Untargeted attack against federated recommendation systems via poisonous item embeddings and the defense

Abstract

- **Context & Issue**: Federated recommendation systems (FedRec) can train personalized recommenders without user data collection. However, the decentralized nature of FedRec makes them prone to poisoning attacks, specifically untargeted attacks that aim to degrade overall system performance.
- **Novel Contributions**:
  - **ClusterAttack**: A new untargeted attack method that converges item embeddings into dense clusters, affecting the recommender's score generation and ranking order.
  - **UNION Defense Mechanism**: regularizes the item embeddings toward a uniform distribution in the space with a contrastive learning task.This method enables the server to filter out malicious gradients based on the uniformity of updated item embeddings.
- **Results**: ClusterAttack can effectively downgrade FedRec systems' performance and evade many defense methods. In contrast, UNION enhances the system's resilience to various untargeted attacks, including ClusterAttack.

Introduction

- **Challenge**: FL's vulnerability to poisoning attacks, where attackers can modify local training data or gradients maliciously.
- **Focus**: **Previous studies largely concentrated on targeted attacks to promote certain items, but untargeted attacks aiming to degrade overall system**

performance were less explored. Such attacks can significantly disrupt user experience and cause financial losses.

ClusterAttack

- **Objective**: To impair the overall performance of FedRec systems by targeting item embeddings.
- **Methodology**:
  - **Adaptive Clustering**: The attack employs k-means clustering to divide item embeddings into clusters and calculates a loss function to minimize within-cluster variance.
  - **Stealthy Gradients**: It uses clipped malicious gradients, estimating the norm of normal item embedding gradients to remain undetected.
- **Impact**: ClusterAttack leads to similar recommendation scores for items in the same cluster, disrupting the ranking order and the system's overall effectiveness.
- **Loss Function**: A critical component of Adaptive Clustering is the calculation of a loss function to measure and minimize the within-cluster variance.
- **Objective of Minimization**: The aim is to make the embeddings within each cluster as close as possible to their respective centroid. This minimization leads to the creation of tightly packed, dense clusters of item embeddings.

UNION Defense Mechanism

- **Purpose**: To counteract attacks like ClusterAttack.
- **Strategy**: Implements a uniformity-based defense by **regularizing item embeddings towards a uniform distribution**.
- **Client-Side Implementation**: Requires benign clients to incorporate a **contrastive learning task** into their local model training.
- **Detection of Malicious Gradients**: The server detects and filters out gradients that cause abnormal distribution of item embeddings.

Detecting and Filtering Malicious Gradients

- **Server's Role**: The server plays a crucial role in the UNION defense mechanism. After each training round, when clients send their updated model parameters (including item embeddings) to the server, the server analyzes these embeddings.
- **Uniformity Analysis**: The server checks for the uniformity of the distribution of item embeddings. Since UNION ensures that under normal conditions, these embeddings should be uniformly distributed, any significant deviation from this uniformity can be a red flag, indicating potential tampering or attack.
- **Filtering Process**: If the server detects item embeddings that deviate from the expected uniform distribution, it can infer that these embeddings might have been manipulated by an attacker. The server can then filter out or disregard these suspicious gradients, preventing them from corrupting the global model.

Detail

Algorithm 1: Adaptive Clustering

Input: Number of clusters K, range of number of clusters $[K_{\min}, K_{\max}]$, threshold R, and decay rate β.

Init: Set $\tilde{\mathcal{L}}_{\text{attack}}^{(0)}, n_{\text{inc}}, n_{\text{dec}}$ and t as 0.

// Repeat after each round of attack

$t \leftarrow t + 1$;

Calculate $\mathcal{L}_{\text{attack}}^{(t)}$ with Equation (2);

$\tilde{\mathcal{L}}_{\text{attack}}^{(t)} \leftarrow \beta \cdot \tilde{\mathcal{L}}_{\text{attack}}^{(t-1)} + (1-\beta) \cdot \mathcal{L}_{\text{attack}}^{(t)}$;

$\hat{\mathcal{L}}_{\text{attack}}^{(t)} \leftarrow \tilde{\mathcal{L}}_{\text{attack}}^{(t)} / \left(1-\beta^t\right)$;

if $\hat{\mathcal{L}}_{\text{attack}}^{(t)} > \hat{\mathcal{L}}_{\text{attack}}^{(t-1)}$ **then** $n_{\text{inc}} \leftarrow n_{\text{inc}} + 1$;

else $n_{dec} \leftarrow n_{\text{dec}} + 1$;

if $n_{\text{inc}} - n_{\text{dec}} \geqslant R$ **then**

  $K \leftarrow \min\left(\left\lfloor K + \sqrt{K_{\max} - K} \right\rfloor, K_{\max}\right)$;

  Reset $n_{\text{inc}}, n_{\text{dec}}$ and t as 0;

end if

if $n_{\text{dec}} - n_{\text{inc}} \geqslant R$ **then**

  $K \leftarrow \max\left(\left\lfloor K - \sqrt{K - K_{\min}} \right\rfloor, K_{\min}\right)$;

  Reset $n_{\text{inc}}, n_{\text{dec}}$ and t as 0;

end if

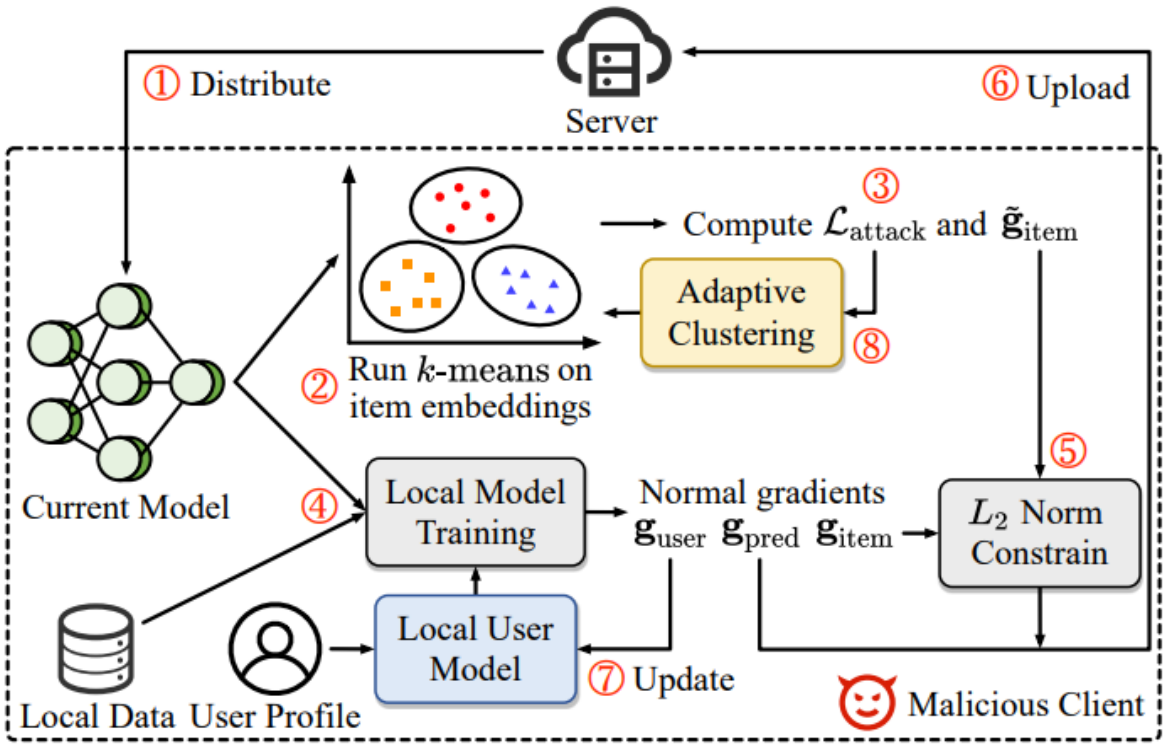

Figure 1: The procedure of our *ClusterAttack*.

When selected for model training, the malicious client receives the latest global model from the server, which contains the item embeddings $\{\boldsymbol{v}_i\}_{i=1}^{M}$ (Step 1). We first apply k-means (Lloyd 1982) to split them into K clusters $\{C_i\}_{i=1}^{K}$ with centroids $\{\boldsymbol{c}_i\}_{i=1}^{K}$ (Step 2). Then we compute the following loss function to measure the within-cluster variance:

$$\mathcal{L}_{\text{attack}} = \sum_{i=1}^{K} \sum_{\boldsymbol{v}_j \in C_i} \|\boldsymbol{v}_j - \boldsymbol{c}_i\|_2^2 . \tag{2}$$

The malicious gradient of each item embedding is computed to minimize the above attack loss, i.e., $\tilde{\mathbf{g}}_{\boldsymbol{v}_i} = \partial \mathcal{L}_{\text{attack}} / \partial \boldsymbol{v}_i$ (Step 3). To make our attack stealthier, we further clip $\tilde{\mathbf{g}}_{\boldsymbol{v}_i}$ with an estimated norm of normal item embedding gradients. Specifically, for each malicious client $u^{(j)} \in \mathcal{U}_{\text{mal}}$, we compute the normal gradient with the original loss function $\mathcal{L}_{\text{rec}}$ and his local training data (Step 4). Then we calculate the mean μ and standard deviation σ of the L_2 norms of all normal item embedding gradients. Assuming these norms follow a Gaussian distribution, we generate a reasonable norm bound $b_i^{(j)} = \mu + \lambda_i^{(j)} \sigma$ for each item embedding $\boldsymbol{v}_i$ on the malicious client $u^{(j)}$, where $\lambda_i^{(j)}$ is a number randomly sampled from $[0, 3]$. Therefore, the clipped malicious item embedding gradients are formulated as follows:

$$\hat{\mathbf{g}}_{\boldsymbol{v}_i}^{(j)} = \frac{\tilde{\mathbf{g}}_{\boldsymbol{v}_i}}{\max\left(1, \|\tilde{\mathbf{g}}_{\boldsymbol{v}_i}\|_2 / b_i^{(j)}\right)} . \tag{3}$$

The malicious gradient of the item model is set as $\hat{\mathbf{g}}_{\text{item}}^{(j)} = \left[\hat{\mathbf{g}}_{\boldsymbol{v}_1}^{(j)}; \hat{\mathbf{g}}_{\boldsymbol{v}_2}^{(j)}; \cdots; \hat{\mathbf{g}}_{\boldsymbol{v}_M}^{(j)}\right]$. Finally, the malicious client uploads

server side:

- measure the uniformity in terms of the average L2 of two item embeddings.adopt the Gap Statistics algorithm to estimate the number of clusters in the set of estimated uniformity.If the algorithm estimates that there is more than one cluster,we believe there are some malicious gradients that lead to abnormally distributed item embeddings. Hence, we further apply k-means to split {di}n i=1 into two clusters and filter
  out all the gradients belonging to the minor one.

Experiments

Research Questions: The study aimed to evaluate the performance of ClusterAttack and UNION, understand their capabilities to bypass or withstand existing attack and defense methods, and assess the impact of various factors like the ratio of malicious clients and the effectiveness of adaptive clustering in ClusterAttack.

- **Datasets and Settings**: The experiments were conducted using two public datasets to answer the outlined research questions.

Conclusion

- **Summary of Findings**: The paper presented ClusterAttack, an effective method for untargeted poisoning attacks on FedRec systems, and UNION, a defense mechanism that counters such attacks.
- **Significance**: The study highlights the security risks in FedRec systems and offers insights into developing robust defense mechanisms against untargeted poisoning attacks.

This detailed summary captures the key aspects of the paper, providing a comprehensive understanding of the research, its methodology, and its significance in the context of FedRec systems.