ES's data integrity by comparing the proof generated from the ES's data replica with the corresponding proof generated from the AV's data replica. Unlike traditional approaches that rely on an untrusted TPA for EDIV, the FedAR-EDIV framework significantly improves system security while eliminating the risk of privacy leakage. For example, as shown in Fig. 4, consider a scenario where ES_2 reports corruption. The AV sends a challenge message to ES_2 to verify data integrity. Both the AV and ES_2 then generate their respective integrity proofs and submit them to the blockchain. This coordinated process allows the VfSc smart contract to verify the proofs and provide the AV with unique identifiers for the corrupted data replicas.

The method for generating integrity proofs can be adapted based on the size of the cached data replica. For smaller data replicas, the SHA-256 hash equation offers a highly efficient option for proof generation. For larger datasets, sampling-based methods, such as Merkle hash tree-based approaches [20], are more suitable due to their efficiency in handling size-sensitive hash operations.

2) Reputation Mechanism: After completing the final verification ('challenge-response' procedure), the system introduces a reputation mechanism designed to support the multi-round EDIV strategy. This mechanism addresses the need for ongoing trust management in dynamic MEC environments by determining which ESs should be re-verified more rigorously and which can be deprioritized in future rounds. The reputation mechanism is triggered after each verification round and leverages both the outcomes of the 'challenge-response' process and the pre-trained global model to reassess the trustworthiness of each ES. Specifically, the algorithm analyzes the post-verification behavior of each ES (as outlined in Algorithm 3) and flags those ESs that exhibit data corruption. Corrupted ESs are penalized by lowering their reputation scores, while ESs that show no data corruption are rewarded with higher scores. These reputation scores are updated dynamically after every EDIV round, forming an adaptive trust evaluation loop. This ensures that future verification efforts are concentrated on high-risk ESs while reducing redundant checks for trustworthy ones—thereby optimizing both computation and communication costs. At the end of the procedure, the algorithm produces an updated reputation score of all ESs in the MEC. Initially, all ESs $\mathcal{MES} = \{ES_1, ES_2, \ldots, ES_n\}$ are assigned equal reputation scores Rep_i^1. After round q finishes, the score of each ES_i is updated based on the following rule:

$$Rep_i^{q+1} = Rep_i^q + \Delta_i^q, \tag{5}$$

where the score change Δ_i^q is computed as:

$$\Delta_i^q = \begin{cases} -\alpha \cdot C_i^q & \text{if } C_i^q = 1 \text{ (corrupted)} \\ +\beta \cdot (1 - C_i^q) & \text{if } C_i^q = 0 \text{ (not corrupted)}, \end{cases} \tag{6}$$

where C_i^q is a binary corruption indicator, $\alpha > 0$ is the penalty factor, and $\beta > 0$ is the reward factor. This formulation is

Algorithm 4: UPDATEREPUTATION: Score Adjustment for ESs.

procedure UPDATEREPUTATION($\{Rep_i^q\}, CoES, \alpha, \beta$)
  for each server $ES_i \in \mathcal{MES}$ **do**
    if $ES_i \in CoES$ **then** ▷confirmed corrupted
      $C_i^q \leftarrow 1$
      $Rep_i^{t+1} \leftarrow Rep_i^q - \alpha \cdot C_i^q$
    else ▷behaved honestly
      $C_i^q \leftarrow 0$
      $Rep_i^{q+1} \leftarrow Rep_i^q + \beta \cdot (1 - C_i^q)$
    end if
  end for
  return $\{Rep_i^{q+1}\}$
end procedure

designed to reinforce long-term trust dynamics: honest ESs that consistently pass verification rounds accumulate progressively higher reputation scores, whereas malicious ESs that repeatedly fail verifications are continually penalized. As a result, the system naturally encourages score convergence over time, enabling a clear separation between trusted and untrusted ESs across successive EDIV rounds.

Importantly, to mitigate the effects of rare false positives and prevent long-term penalization of honest ESs due to isolated misclassifications, the framework allows for gradual recovery. Honest ESs can regain lost reputation through continued correct behavior in subsequent rounds. This adaptive trust evolution ensures that the reputation mechanism remains robust, fair, and aligned with observed integrity performance over time. The entire process is formalized in the UPDATEREPUTATION procedure, as shown in Algorithm 4.

IV. THEORETICAL ANALYSIS

This section provides the theoretical analysis that includes the correctness, effectiveness, security, and convergence analysis of the proposed FedAR-EDIV approach.

A. Correctness Analysis

To validate the correctness of our FedAR-EDIV approach, we first prove its ability to verify the integrity of sampled data replicas accurately. Subsequently, we demonstrate its capability to identify data corruption with a guaranteed probability level.

Theorem 1 (Correctness of Integrity Verification): Given a dataset D consisting of k data replicas, $D = \{d_1, d_2, \ldots, d_k\}$. The FedAR-EDIV approach guarantees the correctness of integrity verification under the assumption that SHA-256 is collision-resistant and resistant to second preimage attacks. If any corrupted replica $d_i \neq d_i'$ exists, it will be detected with probability $P_{\text{detection}} = 1$, provided that the attacker cannot generate a second preimage or collision.

The detailed proof for Theorem 1 is provided in Appendix B.1 of the supplemental document, available online. It illustrates that FedAR-EDIV accomplishes the initial goal outlined in Appendix A.3, available online, namely, the objective of correctness.

B. Efficiency Analysis

Theorem 2: Let $T_{\text{verify},i}$ denote the total verification time for ES_i in the FedAR-EDIV framework. When all ESs operate in parallel, the overall verification latency is

$$T = \max_{i \in S} \{T_{\text{verify},i}\},$$

and this latency satisfies the following bound on the optimal verification time F:

$$T \leq F \leq 2T.$$

The proof for Theorem 2 is provided in Appendix B.2 of the supplemental document, available online. It illustrates that FedAR-EDIV accomplishes the second goal outlined in Appendix A.3, available online, namely, the efficiency objective.

C. Security Analysis

We now demonstrate FedAR-EDIV's effectiveness in addressing the replay attack and the forge attack outlined in Appendix A.2 of the supplemental document, available online.

Theorem 3: It can be concluded that the FedAR-EDIV scheme is secure against replay, replace, and forge attacks by an ES, as long as the $Hash()$ function is collision-resistant and the random functions used are secure.

The proof for Theorem 3 is provided in Appendix B.3 of the supplemental document, available online. It illustrates that FedAR-EDIV accomplishes the third goal outlined in Appendix A.3, available online, namely, the security objective.

D. Convergence Analysis

We now present the convergence guarantee for the federated training in FedAR-EDIV under data heterogeneity settings.

Assumption 1. (*L-smoothness:*) Each local objective function $F_i(w)$ is assumed to be L-smooth, meaning that for all $w, w^t \in \mathbb{R}^d$,

$$\|\nabla F_i(w) - \nabla F_i(w^t)\| \leq L\|w - w^t\|.$$

An equivalent condition is:

$$F_i(w) \leq F_i(w^t) + \nabla F_i(w^t)^\top (w - w^t) + \frac{L}{2}\|w - w^t\|^2.$$

Assumption 2. (*Strong Convexity:*) Each local objective function $F_i(w)$ is assumed to be μ-strongly convex, meaning that for all $w, w^t \in \mathbb{R}^d$,

$$F_i(w) \geq F_i(w^t) + \nabla F_i(w^t)^\top (w - w^t) + \frac{\mu}{2}\|w - w^t\|^2.$$

Theorem 4: Let $F(w) = \frac{1}{n}\sum_{i=1}^n F_i(w) + \lambda \sum_{i=1}^n \mu_i \|w - w_i^t\|^2$ be the global objective function, where each $F_i(w)$ is L-smooth, and $\mu_i = \gamma \|w^t - w_i^t\|^2$. Assume that the gradient dissimilarity is bounded by $\mathbb{E}[\|\nabla F_i(w) - \nabla F(w)\|^2] \leq \beta^2$. Then, for a sufficiently small product $\eta\gamma$, the iterates generated by the update

$$w_i^{t+1} = w^t - \eta \left(\nabla F_i(w^t) + \mu_i (w^t - w_i^t)\right),$$

$$w^{t+1} = \frac{1}{n}\sum_{i=1}^n w_i^{t+1}$$

satisfy:

$$\mathbb{E}[F(w^T)] - F(w^*) = \mathcal{O}\left(\frac{1}{T}\right).$$

The proof for Theorem 4 is provided in Appendix B.4 of the supplemental document, available online.

E. Correctness and Robustness of Reputation System

Theorem 5 (Correctness of Reputation Convergence): Under multiple EDIV rounds, if an ES behaves honestly in all rounds, then its reputation score increases monotonically and eventually exceeds any predefined trust threshold θ, resulting in reduced verification frequency.

The proof for Theorem 5 is provided in Appendix B.5 of the supplemental document, available online.

Theorem 6 (Robustness of Reputation). If an ES behaves dishonestly with probability $p \in (0, 1)$ over time, then its expected reputation score decreases if the penalty factor α satisfies:

$$\alpha > \beta \cdot \frac{1-p}{p}$$

This guarantees that dishonest behavior is penalized in expectation, preventing manipulation of the reputation system.

The proof for Theorem 6 is provided in Appendix B.5 of the supplemental document, available online.

V. Experimental Analysis

We now perform comprehensive experiments on real-world datasets to validate the effectiveness of our proposed FedAR-EDIV framework, particularly by comparing its performance with established EDI solutions.

A. Experimental Settings

We employ two widely used intrusion detection datasets: KDD99 [36] and CIC2017 [37] as ES activity log datasets to conduct FL-based experiments for detecting potentially corrupted ESs. These datasets encompass a wide range of behavior deviations that are not limited to attacks, including misconfigurations and irregular access patterns, phenomena also relevant to edge data corruption scenarios. Following LH-EDCD [10], we adopt a Multi-Layer Perceptron (MLP) as the shared global deep neural network model, trained using cross-entropy loss optimized via Stochastic Gradient Descent (SGD) with a fixed learning rate of 0.01. To assess FL's effectiveness, we perform two experiments: (i) increasing client (ES) participation from 10% to 100%, and (ii) increasing computation per client by varying the number of local epochs and batch size configurations. We compare detection performance against LH-EDCD and FedProx [29] to demonstrate the improvements achieved by FedAR-EDIV. Based on optimal FL outcomes, final EDIV experiments are conducted using the actual data replicas cached at ESs. Each experiment is repeated 50 times, and the results are averaged. All experiments are performed using Python 3.11 on an Intel Core i5 (1.3 GHz) system with 16 GB RAM. The experiments were conducted on a Windows 11 operating system. The Python environment was managed using Anaconda, and all federated learning models were implemented using the PyTorch deep learning framework (version 2.1.0). Additional scientific computing and data preprocessing tasks were handled using NumPy (version 1.26.2) and Pandas (version 2.1.3), respectively. Data visualization and result analysis were supported by Matplotlib (version 3.8.0) and Seaborn (version 0.13.0). All random seeds were fixed to ensure consistent reproducibility across runs. No GPU acceleration was used, ensuring that all experiments reflect performance on CPU-only environments, simulating resource-constrained edge devices.

1) Dataset and Data Distribution: KDD99 contains 44,898,431 training and 311,029 test samples with 42 features across six network flow classes (one benign, five attacks). CIC-IDS2017 includes 2.8 million samples with 78 features spanning eight classes (one benign, seven attacks). To evaluate performance under both ideal and realistic edge conditions, we simulate both IID and non-IID data distributions across 100 clients. For the IID distribution, training samples are randomly shuffled and evenly partitioned to ensure balanced class representation. Each client receives approximately 49,000 training samples for KDD99 and 22,640 for CIC-IDS2017. For non-IID distribution, we adopt a label-skewed partitioning scheme: each dataset is grouped by class labels and split into 200 shards, with each client randomly assigned two shards. Consequently, each client receives data from only two attack classes, resulting in a highly heterogeneous distribution that reflects real-world edge scenarios. Additionally, the AV's actual data replicas are randomly distributed across ESs to assess FedAR-EDIV's efficiency in final EDIV evaluations.

2) Baseline Approaches: In the FL-based corrupted ES detection phase, we evaluate the proposed FedAR-EDIV method against baseline approaches, including LH-EDCD [10] and FedProx [29], to demonstrate its superiority. For the EDIV phase, three baseline EDIV methods, including EDI-V [20], EDI-S [21], and ICL-EDI [7], are considered to assess the efficiency of the proposed FedAR-EDIV approach.

3) Hyperparameters: FL Configuration. We configure both federated optimization and model-related hyperparameters. The client participation fraction (C) is varied over 10%, 20%, 50%, 100%, while the learning rate (η), local epochs (E), and batch size (B) are set to 0.01, 1, 2, 5, and 64, 128, 256, respectively, for both datasets, following standard FL practices [9]. Under a fixed $C = 0.5$, we further explore combinations of E and B to assess their effect on convergence, using the local update count $u = NE/KB$, where N is the number of training samples and K is the number of clients. This allows us to evaluate how local training intensity influences global convergence efficiency.

EDIV Configuration: To assess FedAR-EDIV under varying conditions, we adjust three parameters: the number of ESs (n), the data scale per ES (k), and the replica size (d_s in MB). The `SHA-256` hash function is used throughout, consistent with prior work. For EDI-V, data replicas are partitioned into blocks to achieve 99.9% inspection accuracy, as per Theorem 2 in [20].

4) Evaluation Metrics: FL-Based Metrics. We evaluate FL methods (LH-EDCD and FedProx) by measuring the number of CRs required to achieve a specified test-set detection accuracy. A lower number of CRs indicates reduced communication overhead and improved efficiency, especially under non-IID data conditions, where imbalanced data typically increases convergence time. By reaching the target accuracy faster, FedAR-EDIV demonstrates effective mitigation of data heterogeneity. Additionally, FedAR-EDIV introduces minimal overhead beyond LH-EDCD, as it only augments local training without modifying the baseline procedure. It maintains a convergence rate of $\mathcal{O}(1/T)$.

TABLE I
EFFECT OF EXTENDING CLIENT FRACTION (C)

Dataset	Data Distribution	Method	Detection Accuracy (%)				Communication round (speedup)			
			C				C			
			0.1	0.2	0.5	1	0.1	0.2	0.5	1
KDD99	IID	LH-EDCD	93.4	94.02	95.44	95.44	95	92	86	97
		FedProx	98.33	98.33	98.33	98.33	26(3.65×)	25(3.68×)	25(3.44×)	26(3.73×)
		FedAR-EDIV	99.66	99.66	99.67	99.67	12(7.92×)	11(8.36×)	11(7.82×)	11(8.82×)
	Non-IID	LH-EDCD	91.45	93.65	92.56	92.56	95	93	89	94
		FedProx	96.67	97.09	96.45	98.56	23(4.13×)	24(3.88×)	23(3.87×)	23(4.02×)
		FedAR-EDIV	99.66	99.78	99.78	99.78	9(10.56×)	10(9.3×)	9(9.8×)	10(9.4×)
CIC2017	IID	LH-EDCD	92.48	92.4	92.59	92.34	77	75	76	78
		FedProx	98.33	97.65	97.71	98.41	23(3.35×)	22(3.4×)	20(3.8×)	21(3.71×)
		FedAR-EDIV	99.68	99.88	99.91	99.88	11(7×)	11(6.8×)	10(7.6×)	10(7.8×)
	Non-IID	LH-EDCD	94.33	94.21	94.48	94.26	76	75	74	77
		FedProx	97.78	98.39	98.49	98.29	23(3.30×)	22(3.40×)	19(3.89×)	19(4.05×)
		FedAR-EDIV	99.46	99.43	99.67	99.63	11(6.91×)	10(7.5×)	9(8.22×)	9(8.55×)

TABLE II
IMPACT OF INCREASING COMPUTATION (VARYING E AND/OR B) IN EACH CLIENT OVER THE KDD99 DATASET

Data Distribution	Detection Accuracy				CR Speedup		
	E, B, u	LH-EDCD	FedProx	FedAR-EDIV	LH-EDCD	FedProx	FedAR-EDIV
IID	1, ∞, 1	94.23	98.42	99.68	125	27(4.63×)	13(9.62×)
	5, ∞, 5	94.65	98.54	99.87	120	25(4.8×)	12(10×)
	1, 256, 15	94.87	98.51	99.89	131	25(5.24×)	13(10.07×)
	1, 128, 31	94.84	98.57	99.91	124	26(4.76×)	13(9.53×)
	2, 128, 62	94.89	98.87	99.92	92	25(3.68×)	12(7.67×)
	5, 128, 156	94.87	98.89	99.92	85	26(3.3×)	12(7.08×)
Non-IID	1, ∞, 1	93.54	97.59	99.32	211	24(8.79×)	14(15.07×)
	5, ∞, 5	94.21	97.93	99.45	189	24(7.9×)	13(14.53×)
	1, 256, 15	93.43	97.49	99.41	193	23(8.4×)	13(14.84×)
	1, 128, 31	93.87	97.73	99.49	184	23(8×)	12(15.33×)
	2, 128, 62	94.25	98.31	99.76	176	22(8×)	11(16×)
	5, 128, 156	94.32	98.38	99.84	172	22(7.8×)	12(14.33×)

EDIV Metrics: We focus on minimizing the overall computation cost associated with verifying data replicas, particularly during the interactive 'challenge-response' process between the AV and ESs. Lower computation costs and faster detection times confirm the effectiveness of FedAR-EDIV. Additionally, we evaluate the overall communication cost incurred during EDIV, since an efficient verification scheme must minimize the overhead involved in auditing data replicas.

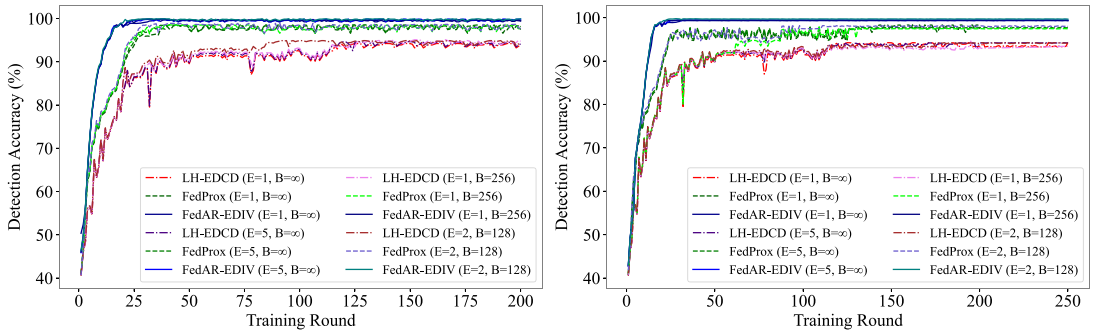
B. Performance Analysis Related to FL-Based Corrupted ESs Detection

1) Increasing Client Parallelism for Training: We evaluate the impact of the client fraction C on the parallelism of multiple clients in the baseline LH-EDCD method, FedProx, and our proposed FedAR-EDIV approach. Experiments are conducted over 100 CRs under both IID and non-IID data distributions, with a batch size of $B = 128$ and local epochs $E = 2$. We record the number of CRs required to achieve the target test accuracy, defined as the maximum accuracy attained by LH-EDCD under the given configuration. Table I presents the quantitative effects of varying C in terms of test-set accuracy and communication speedup. Fig. S1 in the supplementary document illustrates the corresponding trends in test-set detection accuracy over global CRs. The results indicate that FedAR-EDIV consistently outperforms both LH-EDCD and FedProx by requiring fewer CRs and achieving higher test accuracy across all values of C. In terms of communication efficiency, FedAR-EDIV achieves significant speedups—ranging from 7.82× to 8.82× on IID KDD99 and from 9.3× to 10.56× on non-IID KDD99. For the CIC2017 dataset, it achieves speedups of 6.8× to 7.8× under the IID setting and 6.91× to 8.55× under the non-IID setting. Across all values of C for both datasets, FedAR-EDIV consistently delivers the highest communication speedups compared to LH-EDCD and FedProx, demonstrating its effectiveness under both homogeneous and heterogeneous data distributions.

2) Increasing Computation Per Client for Training: In this experiment, we set $C = 0.5$ for the KDD99 dataset and progressively increase the number of SGD operations per client at each CR, where $u = NE/KB$ denotes the total number of SGD updates. This is achieved by increasing E, decreasing B, or adjusting both parameters.

For the IID KDD99 dataset, we conduct 200 global rounds across all values of u, while for the non-IID KDD99 dataset, the number of training rounds is fixed at 250 for every u value. Table II presents the corresponding test-set detection accuracies and communication speedups. Values in parentheses denote the corresponding communication speed-ups, while Fig. 5 displays the plots of test-set accuracy against the number of CRs for both the IID and non-IID KDD99 settings. The results indicate that increasing SGD updates (u) per client by adjusting B and E significantly reduces the number of CRs. Specifically, FedAR-EDIV achieves communication speedups ranging from 7.08× to 10.07× under IID KDD99, and from 14.33× to 16× under non-IID KDD99. Moreover, as u increases, FedAR-EDIV consistently surpasses both LH-EDCD and FedProx in communication efficiency.

For the IID CIC2017 dataset, we conduct 150 global rounds for all values of u, while for the non-IID CIC2017 dataset, 200 global rounds are used consistently across all u values. The detailed results for the CIC2017 dataset are presented in the supplementary document in Appendix C.2, available online.



(a) IID setting with C=0.55 (b) Non-IID setting with C=0.55

Fig. 5. Impact of increasing computation (varying E and/or B) in each client over the KDD99 dataset.

TABLE III
SENSITIVITY ANALYSIS OF META-LEARNING RATE γ ON KDD99 (NON-IID, $C = 0.5, E = 2, B = 128$)

Meta-learning rate γ	Detection Accuracy (%)	CRs to 99% Accuracy
10^{-5}	97.82	52
10^{-4}	98.91	34
10^{-3}	**99.78**	**20**
10^{-2}	99.34	18 (less stable)

C. Sensitivity Analysis of Meta-Learning Rate γ

To assess the impact of the meta-learning rate γ on the effectiveness of the proposed adaptive regularization mechanism, we conducted a sensitivity analysis under the non-IID KDD99 setting. As summarized in Table III, we observe that smaller values of γ (e.g., 10^{-5}) lead to slower convergence and moderately reduced detection accuracy, as the regularization strength adapts conservatively and struggles to align local model updates. As γ increases, both convergence speed and detection performance improve, with $\gamma = 10^{-3}$ achieving the best trade-off, reaching over 99.7% detection accuracy within only 20 communication rounds. However, further increasing γ to 10^{-2} introduces minor instability during early training phases due to overly aggressive adaptation, which can slightly degrade final accuracy. These findings demonstrate that a moderately tuned γ value plays a critical role in balancing convergence efficiency and detection reliability, directly influencing the early filtering of potentially corrupted edge servers and thereby enhancing the overall efficiency of the reputation-guided EDIV process.

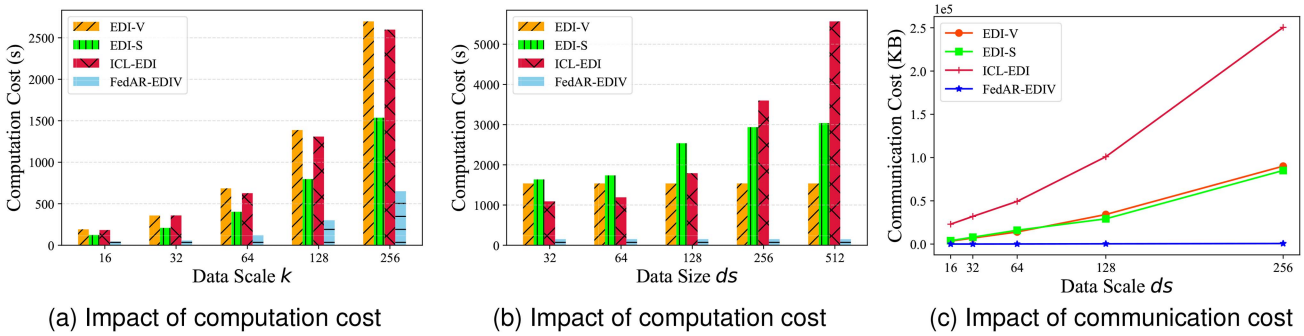
D. Performance Analysis Related to Final EDIV

To ensure a fair comparison and consistent evaluation, we fix the following hyperparameter configuration for Phase 1 of FedAR-EDIV, unless otherwise stated: a total of 100 clients, client fraction $C = 0.5$, local epochs $E = 2$, and batch size $B = 128$. This configuration was selected based on empirical results from Phase 1, demonstrating its effectiveness in achieving high communication speedup and detection accuracy under both IID and non-IID scenarios, as detailed in Section V-B. The output of Phase 1 is subsequently used in Phase 2 to confirm the corruption status of the ESs, which have been identified as potentially corrupted in Phase 1. Specifically, each ES applies the final global model developed in Phase 1 to its local log data to generate a binary prediction. Based on these predictions, only the ESs flagged as potentially corrupted proceed to EDIV confirmation in Phase 2. For the following experiments, we use the KDD99 dataset as the activity log dataset for each ES. Phase 1 achieved statistically indistinguishable outcomes on the KDD99 and CIC2017 datasets, and Phase 2 consumes only the binary corruption labels generated in Phase 1 rather than the log activity dataset. We therefore present Phase 2 results on KDD99 alone, eliminating redundancy while preserving generality. With the optimized configuration and detection pipeline established, we now evaluate the performance of FedAR-EDIV in comparison to baseline EDIV methods.

1) Overall Computation and Communication Cost: To comprehensively evaluate the overall computation and communication efficiency of the proposed FedAR-EDIV framework, we compare its performance against three widely recognized and representative EDIV approaches, namely EDI-V [20], EDI-S [21], and ICL-EDI [7], across various evaluation scenarios involving diverse data scales and data sizes.

Impact of Data Scale on Computation Cost: Fig. 6(a) presents the efficiency of different approaches as the data scale (k) increases from 16 to 256. As expected, a higher k corresponds to a larger number of data replicas requiring verification. The results demonstrate that FedAR-EDIV consistently surpasses EDI-V, EDI-S, and ICL-EDI in verification efficiency across all cases, highlighting its significant performance advantage. This improvement is attributed to FedAR-EDIV's targeted verification strategy, which prioritizes potentially corrupted ESs, thereby reducing the computation cost associated with the more resource-intensive verification methods employed by EDI-V, EDI-S, and ICL-EDI. By limiting the number of replicas undergoing verification, FedAR-EDIV significantly reduces the time required for processing integrity proofs. The efficiency variations observed in different values of k underscore its practical applicability, reinforcing its ability to optimize computation resources during each verification round. On average, FedAR-EDIV outperforms EDI-V by a factor of 3.5 and achieves 2.8 times the efficiency of EDI-S and ICL-EDI. Notably, its scalability remains strong, ensuring stable performance even as k increases. These results establish FedAR-EDIV as an effective and adaptable solution for EDI verification in environments handling multiple data replicas.

Impact of Data Size on Computation Cost: Fig. 6(b) presents the computation cost of each approach as the data size (ds) expands from 16 MB to 1024 MB. The results reveal that ICL-EDI exhibits higher sensitivity to larger data replicas compared to EDI-V, EDI-S, and FedAR-EDIV. Across all data sizes, FedAR-EDIV consistently achieves superior efficiency over its counterparts. Specifically, FedAR-EDIV improves computational efficiency over EDI-S by 0.34 s, 0.37 s, 0.55 s, 0.99 s, and 1.73 s for data sizes of 16 MB, 64 MB, 256 MB,



(a) Impact of computation cost (b) Impact of computation cost (c) Impact of communication cost

Fig. 6. Impact of different parameters on the computation and communication costs.

512 MB, and 1024 MB, respectively. Similarly, it demonstrates reductions of 0.16 s, 0.26 s, 0.30 s, 0.39 s, and 0.59 s compared to EDI-V for the same data sizes. This efficiency gain stems from FedAR-EDIV's selective verification strategy, which focuses on a subset of ESs rather than processing all of them. This targeted approach minimizes the impact of heterogeneous ES resources on verification efficiency. Conversely, EDI-V, EDI-S, and ICL-EDI lack effective preprocessing mechanisms, leading to rigid interactions with all ESs and, consequently, increased computation costs. As ds increases, FedAR-EDIV continues to demonstrate scalability, as verification efficiency remains largely unaffected. For instance, when ds scales from 1 MB to 512 MB, FedAR-EDIV's computation cost rises modestly from 0.03 s to 1 s, whereas EDI-V's increases from 0.3 s to 2.7 s and EDI-S's from 0.2 s to 1.7 s. These results confirm FedAR-EDIV's superior performance in handling varying data sizes while maintaining computational efficiency.

Impact of Data Scale on Communication Cost: Fig. 6(c) compares the communication costs of different integrity verification schemes as the data scale k increases. Unlike EDI-V, EDI-S, and ICL-EDI, which experience significant communication overhead due to intensified ES-AV and inter-ES interactions, FedAR-EDIV maintains a nearly constant communication cost regardless of k.

2) Performance Over Multiple EDIV Rounds: To evaluate the effectiveness of our proposed FedAR-EDIV approach in identifying corrupted ESs across multiple EDIV rounds, we simulated 100 ESs, 30 of which were configured to behave maliciously by tampering with data. The full verification framework was executed over 10 consecutive EDIV rounds to assess how detection performance evolves under the proposed reputation-guided mechanism. We define a true positive (TP) as an ES that is correctly flagged in Phase 1 and subsequently verified in Phase 2 as corrupted. A false positive (FP) corresponds to an honest ES that is incorrectly flagged as corrupted during Phase 1 but is later cleared in the verification phase. A false negative (FN) arises when a truly corrupted ES is missed and not flagged at all, while a true negative (TN) denotes an honest ES that is correctly identified as such and excluded from unnecessary verification. Fig. 7 illustrates the detection summary across all EDIV rounds. Notably, the number of true positives remained consistently at 30 throughout, indicating that all corrupted ESs were successfully identified and confirmed in every round. This result implies a false negative rate of zero across the entire evaluation. However, the framework initially incurred a non-negligible number of false positives—20 in Round 1—which steadily decreased over time due to the influence of the reputation mechanism. By Round 9,

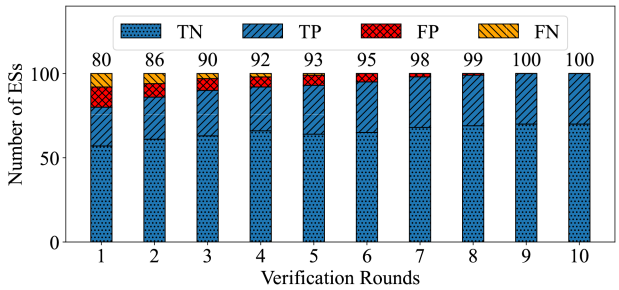


Fig. 7. Performance over multiple EDIV rounds.

the system achieved perfect classification, with zero false positives and zero false negatives, a result that persisted through Round 10. These findings demonstrate that FedAR-EDIV is capable of achieving full detection coverage from the outset, while continuously improving its precision by reducing false alarms over multiple verification rounds. The reputation-based adjustment process thus enables the system to adapt to ES behavior effectively, enhancing both accuracy and efficiency in long-term deployment.

VI. Conclusion and Future Work

In this paper, we propose FedAR-EDIV, an FL-based framework for efficient EDIV in large-scale edge computing environments. By enabling early detection of potentially corrupted ESs, FedAR-EDIV narrows the verification scope and addresses the inefficiencies of traditional 'challenge-response' methods. Unlike baseline FL approaches that assume homogeneous activity data, FedAR-EDIV introduces an adaptive objective regularization strategy to mitigate data heterogeneity across diverse ESs, improving detection accuracy and accelerating FL convergence. This leads to reduced computation and communication costs during final EDIV procedures. To further optimize resource usage across multiple verification rounds, FedAR-EDIV employs a reputation-based mechanism that dynamically prioritizes verifications, assigning fewer challenges to trusted ESs while focusing scrutiny on previously flagged ones. This selective strategy enhances long-term system efficiency. Extensive theoretical analyses validate the correctness, convergence, efficiency, and security of FedAR-EDIV. Experiments across heterogeneous datasets demonstrate substantial improvements over baselines in detection accuracy, convergence speed, and overall EDIV performance. Moreover, leveraging blockchain-based

smart contracts ensures transparency, trust, and data privacy among stakeholders.

Future work will explore advanced adaptive hyperparameter tuning and assess FedAR-EDIV's robustness under evolving adversarial threats. This will further enhance FedAR-EDIV's capability for secure, scalable, and efficient EDIV in decentralized edge intelligence systems.

REFERENCES

[1] IoT Analytics, "Internet of Things (IoT) and non-IoT active device connections worldwide from 2010 to 2025 (in billions)," 2020. [Online]. Available: https://www.statista.com/statistics/1101442/iot-number-of-connected-devices-worldwide/

[2] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: State-of-the-art and research challenges," *J. Internet Serv. Appl.*, vol. 1, pp. 7–18, 2010.

[3] B. Varghese and R. Buyya, "Next generation cloud computing: New trends and research directions," *Future Gener. Comput. Syst.*, vol. 79, pp. 849–861, 2018.

[4] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 450–465, Feb. 2018.

[5] J. Li, H. Yan, and Y. Zhang, "Efficient identity-based provable multi-copy data possession in multi-cloud storage," *IEEE Trans. Cloud Comput.*, vol. 10, no. 1, pp. 356–365, First Quarter, 2022.

[6] W. Tong, B. Jiang, F. Xu, Q. Li, and S. Zhong, "Privacy-preserving data integrity verification in mobile edge computing," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst.*, 2019, pp. 1007–1018.

[7] G. Cui et al., "Efficient verification of edge data integrity in edge computing environment," *IEEE Trans. Services Comput.*, vol. 15, no. 6, pp. 3233–3244, Nov./Dec. 2022.

[8] M. R. Islam, Y. Xiang, M. P. Uddin, Y. Zhao, J. Kua, and L. Gao, "Multiple edge data integrity verification with multi-vendors and multi-servers in mobile edge computing," *IEEE Trans. Mobile Comput.*, vol. 24, no. 6, pp. 4668–4683, Jun. 2025.

[9] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Statist.*, PMLR, 2017, pp. 1273–1282.

[10] Y. Zhao, C. Xu, Y. Qu, Y. Xiang, F. Chen, and L. Gao, "A learning-based hierarchical edge data corruption detection framework in edge intelligence," *IEEE Internet of Things J.*, vol. 11, no. 10, pp. 18366–18380, May 2024.

[11] B. Jia, X. Zhang, J. Liu, Y. Zhang, K. Huang, and Y. Liang, "Blockchain-enabled federated learning data protection aggregation scheme with differential privacy and homomorphic encryption in IIoT," *IEEE Trans. Ind. Informat.*, vol. 18, no. 6, pp. 4049–4058, Jun. 2022.

[12] F. Zafar et al., "A survey of cloud computing data integrity schemes: Design challenges, taxonomy and future trends," *Comput. Secur.*, vol. 65, pp. 29–49, 2017.

[13] L. Zhou, A. Fu, S. Yu, M. Su, and B. Kuang, "Data integrity verification of the outsourced Big Data in the cloud environment: A survey," *J. Netw. Comput. Appl.*, vol. 122, pp. 1–15, 2018.

[14] A. Juels and B. S. Kaliski Jr, "Pors: Proofs of retrievability for large files," in *Proc. 14th ACM Conf. Comput. Commun. Secur.*, 2007, pp. 584–597.

[15] C. B. Tan, M. H. A. Hijazi, Y. Lim, and A. Gani, "A survey on proof of retrievability for cloud data integrity and availability: Cloud storage state-of-the-art, issues, solutions and future trends," *J. Netw. Comput. Appl.*, vol. 110, pp. 75–86, 2018.

[16] G. Ateniese et al., "Provable data possession at untrusted stores," in *Proc. 14th ACM Conf. Comput. Commun. Secur.*, 2007, pp. 598–609.

[17] W. Guo et al., "Outsourced dynamic provable data possession with batch update for secure cloud storage," *Future Gener. Comput. Syst.*, vol. 95, pp. 309–322, 2019.

[18] W. Tong, W. Chen, B. Jiang, F. Xu, Q. Li, and S. Zhong, "Privacy-preserving data integrity verification for secure mobile edge storage," *IEEE Trans. Mobile Comput.*, vol. 22, no. 9, pp. 5463–5478, Sep. 2023.

[19] B. Li et al., "Cooperative assurance of cache data integrity for mobile edge computing," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 4648–4662, 2021.

[20] B. Li, Q. He, F. Chen, H. Jin, Y. Xiang, and Y. Yang, "Auditing cache data integrity in the edge computing environment," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 5, pp. 1210–1223, May 2021.

[21] B. Li, Q. He, F. Chen, H. Jin, Y. Xiang, and Y. Yang, "Inspecting edge data integrity with aggregate signature in distributed edge computing environment," *IEEE Trans. Cloud Comput.*, vol. 10, no. 4, pp. 2691–2703, Fourth Quarter, 2022.

[22] B. Li, Q. He, L. Yuan, F. Chen, L. Lyu, and Y. Yang, "EdgeWatch: Collaborative investigation of data integrity at the edge based on blockchain," in *Proc. 28th ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2022, pp. 3208–3218.

[23] Y. Ding, Y. Li, W. Yang, and K. Zhang, "Edge data integrity verification scheme supporting data dynamics and batch auditing," *J. Syst. Archit.*, vol. 128, 2022, Art. no. 102560.

[24] Y. Zhao, Y. Qu, F. Chen, Y. Xiang, and L. Gao, "Data integrity verification in mobile edge computing with multi-vendor and multi-server," *IEEE Trans. Mobile Comput.*, vol. 23, no. 5, pp. 5418–5432, May 2024.

[25] Y. Ren, J. Qi, Y. Liu, J. Wang, and G.-J. Kim, "Integrity verification mechanism of sensor data based on bilinear map accumulator," *ACM Trans. Internet Technol.*, vol. 21, no. 1, pp. 1–19, 2021.

[26] G. Xu, Y. Yang, C. Yan, and Y. Gan, "A probabilistic verification algorithm against spoofing attacks on remote data storage," *Int. J. High Perform. Comput. Netw.*, vol. 9, no. 3, pp. 218–229, 2016.

[27] A. Blázquez-Garcıá, A. Conde, U. Mori, and J. A. Lozano, "A review on outlier/anomaly detection in time series data," *ACM Comput. Surv.*, vol. 54, no. 3, pp. 1–33, 2021.

[28] M. H. ur Rehman, K. Salah, E. Damiani, and D. Svetinovic, "Towards blockchain-based reputation-aware federated learning," in *Proc. IEEE Conf. Comput. Commun. Workshops*, 2020, pp. 183–188.

[29] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. Mach. Learn. Syst.*, vol. 2, pp. 429–450, 2020.

[30] C. Xu, Y. Qu, P. W. Eklund, Y. Xiang, and L. Gao, "BAFL: An efficient blockchain-based asynchronous federated learning framework," in *Proc. IEEE Symp. Comput. Commun.*, 2021, pp. 1–6.

[31] C. Xu, Y. Qu, T. H. Luan, P. W. Eklund, Y. Xiang, and L. Gao, "An efficient and reliable asynchronous federated learning scheme for smart public transportation," *IEEE Trans. Veh. Technol.*, vol. 72, no. 5, pp. 6584–6598, May 2023.

[32] Z. Wang and Q. Hu, "Blockchain-based federated learning: A comprehensive survey," 2021, *arXiv:2110.02182*.

[33] Y. Liu et al., "Deep anomaly detection for time-series data in industrial IoT: A communication-efficient on-device federated learning approach," *IEEE Internet of Things J.*, vol. 8, no. 8, pp. 6348–6358, Apr. 2021.

[34] B. Wang, F. Wu, Y. Long, L. Rimanic, C. Zhang, and B. Li, "DataLens: Scalable privacy preserving training via gradient compression and aggregation," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2021, pp. 2146–2168.

[35] A. M. Abdelmoniem, A. Elzanaty, M.-S. Alouini, and M. Canini, "An efficient statistical-based gradient compression technique for distributed training systems," in *Proc. Mach. Learn. Syst.*, vol. 3, pp. 297–322, 2021.

[36] K. D. Bowers, A. Juels, and A. Oprea, "Proofs of retrievability: Theory and implementation," in *Proc. ACM Workshop Cloud Comput. Secur.*, 2009, pp. 43–54.

[37] J. Zhang, Z. Zhang, and H. Guo, "Towards secure data distribution systems in mobile cloud computing," *IEEE Trans. Mobile Comput.*, vol. 16, no. 11, pp. 3222–3235, Nov. 2017.

Md Rashedul Islam (Member, IEEE) received the BSc degree in computer science and engineering from Hajee Mohammad Danesh Science and Technology University (HSTU), Bangladesh, the MSc degree in computer science and engineering from the Rajshahi University of Engineering and Technology (RUET), Bangladesh, in 2019. He is currently working toward the PhD degree with the School of Information Technology, Deakin University. He is an academic faculty member with HSTU, Bangladesh. His research direction includes edge computing, machine learning, FL, blockchain applications, and remote sensing image analysis.

Md Palash Uddin (Member, IEEE) received the PhD degree in IT from Deakin University, Australia. He is currently a postdoctoral research fellow with the School of IT, Deakin University, Australia, and also a faculty member with Hajee Mohammad Danesh Science and Technology University, Bangladesh. He has authored more than 75 journal and conference papers in leading venues, including *IEEE Transactions on Parallel and Distributed Systems*, *IEEE Transactions on Services Computing*, *IEEE Transactions on Mobile Computing*, *IEEE Transactions on Information Forensics and Security*, *IEEE Journal on Selected Areas in Communications*, *IEEE Signal Processing Letters*, and *ACM Computing Surveys*. Additionally, he actively reviews for top-tier journals and conferences. His primary research interests lie in FL and machine learning, focusing on their applications in data privacy, security, and integrity verification.

Yong Xiang (Senior Member, IEEE) received the PhD degree in electrical and electronic engineering from The University of Melbourne, Australia. He is a professor with the School of Information Technology, Deakin University, Australia. His research interests include cybersecurity, data science, machine learning & AI, distributed computing, and communications engineering. He has published seven authored books, more than 300 journal articles, and more than 120 conference papers in these areas. He is the associate editor of *IEEE Transactions on Network Science and Engineering*, the associate editor of *IEEE Communications Surveys and Tutorials*, and the associate editor of *Computer Standards and Interfaces*. He was the senior area editor of *IEEE Signal Processing Letters* and the associate editor of *IEEE Access*, and served as guest editor for several journals, such as *IEEE Transactions on Industrial Informatics*, *IEEE Transactions on Intelligent Transportation Systems*, *IEEE Multimedia*, etc. He has served as honorary chair, general chair, program chair, TPC chair, symposium chair and track chair for many conferences, and was invited to give keynotes with numerous conferences.

Kuo-Hui Yeh (Senior Member, IEEE) received the MS and PhD degrees in information management from the National Taiwan University of Science and Technology, Taipei, Taiwan, in 2005 and 2010, respectively. He serves as a professor with the Institute of Artificial Intelligence Innovation, National Yang Ming Chiao Tung University, Hsinchu, Taiwan. Prior to this appointment, he was a professor with the Department of Information Management, National Dong Hwa University, Hualien, Taiwan, from 2012 to 2024. He has contributed more than 150 articles to esteemed journals and conferences, covering a wide array of research interests such as IoT security, Blockchain, NFC/RFID security, authentication, digital signatures, data privacy and network security. Furthermore, He plays a pivotal role in the academic community, serving as an associate editor (or editorial board member) for several journals, including the *Journal of Information Security and Applications (JISA)*, *Human-centric Computing and Information Sciences (HCIS)*, *Digital Health*, *Symmetry*, *Journal of Internet Technology (JIT)* and *CMC-Computers, Materials & Continua*. In the professional realm, he is recognized as a fellow of the British Computer Society (BCS) and a holds memberships with BCS, $(ISC)^2$, ISA, ISACA, CAA, and CCISA. His professional qualifications include certifications like CISSP, CISM, Security+, ISO 27001/27701/42001 Lead Auditor, IEC 62443-2-1 Lead Auditor, and ISA/IEC 62443 Cybersecurity Expert, covering fundamentals, risk assessment, design, and maintenance specialties.

Lu Liu (Member, IEEE) received the PhD degree from Surrey Space Centre, the University of Surrey. He had worked as a research fellow with the WRG e-Science Centre, University of Leeds. He served as the head of the School of Computing and Mathematical Sciences, University of Leicester, U.K., from 2019–2023. His research interests are in the areas of data analytics, service computing, sustainable computing, and the Internet of Things. He has more than 250 scientific publications in reputable journals, academic books, and international conferences. He has been the recipient of 7 Best Paper Awards from international conferences and was invited to deliver 8 keynote speeches with international conferences. He is a fellow of BCS (British Computer Society). He is currently serving as an associate editor for *Peer-to-Peer Networking and Application (PPNA)* and *Big Data Mining and Analytics (BDMA)*.

Jonathan Kua (Member, IEEE) received the BEng (First Class Hons.) degree in telecommunications and network engineering and the PhD degree in telecommunications engineering from the Swinburne University of Technology, Australia, in 2014 and 2019, respectively. He is currently a senior lecturer in the Internet of Things within the School of Information Technology, Deakin University, Australia. His research interests include low-latency data transport protocols, adaptive streaming, content delivery, the Internet of Things, and industrial cyber-physical systems, with an overarching focus on improving their network performance and quality of service. He was the recipient of the Netflix Ph.D. scholarship award (2015–2019) and the second runner-up of the DASH-IF Best PhD Dissertation Award on "Algorithms and Protocols for Adaptive Content Delivery over the Internet" (2019). He currently serves as a guest editor for the *IEEE Journal on Selected Areas in Communications (JSAC)* Special Issue on "Co-Design of Communication, Computing, and Control in Industrial Cyber-Physical Systems".