DETECTING MODEL POISONING ATTACKS IN FEDERATED LEARNING VIA JOINT SPATIO-TEMPORAL INSPECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Detecting model poisoning attacks in federated learning (FL) remains challenging under heterogeneous data distributions and coordinated adversarial behavior. Non-IID data can induce substantial variation among benign updates, leading detectors to misclassify them as malicious, whereas coordinated attackers can submit mutually similar updates to conceal their collective attack behavior. To address these limitations, we propose TSDETECTOR, a joint spatio-temporal model poisoning detection framework that characterizes client behavior from two complementary perspectives. Specifically, from the spatial perspective, TSDETECTOR characterizes the relationships among clients by adaptively fusing an update distance graph which captures discrepancies among client updates with a knowledge discrepancy graph that reflects their similarities in learned knowledge on a reference dataset, from the temporal perspective, TSDETECTOR models client specific update trajectories across communication rounds to identify cross round poisoned attack behaviors compared with normal training dynamics. Extensive experiments across five benchmark datasets and five representative untargeted attacks show that TSDETECTOR reduces average attack impact to 1.16%, with maximum attack induced performance degradation limited to 4.6%. TSDETECTOR avoids severe performance degradation observed in multiple baseline defenses, demonstrating broader applicability across diverse datasets and poisoning strategies.

1 INTRODUCTION

Federated learning (FL) enables multiple clients to collaboratively train a global model without centrally collecting their private data (McMahan et al., 2017). However, the server's limited visibility into clients' local data leaves FL vulnerable to model poisoning attacks (Imteaj et al., 2023), in which compromised clients submit carefully crafted model updates to manipulate global optimization. Depending on the adversarial objective, model poisoning attacks can be categorized as untargeted (Baruch et al., 2019; Fang et al., 2020; Shejwalkar & Houmansadr, 2021; Xie et al., 2025), which aim to degrade overall performance of global model, or targeted (Xie et al., 2020; Mahloujifar et al., 2019; El-Mhamdi et al., 2018; Blanchard et al., 2017), which manipulate predictions on specific inputs while preserving its behavior on others. This paper focus on untargeted model poisoning attacks, which can render the global model unusable and hence pose more severe threats.

Existing defenses against model poisoning attacks broadly fall into two categories: byzantine robust aggregation and server side poisoning detection. Byzantine robust aggregation replaces standard averaging with aggregation rules that select, trim, or reweight client updates according to their statistical relationships or consistency with a trusted reference (Blanchard et al., 2017; Yin et al., 2018; Cao et al., 2021). Server side poisoning detection instead assigns suspicion scores to participating clients based on their reported updates and removes those identified as malicious before aggregation (Zhang et al., 2022; Yan et al., 2023).

Despite their effectiveness in specific settings, existing defenses typically examine client updates through a single view. under non-IID data, benign updates may differ substantially in parameter space, causing distance or coordinate based defenses to misclassify them as malicious. Meanwhile,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

coordinated adversaries can submit mutually similar updates and form a compact cluster, thereby concealing their collective behavior from detectors based solely on relationships among clients. Conversely, defenses relying primarily on historical consistency may overlook contemporaneous client relations and representation level changes induced by malicious updates (Zhang et al., 2022; Yan et al., 2023). These complementary blind spots motivate the joint analysis of spatial relations and client specific temporal trajectories.

To overcome these limitations, we examine client behavior from complementary spatial and temporal perspectives. The spatial perspective exploits the functional consistency that benign clients tend to preserve when optimizing the same task from a common global model. Although non-IID data may induce substantial differences in their parameter updates, benign local models often retain similar relational structures over shared reference inputs. Malicious updates, however, can resemble benign ones in parameter space while inducing distinct representation structures, providing a complementary basis for identifying abnormal behavior. The temporal perspective instead captures how client behavior evolves across communication rounds. An attack may remain inconspicuous within an individual round, yet repeatedly introducing a persistent bias can produce temporal patterns that deviate from those observed during normal training. These two perspectives are complementary: behavior that is ambiguous in one view may remain distinguishable in the other.

In this paper, we propose TSDETECTOR, a joint spatio-temporal untargeted model poisoning attack detection framework. During each communication round, TSDETECTOR constructs an update distance graph and a knowledge discrepancy graph to characterize spatial relations among clients from parameter and functional perspectives, respectively. The two graphs are adaptively fused to produce a spatial anomaly score. In parallel, a recurrent reconstruction model maintains client specific historical states and evaluates whether each current update is consistent with its preceding trajectory, producing a temporal anomaly score. TSDETECTOR then combines the complementary evidence and removes suspicious updates before global aggregation. The entire detection procedure is performed at the server using received model updates and a small reference dataset, without accessing clients' raw training data. We evaluate TSDETECTOR on five benchmark datasets under five representative untargeted model poisoning attacks and compare it with six existing defenses.

Our main contributions are summarized as follows:

- We introduce a spatio-temporal perspective for model poisoning defense that decomposes malicious client identification into spatial relation analysis within each round and cross round trajectory analysis, rather than relying on anomalies in a single statistical representation.
- We propose TSDETECTOR, a joint spatial-temporal detection framework that combines complementary evidence from relationships among clients and update trajectories across rounds to identify and filter malicious updates.
- We conduct an extensive evaluation across five benchmark datasets and five representative untargeted attacks. Across 25 settings, TSDETECTOR reduces average attack impact to 1.16% and limits the maximum performance degradation to 4.6%, demonstrating reliable malicious client detection and preservation of global model utility.

2 RELATED WORK

2.1 FEDERATED LEARNING

Federated learning (FL) enables N clients to collaboratively train a shared global model under coordination of a central server with decentralized data (McMahan et al., 2017). At round t, each client i downloads current global model w^t and trains it on its local dataset $\mathcal{D}_i$, resulting in w_i^t, and submits its model update $g_i^t = w_i^t - w^t$ to the server, which then update the global model applies an aggregation rule:

$$w^{t+1} = w^t + \eta_t \cdot \mathrm{Agg}\left(\{g_i^t\}_{i=1}^N\right), \tag{1}$$

where η_t denotes the learning rate. Different FL methods mainly differ in their aggregation rules (Fang et al., 2025; Cao et al., 2022; Nguyen et al., 2022; Fung et al., 2020; Xie et al., 2024), and this procedure repeats across communication rounds until the global model converges.

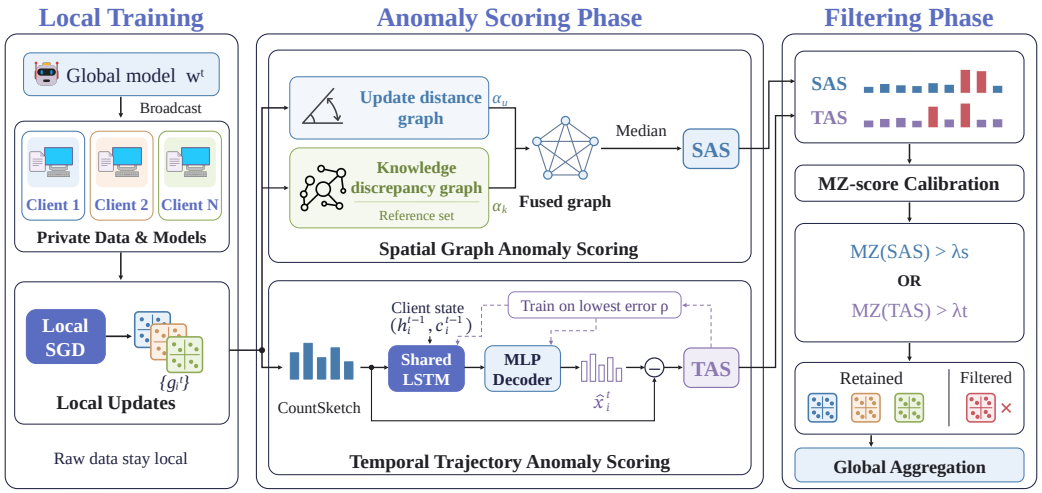


Figure 1: Overview of TSDETECTOR.

2.2 UNTARGETED MODEL POISONING ATTACKS IN FL

Model poisoning attacks allow malicious clients to manipulate global training by submitting adversarial model updates while keeping their local data and optimization procedures hidden from the server. A Little Is Enough (LIE) attack exploits the coordinate variance of benign updates and introduces small perturbations that remain within estimated benign distribution, allowing malicious updates to evade statistical defenses while accumulating harmful effects after aggregation (Baruch et al., 2019). Fang attack formulates model poisoning as an optimization problem that maximizes damage to global model while adapting malicious updates to targeted robust aggregation rule (Fang et al., 2020). Min-Max and Min-Sum maximize the deviation of malicious updates under distance constraints derived from benign updates. Min-Max bounds the maximum distance to benign updates, whereas Min-Sum bounds sum of squared distances (Shejwalkar & Houmansadr, 2021). PoisonedFL maintains a persistent attack direction across communication rounds and dynamically adjusts its magnitude, thereby reducing cancellation of malicious effects without knowledge of benign client updates (Xie et al., 2025). These attacks demonstrate that malicious updates can exploit benign statistical variation, aggregation constraints, and consistency across rounds to evade detection while progressively degrading global model.

2.3 DEFENSES AGAINST MODEL POISONING ATTACKS IN FL

Existing defenses mitigate model poisoning through Byzantine-robust aggregation or explicit malicious client detection. Multi-Krum scores each update according to its distances from neighboring updates and averages those with lowest scores (Blanchard et al., 2017), whereas Trimmed-Mean removes extreme values in each model coordinate before averaging the remaining values (Yin et al., 2018). FLTrust generates a reference update from a small trusted server dataset, assigns trust scores based on directional similarity, and normalizes update magnitudes before weighted aggregation (Cao et al., 2021). FLDetector predicts each client update from historical model information and detects malicious clients using accumulated prediction errors (Zhang et al., 2022). Bant evaluates client updates using a trial loss on server data and combines this signal with trust scores to suppress harmful updates (Molodtsov et al., 2026). Huber obtains a robust aggregate by minimizing a multidimensional Huber loss over client updates (Zhao et al., 2024). Although effective in specific settings, these methods predominantly rely on particular criteria. Their effectiveness may therefore vary across heterogeneous data distributions and attack strategies. In contrast, TSDETECTOR jointly analyzes spatial relations among clients, and client trajectories across rounds, allowing complementary evidence to reduce ambiguities in any individual view.

3 METHODOLOGY

3.1 TSDETECTOR OVERVIEW

TSDETECTOR operates through a spatio-temporal anomaly scoring phase and a complementary evidence filtering phase. In each round, it examines the received local model updates from two complementary perspectives. The spatial perspective quantifies discrepancies among clients in both parameter space and knowledge space, while the temporal perspective assesses the consistency of each client's current update with its historical trajectory. The resulting anomaly scores are then jointly used to filter suspicious clients before server aggregation. The overview of TSDETECTOR is illustrated in Figure 1.

Spatio-temporal anomaly scoring phase. This phase inspects each local model update through complementary spatial and temporal perspectives. Rather than relying on a single update representation, TSDETECTOR combines discrepancies among clients within current round with client specific behavior across communication rounds.

- *Spatial Graph Anomaly scoring.* TSDETECTOR constructs an update distance graph from directional discrepancies between client updates and a knowledge discrepancy graph from their representation topologies induced by server side reference data. Then TSDETECTOR adaptively fuses two graphs according to their anomaly separation strength to derive a spatial anomaly score (SAS) that measures each client's group level deviation.
- *Temporal Trajectory Anomaly scoring.* TSDETECTOR represents each update using CountSketch projections for each layer. A shared LSTM together with client specific historical states are used to models update trajectories across rounds, then computes reconstruction error as temporal anomaly score (TAS), measuring the current update's deviation from historical trajectory.

Complementary evidence filtering phase. This phase is motivated by a key observation: to effectively poison global model, a malicious update typically exhibits anomalous behavior in at least one view, either through abnormal discrepancies among clients within current round or through deviations from historical trajectories across rounds. Accordingly, TSDETECTOR uses MZ-scores to quantify each client's deviation from typical behavior in each view. A update is flagged as suspicious and excluded if its score exceeds corresponding threshold in either view. The remaining updates are then used for global aggregation.

3.2 SPATIAL GRAPH ANOMALY SCORING.

In each round of federated training, the server receives local model updates g_i^t from participating clients. Parameter distance alone is unreliable under heterogeneous data, as benign updates may be dispersed while coordinated malicious updates can form a compact cluster. Therefore, TSDETECTOR constructs an update distance graph and a knowledge discrepancy graph to characterize spatial relations from parameter and functional perspectives.

Knowledge discrepancy graph construction. TSDETECTOR constructs a knowledge topology with a server side reference dataset $\mathcal{D}_{\text{ref}}$ for each client, to characterize the functional discrepancies induced by different client updates. For each participating client, the server first reconstructs its local model w_i^t from submitted update g_i^t and current global model w^t, then feeds $\mathcal{D}_{\text{ref}}$ into w_i^t and extracts their representations from a predefined feature layer. In each knowledge topology $\mathcal{G}$, we define nodes of $\mathcal{G}$ as sample representations, edges as relationships between the samples, and edge weights as the similarity between samples. Then, we model the conditional probability distribution with a cosine similarity based affinity metric for neighborhood graph construction:

$$\mathcal{G} = \left\{ p_{a|b} \;\middle|\; p_{a|b} = \frac{2 - (d_{ab} - \rho_b)}{\sum_{k=1,\, k\neq b}^{|\mathcal{D}_{\text{ref}}|} (2 - (d_{bk} - \rho_b))},\ 0 < a, b \leq |\mathcal{D}_{\text{ref}}| \right\}. \tag{2}$$

where $p_{a|b}$ denotes the conditional probability that the a_{th} sample is a neighbor of the b_{th} sample in the feature space of $\mathcal{G}$. ρ_b represents the distance between the b_{th} data point and its nearest neighbor, whereas d_{ab} denotes cosine distance between the corresponding embeddings of the two samples. By

subtracting ρ_b, we ensure the local connectivity of the manifold, avoid isolated nodes, and thereby better preserve the global structure.

Knowledge discrepancy can be assessed by comparing the topological structures of models on the same input set, we employ the Kullback Leibler divergence to quantify the discrepancy between two knowledge topologies:

$$\mathcal{G}_k^t(i,j) = \sum_a \sum_b \left[p_{a|b} \log \frac{p_{a|b}}{q_{a|b}} + \left(1 - p_{a|b}\right) \log \frac{1 - p_{a|b}}{1 - q_{a|b}} \right], \tag{3}$$

where $\mathcal{G}_k^t$ denotes the knowledge discrepancy graph, $p_{a|b}$ and $q_{a|b}$ denote respective conditional probabilities induced by clients i and j at round t.

Update distance graph construction. The knowledge discrepancy graph characterizes the functional differences among client models, but does not directly reflect differences in their parameter update directions. We therefore construct an update distance graph to provide additional information about relationships among clients in parameter space. Each node represents a client participating in current round, and each edge weight quantifies the directional discrepancy between updates submitted by two clients. Let $\mathbf{g}_i^t$ denote the d-dimensional vectorized model update submitted by client i in round t, we define the edge weight using cosine distance:

$$\mathcal{G}_u^t(i,j) = 1 - \left\langle \mathbf{g}_i^t, \mathbf{g}_j^t \right\rangle / \left(\left\| \mathbf{g}_i^t \right\| \left\| \mathbf{g}_j^t \right\| \right). \tag{4}$$

where $\mathcal{G}_u^t$ denotes the update distance graph.

Spatial anomaly scoring. The knowledge and update graphs capture complementary aspects of client behavior, but their effectiveness may vary across communication rounds. To account for this variation, TSDETECTOR adaptively balances the two graphs according to their ability to distinguish highly deviating clients from the majority.

For graph $v \in \{u, k\}$, the client level discrepancy is summarized by $d_{i,v}^t = \mathrm{TrimMean}_\beta \, \mathcal{G}_v^t(i,j)$, where β denotes the trimming proportion. Let $\mathcal{T}_v$ contain the top r fraction of clients ranked by $d_{i,v}^t$ and $\mathcal{R}_v$ contain the remainder. We quantify the graph discriminative capability as

$$\eta_v = \frac{\dfrac{1}{|\mathcal{T}_v|} \displaystyle\sum_{i \in \mathcal{T}_v} d_{i,v}^t - \dfrac{1}{|\mathcal{R}_v|} \sum_{i \in \mathcal{R}_v} d_{i,v}^t}{\sqrt{\dfrac{1}{|\mathcal{R}_v|} \displaystyle\sum_{i \in \mathcal{R}_v} \left(d_{i,v}^t - \dfrac{1}{|\mathcal{R}_v|} \sum_{j \in \mathcal{R}_v} d_{j,v}^t \right)^2}}. \tag{5}$$

Then, fusion weight of each graph is then computed by $\alpha_v = \lambda + (1 - 2\lambda) \frac{\eta_v}{\eta_u + \eta_k}$, where λ imposes a minimum weight constraint so that neither view is completely discarded.

After normalization $\widetilde{\mathcal{G}}_v^t(i,j) = \mathcal{G}_v^t(i,j) / \max \mathcal{G}_v^t$, the two graphs are fused as $\mathcal{G}_f^t = \alpha_u \widetilde{\mathcal{G}}_u^t + \alpha_k \widetilde{\mathcal{G}}_k^t$.The spatial anomaly score SAS_i^t of client i in communication round t is then defined as

$$SAS_i^t = median_{j \neq i} \mathcal{G}_f^t(i,j). \tag{6}$$

which measures the robust deviation of client i from the dominant client population.

3.3 TEMPORAL TRAJECTORY ANOMALY SCORING.

While SAS_i^t quantifies a client's deviation from participating group within current communication round, but may lose discriminative power under collusive model poisoning attacks. In such attacks, malicious clients deliberately construct mutually similar model updates to reduce pairwise distances within the malicious group. To address this limitation, we further model each client's update trajectory across communication rounds and derive a temporal anomaly score that quantifies deviations from its temporal Trajectory behavior.

Directly modeling temporal evolution of complete high dimensional updates incurs substantial computational and memory overhead, accordingly, for an L-layer DNN trained through federated learning, TSDETECTOR applies layer-wise CountSketch to compress the full model update into a low

dimensional feature representation:

$$x_i^t = \text{Concat}\left(S_1 g_{i,1}^t, \ldots, S_L g_{i,L}^t\right) \in \mathbb{R}^m, \tag{7}$$

where $g_{i,l}^t$ denotes the update of l-th layer for client i at round t, S_l is the CountSketch projection matrix, and m denotes the dimension of compressed feature representation. All projection matrices remain fixed, ensuring a consistent representation for temporal modeling.

The server maintains a LSTM and an MLP decoder shared by all clients. Meanwhile, it independently stores a hidden state $\mathbf{h}_i^t$ and a cell state $\mathbf{c}_i^t$ for each client i. The shared model captures common temporal dynamics across clients, whereas the client-specific states retain their individual historical behavior.

At round t, TSDETECTOR models the temporal consistency of each client by combining its current compressed update $\mathbf{x}_i^t$ with the historical recurrent state. A shared LSTM captures common update dynamics across clients, while client-specific hidden and cell states preserve individual temporal trajectories:

$$\left(\mathbf{h}_i^t, \mathbf{c}_i^t\right) = \text{LSTM}_{\Theta^{t-1}}\left(\mathbf{x}_i^t, \left(\mathbf{h}_i^{t-1}, \mathbf{c}_i^{t-1}\right)\right). \tag{8}$$

The temporal representation is further decoded to reconstruct the current update feature:

$$\widehat{\mathbf{x}}_i^t = D_{\Psi^{t-1}}\left(\mathbf{h}_i^t\right), \tag{9}$$

where Θ^{t-1} and Ψ^{t-1} denote the LSTM and decoder parameters in round $t-1$. We measure temporal anomaly score using the reconstruction error:

$$TAS_i^t = \left\|\mathbf{x}_i^t - \widehat{\mathbf{x}}_i^t\right\|_2^2. \tag{10}$$

A larger TAS_i^t indicates a stronger deviation from the temporal dynamics learned from historical client behavior.

To reduce the influence of anomalous updates, TSDETECTOR updates the temporal model using only the fraction ρ of clients with the smallest reconstruction errors. This selective adaptation allows the model to track normal temporal evolution while limiting contamination from anomalous trajectories.

3.4 COMPLEMENTARY EVIDENCE FILTERING.

SAS and TAS quantify a client's spatial deviation from current round and temporal deviation from historical trajectory, respectively. The two views provide complementary detection coverage, filtering a client only when both scores exceed their thresholds may overlook attacks exposed in single view. TSDETECTOR use MZ-scores to flag a client as malicious. Specifically, given a set of values $X = \{x_1, \ldots, x_n\}$ with median $med(X)$ and standard deviation σ, we define the MZ-score of each element $x_i \in X$ as

$$\lambda_i = \frac{x_i - med(X)}{\sigma}. \tag{11}$$

The MZ-score measures the deviation of an element from the median in units of standard deviation. In TSDETECTOR, the MZ-scores for TAS and SAS values are computed for each local update. Updates with large MZ-scores are regarded as outliers and filtered using thresholds λ_t and λ_s, enabling adaptive outlier detection under varying score distributions. After filtering, the server aggregates remaining updates to obtain updated global model.

4 EXPERIMENTAL EVALUATION

4.1 EXPERIMENTAL SETUP

Dataset: We use MNIST (Lecun et al., 1998), Fashion-MNIST (Xiao et al., 2017), CIFAR-10, CIFAR-100 (Krizhevsky, 2009), and Tiny-ImageNet (Le & Yang, 2015) as evaluation datasets, which are widely used in prior works. Further details on the datasets and model architectures are provided in Appendix C.2. To simulate non-IID FL scenario, we partition the training data among clients using a Dirichlet distribution parameterized by α, where a smaller α indicates stronger data heterogeneity. We set $\alpha = 0.1$ by default and investigate its impact in Figure 3.

Table 1: Attack impact of five model poisoning attacks under TSDETECTOR and baseline defenses on non-IID datasets ($\alpha = 0.1$; 20% malicious clients). Results are shown in %.

Dataset	Defense	LIE	Fang	Min-Max	Min-Sum	PoisonedFL
MNIST	FLTrust	1.23	0.75	0.68	0.63	1.94
	FLDetector	6.16	8.23	72.57	73.92	5.73
	Bant	0.61	0.79	0.93	0.92	8.83
	Huber	0.08	1.36	8.25	7.51	0.72
	Trimmed Mean	0.84	3.81	5.71	4.93	0.39
	Multi-Krum	0.03	0.07	87.36	4.33	1.07
	TSDETECTOR	0.29	0.00	1.01	0.66	3.29
Fashion-MNIST	FLTrust	3.29	1.64	1.64	1.79	4.42
	FLDetector	25.86	13.07	70.43	59.48	19.43
	Bant	0.09	0.07	0.63	0.57	24.24
	Huber	1.81	1.88	10.18	9.71	0.01
	Trimmed Mean	4.72	6.51	6.43	6.43	0.66
	Multi-Krum	0.29	0.43	66.27	2.91	1.71
	TSDETECTOR	3.61	0.03	2.65	2.38	2.25
CIFAR-10	FLTrust	8.13	9.31	9.31	9.31	0.18
	FLDetector	17.11	14.50	12.17	15.01	13.33
	Bant	10.35	2.54	6.23	0.59	0.46
	Huber	1.11	0.24	0.22	0.33	15.00
	Trimmed Mean	12.30	20.97	22.02	10.31	15.09
	Multi-Krum	12.50	1.46	6.19	4.29	14.39
	TSDETECTOR	4.49	2.84	1.26	0.51	0.26
CIFAR-100	FLTrust	15.93	26.00	0.35	0.62	0.17
	FLDetector	22.84	8.62	27.72	27.82	6.78
	Bant	10.00	23.14	0.01	0.06	0.84
	Huber	8.39	23.03	0.31	0.45	0.71
	Trimmed Mean	16.70	25.42	0.65	0.01	3.83
	Multi-Krum	5.54	5.08	5.20	5.26	3.27
	TSDETECTOR	0.62	0.38	0.19	0.21	0.38
Tiny-ImageNet	FLTrust	9.66	28.47	0.23	0.02	0.18
	FLDetector	26.68	15.35	28.63	28.74	19.16
	Bant	16.32	26.03	2.52	2.79	2.51
	Huber	12.36	18.45	0.30	0.26	0.14
	Trimmed Mean	21.22	29.55	4.58	5.04	1.00
	Multi-Krum	23.93	1.24	0.89	0.98	0.24
	TSDETECTOR	0.28	0.27	0.52	0.26	0.25

FL training settings: Following prior works (Fang et al., 2020; Cao et al., 2021), we simulate a cross-silo FL system with 100 clients. We employ a CNN as the global model for MNIST and Fashion-MNIST, ResNet-9 for CIFAR-10, and ResNet-18 (He et al., 2016) for Tiny-ImageNet and CIFAR-100. Default hyper-parameter settings are provided in Appendix C.5.

Evaluated defenses and attacks methods: We compare our method with six representative defenses, including FLTrust (Cao et al., 2021), FLDetector (Zhang et al., 2022), Bant (Molodtsov et al., 2026), Huber (Zhao et al., 2024), Trimmed Mean (Yin et al., 2018), and Multi-Krum (Blanchard et al., 2017). We consider five representative model-poisoning attacks, including LIE (Baruch et al., 2019), Fang (Fang et al., 2020), Min-Max and Min-Sum (Shejwalkar & Houmansadr, 2021), and PoisonedFL (Xie et al., 2025). Further details of those methods are provided in Appendix C.3 and C.4. By default, the malicious client fraction is set to 20%, indicating that 20% of the clients perform attacks in each round. We further vary this fraction to evaluate robustness of TSDETECTOR in Figure 2.

Evaluation metrics: We consider evaluation metrics for both the learned global model and detection. To evaluate the learned global model, we use testing accuracy (TACC), which is the fraction of testing examples correctly classified by the global model, we further quantify attack impact as the decrease in TACC relative to the no attack reference aggregates via FedAvg. For detection, we use detection accuracy (DACC), false-positive rate (FPR), and false-negative rate (FNR). DACC is the fraction of clients correctly classified as benign or malicious. FPR (or FNR) is the fraction of benign (or malicious) clients incorrectly classified as malicious (or benign).

TSDETECTOR settings. For spatial perspective, the server maintains an unlabeled reference set containing 100 samples by default. We set the candidate anomaly ratio to $r = 0.2$ and the minimum weight assigned to each graph during adaptive fusion to $\lambda = 0.25$. For temporal perspective, the server maintains a shared LSTM and an MLP decoder, and the provisional trusted client ratio is set to $\rho = 0.7$. To accumulate sufficient trajectory information, temporal perspective uses five rounds to warm up before starting detection. The default filtering thresholds are set as $\lambda_t = 1.5$ and $\lambda_s = 1.5$. Dataset specific configurations of temporal module are provided in Appendix C.5.

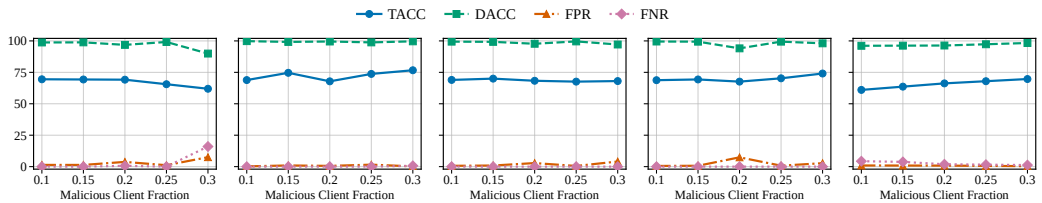


Figure 2: Impact of the malicious client proportion on TSDETECTOR under five attacks on CIFAR-10 with Dirichlet concentration $\alpha = 0.5$.

4.2 EXPERIMENTAL RESULTS

Reduced attack impact. We evaluate TSDETECTOR against five state-of-the-art model poisoning attacks and compare it with six representative defenses. Table 1 summarizes the attack impact under different defenses across five non-IID datasets. Overall, TSDETECTOR consistently limits attack induced performance degradation across diverse attacks and datasets. Its average attack impact over 25 experimental settings is 1.16%, with 16 settings below 1.0% and all settings below 4.5%. Although TSDETECTOR does not achieve the lowest attack impact in every individual setting, these results indicate that its effectiveness is not limited to a particular poisoning strategy or dataset.

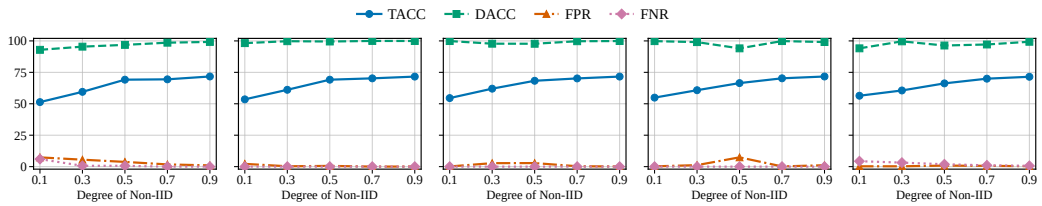
In contrast, several baseline defenses perform well against particular attacks but exhibit considerable performance variation across attacks and datasets. For example, Multi-Krum achieves an attack impact of only 0.03% under LIE on MNIST, but its impact increases to 87.36% under Min-Max on the same dataset. FLDetector similarly incurs an attack impact of 72.57% under Min-Max, whereas TSDETECTOR limits the impact to 1.01%. Similar variations are observed on more complex datasets, Under Fang on Tiny-ImageNet, FLTrust, Bant, and Trimmed Mean incur attack impacts of 28.47%, 26.03%, and 29.55%, respectively, compared with 0.27% for TSDETECTOR. Likewise, Huber achieves a low attack impact of 0.24% under Fang on CIFAR-10, but its impact increases to 23.03% under the same attack on CIFAR-100, while TSDETECTOR maintains a low impact of 0.38%.

These results suggest that relying on single perspective for anomaly detection may introduce detection blind spots, as some malicious updates can appear similar to benign ones under a particular criterion. TSDETECTOR examines model updates from two complementary perspectives, such that anomalies that are less distinguishable in one perspective can still be captured in the other. By leveraging these complementary signals, TSDETECTOR reduces its reliance on specific attack characteristics, enabling effective defense across diverse poisoning strategies.

Impact of the malicious client fraction. Figure 2 shows the performance of TSDETECTOR on CIFAR-10 as the malicious-client fraction varies from 10% to 30%, with $\alpha = 0.5$. We observe that the global-model accuracy remains generally stable as the malicious-client fraction increases. Meanwhile, TSDETECTOR maintains high DACC and low FPR and FNR across most settings, demonstrating its effectiveness under different levels of malicious participation.

Impact of non-IID degree of data distribution. Figure 3 reports the performance of TSDETECTOR on CIFAR-10 as the Dirichlet concentration α varies from 0.1 to 0.9, with 20% malicious clients. As α increases, TACC improves under all five attacks, reflecting the reduced difficulty of federated optimization under weaker data heterogeneity. Meanwhile, TSDETECTOR maintains high DACC and generally low FPR and FNR across the evaluated values of α, demonstrating its effectiveness under different degrees of data heterogeneity.

Ablation study of TSDETECTOR. We examine the contribution and complementarity of the temporal perspective (T), knowledge diatance graph (K), and update distance graph (U). We consider both single component variants and variants obtained by removing one component from the full framework. Table 2 reports their FPR/FNR under five attacks, where lower values indicate better detection.



(a) LIE (b) Fang (c) Min-Max (d) Min-Sum (e) PoisonedFL

Figure 3: Impact of the non-iid degree on TSDetector under five attacks on CIFAR-10 with a malicious client proportion of 20%.

Table 2: Detection performance of different components in TSDETECTOR. Results are shown in %.

Variant	Component			Attack (FPR/FNR)				
	T	K	U	Fang	LIE	Min-Max	Min-Sum	PoisonedFL
T only	✓			0.45/40.10	0.27/88.00	0.35/3.33	0.29/7.33	3.88/2.67
K only		✓		2.88/22.00	0.53/0.00	17.03/79.33	8.61/62.67	6.40/87.00
U only			✓	0.62/18.67	9.93/83.10	0.25/1.33	0.27/0.67	5.08/97.33
TSDETECTOR w/o T		✓	✓	0.30/0.00	0.87/2.67	0.85/8.00	0.34/2.00	6.25/94.67
TSDETECTOR w/o U	✓	✓		1.35/0.00	1.62/0.00	8.59/4.00	7.04/8.00	8.46/6.67
TSDETECTOR w/o K	✓		✓	2.00/0.00	18.57/88.67	0.51/0.00	0.72/0.00	9.60/7.33
TSDETECTOR	✓	✓	✓	2.20/0.00	3.36/3.00	0.15/0.00	0.22/0.00	0.67/3.16

The ablation results demonstrate that temporal and spatial evidence capture different attack behaviors. The temporal perspective is particularly important for PoisonedFL, whose persistent behavior across communication rounds is difficult to identify using spatial relations from a single round. Removing the temporal perspective increases the FNR under PoisonedFL to 94.67%, whereas the full model maintains an FNR of 4.33%. In contrast, using temporal perspective alone produces FNRs of 40.10% and 88.00% under Fang and LIE, showing that spatial relations are essential for identifying abnormal relationships among clients. Combining the two perspectives therefore avoids the detection failures observed when either source of evidence is used alone.

The two spatial graphs also capture different forms of abnormality. Removing knowledge discrepancy graph increases the FNR under LIE to 88.67%, compared with 1.00% for the full model, because malicious updates with subtle parameter deviations may still alter the representation structures learned by local models. The update distance graph contributes more strongly against attacks that manipulate update geometry. For example, removing this graph increases FNR under Min-Sum from 0.00% to 8.00%. Their joint use therefore provides a more complete characterization of spatial anomalies and improves robustness across different poisoning strategies.

Overall, TSDETECTOR achieves a worst case FPR of 3.36% and a worst case FNR of 4.33% across all five attacks. Although individual variants can perform well against specific attacks, none provides consistently low error rates across all settings. These results support the joint use of temporal trajectories, knowledge topologies, and update distances for robust malicious-client detection.

5 CONCLUSION

In this paper, we proposed TSDETECTOR, a joint spatio-temporal model poisoning attacks detection framework in heterogeneous federated learning. Its spatial perspective integrates update distance and knowledge discrepancy graphs to identify abnormal relations among clients, while its temporal perspective models client specific trajectories to capture persistent cross round deviations. Together, these complementary signals reduce reliance on any single anomaly criterion and enable robust detection without accessing clients' raw data. Experiments across five datasets and five representative attacks show that TSDETECTOR consistently limits attack induced performance degradation and maintains reliable detection under varying data heterogeneity and malicious client proportions.

6 AI USE STATEMENT

During the preparation of this manuscript, LLM was used for writing assistance, literature discovery, and research support. For writing assistance, the LLM was used to rephrase sentences, check grammar, improve readability, and refine the organization and presentation of the manuscript. For literature discovery, the LLM was used to suggest search keywords, identify potentially relevant research directions, and provide candidate related works. The authors independently accessed and examined the original publications and manually verified the relevance and accuracy of all cited references. No reference was included solely on the basis of LLM generated information. For research ideation and execution, the LLM was used to discuss possible methodological formulations, experimental settings, and ablation analyses, and to assist with debugging. All methodological decisions were made by the authors. The authors independently implemented the proposed method, conducted the experiments, analyzed the results, and verified all conclusions. The LLM did not generate, modify, or fabricate experimental results. The authors reviewed all LLM assisted content and take full responsibility for the accuracy, originality, and integrity of the manuscript.

REFERENCES

Gilad Baruch, Moran Baruch, and Yoav Goldberg. A little is enough: Circumventing defenses for distributed learning. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong. FLTrust: Byzantine-robust federated learning via trust bootstrapping. In *28th Annual Network and Distributed System Security Symposium (NDSS 2021)*. The Internet Society, 2021. doi: 10.14722/ndss.2021.24434.

Xiaoyu Cao, Zaixi Zhang, Jinyuan Jia, and Neil Zhenqiang Gong. FLCert: Provably secure federated learning against poisoning attacks. *IEEE Transactions on Information Forensics and Security*, 17: 3691–3705, 2022. doi: 10.1109/TIFS.2022.3212174.

El-Mahdi El-Mhamdi, Rachid Guerraoui, and Sébastien Rouault. The hidden vulnerability of distributed learning in Byzantium. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 3521–3530. PMLR, 2018. URL https://proceedings.mlr.press/v80/mhamdi18a.html.

Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Gong. Local model poisoning attacks to Byzantine-Robust federated learning. In *29th USENIX Security Symposium (USENIX Security 20)*, pp. 1605–1622. USENIX Association, aug 2020. ISBN 978-1-939133-17-5. URL https://www.usenix.org/conference/usenixsecurity20/presentation/fang.

Minghong Fang, Seyedsina Nabavirazavi, Zhuqing Liu, Wei Sun, Sundararaja Sitharama Iyengar, and Haibo Yang. Do we really need to design new byzantine-robust aggregation rules? In *32nd Annual Network and Distributed System Security Symposium (NDSS 2025)*. The Internet Society, 2025. doi: 10.14722/ndss.2025.241796.

Clement Fung, Chris J. M. Yoon, and Ivan Beschastnikh. The limitations of federated learning in sybil settings. In *23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020)*, pp. 301–316, San Sebastian, October 2020. USENIX Association. ISBN 978-1-939133-18-2. URL https://www.usenix.org/conference/raid2020/presentation/fung.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90.

Ahmed Imteaj, Khandaker Mamun Ahmed, Urmish Thakker, Shiqiang Wang, Jian Li, and M. Hadi Amini. Federated learning for resource-constrained IoT devices: Panoramas and state of the art. In Roozbeh Razavi-Far, Boyu Wang, Matthew E. Taylor, and Qiang Yang (eds.), *Federated and Transfer Learning*, volume 27 of *Adaptation, Learning, and Optimization*, pp. 7–27. Springer, Cham, 2023. doi: 10.1007/978-3-031-11748-0_2.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Ya Le and Xuan S. Yang. Tiny imagenet visual recognition challenge. CS231n course project report, Stanford University, 2015. URL https://cs231n.stanford.edu/reports/2015/pdfs/yle_project.pdf.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Saeed Mahloujifar, Mohammad Mahmoody, and Ameer Mohammed. Universal multi-party poisoning attacks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4274–4283. PMLR, 2019. URL https://proceedings.mlr.press/v97/mahloujifar19a.html.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In Aarti Singh and Jerry Zhu (eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282. PMLR, 2017. URL https://proceedings.mlr.press/v54/mcmahan17a.html.

Gleb Molodtsov, Daniil Medyakov, Sergey Skorik, Nikolas Khachaturov, Shahane Tigranyan, Vladimir Aletov, Aram Avetisyan, Martin Takáč, and Aleksandr Beznosikov. Bant: Byzantine antidote via trial function and trust scores. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 24431–24440, 2026. doi: 10.1609/aaai.v40i29.39625.

Thien Duc Nguyen, Phillip Rieger, Huili Chen, Hossein Yalame, Helen Möllering, Hossein Fereidooni, Samuel Marchal, Markus Miettinen, Azalia Mirhoseini, Shaza Zeitouni, Farinaz Koushanfar, Ahmad-Reza Sadeghi, and Thomas Schneider. FLAME: Taming backdoors in federated learning. In *31st USENIX Security Symposium (USENIX Security 22)*, pp. 1415–1432, Boston, MA, August 2022. USENIX Association. ISBN 978-1-939133-31-1. URL https://www.usenix.org/conference/usenixsecurity22/presentation/nguyen.

Virat Shejwalkar and Amir Houmansadr. Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In *28th Annual Network and Distributed System Security Symposium (NDSS 2021)*. The Internet Society, 2021. doi: 10.14722/ndss.2021.24498.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017. URL https://arxiv.org/abs/1708.07747.

Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. Fall of empires: Breaking byzantine-tolerant SGD by inner product manipulation. In Ryan P. Adams and Vibhav Gogate (eds.), *Proceedings of the 35th Uncertainty in Artificial Intelligence Conference*, volume 115 of *Proceedings of Machine Learning Research*, pp. 261–270. PMLR, 2020. URL https://proceedings.mlr.press/v115/xie20a.html.

Yueqi Xie, Minghong Fang, and Neil Zhenqiang Gong. FedREDefense: Defending against model poisoning attacks for federated learning using model update reconstruction error. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 54460–54474. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/xie24c.html.

Yueqi Xie, Minghong Fang, and Neil Zhenqiang Gong. Model poisoning attacks to federated learning via multi-round consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15454–15463, June 2025.

Gang Yan, Hao Wang, Xu Yuan, and Jian Li. Defl: Defending against model poisoning attacks in federated learning via critical learning periods awareness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 10711–10719, 2023. doi: 10.1609/aaai.v37i9.26271.

Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5650–5659. PMLR, 2018. URL https://proceedings.mlr.press/v80/yin18a.html.

Zaixi Zhang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. FLDetector: Defending federated learning against model poisoning attacks via detecting malicious clients. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2545–2555. Association for Computing Machinery, 2022. doi: 10.1145/3534678.3539231.

Puning Zhao, Fei Yu, and Zhiguo Wan. A Huber loss minimization approach to byzantine robust federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 21806–21814, 2024. doi: 10.1609/aaai.v38i19.30181.