A Dual Defense Design Against Data Poisoning Attacks in Deep Learning-Based Recommendation Systems

Xiaocui Dang[1], Priyadarsi Nanda[1*], Manoranjan Mohanty[2], Haiyu Deng[1]
[1]*University of Technology Sydney,* Sydney, Australia
[2]*Carnegie Mellon University*, Qatar
Priyadarsi.Nanda@uts.edu.au, mmohanty@andrew.cmu.edu,
{XIAOCUI.DANG-1, haiyu.deng}@student.uts.edu.au

***Abstract*—Deep learning is being extensively utilized across various domains, with deep learning-based recommendation systems gaining prominence due to their exceptional performance. However, these systems are vulnerable to data poisoning attacks, where adversaries introduce carefully crafted fake user ratings to compromise the integrity of the recommendation model. We propose a dual defense to address this threat. The first line of defense, termed active defense, preemptively reduces the system's vulnerability to poisoning attacks by incorporating crafted regularization into the loss function. This approach diminishes the attacker's impact while preserving system performance, thereby lowering the success rate of targeted attacks. To further enhance the system's robustness, we introduce a GAN (Generative Adversarial Network)-based detection model as a passive defense strategy to accurately identify and filter out poisoned data. Empirical evaluations on three distinct datasets demonstrate that our dual defense approach significantly enhances both the proactive defense and passive detection capabilities of recommendation systems, effectively countering data poisoning attacks.**

***Index Terms*—Deep learning, recommendation systems, data poisoning attacks, dual defense**

I. Introduction

With the rapid development of the internet, the overwhelming amount of data presents significant challenges in extracting needed information [1]. The application of deep learning in recommendation systems effectively addresses this issue [2] [3]. Its powerful feature learning and pattern recognition capabilities enable more accurate, personalized recommendations, significantly enhancing user satisfaction and experience [4].

The development of deep learning in recommendation systems brings opportunities and challenges, and these systems are vulnerable to data poisoning attacks, threatening credibility, with scarce defense methods available [5] [6] [7]. Recent research has highlighted active defense approaches in machine learning security, focusing on proactive measures to address various adversarial threats. They propose novel mechanisms to counter data poisoning in deep learning, neural Trojans in pre-trained networks, and attacks on IoT intrusion detectors, demonstrating significant effectiveness in preserving model performance and security across diverse attack scenarios [8] [9] [10]. Additionally, some studies focus on passive defense approaches in network security, emphasizing detection over prevention. These include using data complexity metrics to address causal attacks, employing machine learning classifiers for DDoS (Distributed Denial of Service) attacks on financial systems, and identifying vulnerabilities in industrial protocols like DNP3, demonstrating high accuracy in threat identification [11] [12] [13]. And other studies introduce dual defense frameworks for other areas: for example, one against face swapping via adversarial watermarking [14]. Another mitigating membership inference attacks [15], both balancing security and model utility effectively.

Previous defense research has primarily focused on either active or passive defense strategies in isolation. The existing dual defense applied to fields other than deep learning-based recommendation systems. And related defense methods in tabular data-based deep learning recommendation systems show limitations and inefficiencies, we need to consider all three factors simultaneously:

a) Proactive Defense: active defense, the necessity of preemptive defense measures before an attack succeeds.

b) Model integrity: requiring to protect the original recommendation system model's performance when using the defense methods.

c) Effective Anomaly Detection: passive defense, improving the detection accuracy rate of anomaly detection methods for this data attack.

In this paper, we propose a dual defense mechanism to combat data poisoning attacks on deep learning-based recommendation systems. Our approach includes both active and passive defenses for comprehensive protection. Experiments on MovieLens-100K (ML-100K), MovieLens-1M (ml-1m) and Last.fm show our strategy reduces the attack hit rate $HR(t)$ of target items by up to about 60% in the active defense, and enhances the accuracy of fake user detection by about 90% in the passive defense.

- We propose a first-line defense for deep learning-based recommendation systems: an active defense mechanism using crafted regulation. This preserves original recom-

* Corresponding author.

mendation system performance while effectively reducing data poisoning attack hit rates $HR(t)$.

- We propose a second-line defense, a post-poisoning safeguard that effectively detects fake users. The addition of this line of defense improves the accuracy of fraudulent account identification and creats a comprehensive, dual-layered defense mechanism against data poisoning attacks.
- Experiments on three real-world datasets validate our dual-defense mechanism's effectiveness. Results confirm effective mitigation of data poisoning attacks in recommendation systems.

II. Preliminary

A. Neural Matrix Factorization

This study builds upon the Neural Matrix Factorization (NeuMF) algorithm, a prominent neural collaborative filtering (NCF) approach for recommendation systems, with multi-layer perceptron (MLP) architectures [16]. As illustrated in Fig. 1. This algorithm operates on a dataset comprising m users u and n items i, from which a user-item interaction matrix Y is derived based on observed interaction records $\{u, i, y_{ui}\}$.

NeuMF uses one-hot encoded vectors for users and items, projecting them into dense NeuMF and MLP latent vectors. The model combines a linear NeuMF component (inner product of NeuMF vectors) and a nonlinear MLP component (ReLU activation across X layers). Final predictions $\hat{y_{ui}}$ integrate both outputs. After training, it predicts missing entries in Y to create $\hat{Y}$ for personalized recommendations [17].

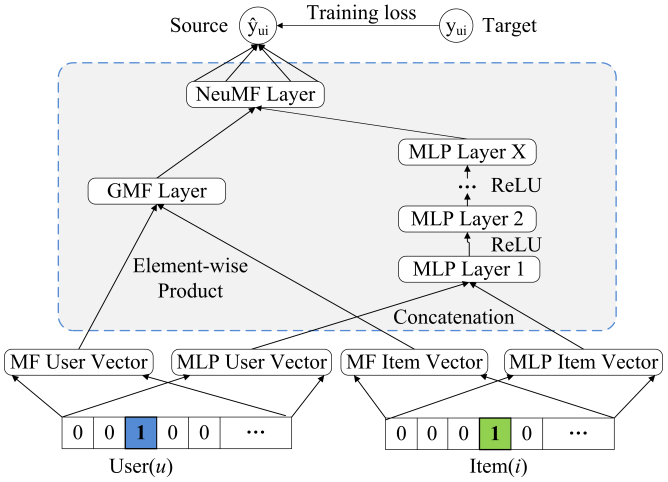


Fig. 1. Neural matrix factorization model (NeuMF).

B. Defensive Strategies for Recommendation Systems

Research indicates that recommendation systems are vulnerable to various attacks aimed at manipulating outcomes by introducing low-quality information [5]. Shilling attacks involve creating multiple copies of an item to enhance its visibility, relying on favorable remarks to boost its recommendation [18]. Furthermore, poisoning attacks, as described by [7], involve malicious data modifications intended to influence system outcomes.

In addition to data quality rules, clustering, outlier detection, L_2 regularization [11], Slab, and loss defense [8], various approaches have been explored to mitigate data poisoning attacks. Each of these methods has its limitations. Liu et al. attempted to neutralize poisoning attacks through input data preprocessing [9]. Pang Wei Koh et al. proposed three attack methods that evade data sanitization, it has shown that using multiple models for fusion and voting or averaging results can reduce the impact of attacks on a single model [10]. Vasiliki Kelli and Islam Umar et al. have also suggested a defense strategy based on multi-model fusion for the data poison attacks [12] [13].

C. Problem Formulation

We begin by analyzing the threat model and attack method, which encompasses the attacker's intentions, capabilities, and expertise. This analysis is crucial for developing effective defense techniques. We scrutinize the threat model considering the attack capabilities and strategy.

Attack capability: Recent research has shown that deep learning is commonly used in recommendation systems to enhance accuracy. However, injecting fake data with meticulously designed ratings into these systems can compromise their performance and accuracy [17]. Utilizing NeuMF, a deep learning-based recommendation system framework, data poisoning attacks can be initiated. Attackers manipulate a small portion of the training data to influence the behavior of learning algorithms, leading to biased recommendations [19].

Attack strategy: This attack strategy approximates the optimization problem, constructs a "poison model" to simulate the compromised recommendation system.

$$G[y(v)] = \|y(v)\|_2^2 + \eta \cdot \sum_{u \in S} \max\{\min_{i \in L_u} \log[\hat{y}_{ui}] - \log[\hat{y}_{ut}], -\kappa\} \tag{1}$$

Here, $y(v)$ is rating vector of user, η is a coefficient, S is unrated users, L_u is u's recommendations, $\hat{y}_{ui}$ and $\hat{y}_{ut}$ are predicted ratings, and κ enhances robustness.

Iteratively selects filler items for fake users using predicted ratings and dynamic probabilities, injecting generated ratings to promote the target item [17]. The attacker proposes this loss function:

$$l = \mathcal{L} + \lambda \cdot G\left[\hat{\mathbf{y}}_{(v)}\right] \tag{2}$$

It includes the original recommendation system loss function $\mathcal{L}$ to ensure model effectiveness, and $G[\hat{y}(v)]$, which is related to the attack objective. Here, $\hat{y}(v)$ represents the predicted rating vector for fake user v, and λ is a positive coefficient that balances the weight between model effectiveness and the attack objective.

III. Dual Defense Framework

A. Overview

Our design presents a dual-defense approach against data poisoning attack in deep learning-based recommendation systems. Following a comprehensive attack analysis, we develop

dual-defense mechanism: the first line of defense, active defense that crafts the original recommendation system loss function before training, reducing attack hit rates $HR(t)$ while maintaining system performance. And second line of defense, passive defense using a GAN-based model to detect fake users. We will discuss our scheme based on the following section *B* and *C*.

B. Active Defense

It is well-designed that L_2 regularization (Tikhonov regularization) can enhance the stability of machine learning algorithms and help mitigate the effects of poisoning attacks [20]. Consequently, we propose the use of a carefully crafted L_2 regularization (CLR) as a response to data poisoning attacks. For the data poisoning attacks of deep learning-based recommendation systems, our goal is to mitigate the data poisoning attacks and improve the privacy and security of the entire recommendation system. Figure 2 illustrates the comprehensive attack mitigation scenario after incorporating the carefully crafted L_2 regularization (CLR).

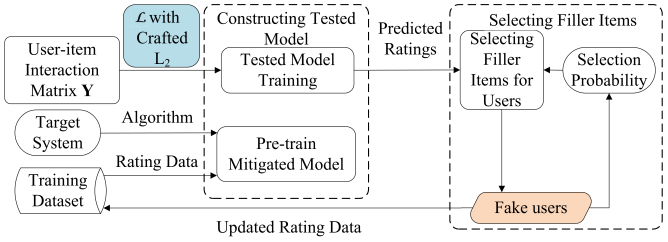


Fig. 2. CLR (crafted L_2 regularization) against data poisoning attacks. The scheme begins with user-item interaction data as input. Defenders then proceed by constructing and training a model based on $\mathcal{L}$ with crafted L_2. Next, they predict ratings and select filler items for users, generating fake users with high ratings for the target item, resulting in updated rating data. Finally, the defenders incorporate this updated data into new training data to produce a recommendation model capable of mitigating data poisoning. This approach is designed to proactively defend against data poisoning attacks while preserving system performance integrity.

To mitigate unknown data poisoning attacks while preserving the recommendation system performance, we incorporate crafted L_2 regularization into the original model's loss function, $\mathcal{L}$. This approach aims to alleviate the impact of data poisoning without compromising the system's effectiveness. Thus, the new loss function is:

$$L = \mathcal{L} + (N_r - 1/2) * 10 + \frac{e^{\lambda}}{2}\|\omega\|_2^2 \tag{3}$$

Here, N_r is random noise-enhancing robustness. The exponential form ensures positive regularization, aiding in learning λ. Thus, the poisoned model's loss function becomes:

$$l' = L + \lambda \cdot G\left[\widehat{\mathbf{y}}_{(v)}\right] \tag{4}$$

Here our goal is to covertly protect the recommendation system before data poison attacks. **Algorithm** 1 uses heuristic active protection against data poisoning.

Algorithm 1: Active Guard: Crafted L_2 Regularization on Training (CLR)

Input: User-item interaction matrix $\boldsymbol{Y}$, Crafted L_2, initial loss function $\boldsymbol{\mathcal{L}}$, pre-train epochs T_{pre}, learning rate η, tested model update schedule S
Output: detection model $\hat{\theta}$

begin
  # STEP 1: Get Training Data $\boldsymbol{D_{trn}}$.
  $\boldsymbol{D_{trn}} \leftarrow \boldsymbol{Y}$;
  # STEP 2: Polish the initial model loss function $\boldsymbol{\mathcal{L}}$.
  Using the item Approximating Hit Ratio as $\boldsymbol{\mathcal{L}}$;
  Get polished model loss function $\boldsymbol{L}$ using Eq. (2);
  $\boldsymbol{L} \leftarrow$ Crafted $\boldsymbol{\mathcal{L}}(N_r, L_2)$;
  # STEP 3: Pre-train model M_t on $\boldsymbol{D_{trn}}$ with $\boldsymbol{L}$.
  Start initial training to get the mitigatory poisoning model M_t based $\boldsymbol{L}$.
  Get mitigatory poisoning model $\theta_t \leftarrow M_t$;
  Initialize $\boldsymbol{L} \Longleftarrow 0$, model θ_t, and random optimizer
  for $t = 1...T_{pre}$ **do**
    $\theta^t \leftarrow \theta^t - \eta \bigtriangledown L(D_{trn}, \theta^t)$
  end
  return mitigatory poisoning model θ_t
  # STEP 4: detection model training for data poisoning defense:
  Get tested model $\hat{\theta}$;
  for $t = T_{pre} + 1...T$ **do**
    if $t \in S$ **then**
      $\hat{\theta} \leftarrow update\ \theta_t(D_{trn}, l')$ based on Eq. (3)
    end
  end
end
return $\hat{\theta}$

C. Passive Defense

To bolster deep learning recommendation systems against data poisoning attacks, we introduce passive defense as a secondary safeguard. This approach identifies and filters out fake users in training data, mitigating attack impacts and enhancing the security of the recommendation system.

We employ a Generative Adversarial Network (GAN)-based detection method to identify fake users D_f by comparing prediction results. The target model trained on real data D_T is compared with a simulated model to measure prediction differences and detect fake users D_f within the dataset $D_d = D_f + D_T$. Figure 3 presents the GAN-based defense detection framework.

Data processing. The sparsity of user-item interaction data presents a significant challenge to the efficacy of recommendation algorithms, as illustrated in Figure 1. We find that Word2Vec, by representing words as dense vectors, captures the semantic relationships between them. Making it widely applicable in Natural Language Processing (NLP) tasks. And since the traditional one-hot encoding method in the recommendation system generates sparse data, which may adversely affect the performance of the detection model. Here, we leverage the advantages of Word2Vec and the need for dense data, we employ Word2Vec to convert sparse data into dense vectors in the data processing phase, mapping users and items into a common vector space to capture their relationships. It shows in Figure 4.

Data enhancement. We recognize the scarcity of reliable

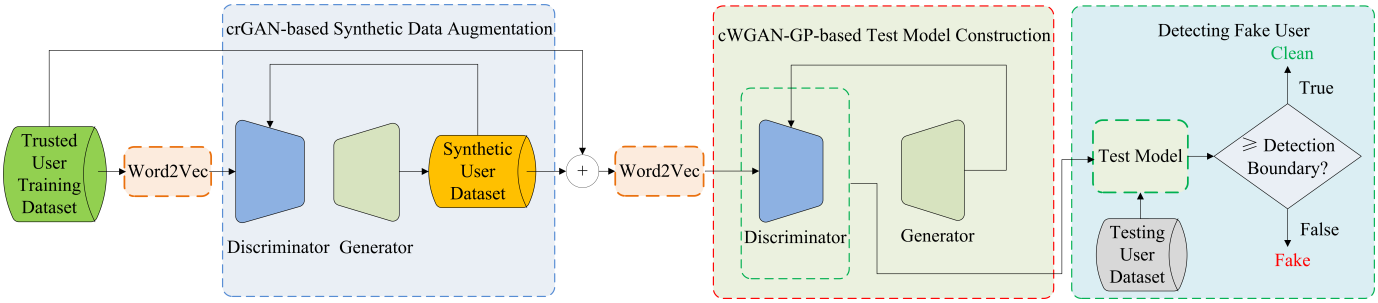


Fig. 3. A framework for defense detection based on Generative Adversarial Network (GAN). In the first step, defenders will use Word2Vec for data processing. In the second step, defenders use data enhancement based on crGAN to compensate for the scarcity of real data. In the third step, defenders use synthetic data to build a test model based on cWGAN-GP. In the last step, using the constructed test model to detect and analyze fake users. It aims to enhance the detection accuracy of fake users by constructing detection models using GAN.

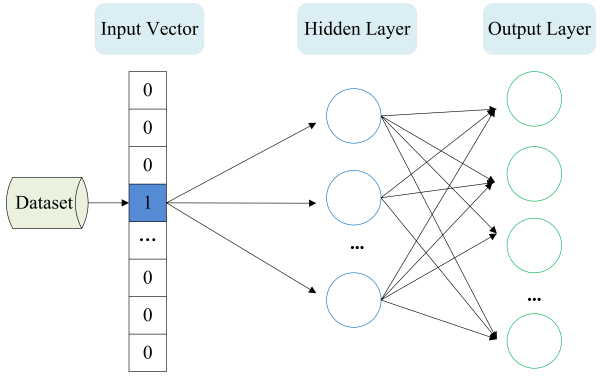


Fig. 4. Word2Vec framework in recommendation system.

real data. The second part of our defense detection process generates synthetic training data matching D_T distribution. Using a consistent regularization Generative Adversarial Network (crGAN)-based framework, the generator (G) produces realistic synthetic samples, enhancing dataset diversity and representativeness for improved model training and analysis. while the discriminator (D) distinguishes between real and synthetic data, helping improve the generator output quality.

During training, the generator (G) and discriminator (D) engage in simultaneous learning to achieve consistency between the augmented data $T(x)$ and the original data x, optimizing the following objectives:

$$\min DLcr = \min D \sum j = m^n \lambda_j \left\| D_j(x) - D_j(T(x)) \right\|^2 \tag{5}$$

Adversarial training generates synthetic data, with the discriminator assessing quality. Objective functions are:

$$\begin{aligned} L_{cr}^{(i)} &= \|D(x) - D(T(x))\|^2 \\ L_D^{(i)} &= D(G(z)) - D(x) \end{aligned} \tag{6}$$

Constructing a simulation model for detection. Here, using the conditional Wasserstein Generative Adversarial Network (cWGAN-GP), we construct a simulation model to enhance training data distribution, addressing overfitting in deep learning due to data insufficiency.

cWGAN-GP uses wasserstein distance to evaluate real-simulated sample distribution discrepancies, incorporating conditional information. wasserstein distance is defined as:

$$\mathrm{W}\left(p_{data}, p_{\mathrm{g}}\right) = \inf_{\gamma \in \Pi\left(p_{data}, p_{\mathrm{g}}\right)} \mathrm{E}_{(x,y)\sim\gamma}[\|x - y\|] \tag{7}$$

Here, p_{data}, p_{g} denote the true data distribution and the generated data distribution, respectively. $\prod\left(p_{\mathrm{data}}, p_{\mathrm{g}}\right)$ represents the joint probability that all edge distributions conform to p_{data} and p_{g}.

The cWGAN-GP generator learns to produce synthetic data matching real data distributions, incorporating conditional information y for personalized simulations. The discriminator combines p_{data}, p_{g} and y in a joint hidden expression. The generator links condition y to p_{g} similarly. This enables conditional data generation:

$$\begin{aligned} \min_G \max_D V(D,G) = \mathrm{E}_{x\sim p_{data}(x)}[D(x \mid y)] - \\ \mathrm{E}_{\tilde{\mathrm{g}}\sim p_{\mathrm{g}}(\mathrm{g})}[D(\tilde{g} \mid y)] - \lambda \mathrm{E}_{\hat{x}\sim \mathrm{P}_{\hat{X}}}\left[\left(\|\nabla_{\hat{x}} D(\hat{x} \mid y)\|_2 - 1\right)^2\right] \end{aligned} \tag{8}$$

The objective optimization functions of cWGAN-GP are:

$$\begin{aligned} L(D) = -\mathrm{E}_{x\sim p_{data}(x)}[D(x \mid \mathrm{y})] + \mathrm{E}_{\tilde{\mathrm{g}}\sim p_{\mathrm{g}}(\mathrm{g})}[D(\tilde{g} \mid \mathrm{y})] + \\ \lambda \mathrm{E}_{\hat{X}\sim \mathrm{P}_{\hat{x}}}\left[\left(\|\nabla_{\hat{x}} D(\hat{\mathrm{x}} \mid \mathrm{y})\|_2 - 1\right)^2\right] \\ L(G) = -\mathrm{E}_{\tilde{\mathrm{g}}\sim p_{\mathrm{g}}(\mathrm{g})}[D(\tilde{\mathrm{g}} \mid \mathrm{y})] \end{aligned} \tag{9}$$

The cWGAN-GP aims to minimize L, reducing the distribution gap between generated and real data. Post-training, the discriminator network serves as our simulation model, leveraging its ability to differentiate real from synthetic data effectively.

Fake user detection. Our aim is to identify fake users and prevent their inclusion in the recommendation system's training data. The simulation model employs a detection threshold to classify users: outputs below the threshold indicate fake users, while those above signify authentic users.

Algorithm 2 summarizes the four-part detection mechanism. Crucially, the detection threshold requires rigorous evaluation to ensure high accuracy and reliability in practical applications.

Algorithm 2: Passive Guard: detection mechanism via GAN

Input: Trusted user data $\boldsymbol{D_T}$, fake user data $\boldsymbol{D_f}$, perturbation vector δ
Output: Detection decision of each user using detection model M_d

begin
  # STEP 1: Get trusted data of dense features $D_T^{'}$.
  $D_T^{'} \leftarrow D_T$ (using Word2Vec);
  # STEP 2: Get augmented training data D_{aug} on crGAN.
  if *Synthetic user D_{sy} exist* **then**
    load D_{Sy}
  else
    Generate D_{Sy} using crGAN
    load Generate D_{Sy}
  end
  $D_{aug} \leftarrow D_T^{'} + D_{Sy}$.
  # STEP 3: Constructing detection simulation model M_d.
  clean data $D_c \leftarrow D_{aug}$;
  $D_G \overset{\mathbf{G}}{\leftarrow} \delta$.
  for *each training iteration* **do**
    Update **D** (**D**_loss (D_c, D_{Sy}))
    Update **G** (**G**_loss)
  end
  Get detection simulation model $M_d \leftarrow$ **D**.
  # STEP 4: Detecting fake user using M_d.
  for *user u in tested dataset* **do**
    if $M_d(u) \geq boundary$ **then**
      $u \Longrightarrow clean$
    else
      $u \Longrightarrow fake$
    end
  end
  return 0
end

IV. Experiment

A. Experimental Design

Selecting datasets and models. Our dual- defense experiment on MovieLens-100K (ML-100K), MovieLens-1M (ml-1m), and Last.fm datasets, detailed in Table I. Especially, Last.fm preprocessing includes binarizing interactions, removing duplicate tags, and filtering to avoid cold start. We target NeuMF for defense, as it effectively models implicit feedback by capturing both linear and nonlinear user-item relationships, enhancing recommendation quality.

TABLE I
The Summary of Three Datasets.

Details	*Datasets*		
	ML-100K	***ml-1m***	***Last.fm***
Users	*5943*	*6040*	*1892*
Items	*1682*	*3706*	*17,632*
Ratings	*100,000*	*1,000,209*	*186,479*

Evaluation metrics. In active defense line, we use the hit rate $HR(t)$ of the target item as the main evaluation metric for defending against data poison attacks. The formula for calculating the hit rate $HR(t)$ is as follows:

$$HR(t) = \frac{\sum_{i=1}^{n} I\left\{t_i \in Top\ K_i\right\}}{n} \tag{10}$$

The indicator function I is defined as 1 when the condition $t_i \in Top\ K_i$ is satisfied, and 0 otherwise. Here, n represents the total number of items.

In the second passive defense line, evaluating GAN detector against poisoning, we use $accuracy$, $recall(TP/(TP+FN))$, and $F1$. $F1$ balances precision and $recall$ for overall performance:

$$F1 = \frac{Precision \times Recall}{2 \times (Precision + Recall)} \tag{11}$$

Where precision is $TP/(TP + FP)$, TP is the number of correctly identified fake users, FN is the number of fake users misclassified as real, and FP is the number of real users misclassified as fake.

Implementation specifics. In our active defense, we conducted simulation experiments on NeuMF, primarily using crafted L_2 regularization (CLR) as a defense mechanism. Various regularization parameters (λ = 0.01, 0.1, 1.0, 3.0) were tested, comparing hit rates $HR(t)$ of target item under raw data poisoning (no defense) [17], local differential privacy [21], and HINT defense [22] methods to evaluate the CLR method's effectiveness. In our passive defense, primarily utilizing Word2vec to process the data, we addressed user-item matrix sparsity with output dimensions of 10x10 for ML-100K, 16x16 for ml-1m, and 20x20 for Last.fm.

B. Results and Analysis

1) First Line of Defense (Active Defense):

Proactive defensive guarantee. Experimental results demonstrate that incorporating a well-crafted L_2 regularization into the original recommendation system model effectively mitigates the hit rate of target item on data poisoning attacks. As shown in Table II, with the insertion of only 0.5% fake users in the ML-100K dataset, our defensive approach reduces the hit rate $HR(t)$ for random target items by 0.08%. This performance surpasses existing defense methods, including HINT, which only achieved a 0.02% reduction in the poisoning hit rate $HR(t)$ for target items. These findings underscore the efficacy of our proposed first line of defense mechanism.

Our proposed defense mechanism demonstrates superior performance compared to existing methodologies, even in scenarios where attackers possess knowledge of only partial common user ratings. Comprehensive experiments conducted on two distinct datasets reveal that when merely 30% of the original matrix is scored, the attack hit rate significantly diminishes to 0.0092. Furthermore, our novel defense strategy effectively reduces this rate to a mere 0.0020 for randomly selected target items. As illustrated in Table III.

Model integrity guarantee. Incorporating crafted L_2 regularization (CLR) into our recommendation system's original loss function serves as an active defense line without compromising performance. We validate this by comparing the

TABLE II
DEFENSIVE RESULTS FOR THE ACTIVE DEFENSE METHOD.

Dataset	Methods	Attack size							
		Random target items				Unpopular target items			
		0.5%	1%	3%	5%	0.5%	1%	3%	5%
ML-100K	Data poisoning attack	0.0034	0.0046	0.0100	0.0151	0.0007	0.0019	0.0111	0.0206
	LDP	0.0030	0.0035	0.0065	0.0087	0.0001	0.0002	0.0012	0.0022
	HINT	0.0032	0.0035	0.0069	0.0080	0.0001	0.0004	0.0014	0.0033
	CLR	**0.0026**	**0.0031**	**0.0044**	**0.0049**	**0.0001**	**0.0002**	**0.0010**	**0.0021**
Last.fm	Data poisoning attack	0.0047	0.0068	0.0144	0.0243	0.0012	0.0026	0.0086	0.0161
	LDP	0.0034	0.0050	0.0120	0.0210	0.0005	0.0017	0.0058	0.0118
	HINT	0.0032	0.0055	0.0069	0.0163	0.0006	0.0014	0.0047	0.0117
	CLR	**0.0031**	**0.0040**	**0.0121**	**0.0183**	**0.0005**	**0.0011**	**0.0061**	**0.0108**

TABLE III
DEFENSE RESULTS OF PARTIAL KNOWLEDGE FOR ATTACKER.

Knowledge level	Methods	Random target items
30%	Data poisoning attack	0.0092
	LDP	0.0086
	HINT	0.0090
	CLR	**0.0072**

recommendation system's common evaluation indicators, Normalized Discounted Cumulative Gain (NDCG), which show negligible impact for the original recommendation system's performance. Under the condition that the default epoch and other parameters remain constant, from Table IV, it is evident that incorporating CLR into the original recommendation system model does not significantly affect its performance, as indicated by the NDCG values. The impact on NDCG across the three datasets remains within 0.002.

TABLE IV
IMPACT OF THE FIRST DEFENSE LINE ON RECOMMENDATION SYSTEM PERFORMANCE.

	ML-100K	ml-1m	Last.fm
Non-CLR	0.31287	0.35960	0.38990
CLR-ed	0.31371	0.35779	0.38912
NDCG Change	**+0.00084**	**-0.00181**	**-0.00078**

2) Second Line of Defense (Passive Defense):

Effective detection guarantee. To comprehensively defend against data poisoning attacks in deep learning-based recommendation systems, we employ a second line of defense, passive defense. This involves evaluating the effectiveness of a GAN-based detection model. such as the ML-100K dataset, we conducted experiments to ensure efficient detection of fake users generated at various scales, critical for countering data poisoning attacks.

Accuracy of GAN detection. For the ML-100K, ml-1m, and Last.fm datasets, we evaluated the accuracy of our second line of defense detection method. Previous rating-based methods had a detection accuracy of only 70% [17], but our GAN detection method achieved around 90%, as shown in Figures 5(a), 5(d), and 5(g), significantly improving defense efficiency against data poisoning attacks.

F1 score of GAN detection. To fully validate our defense detection's effectiveness, we also tested the F1 score. As shown in Figures 5(b), 5(e), and 5(h), our method achieved an average $F1$ score of about 85%, while the previous detection method's $F1$ score was around 65%. This significant improvement demonstrates the superiority of our approach in accurately identifying fake users, combining both precision and recall to provide a more reliable and robust detection mechanism across different datasets.

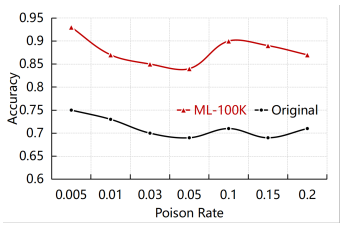
Recall of GAN detection. Additionally, in fake users' detection, recall is used to evaluate the performance of the detection model. We measure our defense model's ability to correctly identify fake users. As shown in Figures 5(c), 5(f), and 5(i), our model achieved a recall of around 90%, indicating highly effective detection of fake users across different datasets. This demonstrates the robustness and reliability of our approach to maintaining the integrity of the recommendation system against fake users.

C. Ablation Study

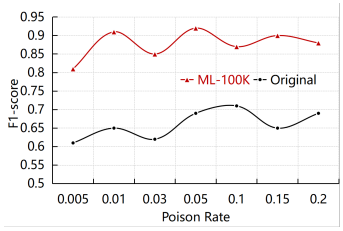
1) First Line of Defense (Active Defense):

The impact for different number of fake users. After implementing active defense, we examined the impact of different numbers of fake users on defense results. Increasing fake users raised the target item's $HR(t)$ across all datasets. For example, in the ML-100K dataset (Table II), injecting 0.5% random fake users resulted in an $HR(t)$ of 0.0026, while 5% increased it to 0.00049. Despite this, our method remains effective in reducing losses, as shown in Table II.

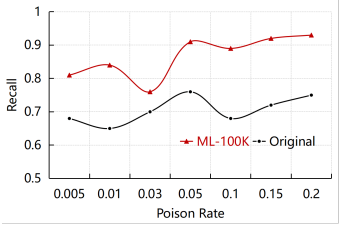
The impact for different numbers of recommended list. Table V illustrates the defense results under varying recommendation list sizes K. We observe that as K increases, the evaluation metric $HR(t)$ also increases. However, our defense method effectively mitigates the impact of data poisoning attacks. For instance, when K = 15, the $HR(t)$ on the ML-100K dataset is reduced to approximately 30% of the original attack's effectiveness. Notably, even at K = 5, our defense strategy successfully neutralizes the poisoning attack on the Last.fm dataset, resulting in an $HR(t)$ of 0.0003.



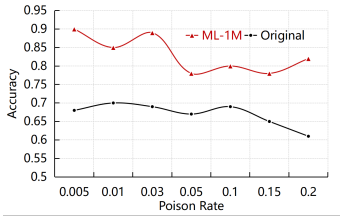
(a) Accuracy of detector on ML-100K.



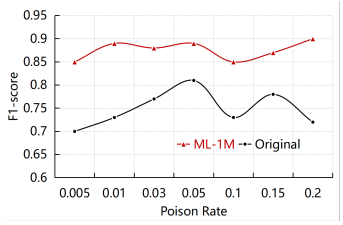
(b) F1 score of detector on ML-100K.



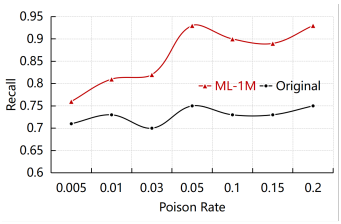
(c) Recall of detector on ML-100K.



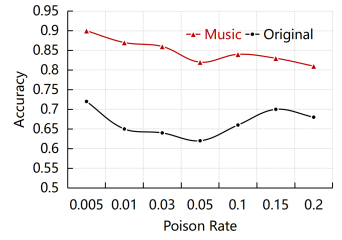
(d) Accuracy of detector on ml-1m.



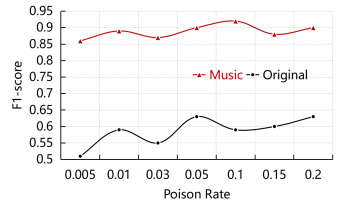
(e) F1 score of detector on ml-1m.



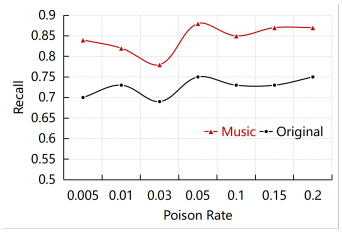
(f) Recall of detector on ml-1m.



(g) Accuracy of detector on Last.fm.



(h) F1 score of detector on Last.fm.



(i) Recall of detector on Last.fm.

Fig. 5. Accuracy-F1 score-recall evaluation.

TABLE V
THE DEFENSE RESULTS FOR DIFFERENT RECOMMENDED LIST SIZE K.

Dataset	Methods	K			
		5	10	15	20
ML-100K	Data poisoning attack	0.0012	0.0019	0.0033	0.0042
	LDP	0.0006	0.0012	0.0024	0.0026
	HINT	0.0006	0.0010	0.0022	0.0028
	CLR	**0.0004**	**0.0006**	**0.0010**	**0.0019**
Last.fm	Data poisoning attack	0.0007	0.0026	0.0042	0.0061
	LDP	0.0006	0.0017	0.0029	0.0040
	HINT	0.0004	0.0021	0.0034	0.0046
	CLR	**0.0003**	**0.0021**	**0.0023**	**0.0037**

2) Second Line of Defense (Passive Defense):

The impact of poison rates on detection accuracy. Additionally, as observed in Figures 5(a), 5(d), and 5(g), a lower poisoning rate for target items corresponds to higher detection accuracy. For instance, in the ML-100K dataset, when $HR(t)$ is 0.005, the detection accuracy peaks at approximately 93%. This is because fewer fake users make it easier for the model to identify fake users and distinguish them from real users.

The impact of poison rates on $F1$-score. Figures 5(b), 5(e), and 5(h) illustrate the variation in $F1$-score results with different poisoning rates. We observed interesting trends: First, an optimal F1 score exists for each dataset and poisoning rate. Secondly, increasing the poisoning rate does not consistently lead to higher or lower F1 score. For the ML-100K dataset, the $F1$ score peaks at a poisoning rate of 0.05, while for the ml-1m and Last.fm datasets, the maximum F1 score occurs at poisoning rates of 0.2 and 0.1, respectively. This non-linear impact indicates varying sensitivity to poisoning attacks across datasets, likely due to differing characteristics and user behavior patterns.

The impact of poison rates on Recall. Figures 5(c), 5(f), and 5(i) illustrate the variations in recall across different data poisoning rates. A notable trend is observed where recall generally increases as the poisoning rate escalates. Significantly, across all three datasets, the highest Recall values are consistently achieved at a poisoning rate of 0.2. This phenomenon indicates that even in environments with high poisoning rates, the model maintains a robust capability to correctly identify a substantial proportion of fake users.

V. RELATED WORK

Data poisoning attacks have garnered significant attention in recent recommendation system research, yet defense strategies against them remain underexplored. However, there is extensive research on defenses against widespread data poisoning attacks [21] [22] [23]. Here, we primarily compare our approach with related works, specifically the Local Differential

Privacy (LDP) defense scheme and the Healthy Influential-Noise based Training (HINT) defense scheme.

Bebensee [24] highlights that Local Differential Privacy (LDP) is a state-of-the-art approach enabling statistical computation while preserving user privacy. As the LDP protocol requires each user to locally obfuscate their raw data before submitting it to the aggregator, it remains susceptible to output poisoning attacks. Consequently, Song et al. propose a practical solution to enhance the reliability of the LDP protocol in real-world applications. Their methods have been validated for effectiveness against data poisoning attacks in practical scenarios.

HINT (Healthy Influential-Noise based Training), a novel defense against data poisoning attacks. HINT uses influence functions to identify training samples that significantly affect model loss, adding "healthy influential noise" to these samples. This approach mitigates harmful impacts and enhances beneficial ones, improving robustness against attacks. And HINT's effectiveness is validated through sensitivity analysis and computational time comparisons [25].

We looked into Huang et al.'s statistical analysis approach to compare methods for spotting fake user data [17]. They extracted key features from the dataset and generated corresponding feature values for each user to train a fake user classifier. However, this detection mechanism is not foolproof, as attackers can evade detection by modifying the method used to create fake users.

VI. Conclusion

In this paper, we propose a dual defense mechanism against data poisoning attacks in deep learning-based recommendation systems. The first line of defense, active defense, is implemented through crafted L_2 regularization (CLR). We found that our CLR method effectively reduces the poisoning attack hit rate under various attack intensities, with a negligible impact on the recommendation system model. Additionally, the second line of defense, passive defense, employs a detection model using Generative Adversarial Network (GAN), significantly improving the accuracy of fake user detection. To further enhance the defense capability of the detection model, incorporating real synthetic data can expand the training dataset, thereby training the simulation model more effectively to identify predictive differences. Future research can explore developing new fake user detection methods and designing more robust recommendation systems to against data poisoning attacks.

References

[1] Hemmati A, Arzanagh H M, Rahmani A M. A taxonomy and survey of big data in social media[J]. Concurrency and Computation: Practice and Experience, 2024, 36(1): e7875.

[2] Singh J, Sajid M, Yadav C S, et al. A novel deep neural-based music recommendation method considering user and song data[C]//2022 6th International Conference on Trends in Electronics and Informatics (ICOEI). IEEE, 2022: 1-7.

[3] Gao C, Wang X, He X, et al. Graph neural networks for recommender system[C]//Proceedings of the fifteenth ACM international conference on web search and data mining. 2022: 1623-1625.

[4] Ahmadian S, Ahmadian M, Jalili M. A deep learning based trust-and tag-aware recommender system[J]. Neurocomputing, 2022, 488: 557-571.

[5] Ahmed N, Amin R, Aldabbas H, et al. Machine learning techniques for spam detection in email and IoT platforms: analysis and research challenges[J]. Security and Communication Networks, 2022, 2022(1): 1862888.

[6] Ren Y, Li Z, Yuan L, et al. Semantic Shilling Attack against Heterogeneous Information Network Based Recommend Systems[J]. IEICE TRANSACTIONS on Information and Systems, 2022, 105(2): 289-299.

[7] Zhang X, Wang Z, Zhao J, et al. Targeted data poisoning attack on news recommendation system[J]. arXiv preprint arXiv:2203.03560, 2022.

[8] Seetharaman S, Malaviya S, Vasu R, et al. Influence based defense against data poisoning attacks in online learning[C]//2022 14th International Conference on COMmunication Systems & NETworkS (COMSNETS). IEEE, 2022: 1-6.

[9] Liu Y, Xie Y, Srivastava A. Neural trojans[C]//2017 IEEE International Conference on Computer Design (ICCD). IEEE, 2017: 45-48.

[10] Jiang H, Lin J, Kang H. FGMD: A robust detector against adversarial attacks in the IoT network[J]. Future Generation Computer Systems, 2022, 132: 194-210.

[11] Chan P P K, He Z, Hu X, et al. Causative label flip attack detection with data complexity measures[J]. International Journal of Machine Learning and Cybernetics, 2021, 12: 103-116.

[12] Islam U, Muhammad A, Mansoor R, et al. Detection of distributed denial of service (DDoS) attacks in IOT based monitoring system of banking sector using machine learning models[J]. Sustainability, 2022, 14(14): 8374.

[13] Kelli V, Radoglou-Grammatikis P, Sesis A, et al. Attacking and defending DNP3 ICS/SCADA systems[C]//2022 18th International Conference on Distributed Computing in Sensor Systems (DCOSS). IEEE, 2022: 183-190.

[14] Zhang Y, Ye D, Xie C, et al. Dual defense: Adversarial, traceable, and invisible robust watermarking against face swapping[J]. IEEE Transactions on Information Forensics and Security, 2024.

[15] Niu J, Liu P, Huang C, et al. Dual Defense: Combining Preemptive Exclusion of Members and Knowledge Distillation to Mitigate Membership Inference Attacks[J]. Journal of Information and Intelligence, 2024.

[16] He X, Liao L, Zhang H, et al. Neural collaborative filtering[C]//Proceedings of the 26th international conference on world wide web. 2017: 173-182.

[17] Huang H, Mu J, Gong N Z, et al. Data poisoning attacks to deep learning based recommender systems[J]. arXiv preprint arXiv:2101.02644, 2021.

[18] Hamidi H, Moradi R. Design of a dynamic and robust recommender system based on item context, trust, rating matrix and rating time using social networks analysis[J]. Journal of King Saud University-Computer and Information Sciences, 2024, 36(2): 101964.

[19] Barreno M, Nelson B, Joseph A D, et al. The security of machine learning[J]. Machine learning, 2010, 81: 121-148.

[20] Carnerero-Cano J, Muñoz-González L, Spencer P, et al. Regularisation can mitigate poisoning attacks: A novel analysis based on multiobjective bilevel optimisation[J]. arXiv preprint arXiv:2003.00040, 2020.

[21] Wang Z, Ma J, Wang X, et al. Threats to training: A survey of poisoning attacks and defenses on machine learning systems[J]. ACM Computing Surveys, 2022, 55(7): 1-36.

[22] Tian Z, Cui L, Liang J, et al. A comprehensive survey on poisoning attacks and countermeasures in machine learning[J]. ACM Computing Surveys, 2022, 55(8): 1-35.

[23] Kasyap H, Tripathy S. Beyond data poisoning in federated learning[J]. Expert Systems with Applications, 2024, 235: 121192.

[24] Bebensee B. Local differential privacy: a tutorial[J]. arXiv preprint arXiv:1907.11908, 2019.

[25] Van M H, Carey A N, Wu X. HINT: Healthy Influential-Noise based Training to Defend against Data Poisoning Attacks[C]//2023 IEEE International Conference on Data Mining (ICDM). IEEE, 2023: 608-617.