# Untargeted Attack against Federated Recommendation Systems via Poisonous Item Embeddings and the Defense

**Yang Yu[1,2], Qi Liu[1,2*], Likang Wu[1,2], Runlong Yu[1,2], Sanshi Lei Yu[1,2], Zaixi Zhang[1,2]**

[1]Anhui Province Key Laboratory of Big Data Analysis and Application, University of Science and Technology of China

[2]State Key Laboratory of Cognitive Intelligence

{yflyl613, wulk, yrunl, zaixi}@mail.ustc.edu.cn, qiliuql@ustc.edu.cn, meet.leiyu@gmail.com

Paper Code

## Introduction

### Background

- Most existing recommenders are trained on centralized user data, which has the risk of data leakage and raises privacy concerns.
- Several studies have applied federated learning (FL) to train privacy-preserving federated recommendation (FedRec) systems.
- Unfortunately, FL is known to be vulnerable to poisoning attacks.
- The untargeted attack that aims to degrade the overall performance of the FedRec system and its defense remains less explored.

### Challenges

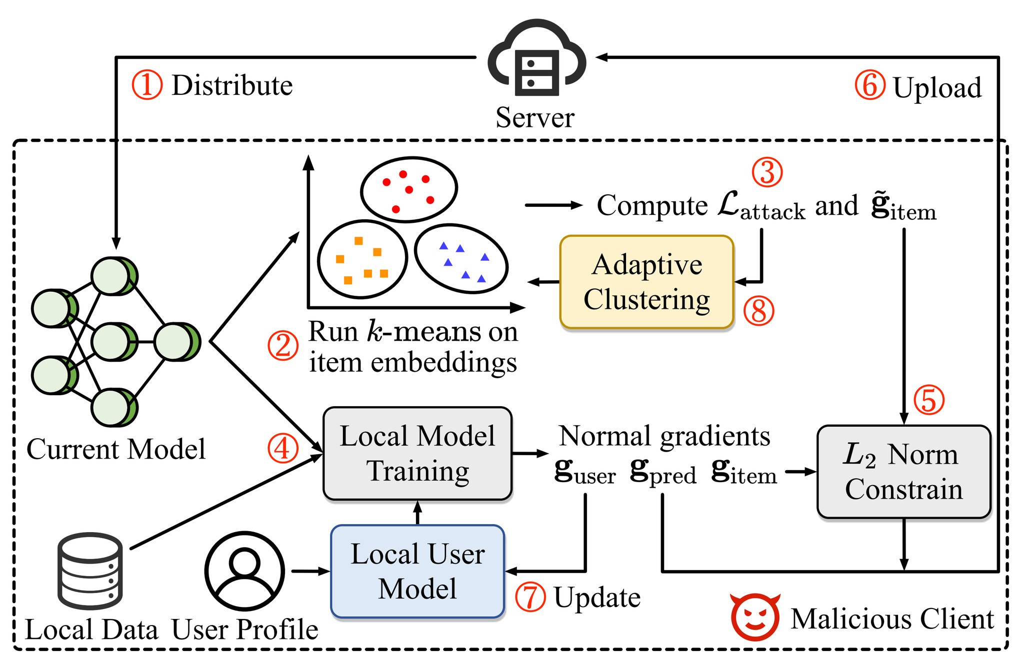
- The attack method must be effective even with a small fraction of malicious clients.
- The attacker can only access a small set of data stored on the malicious clients.
- The attack method needs to manipulate the model output on arbitrary inputs.
- Many recommenders are naturally robust to malicious perturbation to a certain degree since they are trained on implicit user feedback with heavy noise.

## Attack: *ClusterAttack*

### Main Idea

- Upload malicious gradients that converge item embeddings into several dense clusters.
- The recommender tends to generate similar scores for these close items in the same cluster and mess up the ranking order.

### Attack Procedure (See the paper for details)




- Apply $k$-means to split the item embeddings $\{\boldsymbol{v}_i\}_{i=1}^{M}$ into $K$ clusters $\{C_i\}_{i=1}^{K}$ with centroids $\{\boldsymbol{c}_i\}_{i=1}^{K}$.
- Compute the within-cluster variance and the malicious item embedding gradient.

$$\mathcal{L}_{\text{attack}} = \sum_{i=1}^{K} \sum_{\boldsymbol{v}_j \in C_i} \|\boldsymbol{v}_j - \boldsymbol{c}_i\|_2^2 \qquad \tilde{\mathbf{g}}_{\boldsymbol{v}_i} = \frac{\partial \mathcal{L}_{\text{attack}}}{\partial \boldsymbol{v}_i}$$

- Compute the normal gradient $[\mathbf{g}_{\text{item}}; \mathbf{g}_{\text{user}}; \mathbf{g}_{\text{pred}}]$ of each malicious client.
- Clip the malicious gradient with an estimated norm of normal item embedding gradients.
- Upload $\mathbf{g}_{\text{pred}}$ and the clipped malicious item embedding gradient $\hat{\mathbf{g}}_{\text{item}}$ to the server, and update the local user model with $\mathbf{g}_{\text{user}}$.
- Adjust the number of clusters $K$ with the **adaptive clustering mechanism** based on $\mathcal{L}_{\text{attack}}$ after each round of attack.

## Defense: *UNION*

### Client Side

- Train the local recommendation model with an **additional contrastive learning task**.
- Denote the item set interacted by the user as $\mathcal{V}_u = \{v_i\}_{i=1}^{L}$ and the entire item set as $\mathcal{V}$.
- For each $v_i \in \mathcal{V}_u$, randomly select another positive item $v_i^+ \in \mathcal{V}_u$ and $P$ negative items $\{v_i^-\}_{i=1}^{P} \subseteq \mathcal{V} \backslash \mathcal{V}_u$.

$$\mathcal{L}_{\text{cl}} = -\sum_{i=1}^{L} \log \frac{e^{f(v_i)^\top f(v_i^+)}}{e^{f(v_i)^\top f(v_i^+)} + \sum_{j=1}^{P} e^{f(v_i)^\top f(v_j^-)}}$$

- $\mathcal{L}_{\text{cl}}$ can regularize the item embedding toward a uniform distribution in the space while training with the recommendation task.

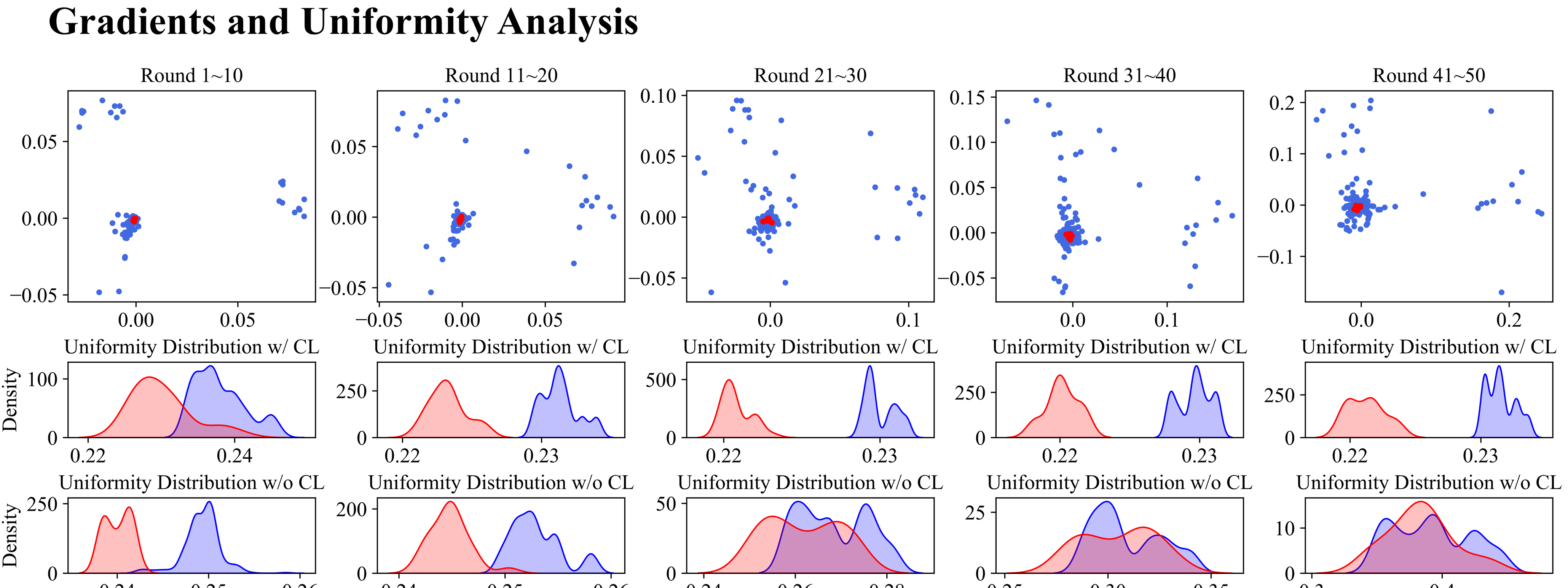
### Server Side

- Estimate the uniformity of updated item embeddings for each received gradient.

$$d_i = \mathbb{E}_{x,y \overset{\text{i.i.d}}{\sim} p_{\text{data}}} \|f(x) - f(y)\|_2^2$$

- Use the **Gap Statistics algorithm** to estimate the number of clusters in the set of estimated uniformity $\{d_i\}_{i=1}^{n}$.
- If the algorithm estimates that there is more than one cluster, we apply $k$-means to split $\{d_i\}_{i=1}^{n}$ into two clusters and filter out all the gradients belonging to the minor one.

### Gradients and Uniformity Analysis




Visualization of the uploaded gradients and the uniformity distribution in different rounds of training (blue: benign clients, red: malicious clients).

## Experiments

### Datasets

- MovieLens-1M: a public movie recommendation dataset.
- Gowalla: a public check-in dataset obtained from the Gowalla website.

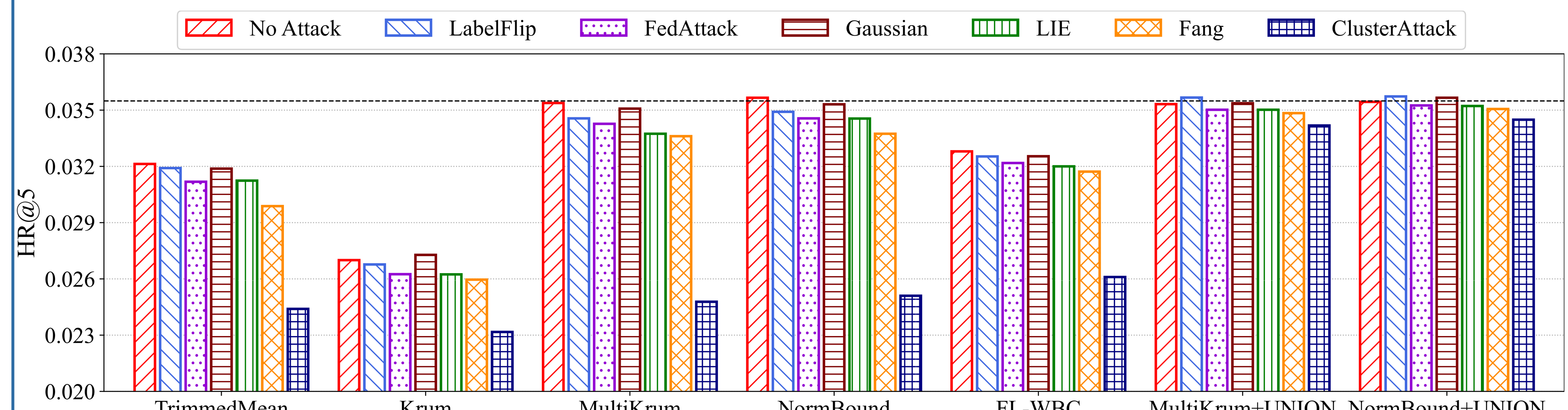
| Dataset | #Users | #Items | #Actions | Avg. length | Density |
|---|---|---|---|---|---|
| ML-1M | 6,040 | 3,706 | 1,000,209 | 165.6 | 4.47% |
| Gowalla | 29,858 | 40,981 | 1,585,043 | 53.1 | 0.13% |

Detailed statistics of the two datasets.

### Performance Comparisons

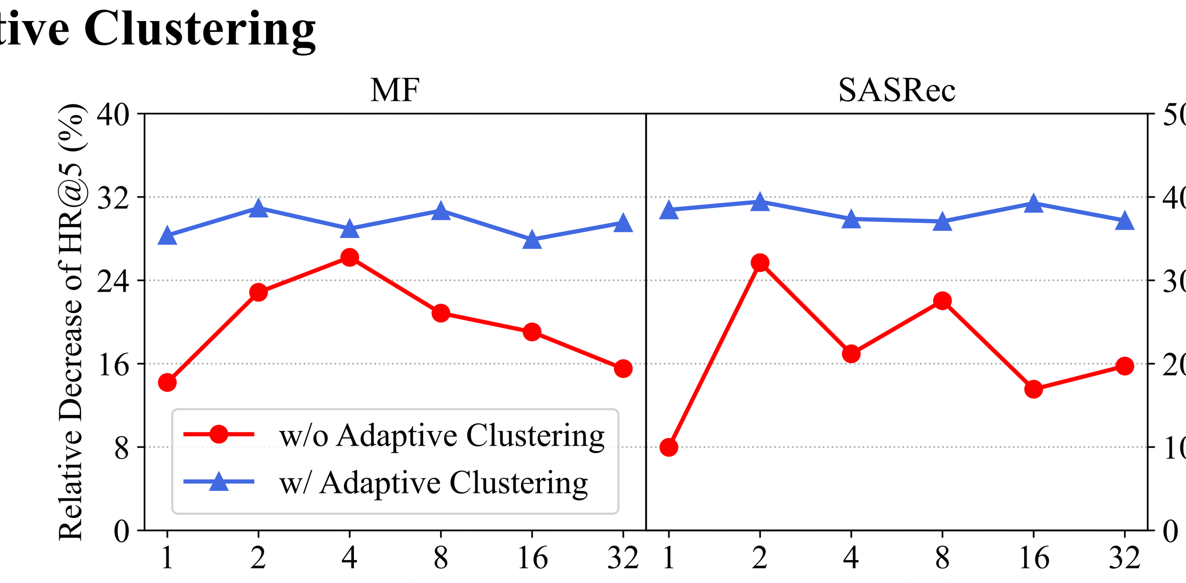
| Model | Attack Method | ML-1M HR@5 | ML-1M NDCG@5 | Gowalla HR@5 | Gowalla NDCG@5 |
|---|---|---|---|---|---|
| MF | No Attack | 0.03549 (-) | 0.02226(-) | 0.02523 (-) | 0.01697 (-) |
| | LabelFlip | 0.03561 (-0.34%) | 0.02238 (-0.54%) | 0.02541 (-0.71%) | 0.01711 (-0.82%) |
| | FedAttack | 0.03358 (5.38%) | 0.02118 (4.85%) | 0.02371 (6.02%) | 0.01585 (6.60%) |
| | Gaussian | 0.03555 (-0.17%) | 0.02224 (0.09%) | 0.02528 (-0.20%) | 0.01701 (-0.24%) |
| | LIE | 0.03259 (8.17%) | 0.02062 (7.37%) | 0.02316 (8.20%) | 0.01571 (7.42%) |
| | Fang | 0.03038 (14.40%) | 0.01897 (14.78%) | 0.02131 (15.54%) | 0.01448 (14.67%) |
| | ClusterAttack | **0.02451 (30.94%)** | **0.01545 (30.59%)** | **0.01664 (34.05%)** | **0.01117 (34.18%)** |
| SASRec | No Attack | 0.10810 (-) | 0.07053 (-) | 0.03251 (-) | 0.02217 (-) |
| | LabelFlip | 0.10857 (-0.43%) | 0.07071 (-0.26%) | 0.03270 (-0.58%) | 0.02222 (-0.23%) |
| | FedAttack | 0.10013 (7.37%) | 0.06572 (6.82%) | 0.03054 (6.06%) | 0.02087 (5.86%) |
| | Gaussian | 0.10769 (0.38%) | 0.07055 (-0.03%) | 0.03226 (0.77%) | 0.02222 (-0.23%) |
| | LIE | 0.09677 (10.48%) | 0.06281 (10.95%) | 0.03008 (7.47%) | 0.02021 (8.84%) |
| | Fang | 0.08964 (17.08%) | 0.05909 (16.22%) | 0.02797 (13.96%) | 0.01883 (15.07%) |
| | ClusterAttack | **0.06547 (39.44%)** | **0.04130 (41.44%)** | **0.02223 (31.62%)** | **0.01544 (30.36%)** |

Model performance under different attack methods with different defense mechanisms.




Model performance under different attack methods with no defense.

### Impact of Adaptive Clustering




### Influence of the Ratio of Malicious Clients

| $m$% | No Attack | FedAttack | LIE | Fang | ClusterAttack |
|---|---|---|---|---|---|
| 0.5% | 0.03549 | 0.03491 | 0.03465 | 0.03426 | 0.03001 |
| 1% | 0.03549 | 0.03358 | 0.03259 | 0.03038 | 0.02451 |

Performance of *ClusterAttack* with different ratios of malicious clients.

| Defense Method | FedAttack | LIE | Fang | ClusterAttack |
|---|---|---|---|---|
| No Defense | 0.03195 | 0.03147 | 0.02793 | 0.01950 |
| MultiKrum+UNION | 0.03438 | 0.03447 | 0.03454 | 0.03291 |
| MormBound+UNION | 0.03490 | 0.03464 | 0.03464 | 0.03351 |

Performance of *UNION* against different attacks with 5% malicious clients.