EUROPEAN JOURNAL OF EMERGING CYBERSECURITY AND INFORMATION PROTECTION

A TRUST-BASED INCENTIVE FRAMEWORK FOR FEDERATED LEARNING IN EDGE ENVIRONMENTS WITH DIVERSE PARTICIPANTS

Dr. Tobias Muller
Chair of Network Architectures and Services, Technical University of Munich, Germany

Chan Chiang
School of Computer Science, Tsinghua University, China

Yuan Hsiao
School of Computer Science, Tsinghua University, China

V0LUME01 ISSUE01 (2024)
Published Date: 17 December 2024 // Page no.: - 14-34

ABSTRACT

Federated Learning (FL) offers a privacy-preserving paradigm for collaborative model training, particularly well-suited for edge computing. However, the inherent heterogeneity of edge clients—encompassing data distribution, computational capabilities, and reliability—poses significant challenges to FL's effectiveness, including performance degradation and vulnerability to malicious participants. This article proposes a novel trust-based incentive mechanism designed to address these issues by dynamically evaluating client trustworthiness and adjusting incentives accordingly. Our framework integrates a multi-faceted trust score that considers contribution quality, reliability, and the detection of malicious behavior. By rewarding trustworthy contributions and penalizing unreliable actions, the proposed mechanism aims to enhance global model accuracy, improve robustness against attacks, and foster fairness among diverse participants. This approach encourages consistent, high-quality contributions while mitigating risks from untrustworthy clients, paving the way for more resilient and efficient federated learning deployments in real-world edge environments.

Keywords: Federated Learning, Edge Computing, Incentive Mechanism, Trust Management, Heterogeneous Clients, Data Heterogeneity, Model Poisoning, Client Selection.

INTRODUCTION

The rapid advancements in artificial intelligence (AI) and the proliferation of edge devices have ushered in an era where intelligent systems are increasingly deployed at the network's periphery. Concurrently, growing concerns over data privacy and stringent regulations, such as the General Data Protection Regulation (GDPR) and the Health Insurance Portability and Accountability Act (HIPAA), have made traditional centralized machine learning approaches less viable due to the necessity of collecting vast amounts of sensitive user data [1]. Federated Learning (FL) has emerged as a groundbreaking distributed machine learning paradigm that addresses these challenges by enabling collaborative model training without requiring raw data to leave the client devices [2]. This "data-at-source" principle significantly enhances data locality and privacy, making FL particularly appealing for sensitive applications in healthcare, finance, and smart city infrastructures [3].

Despite its immense potential, the real-world deployment of federated learning, especially in heterogeneous edge computing environments, faces a myriad of critical challenges. These challenges primarily stem from three fundamental types of heterogeneity: statistical, system, and behavioral [4, 5].

Firstly, statistical heterogeneity, often referred to as non-Independent and Identically Distributed (non-IID) data, is a pervasive issue in FL. Data collected by diverse edge clients typically originates from distinct users, geographical locations, or operational contexts, leading to significant variations in data distributions. This can manifest as label skew (clients having data predominantly from certain classes), feature skew (different feature distributions), or quantity skew (imbalanced data sizes across clients) [6, 7]. Such non-IID data severely impedes the convergence rate and generalization capabilities of the global model, as local updates from disparate distributions may pull the global model in conflicting directions [8, 39, 40].

Secondly, system heterogeneity arises from the vast differences in the computational capabilities, memory, battery life, and network bandwidth of edge devices [38]. Clients may experience varying communication delays, intermittent connectivity, or even training interruptions due to resource constraints or dynamic network conditions. These disparities create "stragglers"—clients

that are significantly slower than others—which can prolong training times and widen performance gaps at the system level, making it challenging to synchronize model updates efficiently [43].

Most critically, behavioral heterogeneity introduces a layer of complexity related to client motivations and trustworthiness. In an open and decentralized FL environment, clients are often self-interested entities that may prioritize their own utility (e.g., minimizing resource consumption) over the collective good of the global model [9]. This can lead to strategic participation, where clients contribute minimally (free-riding) or even upload perturbed gradients to reduce their computational burden while still benefiting from the aggregated model. Furthermore, the presence of malicious clients poses a severe threat, as they can intentionally inject corrupted data (data poisoning) or manipulate model updates (model poisoning, backdoor attacks) to degrade the global model's performance, compromise its integrity, or introduce specific vulnerabilities [10, 11, 12, 13, 14, 34, 35, 36]. The dynamic nature of these behaviors, including on-off attacks or mimicry attacks, makes them particularly difficult to detect and counter effectively [44, 45].

To address these multifaceted challenges, extensive research has been conducted across various domains, including robust aggregation algorithms, security mechanisms, trust management, and incentive design [11, 12]. While robust aggregation techniques like Krum [13] and Trimmed Mean [14] can filter outlier updates and enhance resilience, they often employ rigid filtering strategies that may inadvertently penalize benign but high-variance clients, thus sacrificing model diversity. Similarly, existing trust mechanisms, such as FedTrust [31] and TrustFL [32], assess client credibility based on model similarity or behavioral patterns but are frequently decoupled from the incentive schemes. Current incentive mechanisms typically allocate rewards based on static indicators like data volume or training time, which are susceptible to manipulation by strategic clients and can lead to unfair reward distribution and incentive abuse under adversarial conditions [17, 18].

A significant limitation in the current landscape is the lack of a unified, integrated approach where trust modeling, robust aggregation, and incentive mechanisms are systematically coupled. This decoupling often results in persistent internal incentive biases, distorted client behaviors, and a loss of control over the FL training process. Moreover, the inherently dynamic nature of client behaviors in edge environments means that one-shot scoring or static thresholds are insufficient to capture long-term trends and fluctuations, leading to misjudgments and misallocated rewards that compromise overall fairness and stability.

This article proposes a novel Trust-Aware Incentive Mechanism (TAIM) that unifies client trust modeling, incentive feedback, and robust aggregation within a single framework. Guided by the principle that "trust drives participation, incentive motivates resource investment, and aggregation ensures robustness," TAIM aims to achieve deep coupling between strategic and optimization layers. This integration is designed to enhance system robustness, ensure fairness in resource allocation, and promote sustainable, high-quality participation from diverse clients in heterogeneous edge computing environments.

The main contributions of this paper are summarized as follows:

- We develop a comprehensive dynamic trust modeling framework that integrates multiple behavioral indicators, including participation frequency, gradient consistency, and contribution effectiveness. This framework is designed to capture the dynamic behavioral trajectories of clients and quantify their stability and reliability over time.
- We formulate a trust-driven incentive mechanism based on Stackelberg game theory. This mechanism allows the server (leader) to strategically allocate rewards, guiding clients (followers) to invest optimal resources and converge towards rational strategies that align with the system's objectives, ultimately enhancing their participation and resource commitment.
- We introduce a confidence-aware smoothing aggregation algorithm that incorporates a novel soft filtering function. This function intelligently suppresses the influence of low-trust updates while allowing for their potential recovery, thereby striking a crucial balance between maintaining robustness against malicious attacks and preserving beneficial client diversity.
- We conduct extensive experimental validation across multiple non-IID datasets (FEMNIST, CIFAR-10, Sent140) and various adversarial scenarios. Through comparative analysis against existing baselines, we demonstrate the superior robustness, fairness, and convergence performance of the proposed TAIM.

The remainder of this paper is organized as follows: Section 2 provides a detailed review of related works and key methodologies in federated learning. Section 3 formalizes the system model, characterizes client heterogeneity, and defines the problem formulation. Section 4 elaborates on the design, theoretical analysis, and algorithmic details of the proposed TAIM. Section 5 presents the comprehensive experimental validation and comparative evaluation of TAIM's performance. Finally, Section 6 discusses the limitations of the current work and outlines promising directions for future research.

2. RELATED WORK

To construct a resilient and efficient federated learning system tailored for the complexities of heterogeneous edge environments, researchers have extensively explored three interconnected yet often compartmentalized research directions: client incentive

mechanisms, trust modeling, and robust aggregation algorithms. This section systematically reviews the state-of-the-art in each of these areas, identifies their inherent limitations, and highlights the unique contributions and distinctiveness of our proposed unified framework.

2.1. Incentive Mechanism Design in Federated Learning

In practical FL deployments, the voluntary participation of numerous clients, each possessing varying resource capabilities and self-interests, is crucial. However, concerns regarding privacy, computational resource consumption, and communication overhead often make clients reluctant to participate consistently or contribute high-quality updates [19, 20]. Therefore, designing effective incentive mechanisms to enhance client participation, ensure the quality of their contributions, and prevent free-riding has become a paramount research challenge [21, 22]. Comprehensive surveys, such as that by Zhou [23], categorize these mechanisms, emphasizing the prominence of economic theories like game theory, auction theory, and contract theory in addressing the challenge of motivating self-interested clients.

Early studies in FL incentive design primarily focused on resource-driven models. These models often rewarded clients based on quantifiable metrics like the volume of data contributed or the duration of training time. For instance, Zhang et al. [24] proposed an incentive model based on a Stackelberg game, where rewards were allocated based on the quality of uploaded models rather than merely data volume or training time. While such methods could improve system efficiency to some extent by encouraging resource commitment, they frequently overlooked the inherent heterogeneity in model quality and the strategic participation behaviors of clients. This oversight could lead to disproportionate rewards for strategically behaving clients who might appear to contribute significantly but whose updates offer minimal actual benefit to the global model.

To enhance fairness and robustness, some works introduced more sophisticated economic frameworks, particularly game theory and marginal contribution analysis. Huang et al. [25] and Xia et al. [26] designed demand-based reward allocation strategies that leveraged Shapley value estimation. The Shapley value, a concept from cooperative game theory, provides a fair way to distribute the total gains from a cooperative endeavor among its participants, based on their marginal contributions. In FL, it quantifies each client's marginal improvement to the global model's performance. While theoretically sound for fairness, computing exact Shapley values is computationally intensive (NP-hard) for large numbers of clients, necessitating approximations that can introduce their own biases or computational overhead. Different game-theoretic approaches have also been explored; for example, Pang et al. [27] designed an incentive auction specifically for heterogeneous client selection, aiming to create a market-based environment for efficient resource allocation. Auction mechanisms, while effective in competitive bidding scenarios, differ from our approach by emphasizing short-term competitive bidding rather than fostering long-term trust and collaborative behavior.

Recognizing the limitations of static or purely contribution-based metrics, recent works have shifted towards evaluating client behavior over time. For example, Al-Saedi et al. [28] proposed a method to predict client contributions by evaluating their past behaviors, with the goal of proactively selecting more reliable participants for future rounds. This predictive approach offers a valuable complement to reactive trust-scoring mechanisms, which assess credibility after each round to dynamically adjust rewards and aggregation weights. Furthermore, some research has ventured into using reinforcement learning (RL) for dynamic incentive strategy generation. Ma et al. [29] introduced a deep reinforcement learning algorithm for incentive-based demand response, which continuously optimizes interaction strategies using client states and feedback signals, even under conditions of incomplete information. While RL-based approaches improve adaptability by learning optimal incentive policies, they often rely on global reward signals and may struggle to capture fine-grained individual trustworthiness or defend against sophisticated strategic manipulation attempts by individual clients. The increasing complexity and diversity of these mechanisms also highlight the growing need for standardized evaluation frameworks, a gap addressed by platforms like FLWB [30], which facilitate reproducible performance comparisons of FL algorithms.

In summary, a critical limitation of current incentive mechanisms is their insufficient modeling and utilization of client behavioral credibility. This deficiency often results in misallocated or abused rewards, leading to suboptimal global model performance and reduced client retention. Our study aims to overcome this by deeply embedding dynamic trust scores into the game-theoretic incentive function, thereby constructing a behavior-driven resource allocation mechanism that enhances system security and participation stability.

2.2. Trust Modeling and Robust Aggregation in Federated Learning

Security threats in federated learning are a major concern, primarily stemming from clients uploading malicious or low-quality updates that can significantly degrade global model performance, introduce backdoors, or compromise data privacy. To mitigate these pervasive threats, trust modeling and robust aggregation have emerged as central research topics.

In the realm of trust modeling, various methods have been proposed to evaluate client credibility from different perspectives. FedTrust [31] calculates trust scores based on the similarity among uploaded models, adjusting aggregation weights accordingly. The underlying assumption is that benign clients will produce similar

model updates, whereas malicious ones will deviate significantly. TrustFL [32] takes a more dynamic approach, adjusting client weights based on performance fluctuations observed on a public validation set and the consistency of feature representations learned by local models. This allows for a more direct assessment of a client's actual impact on model utility. Lyubchyk et al. [33] further advanced this by constructing a composite trust scoring system that leverages multi-dimensional indicators to reflect a client's long-term behavioral stability and reliability, moving beyond single-metric assessments. These approaches aim to identify and isolate untrustworthy participants before their updates can harm the global model.

Concurrently, robust aggregation algorithms provide a crucial line of defense against various poisoning attacks. Methods such as Krum [34] and Trimmed Mean [35] are designed to eliminate outlier gradients or select consistent subsets of updates to enhance robustness. Krum, for instance, selects the update that is closest to its k-nearest neighbors in the gradient space, effectively discarding outliers. Trimmed Mean, as its name suggests, sorts the gradients and discards a certain percentage of the largest and smallest values before averaging. While these methods offer effective defenses against simple poisoning attacks, most rely on static thresholds or distance-based filtering, which struggle to adapt to dynamic client behaviors and often overlook the strategic interactions among participants. More sophisticated attacks, like those that mimic benign behavior or operate intermittently, can often bypass these rigid filtering strategies.

Recently, some works have attempted to couple trust mechanisms with robust aggregation to create more adaptive defense systems. Perry et al. [36] introduced update correlation analysis for dynamic detection of collusive poisoning attacks, where multiple malicious clients coordinate their actions. Abri et al. [37] modeled the trust learning process as a Markov decision process to recognize potential attack states and adapt defense strategies. However, a persistent limitation in these integrated approaches is their neglect of client responses to incentive feedback. Without a proper incentive regulation mechanism that encourages honest behavior and penalizes malicious acts, the effectiveness of even sophisticated trust scoring and aggregation strategies can be compromised in the long run. Malicious clients, if not appropriately disincentivized, may continue to find new ways to exploit the system.

This work distinguishes itself by proposing a soft trust filtering mechanism that introduces a smoothing suppression function during aggregation. This function intelligently attenuates the impact of low-trust updates without completely discarding them, thereby avoiding overly harsh penalties for edge clients with transient behavioral fluctuations. More importantly, our trust evaluation is deeply coupled with the incentive allocation function, forming a "high trust-high incentive-high participation" positive feedback loop. This holistic integration enhances adaptive defense capabilities and promotes long-term strategy stability within the FL ecosystem.

2.3. Federated Modeling Mechanisms for Heterogeneous Edge Environments

The deployment of federated systems in real-world, heterogeneous edge environments presents unique challenges that go beyond statistical and behavioral aspects. These challenges encompass non-ideal conditions such as diverse device capabilities, resource imbalances, frequent communication disruptions, and highly dynamic client availability [38, 39, 40]. These factors significantly amplify the complexities related to fairness, robustness, and overall system efficiency.

To address system-level heterogeneity, various FL algorithms have been proposed. FedProx [41] introduces a proximal regularization term into the local training objective function. This term penalizes local model updates that deviate significantly from the global model, thereby limiting model divergence and improving global convergence, especially in non-IID settings. FedNova [42] focuses on normalizing updates to unify contribution scales across clients, ensuring that clients with varying computational speeds or local training steps contribute proportionally to the global model. FedCS [43] proposes a bandwidth-aware client selection strategy, optimizing training efficiency by prioritizing clients with better network connectivity under communication constraints. Other works have concentrated on the timeliness of information, proposing Age of Information (AoI)-aware client selection or update weighting schemes to prioritize fresher updates from clients with better connectivity, thereby mitigating the negative impact of stragglers (slow clients) and stale models [22]. While these approaches have achieved notable progress in system optimization and efficiency, they largely ignore the dynamic nature of client participation and the strategic evolution of client behaviors. This oversight makes them less effective in open edge environments characterized by frequent malicious behaviors and self-interested participants.

In particular, under the presence of strategic participants, clients may actively seek to evade detection and manipulate rewards through sophisticated tactics. These tactics include mimicry attacks, where malicious clients attempt to imitate the gradients of high-trust benign clients to evade detection [44], or intermittent poisoning, where attacks are launched sporadically to avoid consistent detection patterns. Clients might also engage in frequent switching between honest and malicious behaviors, or between different attack types, ultimately undermining the long-term stability and trustworthiness of the FL system [45]. Therefore, "behavioral trustworthiness" must be considered a core constraint and an integral component in federated learning systems to enable multi-objective optimization under trustworthy

guidance.

Our work integrates edge heterogeneity modeling, dynamic trust evaluation, and incentive-response mechanisms to construct a comprehensive trust-driven game-theoretic regulation framework at the strategic level. By incorporating a soft suppression strategy during aggregation, our approach achieves a delicate balance between robustness, incentive compatibility, and resource adaptation. This provides a systematic modeling paradigm for building secure, controllable, and truly trustworthy federated learning systems at the edge, capable of operating effectively in dynamic and untrusted open environments.

3. SYSTEM MODEL AND PROBLEM FORMULATION

In this section, we lay the foundational groundwork for our proposed Trust-Aware Incentive Mechanism (TAIM). We begin by formalizing the basic structure of a federated learning system. Subsequently, we construct a comprehensive system modeling framework specifically tailored to capture the multifaceted heterogeneity prevalent in edge computing environments, introducing the concepts of dynamic trust modeling and incentive allocation mechanisms. Finally, we define a unified optimization objective that TAIM aims to achieve.

3.1. Federated Learning Task Modeling

We consider a cross-device federated learning scenario comprising a central server and a set of N edge clients, denoted as $C=\{c1,c2,\ldots,cN\}$. Each client ci possesses a local dataset Di, which is typically characterized by a high degree of non-Independent and Identically Distributed (non-IID) properties. The overarching objective of the system is to collaboratively train a robust and high-performing global machine learning model, denoted as w, without requiring any client to directly share its raw, sensitive data with the central server or other clients.

The global optimization problem in federated learning can be formulated as minimizing a global loss function F(w):

$$\min_w F(w)=\sum_{i=1}^{N}\frac{|D_i|}{|D|}F_i(w)$$

where $D=\bigcup_{i=1}^{N}D_i$ represents the total dataset across all clients, |Di| is the size of client ci's local dataset, and Fi(w) is the local loss function for client ci, typically defined as $F_i(w)=\frac{1}{|D_i|}\sum_{(x,y)\in D_i}\ell(w;x,y)$, where ℓ is the loss incurred on a single data sample (x,y).

The collaborative training process unfolds over a series of discrete communication rounds, denoted by t. During each round t, the central server orchestrates the following sequence of steps:

1. Client Selection: The server selects a subset of clients, $S_t\subseteq C$, to participate in the current training round. This selection can be random or based on specific criteria (e.g., client availability, historical performance, or as we propose, trust scores).

2. Global Model Broadcast: The server broadcasts the current global model parameters, wt, to all selected clients in St.

3. Local Training: Each selected client $c_i\in S_t$ downloads wt and performs local model training using its private dataset Di. This typically involves several epochs of stochastic gradient descent (SGD) or a similar optimization algorithm to minimize its local loss function Fi(w).

$$w_{i}^{t+1}=w^t-\eta\nabla F_i(w^t)$$

where η is the local learning rate. The client then computes its local model update, $\Delta w_i^t=w_i^{t+1}-w^t$ (or often, the local gradient $\nabla F_i(w^t)$).

4. Local Update Upload: Each client $c_i\in S_t$ uploads its computed model update Δw_i^t (or its local model w_i^{t+1}) to the central server.

5. Server Aggregation: Upon receiving updates from all participating clients in St, the server aggregates these updates to produce a new global model for the next round, wt+1. The most common aggregation method is Federated Averaging (FedAvg), which performs a weighted average of the local updates:

$$w^{t+1}=w^t+\sum_{i\in S_t}\frac{|D_j|}{\sum_{j\in S_t}|D_j|}|D_i|\Delta w_i^t$$

This new global model wt+1 is then ready for the next communication round.

3.2. Heterogeneity and Behavior Modeling

The inherent characteristics of edge computing environments introduce significant complexities to the standard FL paradigm, primarily due to various forms of heterogeneity. To accurately model and address these challenges, we categorize client states and behaviors from three critical perspectives:

1. Statistical Heterogeneity:

This refers to the variations in data distributions across different clients' local datasets (Di). Unlike the idealized Independent and Identically Distributed (IID) assumption in traditional distributed learning, real-world FL datasets are almost always non-IID. This heterogeneity can manifest in several ways:

- Label Skew (Concept Drift): Clients may have data predominantly from a subset of classes. For example, in an image classification task, one mobile phone user might primarily take pictures of pets, while another takes pictures of landscapes. This leads to local models specializing in different classes, potentially hindering global generalization.
- Feature Skew (Covariate Shift): Even if label distributions are similar, the feature distributions for a given label might vary. For instance, images of the same object taken under different lighting conditions or from different angles by various devices.
- Quantity Skew (Data Imbalance): Clients may

possess vastly different amounts of local data. Some edge devices might have extensive historical data, while others have very limited samples, leading to varying contributions in terms of data volume.

- Concept Shift: The relationship between features and labels might change across clients or over time. For example, the definition of "spam" might evolve differently for different users.

These disparities can cause local models to diverge significantly, making global model convergence slow, unstable, or leading to a poorly generalized model that performs suboptimally on unseen data.

2. System Heterogeneity:

This encompasses the variations in hardware capabilities and network conditions among edge devices.

- Computational Capabilities (ρi): Clients differ widely in their processing power (CPU/GPU), memory capacity, and battery life. A high-end smartphone can train a local model much faster than a low-power IoT sensor. This leads to varying local training times.

- Communication Latency (li): Network conditions (Wi-Fi, 5G, cellular) and geographical locations introduce varying communication delays and bandwidth constraints. Some clients might be connected via high-speed, low-latency networks, while others operate on unstable or congested links.

- Availability and Connectivity: Edge devices are often mobile and may experience intermittent connectivity or go offline frequently. This dynamic availability means that the set of active clients can change unpredictably from round to round, leading to "stragglers" (clients that are too slow to complete their updates within the round deadline) or complete dropouts.

These system-level heterogeneities can significantly impact the efficiency and synchronization of the FL process, leading to delays, resource underutilization, or even system failures if not properly managed.

3. Behavioral Heterogeneity:

This refers to the diverse motivations and potential anomalies in client behavior within an open FL environment. This is particularly critical as clients are not necessarily benevolent and may act strategically.

- Strategic Participation: Clients are rational agents aiming to maximize their utility. This utility is often a trade-off between the incentives received (e.g., computational resources, monetary rewards, improved model performance) and the costs incurred (e.g., computational effort, energy consumption, communication bandwidth). This can lead to:

  - Free-riding: Clients participate but contribute minimal effort or low-quality updates to benefit from the global model without incurring significant costs.
  - Opportunistic Behavior: Clients might strategically adjust their contribution level based on perceived rewards or system state.

- Malicious Behavior (Adversarial Attacks): Beyond mere self-interest, some clients might be actively malicious, aiming to sabotage the FL process. Common attack types include:

  - Data Poisoning: Injecting corrupted or mislabeled data into their local datasets to degrade the global model's performance or introduce specific vulnerabilities.
  - Model Poisoning (Backdoor Attacks): Uploading maliciously crafted model updates (gradients) that, when aggregated, degrade the global model's accuracy, cause it to misclassify specific inputs (backdoors), or lead to convergence to a suboptimal state.
  - Sybil Attacks: A single malicious entity controls multiple client identities to amplify its influence on the aggregation process.
  - Mimicry Attacks: Malicious clients attempt to imitate the behavior of benign, high-trust clients to evade detection, while still subtly perturbing the global model.
  - On-Off Attacks: Malicious clients alternate between honest and malicious behavior to make detection more difficult and to accumulate trust over time, only to launch a significant attack later.

These behavioral anomalies, especially malicious ones, can destabilize FL training, significantly degrade the global model's performance, or even cause it to crash. To effectively manage these dynamic and often adversarial characteristics, we introduce a Trust-Aware Incentive Mechanism designed to enable continuous behavior perception and adaptive regulation throughout the training process.

3.3. Dynamic Trust Score Modeling

To capture the evolving and dynamic behavioral characteristics of each client, we assign a trust score $\tau_i^t \in [0,1]$ to each client c_i at the beginning of each round t. This score quantitatively represents the overall reliability and trustworthiness of client c_i's recent behavior, with 1 indicating perfect trust and 0 indicating complete untrustworthiness.

The trust score is not static but dynamically updated in each round using an exponential decay rule:

$$\tau_i^t = \gamma \cdot \tau_i^{t-1} + (1-\gamma) \cdot \tilde{\tau}_i^t \quad (1)$$

where $\gamma \in [0,1)$ is the memory decay coefficient. This coefficient determines the influence of past trust scores on the current score. A higher γ implies a longer memory, meaning past behavior has a more lasting impact, while a lower γ emphasizes recent behavior. This allows the system to adapt to changes in client behavior over time. $\tilde{\tau}_i^t$ is the instantaneous trust score for client c_i in round t, reflecting its behavior in the most recent round.

The instantaneous trust score $\tau \sim it$ is defined as a weighted sum of multiple behavioral indicators, providing a multi-dimensional assessment of trustworthiness:

$$\tau \sim it = \lambda 1 \phi it + \lambda 2 \psi it + \lambda 3 \omega it, j=1 \sum \lambda 3 j=1 \quad (2)$$

Here, $\lambda 1, \lambda 2, \lambda 3$ are non-negative weighting coefficients that sum to 1, allowing the system to prioritize different aspects of trust based on the application's requirements. The individual behavioral indicators are:

- Participation Frequency (ϕit): This metric quantifies the consistency of client ci's participation. It is typically defined as the proportion of active rounds (where the client successfully submitted an update) within a predefined sliding window of the past Twindow rounds. A higher ϕit indicates a more reliable and consistently available client, crucial for maintaining the FL process's continuity.

- Gradient Consistency (ψit): This measures how consistent a client's local model update is with the aggregated global direction of updates. It is calculated as the cosine similarity between the local update Δwit and the global average update direction Δwt (or the global model's gradient).

$$\psi it = ||\Delta wit||2 \cdot ||\Delta wt||2 \Delta wit \cdot \Delta wt$$

A high cosine similarity (close to 1) suggests that the client's update aligns with the general consensus, indicating benign behavior. A low or negative similarity might signal a malicious update or a significantly divergent data distribution.

- Contribution Effectiveness (ωit): This metric directly assesses the positive impact of client ci's update on the global model's performance. It is quantified as the improvement in validation error (or reduction in loss) brought about by the client's update. This can be measured by applying the client's local model (or a hypothetical global model incorporating only its update) on a small, publicly available validation dataset.

$$\omega it = L(wt) L(wt) - L(wt + \Delta wit)$$

where $L(wt)$ is the loss of the global model wt on the validation set, and $L(wt+\Delta wit)$ is the hypothetical loss if only client i's update were applied. A higher ωit implies a more beneficial contribution to the global model's learning objective.

The dynamic trust score τit serves a dual purpose: it acts as a descriptive indicator of client behavior and, more importantly, as a crucial control variable that influences both the aggregation process and the incentive allocation mechanism.

3.4. Incentive Mechanism and Optimization Objective

The design of an effective incentive mechanism is paramount for motivating clients to participate actively and ensure the quality of their contributions in FL. The server, as the orchestrator, allocates incentives to participating clients. Let rit denote the incentive allocated by the server to client ci in round t. This allocation is subject to a total budget constraint for each round:

$$i \in St \sum rit \leq Rt, rit \geq 0 \quad (3)$$

where Rt is the total incentive budget available to the server in round t.

Each client, in response to the incentives, decides on its local resource investment, xi, to complete the training task. This resource investment could represent computational cycles, energy consumption, or the duration of local training. The client's utility function, Ui(xi), is defined to capture the trade-off between the benefits received (incentives) and the costs incurred (resource consumption):

$$Ui(xi) = \eta \cdot \tau it \cdot \sum j \in Stxjxi - (aixi2 + bixi) \quad (4)$$

In this utility function:

- The first term, $\eta \cdot \tau it \cdot \sum j \in Stxjxi$, represents the trust-weighted share of incentives. η is a scaling factor for the incentive. This term signifies that the incentive received by client ci is proportional to its resource investment xi relative to the total investment of all participating clients, and is weighted by its current trust score τit. This weighting ensures that clients with higher trust receive a larger share of the incentive pool for the same level of resource investment, thus encouraging trustworthy behavior.

- The second term, $(aixi2+bixi)$, captures the cost of resource consumption. This quadratic cost function is widely adopted in economics and resource allocation models [46]. It ensures convexity, which facilitates mathematical analysis, and realistically reflects diminishing returns—meaning the cost per unit of resource consumption increases as more resources are invested. This accurately models the non-linear relationship between energy expenditure, training time, and performance on resource-constrained edge devices.

A critical aspect of this formulation is that each client's best response (optimal xi) depends on the global term $\sum jxj$, which represents the total resource investment of all clients. In decentralized settings, this total sum is generally unknown to individual clients. We address this by assuming that the server provides an aggregated signal (e.g., an estimate of total resource investment from the previous round) during each communication round. This approximation is consistent with many Stackelberg-based FL mechanisms [47], where clients respond based on coarse-grained information rather than full observability of all other clients' actions. Future work could explore more sophisticated distributed best-response estimation or local belief updates to relax this assumption.

The server's objective is to design the incentive allocation $\{rit\}$ and the aggregation weights $\{\alpha it\}$ in each round t to achieve a dual goal:

1. Ensure high model quality and convergence: This

implies minimizing the global loss and achieving robust performance.

2. Promote high-trust participation and suppress malicious updates: This involves incentivizing reliable clients and effectively mitigating the impact of untrustworthy ones.

The server's utility function, US, is defined as the net benefit derived from the improved model accuracy minus the total incentive cost:

US=ΔL(wt)−λ·Rt(10)

Here, ΔL(wt) represents the reduction in global model loss (or improvement in accuracy) after the aggregation of client updates in round t. λ is a balancing coefficient that controls the server's sensitivity to the total incentive budget Rt. A higher λ means the server is more cost-averse. The server's ultimate objective is to choose the optimal total budget Rt and the individual incentive allocations {rit} such that US is maximized, while simultaneously encouraging high-trust participation from clients. By using backward induction in the Stackelberg game, the server can derive its optimal reward strategy, thereby establishing a closed-loop linkage between incentive allocation and adaptive client behavior.

4. Trust-Driven Incentive and Aggregation Mechanism

In this section, we systematically present the proposed Trust-Aware Incentive Mechanism (TAIM) and its accompanying robust aggregation algorithm. The fundamental objective of TAIM is to establish a triple control logic within the federated learning ecosystem: incentives must be guided by trust, aggregation processes must enhance robustness, and client behaviors should be driven by incentives to form a positive-feedback convergence loop. Unlike traditional methods that often decouple trust modeling, incentive mechanisms, and aggregation strategies, our design achieves a unified modeling of these three critical components. We also introduce corresponding game-theoretic solution strategies and adaptive weight adjustment mechanisms to realize this integrated control.

4.1. Trust-Aware Incentive Allocation Modeling

The design of an effective incentive mechanism is paramount for motivating clients to participate diligently and ensuring the quality of their behavioral contributions in federated learning systems. Building upon our dynamic trust modeling, we utilize the trust score τit and a refined measure of the client's contribution, vit, as the primary factors in reward allocation. This approach aims to circumvent the manipulation that can arise from using static or easily falsifiable metrics, such as declared data volume or reported training epochs.

First, we define the raw contribution vit of client ci in round t as the normalized L2 norm of its uploaded model update Δwit:

vit=∑j∈St||Δwjt||2||Δwit||2(5)

This metric reflects the magnitude of the change a client's update introduces to the global model space. A larger norm might indicate a more significant update. However, relying solely on this raw contribution can be problematic, as malicious clients might inflate their update norms (e.g., by adding noise) to appear more active, even if their updates are detrimental.

To counteract such potential manipulations and ensure that incentives are tied to genuinely beneficial contributions, we introduce a validation-based actual gain function, git. This function quantifies the true positive impact of a client's update on the global model's performance. It is calculated as the reduction in the global model's loss (or improvement in accuracy) on a small, representative public validation dataset if only client ci's update were hypothetically applied:

git=L(wt)−L(wt+Δwit)(7)

Here, L(wt) is the loss of the global model wt on the validation set before aggregation, and L(wt+Δwit) is the loss after hypothetically applying only client ci's update. A higher git indicates a more effective contribution. This validation-based approach ensures that clients are rewarded for updates that actually improve the model's generalization, rather than just for their size.

Using this actual gain, the corrected contribution v^it is then computed by scaling the raw contribution vit by the relative actual gain:

v^it=vit·L(wt)git

This corrected contribution v^it serves as a more reliable indicator of a client's value and is used in both the incentive allocation and aggregation processes.

Finally, the incentive reward function rit for client ci in round t is defined as follows:

rit=∑j∈Stτjt·v^jtRt·τit·v^it(6)

This function ensures incentive compatibility and prioritizes clients that demonstrate both high trust and high corrected contribution under the total budget constraint Rt. Clients with higher trust scores and more effective updates receive a proportionally larger share of the available incentive budget. This mechanism directly links rewards to verifiable, beneficial behavior, discouraging free-riding and malicious activities that do not result in actual model improvement.

4.2. Stackelberg Game-Based Solution Strategy

To formally model the strategic interaction between the central server and the participating clients, we adopt a Stackelberg game formulation. In this hierarchical game, the server acts as the leader, making its decisions (total budget Rt and reward strategy {rit}) first. The clients, as

followers, then observe the server's decisions and choose their optimal resource investment xi to maximize their individual utility. This leader-follower dynamic is a common and effective way to model such interactions in decentralized systems [47].

Each client ci's utility function, as introduced in Equation (8), is given by:

$$Ui(xi)=\eta\cdot\tau it\cdot\sum j\in Stxjxi-(aixi2+bixi)(8)$$

To find the client's best-response function, each client aims to maximize its utility Ui(xi) by choosing its optimal resource investment xi. Since the utility function is concave with respect to xi (due to the quadratic cost term), we can find the optimal xi∗ by taking the first-order derivative of Ui(xi) with respect to xi and setting it to zero:

$$\partial xi\partial Ui(xi)=\eta\cdot\tau it\cdot((\sum j\in Stxj)2\sum j\in Stxj-xi)-(2aixi+bi)=0$$

Solving this equation for xi (assuming $\sum j\in Stxj$ is treated as a constant by client i for its local optimization, as is typical in Stackelberg follower problems), we obtain the closed-form best-response function for client ci:

$$xi*=2ai\cdot\sum j\in Stxj\eta\cdot\tau it-bi\cdot\sum j\in Stxj(9)$$

This equation reveals that a client's optimal resource investment xi∗ is directly proportional to its trust score τit and inversely related to its cost parameters (ai,bi) and the total resource investment of other clients. This implies that clients with higher trust scores or lower resource costs will be incentivized to contribute more.

For the server, its utility function is defined as the net benefit of model improvement minus the incentive cost:

$$US=\Delta L(wt)-\lambda\cdot Rt(10)$$

where ΔL(wt) is the loss reduction after aggregation (reflecting model quality improvement), and λ is the server's cost sensitivity. The server's objective is to choose the optimal total budget Rt and the reward distribution {rit} that maximize US, while simultaneously encouraging high-trust participation from clients.

To solve this Stackelberg game, the server employs backward induction. First, it determines the clients' best responses (as derived above). Then, it substitutes these best responses into its own utility function. This allows the server to anticipate how clients will react to its incentive strategy. The server then optimizes its own decisions (budget and reward allocation) to maximize its utility, knowing the clients' rational responses. This establishes a closed-loop linkage between the server's incentive allocation and the clients' adaptive behavior, driving the system towards an equilibrium where both server and clients optimize their objectives. While solving this non-linear optimization problem can be complex, iterative algorithms or approximations can be employed for practical deployment.

4.3. Trust-Guided Soft Aggregation Mechanism

Traditional robust aggregation methods, such as Krum or Trimmed Mean, often employ rigid techniques like outlier removal or hard thresholds. While effective against blatant attacks, these methods can inadvertently harm model diversity and inclusiveness by discarding updates from benign clients with highly non-IID data distributions or those experiencing temporary network fluctuations. Such rigid filtering might also excessively penalize edge clients with legitimate behavioral fluctuations, hindering long-term trust evolution.

To address these limitations, we propose a trust-guided non-linear soft suppression strategy. This approach attenuates the impact of low-trust updates using a continuous weighting function, rather than completely discarding them. This allows for a more nuanced control over the influence of each client's update.

We define a sigmoid-based suppression function σ(τ) as follows:

$$\sigma(\tau)=1+e-k(\tau-\mu)1(11)$$

where:

- τ is the client's trust score.
- k controls the steepness of the sigmoid curve. A larger k results in a steeper curve, leading to a more aggressive suppression of updates from clients with trust scores below the threshold.
- μ controls the suppression threshold. Clients with trust scores significantly below μ will have their updates heavily suppressed, while those above μ will have their updates weighted more favorably.

This sigmoid function ensures that the suppression is continuous and smooth. Updates from highly trusted clients ($\tau\gg\mu$) receive a weight close to 1, while updates from very low-trust clients ($\tau\ll\mu$) receive a weight close to 0. Clients with trust scores around μ experience a gradual suppression, allowing for potential re-evaluation and recovery of their influence if their trust score improves in subsequent rounds.

The final aggregation weight αit for client ci's update in round t is then determined by combining its trust score (via the sigmoid suppression function) and its corrected contribution v^it:

$$\alpha it=\sum j\in St\sigma(\tau jt)\cdot v\hat{}jt\sigma(\tau it)\cdot v\hat{}it(12)$$

This formula ensures that the aggregation weight is not only proportional to a client's effective contribution but also modulated by its trustworthiness. High-trust, high-contribution clients receive larger weights, while low-trust clients have their influence reduced. The sum of all aggregation weights αit for i∈St equals 1.

Finally, the global model update for the next round wt+1 is computed by applying these trust-guided soft aggregation weights to the local model updates:

$$wt+1=wt+i\in St\sum\alpha it\cdot\Delta wit(13)$$

This aggregation scheme effectively suppresses the influence of malicious or unreliable updates while still allowing low-trust clients to be re-evaluated and potentially regain weight if their behavior improves. This enhances long-term fairness, maintains client diversity, and promotes robust convergence of the global model.

To ensure practical deployability and minimize computational overhead, the sigmoid-based suppression function can be implemented using precomputed lookup tables or efficient approximate activation functions, avoiding expensive real-time exponential calculations. Similarly, trust score updates, being server-side vector operations, introduce minimal computational cost compared to the overall model training and communication. These design choices ensure that the trust-aware mechanism does not introduce significant delays compared to standard aggregation methods, making TAIM practical for large-scale deployments.

4.4. Robustness Enhancement and Anomaly Detection Mechanisms

While the dynamic trust score and soft aggregation provide a strong foundation for robustness, sophisticated adversaries might attempt to mimic trustworthy patterns or frequently switch their strategies to evade detection and manipulate the system. To counter such advanced attacks and further enhance the behavioral sensitivity and anomaly adaptability of our trust model, we introduce two additional robustness enhancement modules, forming a layered defense framework.

1. The Deviation Penalty Mechanism:

This mechanism is designed to immediately penalize clients whose updates significantly deviate from the expected global trend, which is a common characteristic of poisoning attacks. We define the relative deviation ζ_i^t of client c_i's update Δw_i^t from the average global update direction Δw^t (or the global model's gradient) as:

$$\zeta_i^t = ||\Delta w^t||_2 ||\Delta w_i^t - \Delta w^t||_2 \quad (14)$$

This metric quantifies how far a client's update is from the collective movement of the global model. If this relative deviation ζ_i^t exceeds a predefined threshold ϵ, it indicates a potentially anomalous or malicious update. In such cases, the client's trust score is immediately penalized using an exponential decay factor:

$$\tau_i^t \leftarrow \tau_i^t \cdot \exp(-\beta \cdot \zeta_i^t) \quad (15)$$

Here, β is a positive penalty coefficient. This exponential penalty ensures that larger deviations lead to a more severe and immediate reduction in the trust score. This mechanism acts as a rapid response system, quickly reducing the influence of potentially harmful updates and discouraging clients from submitting drastically perturbed gradients.

2. Sliding Window-Based Trust Correction:

This module addresses the challenge of strategic clients who might exhibit drastic, non-monotonic variations in their behavior (e.g., alternating between honest and malicious, or suddenly improving their trust to gain rewards). A sliding window mechanism is employed to track the historical fluctuations of each client's trust scores over a longer period.

If a client's trust score exhibits sudden, significant, and non-monotonic increases (e.g., rapidly jumping from very low to very high trust without a consistent history of good behavior), we introduce a mechanism to slow down the growth of its aggregation weight. This prevents short-term strategic speculation from immediately receiving high incentives and disproportionate influence. For instance, instead of directly using the newly updated τ_i^t for aggregation weight calculation, a smoothed version or a lower bound derived from its historical window might be used if suspicious patterns are detected. This adds a layer of memory and cautiousness to the system, making it harder for "on-off" attackers or those attempting to "mimic" good behavior for a short period to gain undue influence. This mechanism improves the behavioral sensitivity and anomaly adaptability of the trust model, contributing to a robust and multi-layered defense framework for the FL system.

4.5. Integrated Federated Training Procedure

By seamlessly combining the dynamic trust modeling, the game-theoretic incentive allocation, and the trust-guided soft aggregation mechanism, the overall TAIM training process forms a comprehensive and adaptive federated learning framework. The integrated procedure is summarized in Algorithm 1 (from the provided PDF):

Algorithm 1 TAIM: Trust-Aware Incentive and Robust Aggregation Algorithm

Require: Initial global model w^0, total training rounds T, initialize client trust scores $\tau_i^0 = 0.5$ for all $i \in C$.

1: for each round $t = 1$ to T do

2: Server selects client set S_t (e.g., randomly or based on previous trust scores) and broadcasts current global model w^t.

3: for each client $c_i \in S_t$ do in parallel

4: Client c_i downloads w^t.

5: Client c_i trains locally on its dataset D_i to obtain local model update Δw_i^t.

6: Client c_i uploads Δw_i^t to the server.

7: end for

8: Server receives updates $\{\Delta w_i^t\}_{i \in S_t}$ from selected clients.

9: Server computes raw contribution v_i^t for each client $c_i \in S_t$ using Equation (5).

10: Server computes contribution effectiveness g_i^t for each client $c_i \in S_t$ using Equation (7).

11: Server computes corrected contribution v^it for each client ci∈St.

12: Server updates trust score τit for each client ci∈St using Equations (1) and (2), incorporating ϕit,ψit,ωit.

13: Server applies Deviation Penalty Mechanism (Equation (15)) and Sliding Window-Based Trust Correction to τit if anomalies are detected.

14: Server computes incentive reward rit for each client ci∈St using Equation (6).

15: Server computes aggregation weight αit for each client ci∈St using Equation (12).

16: Server aggregates local updates: wt+1=wt+∑i∈Stαit·Δwit (Equation (13)).

17: Server broadcasts incentives {rit} to participating clients (and potentially updated trust scores for transparency).

18: end for

19: return wT (the final global model).

This integrated training process maintains the fundamental deployability of the standard FedAvg framework while constructing a complete and dynamic trust-incentive-aggregation feedback loop. This loop ensures that client behavior directly influences their trust, which in turn dictates their incentives and their impact on the global model. This adaptive system offers enhanced security and strategy adaptiveness, making it particularly suitable for heterogeneous, dynamic, and untrusted open edge environments.

Complexity Analysis:

The computational overhead introduced by TAIM is manageable and does not fundamentally alter the overall complexity of the federated learning process. Let |St| be the number of selected clients per round and d be the dimensionality of the model parameters (i.e., the number of weights in the model). The primary computations introduced by TAIM, executed on the server-side, include the following:

1. Trust Score Update (Lines 9-13):

○ Raw Contribution (vit): Calculating the L2 norm and normalization for each client requires O(d) operations per client, totaling O(|St|·d).

○ Contribution Effectiveness (git): This involves running a forward pass of the model on a small public validation set. If the validation set size is Dval and the model inference complexity is O(d), then this is O(Dval·d) per client, totaling O(|St|·Dval·d). However, typically Dval is very small, or a proxy is used, making this overhead minimal.

○ Gradient Consistency (ψit): Calculating the cosine similarity between a client's update and the global average update requires a dot product and two norm calculations, which is O(d) operations per client. Summing over all clients, this is O(|St|·d).

○ Exponential Decay and Weighting: These are constant time operations per client, O(|St|).

○ Deviation Penalty: Similar to gradient consistency, involves norm calculations, O(|St|·d).

○ Sliding Window Correction: Primarily involves array manipulations, O(|St|) or O(|St|·Twindow).

2. Incentive Allocation (Line 14): This step involves summations and divisions over |St| clients, resulting in an overhead of O(|St|).

3. Soft Aggregation (Line 15): Computing the aggregation weights using the sigmoid function and normalization also requires O(|St|) operations. The sigmoid function itself can be implemented efficiently using precomputed lookup tables or fast approximations, adding negligible overhead.

4. Global Model Aggregation (Line 16): The final weighted aggregation of the model updates remains O(|St|·d), which is the dominant operation in standard FedAvg as well.

Therefore, the total computational complexity per round for TAIM remains dominated by the model-related vector operations, which is O(|St|·d). The additional trust and incentive calculations introduce a constant factor increase in server-side computation but do not scale with model dimensionality or client count in a prohibitive way. This makes TAIM practical for large-scale deployments, as further supported by the empirical overhead analysis in Section 5.5. The client-side overhead for generating Δwit is unchanged from standard FL.

5. EXPERIMENTAL EVALUATION

To rigorously validate the effectiveness and robustness of the proposed Trust-Aware Incentive Mechanism (TAIM) in realistic federated learning environments, this section presents a comprehensive empirical study. The experiments are conducted across multiple representative datasets, various attack types, and against several established baseline methods. The evaluation primarily focuses on answering the following key research questions:

1. Accuracy and Convergence: Can TAIM significantly improve the global model's accuracy and accelerate its convergence efficiency, especially under conditions of heterogeneous client participation?

2. Robustness: Is TAIM more robust against a diverse range of attack types, including sophisticated adaptive adversaries, and is it capable of accurately identifying malicious client behaviors?

3. Fairness and Overhead: Does TAIM ensure a fairer distribution of incentives among clients, and is its computational and communication overhead acceptable under realistic edge computing constraints?

To ensure the reproducibility and credibility of our findings, we provide detailed descriptions of the experimental setup, attack modeling strategies, evaluation metrics, and the baseline methods used for comparison. The subsequent subsections present a thorough analysis of the experimental results.

5.1. Experimental Setup and Datasets

We select three distinct and representative federated learning tasks to validate TAIM's effectiveness across diverse data modalities and types of heterogeneity that mirror real-world challenges. This selection ensures a comprehensive assessment of the framework's generalizability.

1. FEMNIST (Federated Extended MNIST): This dataset is derived from the LEAF benchmark suite and involves handwritten character recognition (digits and English letters). It is characterized by a highly non-IID user-based split, where each client corresponds to a single writer. This inherent partitioning naturally models the significant statistical heterogeneity (label and feature skew) found in real-world user-generated data, where individuals have distinct writing styles and character frequencies.

2. CIFAR-10: A classical image classification dataset consisting of 60,000 32x32 color images in 10 classes. To simulate non-IID conditions, we generate partitions using a Dirichlet distribution with parameter α=0.3. A smaller α value leads to a higher degree of non-IIDness, meaning clients will have more imbalanced class distributions. This systematic partitioning allows us to precisely control and evaluate the impact of statistical heterogeneity on model performance.

3. Sent140: A large-scale Twitter sentiment analysis task, comprising 1.6 million tweets. Each client in this dataset reflects individual language styles, vocabulary choices, and sentiment expression patterns. This makes Sent140 an ideal testbed for modeling and evaluating the behavioral heterogeneity that TAIM is specifically designed to manage, as client contributions can vary significantly in quality and consistency.

For all experiments, the datasets are consistently split into training, validation, and test sets with a ratio of 80:10:10, respectively, unless otherwise specified. This standardized split ensures fair and consistent evaluation across all baseline and proposed methods.

Model Configuration and Training Details:

To ensure a fair comparison and isolate the effects of TAIM, we adopt standard neural network architectures commonly used for these tasks:

- For CIFAR-10, we employ a Convolutional Neural Network (CNN) consisting of two convolutional layers (e.g., 32 filters, 64 filters, each followed by ReLU activation and max-pooling), followed by two fully connected layers.
- For FEMNIST, a two-layer CNN architecture is used, similar to the CIFAR-10 model but adapted for the grayscale and character-specific features.
- For Sent140, a Long Short-Term Memory (LSTM) network is utilized, which is well-suited for sequential text data. The LSTM model typically includes an embedding layer, one or more LSTM layers, and a final dense layer for classification.

The training process for all methods adheres to the following parameters:

- In each communication round, 10% of the total clients are randomly selected to participate. This simulates a realistic FL scenario where only a subset of devices is active at any given time.
- Local training on each selected client runs for five epochs using the Stochastic Gradient Descent (SGD) optimizer.
- The learning rate for SGD is set to 0.01, with a momentum of 0.9 to accelerate convergence.
- The trust parameters for TAIM are carefully tuned based on preliminary experiments and set as follows: memory decay coefficient γ=0.8, and instantaneous trust score weights $\lambda1$=0.3 (for participation frequency), $\lambda2$=0.4 (for gradient consistency), and $\lambda3$=0.3 (for contribution effectiveness). This configuration emphasizes gradient consistency slightly more, reflecting its direct link to model quality.
- The sigmoid suppression function parameters for soft aggregation are set to steepness k=10 and threshold μ=0.5. These values ensure a balance between aggressive suppression of very low-trust updates and a smooth transition for moderately trusted clients.

5.2. Attack Modeling and Client Behavior Settings

To simulate realistic and challenging threats in federated environments, we inject a varying proportion of malicious clients per round and implement four distinct types of adversarial behaviors. The client composition in our simulations is designed to reflect a diverse real-world scenario:

- Malicious Clients: Between 10% and 30% of the total client pool are designated as malicious. The specific attack types are evenly distributed among these malicious clients.
- Resource-Constrained Clients: 20% of the clients are modeled as resource-constrained. These clients have reduced upload frequency (e.g., they only participate and upload updates every 2 or 3 rounds, simulating intermittent connectivity or energy-saving modes).
- Benign Clients: The remaining clients are considered benign and contribute honestly to the FL process.

The four types of adversarial behaviors implemented are:

1. Label Flip Attack: This is a common data poisoning attack where a portion of labels in the malicious client's local dataset are flipped to incorrect classes. For instance, in CIFAR-10, a malicious client might flip 20% of images labeled "cat" to "dog." This attack aims to mislead the global model's convergence by introducing erroneous gradients.

2. Gaussian Noise Attack: Malicious clients add Gaussian noise (with a mean of 0 and a standard deviation of 5, for example) directly to their computed gradients before uploading them. This attack aims to degrade the global model's performance by injecting random, high-variance updates, hindering stable convergence.

3. On-Off Attack: This is a behavioral attack designed to evade trust accumulation mechanisms. Malicious clients alternate between honest behavior (uploading benign updates) and malicious behavior (e.g., performing a label flip or Gaussian noise attack). For example, a client might behave honestly for 5 rounds, then maliciously for 2 rounds, and then honestly again. This makes it difficult for trust models with short memory spans to consistently identify them as malicious.

4. Mimic Attack: This is a sophisticated attack where malicious clients attempt to imitate the gradients of high-trust benign clients to evade detection. They might calculate their malicious gradient, then scale or shift it to resemble the average or a specific benign client's gradient, while still subtly disturbing the aggregation. This attack specifically targets trust mechanisms that rely heavily on gradient similarity for detection.

By incorporating these diverse attack types and client behaviors, our experimental setup rigorously tests TAIM's ability to maintain performance and security in challenging, heterogeneous FL environments.

5.3. Evaluation Metrics

To provide a comprehensive assessment of TAIM's performance, we evaluate all methods from four critical perspectives: accuracy, robustness, fairness, and detection ability. We also consider system-level overhead.

1. Final Accuracy (Acc): This refers to the test accuracy achieved by the global model on the unseen test set after the federated learning process has converged. It is the primary indicator of the model's overall predictive performance.

2. Robustness Drop (RD): This metric quantifies the performance degradation caused by the presence of adversarial clients. It is calculated as the percentage drop in accuracy compared to a baseline scenario where no attacks are present (i.e., only benign clients).

RD=AccuracybenignAccuracybenign−Accuracyattack×100%

A lower Robustness Drop indicates a more resilient and robust FL system.

3. Fairness (Gini Coefficient): The Gini coefficient is a widely used measure of statistical dispersion intended to represent the income or wealth distribution within a nation or any other group. In our context, it quantifies the inequality in the cumulative reward distribution among clients.

G=2n∑i=1nri∑i=1n∑j=1n|ri−rj|(16)

where ri is the cumulative reward received by client i over all training rounds, and n is the total number of clients.

- A Gini coefficient of 0 indicates perfect equality (all clients receive the same reward).
- A Gini coefficient of 1 (or 100%) indicates maximum inequality (one client receives all the reward).

A lower Gini value signifies a more balanced and equitable incentive distribution, which is crucial for fostering long-term client participation and preventing resentment among contributors.

4. Detection Ability: This assesses how effectively the system identifies malicious clients. We use two standard classification metrics:

- Recall (%): Also known as sensitivity or true positive rate. It measures the proportion of actual malicious clients that were correctly identified as malicious.

Recall=TruePositives+FalseNegativesTruePositives

where True Positives (TP) are malicious clients correctly identified, and False Negatives (FN) are malicious clients incorrectly classified as benign.

- False-Positive Rate (FPR) (%): It measures the proportion of benign clients that were incorrectly identified as malicious.

FPR=FalsePositives+TrueNegativesFalsePositives

where False Positives (FP) are benign clients incorrectly classified as malicious, and True Negatives (TN) are benign clients correctly classified as benign.

High recall is desirable to catch most attackers, while low FPR is crucial to avoid penalizing honest clients.

5. System-level Overhead: We report the total Training Time (s) and Communication Volume (MB) to assess the practical feasibility and scalability of the proposed method.

- Local Training Time (s): Average time spent by clients on local model training per round.
- Communication Volume (MB): Total data transferred between clients and the server (upload and download) over all rounds.
- Server Aggregation Time (s): Time taken by the server to perform aggregation and trust calculations per

round.

These comprehensive metrics enable a thorough assessment of whether TAIM's trust-guided reward allocation and robust aggregation contribute to reducing reward centralization, enhancing security, and maintaining fairness among heterogeneous clients without imposing excessive computational or communication burdens.

5.4. Baseline Methods

To provide a robust comparative analysis, we evaluate TAIM against several mainstream federated learning strategies that represent different approaches to handling heterogeneity and robustness:

1. FedAvg (Federated Averaging) [48]: This is the foundational and most widely used FL algorithm. It performs a simple weighted average of client model updates, where weights are typically proportional to the size of client datasets. FedAvg serves as a crucial baseline to demonstrate the performance improvements achieved by more advanced mechanisms, especially under non-IID data and adversarial conditions. It does not inherently handle heterogeneity or malicious clients.

2. FedProx [49]: This method extends FedAvg by introducing a proximal term to the local objective function during client training. This term penalizes local model updates that deviate significantly from the global model, aiming to mitigate the effects of statistical heterogeneity (non-IID data) and improve global convergence. While it addresses data heterogeneity, it does not explicitly account for behavioral heterogeneity or malicious attacks.

3. FedTrust [50]: This algorithm incorporates a basic trust-weighted aggregation mechanism. It calculates trust scores based on the similarity of uploaded models (e.g., cosine similarity of gradients) and adjusts aggregation weights accordingly. It is designed to give more influence to seemingly trustworthy clients and less to outliers. However, its trust model is often simpler and may not be robust against sophisticated, adaptive attacks that mimic benign behavior.

4. Krum [51]: This is a well-known robust aggregation algorithm designed to defend against Byzantine attacks (including model poisoning). Krum selects a subset of client updates that are "closest" to each other in the parameter space, effectively discarding outlier updates that are likely to be malicious. While effective against certain types of attacks, Krum can be computationally intensive due to distance calculations and may be overly aggressive in filtering, potentially discarding valuable updates from benign clients with highly non-IID data. It also does not incorporate an incentive mechanism.

These baselines collectively serve as the foundation for our comprehensive comparison and analysis, allowing us to highlight TAIM's advantages in integrating trust, incentives, and robustness within a unified framework.

5.5. Overall Performance Comparison

Our extensive experimental results clearly demonstrate the superior performance of TAIM across various metrics, particularly under challenging conditions of high heterogeneity and adversarial attacks.

1. Final Accuracy and Convergence (Figure 5):

Figure 5 illustrates the final test accuracy achieved by each method on FEMNIST and CIFAR-10 datasets under both 10% and 30% attacker ratios.

- As expected, all methods experience a significant drop in accuracy when the proportion of attackers increases from 10% to 30%, highlighting the severe impact of adversarial clients.

- TAIM consistently achieves the best performance across all scenarios. On FEMNIST, TAIM reaches 79.5% accuracy with 10% attackers and a robust 76.1% with 30% attackers. On CIFAR-10, it achieves 75.2% with 10% attackers and 71.9% with 30% attackers.

- Compared to FedAvg and FedProx, which show significant degradation under attacks, TAIM demonstrates up to 9.2% higher accuracy under heavy attack conditions (e.g., FEMNIST with 30% attackers, FedAvg at 65.5% vs. TAIM at 76.1%). This substantial improvement underscores TAIM's effectiveness in suppressing adversarial disturbances through its integrated trust modeling and incentive mechanisms.

- Even against robust baselines like FedTrust and Krum, TAIM maintains a noticeable edge (e.g., 79.5% vs. Krum's 77.9% on FEMNIST with 10% attackers). This suggests that TAIM's multi-dimensional trust assessment and soft aggregation are more effective at discerning and mitigating the impact of sophisticated adversaries compared to methods relying on simpler trust models or rigid filtering.

2. Model Robustness Under Adaptive Attacks (Figure 6):

Figure 6 quantifies the model robustness, measured as performance drop (lower is better), under various adaptive attack types on CIFAR-10.

- Traditional methods like FedAvg and FedProx exhibit severe performance fluctuations and significant accuracy drops across all attack types, confirming their vulnerability.

- Even robust methods like FedTrust and Krum are significantly affected by Mimic attacks (22.5% drop for FedAvg, 19.8% for FedProx, 14.9% for FedTrust, 12.1% for Krum). This highlights the challenge posed by adversaries that attempt to blend in with benign clients.

- In stark contrast, TAIM maintains stable accuracy with minimal performance drop across all attack types. Notably, under Mimic attacks, TAIM's performance drop is only 8.4%, outperforming Krum by a substantial 3.7%

(12.1% vs. 8.4%).

- TAIM also shows superior resilience against On-Off attacks (4.9% drop), which are designed to evade trust accumulation. This indicates that the multi-dimensional trust model in TAIM, with its memory decay and anomaly detection mechanisms, effectively filters disguised adversaries and adapts to dynamic attack strategies. The consistent low performance drop across diverse attacks confirms TAIM's enhanced robustness.

3. Malicious Client Detection Performance:

- Traditional methods like FedAvg and FedProx are not designed for explicit malicious client detection, hence their performance is denoted by "-".

- FedTrust and Krum achieve reasonable detection capabilities, with FedTrust showing 82.4% Recall and 9.6% FPR, and Krum achieving 85.7% Recall with 12.1% FPR.

- TAIM significantly outperforms these baselines with a Recall of 91.3% and a remarkably low FPR of only 5.8%. This superior detection capability is directly attributable to TAIM's comprehensive trust modeling, which integrates participation frequency, gradient consistency, and contribution effectiveness. By leveraging these multiple dimensions, TAIM can more accurately distinguish between benign clients (even those with non-IID data) and malicious ones, minimizing both missed detections and false alarms.

4. Fairness of Incentive Distribution:

Figure 7 presents the Gini coefficient during training, which reflects the fairness of incentive distribution among clients. A lower Gini coefficient indicates a more balanced and equitable reward allocation.

- FedAvg and FedProx exhibit high Gini values (0.52 and 0.48 respectively), indicating significant reward centralization and unfairness. This is expected as they do not explicitly consider contribution quality or trust, potentially rewarding free-riders or clients with large datasets disproportionately.

- FedTrust and Krum show some improvement (0.35 and 0.33), but still indicate a degree of inequality.

- TAIM consistently maintains a Gini coefficient below 0.3 throughout the training process. This remarkable fairness is a direct result of TAIM's balanced trust design, which considers both consistent participation and verifiable contribution quality. By linking incentives directly to trustworthiness and effective updates, TAIM prevents reward centralization and ensures a more equitable distribution, fostering long-term engagement from all types of heterogeneous clients.

5. System Resource Overhead Comparison (Table 3):

Table 3 compares the system resource overhead of TAIM against the baselines, focusing on local training time, communication volume, and server aggregation time.

- Despite integrating additional trust computation and soft aggregation mechanisms, TAIM's overhead remains highly comparable and acceptable.

- The communication volume for TAIM is only marginally higher (9.0 MB) compared to baselines (8.5 MB), a negligible increase given the significant performance gains. This is because TAIM does not require additional communication rounds or large data transfers.

- The server aggregation time for TAIM (0.16 s) is slightly higher than FedAvg (0.03 s) and FedProx (0.05 s), but it is comparable to Krum (0.15 s), which also involves additional computations (distance calculations).

- The local training time for TAIM (3.9 s) is similar to FedTrust and well within acceptable limits.

This analysis confirms that the computational overhead introduced by TAIM is manageable and does not introduce significant delays compared to standard aggregation, making TAIM practical for large-scale deployments. The efficiency stems from the lightweight nature of trust score updates and the optimized implementation of the soft aggregation function.

6. Accuracy Comparison under Unified Model Architecture:

To eliminate any confounding effects caused by differences in model architecture across datasets, we conducted a control experiment. For FEMNIST and CIFAR-10, we used a unified lightweight CNN (two convolutional layers and two fully connected layers, approximately 0.5 M parameters). For Sent140, we retained the LSTM-based model for all methods due to the inherent sequential nature of the data, ensuring consistency and fairness in comparison.

- presents these results, showing that TAIM consistently outperforms other methods across all three datasets even under unified or consistent model settings. For instance, on FEMNIST, TAIM achieves 78.5% compared to Krum's 75.6%; on CIFAR-10, TAIM reaches 74.9% against FedTrust's 71.0%; and on Sent140, TAIM scores 75.8% compared to FedProx's 72.3%.

- This crucial control experiment confirms that the observed performance gains are directly attributable to the effectiveness of TAIM's trust-aware incentive and aggregation mechanisms, rather than being influenced by specific model architecture advantages. This strengthens the validity and generalizability of our conclusions.

5.6. Ablation Study and Parameter Sensitivity

To understand the individual contributions of TAIM's components and evaluate its robustness to hyperparameter settings, we conducted an ablation study and parameter sensitivity analysis.

1. Ablation Study on Trust Dimensions:

- Full TAIM (with all three components) achieves the highest accuracy of 75.2%.

- Removing participation frequency (ϕ) (denoted as "w/o ϕ") drops the accuracy to 70.5%. This indicates that consistent participation is a vital indicator of client reliability and its absence significantly degrades performance.

- Removing gradient consistency (ψ) (denoted as "w/o ψ") further lowers the accuracy to 67.9%. This highlights the critical role of gradient alignment in identifying benign updates and suppressing malicious ones. Without this component, the model becomes highly susceptible to attacks that perturb gradients.

- Excluding contribution effectiveness (ω) (denoted as "w/o ω") results in an accuracy of 69.4%. This shows that directly measuring the positive impact of an update on model performance is essential for ensuring that incentives are tied to valuable contributions and for filtering out updates that might look benign but are ineffective.

This ablation study unequivocally confirms the complementary and crucial roles of all three components (ϕ, ψ, ω) in achieving accurate trust assessment and, consequently, the overall superior performance of TAIM. Each dimension captures a distinct aspect of client behavior, and their synergistic combination provides a robust and comprehensive trust score.

2. Sensitivity to Hyperparameter Configurations (Figure 8):

Figure 8 illustrates the sensitivity of the TAIM model to two key hyperparameter configurations: the sigmoid suppression steepness parameter k and the weighting coefficients of the trust components (λ1,λ2,λ3).

- Sensitivity to k Parameter (Sigmoid Steepness) - Left Subfigure:

  - The graph shows how the steepness parameter k in the sigmoid suppression function (Equation 11) affects the final model accuracy.

  - When k is set to a small value (e.g., k=2.5), the model accuracy drops to 70.8%. This is because a small k results in an overly smooth sigmoid function, which fails to effectively differentiate between high-trust and low-trust clients. The suppression is too weak to mitigate malicious updates.

  - As k increases, the accuracy gradually improves, reaching a peak of 75.2% at k=10. This suggests that a moderately enhanced steepness helps amplify trust-based differentiation in the aggregation process, allowing TAIM to effectively suppress detrimental updates without being too rigid.

  - Beyond k=10, the accuracy slightly decreases but remains relatively high (e.g., 74.6% at k=15, 73.9% at k=20). This indicates that excessively steep functions might overfit to the trust estimates or become too sensitive to minor trust fluctuations, potentially impairing generalization or discarding updates that are only slightly below the threshold but still beneficial.

  - Overall, the system demonstrates robustness to a wide range of k values around the optimal point, implying that precise tuning is not overly burdensome.

- Sensitivity to Trust Component Weights (λ1,λ2,λ3) - Right Subfigure:

  - This bar chart examines the influence of different trust component weight configurations on accuracy.

  - With an equal weight setting (Equal: λ1=0.33,λ2=0.33,λ3=0.33), the model achieves 73.8% accuracy. This confirms that each trust dimension independently contributes to performance, and even a simple equal weighting provides reasonable results.

  - However, when one dimension is overly emphasized, the accuracy can significantly drop. For instance, emphasizing participation frequency (Freq. Heavy: λ1=0.6,λ2=0.2,λ3=0.2) leads to a lower accuracy of 72.1%. This suggests that while participation is important, over-relying on it without considering quality or consistency can be detrimental.

  - In contrast, emphasizing gradient consistency (Consist. Heavy: λ1=0.2,λ2=0.6,λ3=0.2) leads to a better result of 74.3%. This highlights the crucial importance of gradient-level behaviors in robust modeling, as it directly reflects the alignment of updates with the global objective.

  - Similarly, emphasizing contribution effectiveness (Contrib. Heavy: λ1=0.2,λ2=0.2,λ3=0.6) yields 73.6%.

  - TAIM's default setting (λ1=0.3,λ2=0.4,λ3=0.3) achieves the highest accuracy of 75.2%. This empirically confirms the effectiveness of our proposed joint modeling strategy in balancing the influence of different trust dimensions, leading to optimal performance.

These sensitivity analyses demonstrate that TAIM's performance is robust across a reasonable range of hyperparameters, and the chosen default settings effectively leverage the complementary nature of the trust components.

6. Conclusions and Future Work

6.1. Conclusions

The increasing adoption of federated learning in edge computing environments, while promising for privacy-preserving collaborative AI, introduces significant complexities stemming from client heterogeneity, incentive imbalances, and the pervasive threat of adversarial attacks. This paper has addressed these critical challenges by proposing a novel and unified framework called the Trust-Aware Incentive Mechanism (TAIM).

TAIM's core innovation lies in its integrated approach, which jointly models dynamic multi-dimensional client

trust, leverages game theory for incentive allocation, and employs a trust-guided soft aggregation algorithm. Specifically, our framework meticulously evaluates client reliability by incorporating three key behavioral indicators: participation frequency, gradient consistency, and contribution effectiveness. This comprehensive trust assessment provides a nuanced understanding of each client's value and potential risk. Building upon these dynamic trust scores, we formulated a Stackelberg game-based incentive allocation strategy that strategically guides clients to optimize their resource investment, fostering a rational and self-sustaining participation ecosystem. Furthermore, the introduction of a confidence-aware smoothing aggregation algorithm, featuring a soft filtering function, enables TAIM to intelligently attenuate the influence of low-trust updates without resorting to rigid, diversity-harming filtering.

Our extensive experimental evaluations, conducted across diverse non-IID datasets (FEMNIST, CIFAR-10, Sent140) and various adversarial scenarios (including label flip, Gaussian noise, on-off, and mimic attacks), unequivocally demonstrate TAIM's superior performance compared to mainstream baseline methods. The results consistently show that TAIM significantly improves global model accuracy (achieving up to +9.2% higher accuracy under heavy attacks), substantially reduces performance degradation under adaptive attacks (maintaining robustness degradation within 3%), and ensures a remarkably fairer incentive distribution among clients (consistently achieving a Gini coefficient below 0.3). Moreover, TAIM exhibits high malicious client detection capabilities (over 91% recall with low false positives) while maintaining acceptable computational and communication overhead, confirming its practical deployability in resource-constrained edge environments.

In essence, TAIM achieves a crucial balance between incentive rationality, fairness, and system robustness. The proposed soft filtering strategy not only enhances system security but also preserves client diversity, enabling a dynamic long-term trust evolution and potential reputation recovery for clients whose behavior improves. This holistic framework provides a robust and equitable solution for building future-ready federated learning systems characterized by openness, dynamism, and strategic participation.

6.2. Future Work

Despite the significant advancements offered by TAIM, several promising avenues for future research remain to further enhance its capabilities and address lingering challenges:

1. Decentralized Trust Evaluation and Management: The current trust evaluation process in TAIM primarily relies on centralized server control. While efficient, this poses potential risks of data linkage leakage (if the server is compromised) and represents a single point of failure. Future research should explore decentralized technologies to enhance privacy and system resilience.

   - Blockchain Integration: Utilizing blockchain as a distributed, immutable ledger for storing and managing client trust scores could eliminate the need for a centralized trust authority, enhancing transparency and auditability. This would require designing efficient consensus mechanisms for trust updates.

   - Secure Multi-Party Computation (SMC): Investigating SMC techniques for privacy-preserving trust score computation would allow multiple parties (e.g., other trusted clients or intermediate edge servers) to jointly compute a client's trust score without revealing their private data or individual contributions.

   - Federated Trust Learning: Developing a federated approach to learning the trust model itself, where clients collaboratively train a trust prediction model without sharing raw behavioral data, could further decentralize the trust mechanism.

2. Sophisticated Client Response Modeling and Game Learning: Our current Stackelberg game formulation simplifies client responses, assuming fully rational and immediate reactions. This might not fully capture the complexities of real-world client dynamics, which can involve constrained strategy spaces, delayed behavioral adaptations, or even irrational decisions.

   - Evolutionary Game Theory: Incorporating evolutionary game theory could model how client strategies evolve over time based on past rewards and observations of other clients' behaviors, leading to more realistic and adaptive incentive designs.

   - Reinforcement Learning for Client Behavior: Using Q-learning or other reinforcement learning techniques within the Stackelberg framework could allow the server to learn optimal incentive policies by observing client responses, even without a predefined utility function for clients.

   - Bounded Rationality: Exploring models that account for clients' bounded rationality, where they make decisions based on simplified heuristics rather than perfect optimization, could lead to more practical incentive mechanisms.

3. Multi-Modal and Multi-Task Federated Learning: This work primarily focuses on single-task, unimodal scenarios. The applicability and effectiveness of TAIM in more complex settings remain largely unexplored.

   - Cross-Modal Trust: Investigating how to define and measure trust when clients contribute data from different modalities (e.g., joint modeling of vision and language, or audio and sensor data). This would require developing modality-specific trust indicators and aggregation strategies.

   - Multi-Task FL: Extending TAIM to scenarios where clients participate in multiple federated learning tasks

simultaneously. How does trust in one task influence participation or incentives in another? This could involve transfer learning for trust or shared trust representations.

4. Integration with Differential Privacy (DP): The current trust mechanism is not explicitly integrated with differential privacy (DP), which is a strong privacy-preserving technique that adds noise to data or gradients. This raises potential concerns about privacy leakage during trust computation, or conversely, how DP's noise might impact the accuracy of trust assessment.

○ Privacy-Preserving Trust Metrics: Researching methods to compute trust indicators (e.g., gradient consistency, contribution effectiveness) under DP budget constraints. This would involve understanding the trade-offs between the level of privacy guaranteed and the accuracy of the trust score.

○ Joint Optimization: Developing frameworks that jointly optimize for model utility, trust, and differential privacy, considering how each component influences the others.

5. Real-world Deployment and Long-term Evaluation: While our simulations provide strong evidence, deploying TAIM in real-world edge environments and conducting long-term evaluations would be crucial.

○ Resource Management: Addressing practical challenges related to dynamic resource management, energy consumption, and network fluctuations in live edge deployments.

○ User Interface and Transparency: Designing user interfaces that clearly communicate trust scores and incentive mechanisms to clients, fostering greater transparency and encouraging honest participation.

○ Scalability to Massive Clients: Further optimizing the server-side computations and communication protocols to handle millions of clients, as envisioned in large-scale IoT deployments.

In summary, TAIM provides an effective trust-driven solution for building future-ready federated learning systems characterized by openness, dynamism, and strategic participation. Ongoing efforts will focus on enhancing the generality, security, and distributed capability of the mechanism to enable wide deployment of trustworthy federated intelligence across an even broader spectrum of applications.

REFERENCES

1. Aminifar, A.; Shokri, M.; Aminifar, A. Privacy-preserving edge federated learning for intelligent mobile-health systems. Future Gener. Comput. Syst. 2024, 161, 625–637.
2. Lazaros, K.; Koumadorakis, D.E.; Vrahatis, A.G.; Kotsiantis, S. Federated Learning: Navigating the Landscape of Collaborative Intelligence. Electronics 2024, 13, 4744.
3. Ivanovic, M. Influence of Federated Learning on Contemporary Research and Applications. In Proceedings of the 2024 International Conference on INnovations in Intelligent SysTems and Applications (INISTA), Craiova, Romania, 4–6 September 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 1–6.
4. Iyer, V.N. A review on different techniques used to combat the non-IID and heterogeneous nature of data in FL. arXiv 2024, arXiv:2401.00809.
5. Hartmann, M.; Danoy, G.; Bouvry, P. FedPref: Federated Learning Across Heterogeneous Multi-objective Preferences. ACM Trans. Model. Perform. Eval. Comput. Syst. 2024, 10, 1–40.
6. Chen, Y. Advancing Federated Learning by Addressing Data and System Heterogeneity. In Proceedings of the AAAI Symposium Series, Stanford, CA, USA, 25–27 March 2024; Volume 3, p. 294.
7. Liu, C.; Alghazzawi, D.M.; Cheng, L.; Liu, G.; Wang, C.; Zeng, C.; Yang, Y. Disentangling Client Contributions: Improving Federated Learning Accuracy in the Presence of Heterogeneous Data. In Proceedings of the 2023 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), Wuhan, China, 21–24 December 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 381–387.
8. Taghiyarrenani, Z.; Alabdallah, A.; Nowaczyk, S.; Pashami, S. Heterogeneous Federated Learning via Personalized Generative Networks. arXiv 2023, arXiv:2308.13265.
9. Ma, C.; Li, J.; Ding, M.; Wei, K.; Chen, W.; Poor, H.V. Federated learning with unreliable clients: Performance analysis and mechanism design. IEEE Internet Things J. 2021, 8, 17308–17319.
10. Xia, G.; Chen, J.; Huang, X.; Yu, C.; Zhang, Z. FL-PTD: A Privacy Preserving Defense Strategy Against Poisoning Attacks in Federated Learning. In Proceedings of the 2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC), Torino, Italy, 26–30 June 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 735–740.
11. Manzoor, H.U.; Shabbir, A.; Chen, A.; Flynn, D.; Zoha, A. A survey of security strategies in federated learning: Defending models, data, and privacy. Future Internet 2024, 16, 374.
12. Zhang, H.; Elsayed, M.; Bavand, M.; Gaigalas, R.; Ozcan, Y.; Erol-Kantarci, M. Federated learning with dual attention for robust modulation

classification under attacks. In Proceedings of the ICC 2024-IEEE International Conference on Communications, Denver, CO, USA, 9–13 June 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 5238–5243.

13. Wang, T.; Zheng, Z.; Lin, F. Federated learning framework based on trimmed mean aggregation rules. Expert Syst. Appl. 2025, 270, 126354.

14. Nabavirazavi, S.; Taheri, R.; Shojafar, M.; Iyengar, S.S. Impact of aggregation function randomization against model poisoning in federated learning. In Proceedings of the 22nd IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2023, Exeter, UK, 1–3 November 2023; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2024; pp. 165–172.

15. Xiong, A.; Chen, Y.; Chen, H.; Chen, J.; Yang, S.; Huang, J.; Li, Z.; Guo, S. A truthful and reliable incentive mechanism for federated learning based on reputation mechanism and reverse auction. Electronics 2023, 12, 517.

16. Han, K.; Zhang, G.; Yang, L.; Bai, J. Client dependability evaluation in federated learning framework. In Proceedings of the Third International Conference on Communications, Information System, and Data Science (CISDS 2024), Nanjing, China, 22–24 November 2024; SPIE: Bellingham, WA, USA, 2025; Volume 13519, pp. 71–79.

17. Chen, Z.; Zhang, H.; Li, X.; Miao, Y.; Zhang, X.; Zhang, M.; Ma, S.; Deng, R.H. FDFL: Fair and discrepancy-aware incentive mechanism for federated learning. IEEE Trans. Inf. Forensics Secur. 2024, 19, 8140–8154.

18. Wu, R.; Chen, Y.; Tan, C.; Luo, Y. MDIFL: Robust federated learning based on malicious detection and incentives. Appl. Sci. 2023, 13, 2793.

19. Yellampalli, S.S.; Chalupa, M.; Wang, J.; Song, H.J.; Zhang, X.; Yue, H.; Pan, M. Client Selection in Federated Learning: A Dynamic Matching-Based Incentive Mechanism. In Proceedings of the 2024 International Conference on Computing, Networking and Communications (ICNC), Big Island, HI, USA, 19–24 February 2024; IEEE: Piscataway, NJ, USA, 2024, pp. 989–993.

20. Han, J.; Khan, A.F.; Zawad, S.; Anwar, A.; Angel, N.B.; Zhou, Y.; Yan, F.; Butt, A.R. Tiff: Tokenized incentive for federated learning. In Proceedings of the 2022 IEEE 15th International Conference on Cloud Computing (CLOUD), Barcelona, Spain, 10–16 July 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 407–416.

21. Wenya, L.; Bo, L.; Weiwei, L.; Yuanchao, Y. Survey of incentive mechanism for federated learning. Comput. Sci. 2022, 49, 7.

22. Ling, X.; Li, R.; Ouyang, T.; Chen, X. Time is Gold: A Time-Dependent Incentive Mechanism Design for Fast Federated Learning. In Proceedings of the 2022 IEEE/CIC International Conference on Communications in China (ICCC), Foshan, China, 11–13 August 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1038–1043.

23. Zhou, Y. A Survey of Incentive Mechanisms for Federated Learning. Appl. Comput. Eng. 2024, 10, 1035–1044.

24. Zhang, W.; Wang, Q.; Zhao, H.; Xia, W.; Zhu, H. Incentivizing Quality Contributions in Federated Learning: A Stackelberg Game Approach. In Proceedings of the 2024 IEEE 99th Vehicular Technology Conference (VTC2024-Spring), Singapore, 24–27 June 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 1–5.

25. Huang, J.; Hong, C.; Chen, L.Y.; Roos, S. Is shapley value fair? improving client selection for mavericks in federated learning. arXiv 2021, arXiv:2106.10734.

26. Xia, H.; Li, X.; Pang, J.; Liu, J.; Ren, K.; Xiong, L. P-Shapley: Shapley Values on Probabilistic Classifiers. Proc. VLDB Endow. 2024, 17, 1737–1750.

27. Pang, J.; Yu, J.; Zhou, R.; Lui, J.C.S. An Incentive Auction for Heterogeneous Client Selection in Federated Learning. IEEE Trans. Mob. Comput. 2023, 22, 5733–5750.

28. Al-Saedi, A.A. Contribution prediction in federated learning via client behavior evaluation. Future Gener. Comput. Syst. 2024, 166, 107639.

29. Ma, S.; Liu, H.; Wang, N.; Xie, H.; Huang, L.; Li, H. Deep reinforcement learning for an incentive-based demand response model. In Proceedings of the 2022 IEEE 6th Conference on Energy Internet and Energy System Integration (EI2), Chengdu, China, 11–13 November 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 246–250.

30. Casalicchio, E.; Esposito, S.; Al-Saedi, A.A. FLWB: A Workbench Platform for Performance Evaluation of Federated Learning Algorithms. In Proceedings of the 2023 IEEE International Workshop on Technologies for Defense and Security (TechDefense), Rome, Italy, 20–22 November 2023.

31. Hsu, C.F.; Huang, J.L.; Liu, F.H.; Chang, M.C.; Chen, W.C. Fedtrust: Towards building secure robust and trustworthy moderators for federated learning. In Proceedings of the 2022 IEEE 5th International Conference on Multimedia Information Processing and Retrieval (MIPR), Virtual, 2–4 August 2022;

IEEE: Piscataway, NJ, USA, 2022; pp. 318–323.

32. Zhang, X.; Li, F.; Zhang, Z.; Li, Q.; Wang, C.; Wu, J. Enabling execution assurance of federated learning at untrusted participants. In Proceedings of the IEEE INFOCOM 2020-IEEE Conference on Computer Communications, Virtual, 6–9 July 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1877–1886.

33. Lyubchyk, L.; Grinberg, G.; Konokhova, Z.; Yamkovyi, K. Composite Indicators Building Based on Concordant of Expert-Statistical Information Using Biased Ridge Kernel Regression. In Proceedings of the 2023 IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Dortmund, Germany, 7–9 September 2023; IEEE: Piscataway, NJ, USA, 2023; Volume 1, pp. 617–620.

34. Yang, H.; Gu, D.; He, J. A robust and efficient federated learning algorithm against adaptive model poisoning attacks. IEEE Internet Things J. 2024, 11, 16289–16302.

35. Yazdinejad, A.; Dehghantanha, A.; Karimipour, H.; Srivastava, G.; Parizi, R.M. A robust privacy-preserving federated learning model against model poisoning attacks. IEEE Trans. Inf. Forensics Secur. 2024, 19, 6693–6708.

36. Perry, S.; Jiang, Y.; Zhong, F.; Huang, J.; Gyawali, S. DynaDetect2.0: Improving Detection Accuracy of Data Poisoning Attacks. In Proceedings of the 2024 Cyber Awareness and Research Symposium (CARS), Grand Forks, ND, USA, 28–29 October 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 1–8.

37. Abri, F.; Zheng, J.; Namin, A.S.; Jones, K.S. Markov decision process for modeling social engineering attacks and finding optimal attack strategies. IEEE Access 2022, 10, 109949–109968.

38. Malik, P.; Alirajpurwala, T.; Kaushal, S.; Patidar, T.; Indore, I.I.I.; Padlak, S. Scalability and robustness of federated learning systems: Challenges and solutions. Int. J. Sci. Res. Eng. Manag. IJSREM 2024, 8.

39. Abdelmoniem, A.M.; Ho, C.Y.; Papageorgiou, P.; Canini, M. A comprehensive empirical study of heterogeneity in federated learning. IEEE Internet Things J. 2023, 10, 14071–14083.

40. Abdelmoniem, A.M.; Ho, C.Y.; Papageorgiou, P.; Canini, M. Empirical analysis of federated learning in heterogeneous environments. In Proceedings of the 2nd European Workshop on Machine Learning and Systems, Rennes France, 5–8 April 2022; pp. 1–9.

41. Yu, X.; He, Z.; Sun, Y.; Xue, L.; Li, R. The Effect of Personalization in FedProx: A Fine-grained Analysis on Statistical Accuracy and Communication Efficiency. arXiv 2024, arXiv:2410.08934.

42. Kang, H.; Kim, M.; Lee, B.; Kim, H. FedAND: Federated learning exploiting consensus ADMM by nulling drift. IEEE Trans. Ind. Inform. 2024, 20, 9837–9849.

43. Mahmoud, M.H.; Albaseer, A.; Abdallah, M.; Al-Dhahir, N. Federated learning resource optimization and client selection for total energy minimization under outage, latency, and bandwidth constraints with partial or no CSI. IEEE Open J. Commun. Soc. 2023, 4, 936–953.

44. Antonioli, D.; Tippenhauer, N.O.; Rasmussen, K. Bias: Bluetooth impersonation attacks. In Proceedings of the 2020 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 18–20 May 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 549–562.

45. Ebrahimabadi, M.; Lalouani, W.; Younis, M.; Karimi, N. Countering PUF modeling attacks through adversarial machine learning. In Proceedings of the 2021 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), Tampa, FL, USA, 7–9 July 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 356–361.

46. Wang, B.; Sun, S.; Ren, W. Distributed continuous-time algorithms for optimal resource allocation with time-varying quadratic cost functions. IEEE Trans. Control Netw. Syst. 2020, 7, 1974–1984.

47. Javaherian, S.; Turney, B.; Chen, L.; Tzeng, N.F. Incentive-Compatible Federated Learning with Stackelberg Game Modeling. arXiv 2025, arXiv:2501.02662.

48. Collins, L.; Hassani, H.; Mokhtari, A.; Shakkottai, S. Fedavg with fine tuning: Local updates lead to representation learning. Adv. Neural Inf. Process. Syst. 2022, 35, 10572–10586.

49. Mora, A.; Bujari, A.; Bellavista, P. Enhancing generalization in federated learning with heterogeneous data: A comparative literature review. Future Gener. Comput. Syst. 2024, 157, 1–15.

50. Chakravarthy, V.; Bell, D.; Bhaskaran, S. Emergent Intrusion Detection System for Fog Enabled Smart Agriculture Using Federated Learning and Blockchain Technology: A Review. In Proceedings of the 2024 International Conference on IT Innovation and Knowledge Discovery (ITIKD), Manama, Bahrain, 13–15 April 2025; IEEE: Piscataway, NJ, USA, 2025; pp. 1–7.

51. Yang, K.; Imam, N. Secure and Private Federated Learning: Achieving Adversarial Resilience through Robust Aggregation. arXiv 2025,

arXiv:2505.17226.