A Bayesian Incentive Mechanism for Poison-Resilient Federated Learning

Daniel Commey*, Rebecca A. Sarpong†, Griffith S. Klogo‡, Winful Bagyl-Bac§, and Garth V. Crosby¶
*Department of Multidisciplinary Engineering, Texas A&M University, College Station, TX, USA
dcommey@tamu.edu
†Department of Statistics and Actuarial Science, KNUST, Kumasi, Ghana
rasarpong5@st.knust.edu.gh
‡Department of Computer Engineering, KNUST, Kumasi, Ghana
gsklogo.coe@knust.edu.gh
§Department of Computer Science, George Washington University, Washington, DC, USA
winful.bagylbac@gwu.edu
¶Department of Engineering Technology and Industrial Distribution, Texas A&M University, College Station, TX, USA
gvcrosby@tamu.edu

***Abstract*—Federated learning (FL) enables collaborative model training across decentralized clients while preserving data privacy. However, its open-participation nature exposes it to data-poisoning attacks, in which malicious actors submit corrupted model updates to degrade the global model. Existing defenses are often *reactive*, relying on statistical aggregation rules that can be computationally expensive and that typically assume an honest majority. This paper introduces a *proactive*, economic defense: a lightweight Bayesian incentive mechanism that makes malicious behavior economically irrational. Each training round is modeled as a Bayesian game of incomplete information in which the server, acting as the principal, uses a small, private validation dataset to verify update quality before issuing payments. The design satisfies Individual Rationality (IR) for benevolent clients, ensuring their participation is profitable, and Incentive Compatibility (IC), making poisoning an economically dominated strategy. Extensive experiments on non-IID partitions of MNIST and FashionMNIST demonstrate robustness: with 50 % label-flipping adversaries on MNIST, the mechanism maintains 96.7 % accuracy, only 0.3 percentage points lower than in a scenario with 30 % label-flipping adversaries. This outcome is 51.7 percentage points better than standard FedAvg, which collapses under the same 50 % attack. The mechanism is computationally light, budget-bounded, and readily integrates into existing FL frameworks, offering a practical route to economically robust and sustainable FL ecosystems.**

***Index Terms*—Federated learning, mechanism design, Bayesian games, data poisoning, incentive compatibility, robust aggregation.**

I. Introduction

Federated learning (FL) has emerged as a key paradigm for privacy-preserving machine learning, allowing multiple parties to train a shared model without centralizing their raw data [1], [2]. While promising for sensitive applications such as healthcare [3], its distributed and open nature creates a significant vulnerability: data poisoning attacks [4], [5]. Malicious participants can intentionally submit corrupted model updates to degrade the global model's performance or introduce targeted backdoors [4], [5]. The breadth and severity of these vulnerabilities are well-documented in recent surveys [6].

The predominant line of defense has been the development of Byzantine-robust aggregation rules. Methods like Krum [7], Trimmed Mean [8], and geometric median-based approaches like RFA [9] aim to filter or down-weight malicious updates at the server. However, these methods are fundamentally *reactive*. They often require strong assumptions (e.g., an honest majority), can be computationally intensive, and may discard valuable information from honest clients, thereby slowing convergence. More critically, they fail to address the underlying economic misalignment: honest clients who contribute valuable resources (computation, data, communication) are treated no differently from attackers who seek to sabotage the system.

This economic imbalance threatens the long-term sustainability of open FL ecosystems. If honest participation is not properly incentivized and malicious behavior is not penalized, the system becomes prone to collapse. This leads us to our research question: *Can we design an FL system where participants' economic incentives are aligned with the goal of training a high-quality model, making poisoning attacks unprofitable at equilibrium?*

In this work, we draw upon the principles of mechanism design and game theory to provide an affirmative answer. We propose a proactive, lightweight Bayesian incentive mechanism that shifts the defense from a purely algorithmic problem to a socio-economic one.

Contributions. Our main contributions are as follows:

- We formulate the FL training process as a repeated Bayesian game of incomplete information, formally capturing the strategic decisions of clients who can be either benevolent or malicious.
- We design a simple yet powerful incentive mechanism where the server uses a small, private validation set to assess the quality of submitted updates. Based on this verification, it issues rewards, effectively creating a market for high-quality model contributions.
- We provide formal proofs demonstrating that our mechanism is **Individually Rational (IR)**, ensuring benevolent clients have a positive expected utility, and **Incentive Compatible (IC)**, making poisoning an economically

dominated strategy for rational attackers.

- We conduct extensive experiments on the MNIST and FashionMNIST datasets with non-IID data distributions. Our mechanism demonstrates exceptional robustness, maintaining high accuracy (over 96% on MNIST and 80% on FashionMNIST) even when 50% of clients are malicious—a scenario where standard FedAvg's accuracy catastrophically collapses.

II. Related Work

A. *Robust Aggregation in Federated Learning*

The primary defense against poisoning in FL has centered on Byzantine-robust aggregation. These server-side methods aim to identify and mitigate the impact of malicious updates during the aggregation phase.

Federated Averaging (FedAvg) [1] is the standard, non-robust baseline. The server aggregates updates by taking a weighted average of the model parameters from participating clients. While simple and effective in non-adversarial settings, it is highly susceptible to even a single malicious client.

Byzantine-Robust Methods have been developed to counter this vulnerability. **Krum** [7] computes a score for each client update based on its sum of squared Euclidean distances to its nearest neighbors and selects only the single update with the lowest score. This is robust but highly inefficient, as it discards the contributions of all other honest clients. **Coordinate-wise methods** like Trimmed Mean and Median [8] compute the median or a trimmed mean for each coordinate of the model-weight vectors across all clients. These are robust to extreme values but can be distorted by more subtle attacks. **Geometric median-based methods** like RFA [9] compute the geometric median of the client updates, which is more robust to high-dimensional outliers than the arithmetic mean but is computationally expensive. **Trusted-source methods** like FLTrust [10] require the server to have a small, clean "root" dataset. The server trains a baseline update on this set and re-weights client updates based on their cosine similarity to this trusted update. Another line of work uses redundancy and coding theory, such as **DRACO** [11], which uses coded computations to detect and correct errors from stragglers or Byzantine workers, though this often requires significant overhead.

Our work is distinct from these approaches. Instead of relying on statistical properties or a trusted data source, we use a performance-based economic filter that is agnostic to the attack's specific structure.

B. *Incentive Mechanisms for FL*

Recognizing the need to motivate participation, researchers have explored economic incentives for FL. These works primarily focus on encouraging high-quality participation from rational, self-interested clients. Reputation-based systems [12] and contract theory [13] have been proposed to model the contributions of clients and offer tailored rewards. Auction theory has also been applied, for example, in [14], where the server runs an auction to select clients with the best data quality for a given budget. Some works use blockchain to create decentralized and transparent reward systems, like FedCoin [15].

Our work's novelty lies in its direct focus on the security dimension of incentives. We design a mechanism with formal game-theoretic guarantees (IR and IC) to not only encourage honest participation but to actively and provably *discourage* poisoning attacks by making them economically non-viable. The work closest in spirit is perhaps VeriFL [16], which also uses a validation set, but its goal is post-hoc verification and attribution rather than proactive, in-round economic deterrence.

III. System and Threat Model

A. *System Model*

We consider a standard synchronous FL architecture comprising a central server and a population of N clients. Training proceeds in discrete communication rounds. In each round t, the server broadcasts the current global model, w_t, to a subset of clients. These clients train the model on their local data and submit their updated model parameters, $w_{i,t+1}$, back to the server. The server then aggregates these updates to produce the next global model, w_{t+1}. Our key innovation lies in the verification and payment logic applied before aggregation, as depicted in Figure 1.

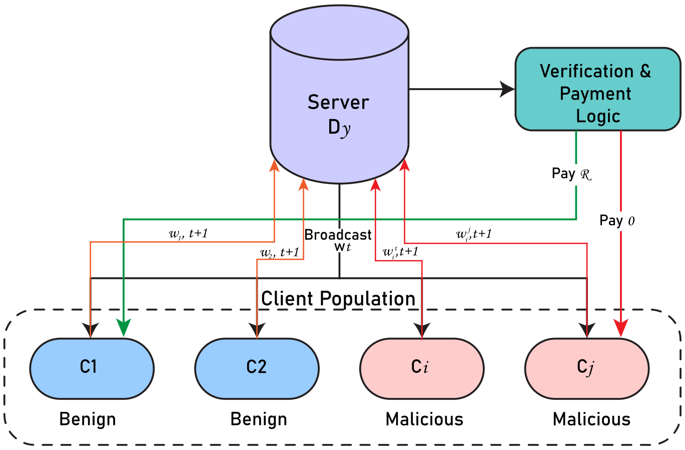


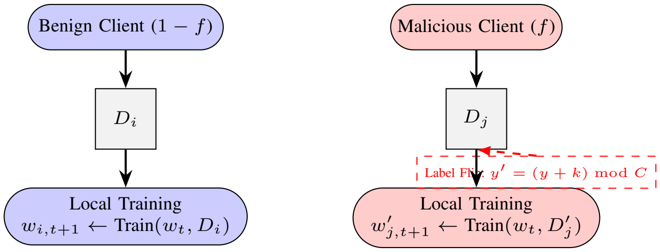
Fig. 1. System architecture with the proposed incentive mechanism. The server broadcasts the global model w_t. Benign clients (C1, C2) and malicious clients (Ci, Cj) submit their updates. The server uses a private validation set (D_y) to assess update quality. High-quality updates are accepted and paid a reward $\mathcal{R}$, while malicious updates are rejected and receive no payment. Only verified updates are used for aggregation.

For our experiments, we use a population of 100 clients. Data from the MNIST and FashionMNIST datasets is partitioned among clients in a non-IID fashion using a Dirichlet distribution with $\alpha = 0.5$ to simulate realistic data heterogeneity.

B. *Threat Model*

We assume an honest-but-curious server that faithfully executes the protocol but may try to infer information from client updates. A fraction f of the clients are malicious, while the remaining $1 - f$ are benevolent. The server has incomplete information; it knows the overall fraction f but does not know the type of any individual client a priori.

The malicious clients aim to degrade the global model's performance on the primary task. To this end, they employ a **label-flipping** attack, a potent form of data poisoning [4]. During local training, a malicious client maps each true label y to a target label y', effectively training its model on deliberately mislabeled data. For our 10-class datasets, we use the mapping $y' = (y + k) \bmod 10$, where k is an offset (we use $k = 1$ for our attacks). This process, illustrated in Figure 2, forces the client's local model to learn incorrect associations, and when aggregated, these poisoned updates corrupt the global model. More sophisticated attacks, such as targeted backdooring [5], follow a similar principle of manipulating local training to achieve a malicious objective.



Benign Client $(1 - f)$
D_i
Local Training
$w_{i,t+1} \leftarrow \text{Train}(w_t, D_i)$
Malicious Client (f)
D_j
Label Flip: $y' = (y + k) \bmod C$
Local Training
$w'_{j,t+1} \leftarrow \text{Train}(w_t, D'_j)$

Fig. 2. The threat model. A fraction f of clients are malicious. They poison their local dataset D_j by flipping labels to create D'_j before training their local model. Benign clients train on their original, clean data D_i.

IV. Bayesian Incentive Mechanism

A. Game Formulation

We model each round of federated training as a Bayesian game of incomplete information, defined by the tuple $\Gamma = \langle \mathcal{N}, \{\Theta_i\}, \{A_i\}, \{u_i\}, p \rangle$:

- $\mathcal{N}$: The set of players, consisting of the server and N clients.
- Θ_i: The set of types for each client i, $\Theta_i = \{\text{benevolent}, \text{malicious}\}$. A client's type is private information.
- A_i: The action space for client i. A client chooses an action $a_i \in \{\text{honest\_update}, \text{poisoned\_update}\}$.
- $p(\theta_i)$: The server's prior belief about the probability that client i is of a certain type. We assume a common prior where $P(\theta_i = \text{malicious}) = f$.
- u_i: The utility function for client i. The utility is determined by the payment received from the server, p_i, minus the operational cost incurred, C_i. Thus, $u_i = p_i - C_i$. We assume a uniform cost C for all clients for simplicity.

B. The Verification and Payment Mechanism

The core of our defense is a direct revelation mechanism where the server incentivizes clients to reveal their "true" contribution quality.

Definition 1 (Verification Mechanism)**.** The server holds a small, private, and clean validation dataset, D_y. Upon receiving a model update $w_{i,t+1}$ from client i, the server evaluates its loss on this set: $L_i = \mathcal{L}(w_{i,t+1}; D_y)$. The update is considered "verified" if its loss is below a predefined quality threshold τ. The payment rule is:

$$p_i(w_{i,t+1}) = \begin{cases} \mathcal{R} & \text{if } \mathcal{L}(w_{i,t+1}; D_y) < \tau \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $\mathcal{R}$ is a fixed reward for a verified update. Only the set of verified updates, $\mathcal{V}_t$, are aggregated to form the next global model:

$$w_{t+1} = \frac{1}{|\mathcal{V}_t|} \sum_{w_j \in \mathcal{V}_t} w_j \tag{2}$$

This mechanism is lightweight, as it only requires a single forward pass on a small validation set for each client, a negligible cost compared to the training itself. The logic is outlined in Algorithm 1.

C. Mechanism Properties

A robust mechanism must make it profitable for honest players to participate and unprofitable for malicious players to attack. These correspond to the game-theoretic properties of Individual Rationality (IR) and Incentive Compatibility (IC).

Theorem 1 (Individual Rationality)**.** *For a benevolent client choosing the honest action, the expected utility is positive if the reward $\mathcal{R}$ and cost C are set such that $\mathcal{R} > C/P(verification|honest)$.*

Proof. A benevolent client's action is to submit an honestly trained update. Let this action be a_h. The client's update will be verified if its loss on D_y is less than τ. Let the probability of this event be $P_v^h = P(\mathcal{L}(w_{honest}) < \tau)$. The expected utility for a benevolent client is:

$$\mathbb{E}[u_i|\theta_i = \text{benevolent}, a_i = a_h] = P_v^h \cdot \mathcal{R} + (1 - P_v^h) \cdot 0 - C \tag{3}$$

For participation to be rational, this expected utility must be greater than 0 (the utility of not participating).

$$P_v^h \cdot \mathcal{R} - C > 0 \implies \mathcal{R} > \frac{C}{P_v^h} \tag{4}$$

Since an honest update is designed to minimize the loss, its loss on a clean validation set will be low. Therefore, for a reasonably set τ, $P_v^h \approx 1$. With our parameters $\mathcal{R} = 10$ and $C = 2$, the condition becomes $10 > 2/P_v^h$, which holds easily. Thus, honest participation is economically rational. □

Theorem 2 (Incentive Compatibility)**.** *For a rational, self-interested client, choosing a poisoned action is an economically dominated strategy if the attack significantly increases the model's loss on a clean validation set.*

Proof. A malicious client's goal is to submit a poisoned update, a_p, to degrade the model. This action inherently increases the model's true loss. Let the probability of a poisoned update passing verification be $P_v^m = P(\mathcal{L}(w_{poisoned}) < \tau)$. The expected utility for taking the poisoned action is:

$$\mathbb{E}[u_i|a_i = a_p] = P_v^m \cdot \mathcal{R} - C \tag{5}$$

For a label-flipping attack, the resulting model will perform poorly on the correctly labeled validation set D_y. Thus, its loss will be high, and for a reasonable τ, the probability of verification will be near zero, $P_v^m \approx 0$. The expected utility becomes:

$$\mathbb{E}[u_i|a_i = a_p] \approx 0 \cdot \mathcal{R} - C = -C \tag{6}$$

A rational agent will compare this negative utility to the utility of not participating (utility 0) or participating honestly (positive utility, from Theorem 1). Since $-C < 0$, the poisoning strategy is strictly dominated by non-participation. This demonstrates that the mechanism is incentive-compatible, as it disincentivizes the malicious action. □

V. Experimental Setup

We implemented our system in PyTorch and conducted experiments to evaluate its performance against standard baselines.

- **Datasets:** We used two benchmark datasets: **MNIST** (handwritten digits) and **FashionMNIST** (apparel images). Both have 10 classes. Data was distributed among 100 clients using a Dirichlet distribution ($\alpha = 0.5$) to simulate a non-IID environment.
- **Model:** A Convolutional Neural Network (CNN) with two convolutional layers (32 and 64 filters, 5x5 kernel), each followed by max-pooling, and two fully-connected layers (1024 units and 10 units for the output).
- **Training Protocol:** 40 communication rounds, 3 local epochs per round, batch size 32, and SGD with a learning rate of 0.01.
- **Attack Scenarios:** We tested with malicious client fractions (f) of 30%, 40%, and 50%. Malicious clients performed a label-flipping attack ($y' = (y + 1) \bmod 10$).
- **Baselines for Comparison:**
  - **FedAvg:** The standard, non-robust federated averaging algorithm.
  - **Krum:** A well-known Byzantine-robust aggregation rule that selects the single "best" update.
- **Mechanism Parameters:** Reward $\mathcal{R} = 10$, Cost $C = 2$, and verification threshold $\tau = 2.5$. The server's private validation set D_y contained 200 randomly sampled examples.

The pseudocode for the server's logic in our proposed mechanism is detailed in Algorithm 1.

Algorithm 1 Federated Learning with Bayesian Incentive Mechanism (Server-Side Logic)

```
Require: Reward $\mathcal{R}$, Cost $C$, Threshold $\tau$, Validation set $D_y$
Initialize global model $w_0$
for each communication round $t = 0, 1, \ldots, T-1$ do
    Broadcast global model $w_t$ to all clients
    $\mathcal{U}_t \leftarrow \emptyset$ {Collect all incoming updates}
    for all clients $i \in \{1, \ldots, N\}$ in parallel do
        $w_{i,t+1} \leftarrow \textsc{ClientUpdate}(w_t, D_i)$
        $\mathcal{U}_t \leftarrow \mathcal{U}_t \cup \{w_{i,t+1}\}$
    end for
    $\mathcal{V}_t \leftarrow \emptyset$ {Set of verified updates}
    for each update $w_i \in \mathcal{U}_t$ do
        $L_i \leftarrow \textsc{EvaluateLoss}(w_i, D_y)$
        if $L_i < \tau$ then
            Pay $\mathcal{R}$ to client $i$
            $\mathcal{V}_t \leftarrow \mathcal{V}_t \cup \{w_i\}$
        else
            Pay 0 to client $i$ {Client incurs cost C}
        end if
    end for
    if $|\mathcal{V}_t| > 0$ then
        $w_{t+1} \leftarrow \frac{1}{|\mathcal{V}_t|} \sum_{w \in \mathcal{V}_t} w$ {Aggregate verified updates}
    else
        $w_{t+1} \leftarrow w_t$ {No updates verified, maintain model}
    end if
end for
```

VI. Experimental Results

Our experiments provide a comprehensive view of the mechanism's performance, robustness, and economic effects across both MNIST and FashionMNIST datasets. We compare our proposed Bayesian Incentive Mechanism (which we will refer to as "Mechanism") against standard FedAvg and the well-known robust aggregation rule, Krum.

A. Overall Performance and Robustness

We first present a high-level analysis of the mechanism's resilience to an increasing number of attackers. Figure 3 summarizes the key outcomes. The top-left panel shows the final test accuracy on MNIST as the fraction of malicious clients increases from 30% to 50%. While FedAvg's accuracy plummets from 95.3% to 43.5%, and Krum's performance degrades, our mechanism's accuracy remains exceptionally stable above 96.7%. The top-right panel reinforces this, showing our mechanism achieving 97.0% accuracy on MNIST and 80.3% on FashionMNIST in the challenging 40% malicious scenario, significantly outperforming both baselines.

The robustness of our approach is quantified in Table I. When the attacker population grows from 30% to 50%, FedAvg's accuracy suffers a catastrophic drop of 51.8 percentage points on MNIST and 45.3 points on FashionMNIST. Krum also proves vulnerable, especially on FashionMNIST, where its accuracy collapses. In stark contrast, our mechanism's performance degrades by a negligible 0.24 points on MNIST and 1.22 points on FashionMNIST, demonstrating that its economic filtering effectively insulates the global model from the number of attackers.

Finally, the bottom-right panel of Figure 3 validates our economic model. The average utility for an honest client quickly converges to the theoretical maximum of $\mathcal{R} - C = 8$, confirming that honest participation is consistently and profitably rewarded.

B. Detailed Analysis on MNIST

Figure 4 provides a detailed view of the training dynamics on MNIST. The final accuracy (top-left panel) remains consistently high for our mechanism, achieving 96.7% even with 50%

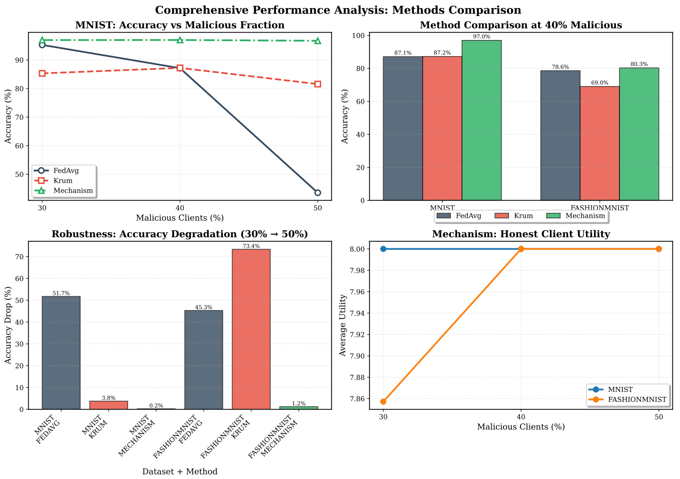


Fig. 3. Comprehensive performance analysis comparing FedAvg, Krum, and our proposed mechanism. (Top-Left) Final test accuracy on MNIST vs. the fraction of malicious clients. (Top-Right) A direct comparison of final accuracy at 40% malicious for both MNIST and FashionMNIST. (Bottom-Left) The drop in accuracy when increasing malicious clients from 30% to 50%, highlighting robustness. (Bottom-Right) The average utility for honest clients participating in our mechanism, showing convergence to the theoretical maximum of 8.

TABLE I
ROBUSTNESS ANALYSIS: ACCURACY DEGRADATION WHEN INCREASING MALICIOUS CLIENT FRACTION FROM 30% TO 50%.

Method	**Acc. @ 30%**	**Acc. @ 50%**	**Degrad. (% pts)**
		MNIST Dataset	
FedAvg	95.27%	43.52%	51.75
Krum	85.31%	81.56%	3.75
Mechanism	**96.96%**	**96.72%**	**0.24**
		FashionMNIST Dataset	
FedAvg	80.74%	35.44%	45.30
Krum	73.68%	0.33%	73.35
Mechanism	**81.89%**	**80.67%**	**1.22**

attackers, as also detailed in Table II. This stability is a direct result of the economic filter. The training progress (bottom-right panel) for the 40% attack scenario shows our mechanism's smooth and rapid convergence, while FedAvg is erratic and Krum is noisy.

From an economic perspective, the top-right panel shows that the average utility for honest clients is stable and positive, rapidly converging towards the ideal payoff of 8.0. This empirically validates Theorem 1 (Individual Rationality). Malicious clients, whose updates are consistently rejected, receive zero payment and incur the cost C, yielding a negative utility. This validates Theorem 2 (Incentive Compatibility), as attacking is an economically irrational choice. The total server expenditure (bottom-left panel) remains bounded and stable, demonstrating the economic sustainability of the system.

C. Detailed Analysis on FashionMNIST

The results on the more challenging FashionMNIST dataset (Figure 5) further underscore our mechanism's strengths. While our mechanism maintains over 80% accuracy across all attack levels, FedAvg's performance degrades sharply, and Krum fails catastrophically, with its accuracy dropping to near-random levels (0.33% with 50% attackers), highlighting the fragility of distance-based metrics on more complex tasks. These final accuracy numbers are compiled in Table II.

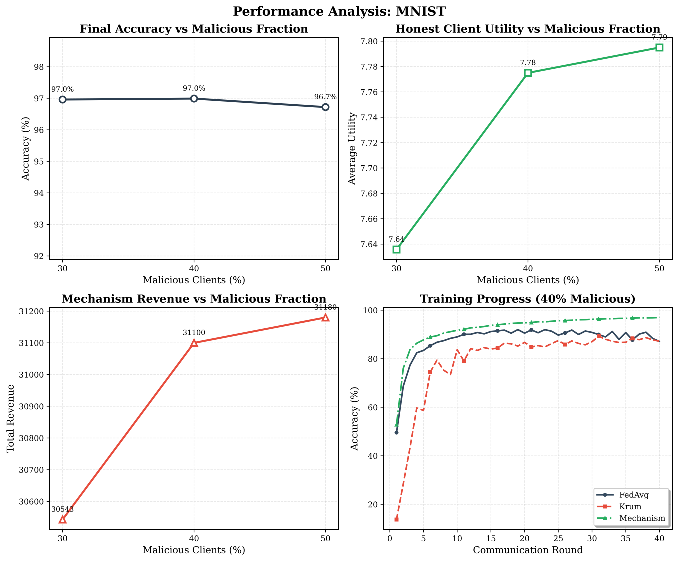


Fig. 4. Detailed performance and economic analysis on the MNIST dataset. (Top-Left) Final accuracy remains high and stable. (Top-Right) Average utility for honest clients is stable and positive. (Bottom-Left) Total server revenue remains bounded. (Bottom-Right) Training progress at 40% malicious shows superior convergence.

The economic outcomes are equally strong. The utility for honest clients (top-right panel) remains positive, ensuring participation is viable. The training curve at 40% malicious (bottom-right panel) again confirms our mechanism's stability and superior convergence. The total server revenue (bottom-left panel) is stable, showing the mechanism is not just robust but also budget-conscious, as it avoids paying for low-quality or malicious contributions.

TABLE II
DETAILED FINAL PERFORMANCE AND ECONOMIC METRICS.

Dataset	**Malicious %**	**Method**	**Final Acc. %**	**Honest Utility**	**Total Revenue**
MNIST	30%	FedAvg Krum **Mechanism**	95.27 85.31 **96.96**	— — **7.64**	— — **30.5k**
	40%	FedAvg Krum **Mechanism**	87.13 87.22 **97.00**	— — **7.78**	— — **31.1k**
	50%	FedAvg Krum **Mechanism**	43.52 81.56 **96.70**	— — **7.79**	— — **31.2k**
FMNIST	30%	FedAvg Krum **Mechanism**	80.74 73.68 **81.90**	— — **7.67**	— — **30.7k**
	40%	FedAvg Krum **Mechanism**	78.60 69.03 **80.30**	— — **7.56**	— — **30.2k**
	50%	FedAvg Krum **Mechanism**	35.44 0.33 **80.70**	— — **7.56**	— — **30.2k**

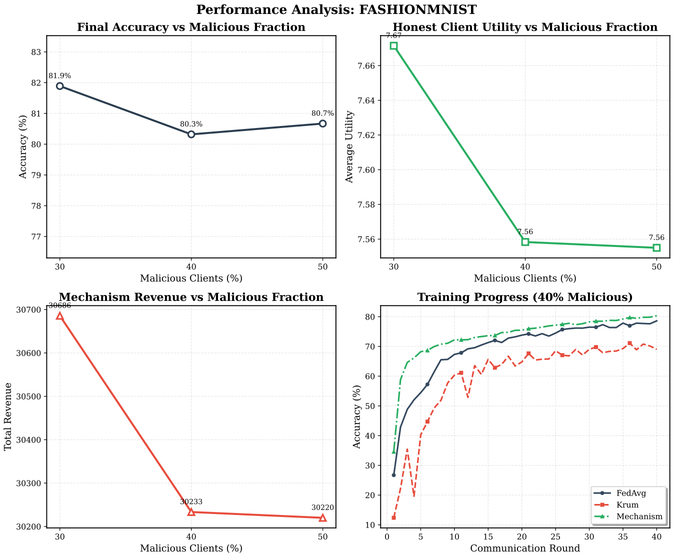


Fig. 5. Detailed performance and economic analysis on the FashionMNIST dataset. (Top-Left) Our mechanism maintains high accuracy while baselines fail. (Top-Right) Average utility for honest clients remains positive and viable. (Bottom-Left) Total server revenue shows controlled expenditure. (Bottom-Right) Training progress confirms the stability and effectiveness of our mechanism.

VII. Conclusion

In this paper, we introduced a game-theoretic incentive mechanism that provides a proactive, economic defense against data poisoning attacks in federated learning. By framing the FL process as a Bayesian game and implementing a simple, low-cost verification step, our mechanism successfully aligns client incentives with the global objective of training an accurate model.

Our extensive experiments on MNIST and FashionMNIST demonstrate the mechanism's remarkable effectiveness. It maintains high accuracy and stability even under extreme attack conditions (50% malicious clients) where standard FedAvg fails completely and the popular robust aggregator Krum struggles or fails. Our robustness analysis (Table I) quantifies this resilience, showing negligible performance degradation under increased attack intensity. We formally proved, and empirically validated through detailed results (Figures 4 and 5, and Table II), that the mechanism is individually rational for honest participants and incentive-compatible for deterring attackers by making malicious behavior economically non-viable.

This work shows that shifting focus from purely algorithmic defenses to socio-economic ones is a powerful and practical strategy for building secure, robust, and sustainable federated learning systems. Future work could explore adaptive verification thresholds, extend the mechanism to defend against more subtle attack strategies like model backdooring, and investigate its application in fully decentralized settings.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282. [Online]. Available: https://proceedings.mlr.press/v54/mcmahan17a?ref=https://githubhelp.com

[2] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE signal processing magazine*, vol. 37, no. 3, pp. 50–60, 2020, publisher: IEEE. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9084352/

[3] D. Commey, S. Hounsinou, and G. V. Crosby, "Securing Health Data on the Blockchain: A Differential Privacy and Federated Learning Framework," May 2024, arXiv:2405.11580 [cs]. [Online]. Available: http://arxiv.org/abs/2405.11580

[4] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *International conference on machine learning*. PMLR, 2019, pp. 634–643. [Online]. Available: https://proceedings.mlr.press/v97/bhagoji19a.html

[5] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *International conference on artificial intelligence and statistics*. PMLR, 2020, pp. 2938–2948. [Online]. Available: https://proceedings.mlr.press/v108/bagdasaryan20a.html

[6] L. Lyu, H. Yu, and Q. Yang, "Threats to Federated Learning: A Survey," Mar. 2020, arXiv:2003.02133 [cs]. [Online]. Available: http://arxiv.org/abs/2003.02133

[7] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," *Advances in neural information processing systems*, vol. 30, 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/f4b9ec30ad9f68f89b29639786cb62ef-Abstract.html

[8] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *International conference on machine learning*. Pmlr, 2018, pp. 5650–5659. [Online]. Available: https://proceedings.mlr.press/v80/yin18a

[9] K. Pillutla, S. M. Kakade, and Z. Harchaoui, "Robust aggregation for federated learning," *IEEE Transactions on Signal Processing*, vol. 70, pp. 1142–1154, 2022, publisher: IEEE. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9721118/

[10] X. Cao, M. Fang, J. Liu, and N. Z. Gong, "FLTrust: Byzantine-robust Federated Learning via Trust Bootstrapping," Apr. 2022, arXiv:2012.13995 [cs]. [Online]. Available: http://arxiv.org/abs/2012.13995

[11] L. Chen, H. Wang, Z. Charles, and D. Papailiopoulos, "Draco: Byzantine-resilient distributed training via redundant gradients," in *International Conference on Machine Learning*. PMLR, 2018, pp. 903–912. [Online]. Available: http://proceedings.mlr.press/v80/chen18l

[12] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, "Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 700–10 714, 2019, publisher: IEEE. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8832210/

[13] Y. Zhan, P. Li, Z. Qu, D. Zeng, and S. Guo, "A learning-based incentive mechanism for federated learning," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6360–6368, 2020, publisher: IEEE. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8963610/

[14] J. Zhang, Y. Wu, and R. Pan, "Incentive Mechanism for Horizontal Federated Learning Based on Reputation and Reverse Auction," in *Proceedings of the Web Conference 2021*. Ljubljana Slovenia: ACM, Apr. 2021, pp. 947–956. [Online]. Available: https://dl.acm.org/doi/10.1145/3442381.3449888

[15] Y. Liu, Z. Ai, S. Sun, S. Zhang, Z. Liu, and H. Yu, "FedCoin: A Peer-to-Peer Payment System for Federated Learning," in *Federated Learning*, Q. Yang, L. Fan, and H. Yu, Eds. Cham: Springer International Publishing, 2020, vol. 12500, pp. 125–138, series Title: Lecture Notes in Computer Science. [Online]. Available: http://link.springer.com/10.1007/978-3-030-63076-8_9

[16] X. Guo, Z. Liu, J. Li, J. Gao, B. Hou, C. Dong, and T. Baker, "VeriFL: Communication-Efficient and Fast Verifiable Aggregation for Federated Learning," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1736–1751, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9285303