Is There A Bottom Line for Poisoning? Detecting High-Concealed Injection Attacks for Recommendation

Yan Feng, Zhihai Yang, Kexin Li, Jianhua He, Jianxin Li, Pinghui Wang, and Zhiquan Liu

***Abstract*—Recommender systems (RSs) are widely adopted due to their effectiveness in modeling user preferences and generating personalized recommendations. However, data poisoning attacks (PAs), which manipulate recommendation results by injecting fake user profiles, thereby affecting the quality and accuracy of the RS. Moreover, emerging high-concealed PAs (HCPAs) achieve greater evasion of detection by controlling the cost of the attack, simulating the behavior patterns of benign users, and carrying out the attack with less prior knowledge. The HCPAs bring potential challenges: (1) the very low cost for attacks not only leads to an imbalance in data distribution but also introduces a large amount of accidental co-occurrence noise; (2) the behavioral patterns similar to benign users make it difficult to describe the characteristics of HCPAs; and (3) the prior knowledge for detecting HCPAs in real scenarios is very limited. To address these challenges, we propose STOP, an orthogonal projection bi-hypersphere detection method built on multi-view relational disentanglement and information-consistent fusion. First, we model the distributional preferences of user ratings to eliminate rating and popularity bias, and further construct a co-occurrence association graph to suppress accidental overlaps. To address data imbalance caused by HCPAs, second, we introduce a distributional-consensus importance screening method that filters out benign users weakly associated with potential attackers. To address the issues of noise and the difficulty in feature characterization, third, we propose a multi-view relational disentanglement and information-consistent fusion method, which can eliminate redundant relationships, separate key relationships into dynamic and static ones, and retain task-related relationships. Finally, inspired by the "convergence theorem", we design an orthogonal projection bi-hypersphere boundary learning detection method to reduce the high false alarm rate (FAR). We extensively evaluate STOP under various HCPA scenarios, demonstrating its superiority over existing methods with an average 12.34% improvement in detection rate and an average 2.75% reduction in FAR. Furthermore, forensic analysis on real-world unlabeled data reveals distinct attacker "fingerprints", such as extreme ratings, contradictory review styles, and analysis of target items, validating STOP's reliability in practical applications.**

***Index Terms*—Injection attack, Behavior representation, Attack detection, Abnormality forensics.**

This work was supported in part by the National Natural Science Foundation of China under Grant 62172331 and in part by the Fundamental Research Funds for the Central Universities under Grant 300102404301, CHD.

Y. Feng, Z. Yang, and K. Li are with the School of Data Science and Artificial Intelligence, Chang'an University, Xi'an, China. J. He is with the School of Computer Science and Electronic Engineering, Essex University, UK. J. Li is with the School of Business and Law, Edith Cowan University, Australia. P. Wang is with the School of Cyber Science and Engineering, Xi'an Jiaotong University, Xi'an, China. Z. Liu is with the College of Cyber Security, Jinan University, Guangzhou, China. E-mail: {yanfeng, zhihaiyang, kexinli}@chd.edu.cn; j.he@essex.ac.uk; jianxin.li@ecu.edu.au; phwang@mail.xjtu.edu.cn; zqliu@jnu.edu.cn.

(Corresponding author: Zhihai Yang.)

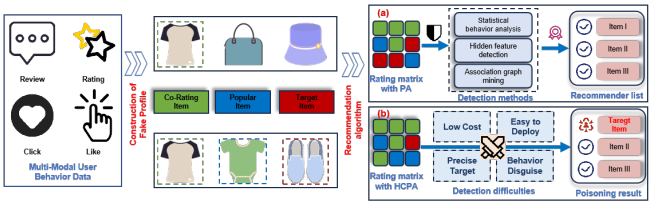


Fig. 1. An example of poisoning attacks (PAs) in recommendation systems. (a) represents traditional PAs, which are easily identified by detection methods; (b) represents high-concealed poisoning attacks (HCPAs), which are implemented at low cost and are easy to deploy in real-world environments, thus being difficult to detect.

I. Introduction

RECOMMENDER systems (RSs) have become a core component of online services. As a cutting-edge technology in the field of artificial intelligence, they are applied in multiple domains, including e-commerce, finance, education, and news [1]. Their widespread application significantly improves user experience, user stickiness, and business conversion efficiency [2]. RSs are mainly classified into content-based RSs [3], collaborative filtering RSs (CFRSs) [4], [5], and hybrid filtering RSs [6], [7]. In recent years, they have become more intelligent, precise, and interpretable, serving as a key bridge connecting users with online services. According to a report by Mordor Intelligence [8], personalized RSs have led to a 31% improvement in annual revenue and a 15% to 35% improvement in turnover ratio for e-commerce enterprises. However, alongside these benefits, the rapid development of RSs has also brought about emerging security threats.

Due to the openness and vulnerability of RSs, they are highly susceptible to profile injection attacks [9], [10] and poisoning attacks (PAs) [1], [11], [12]. Both types of attacks aim to manipulate the input data of RSs to affect the quality and accuracy of recommendation results, thereby achieving specific attack goals. In the field of e-commerce (e.g., Amazon, JD), for example, attackers create fake associations between popular commodities and poor-quality ones, thereby misleading the RS to push poor-quality commodities [13]. In the finance sector (e.g., eBay, Bloomberg), attackers attract users to invest through fake advertising to obtain illegal benefits [14], [15]. Regarding the education domain (e.g., Coursera, edX), attackers make fake ratings of specific courses or teaching materials to exaggerate their value [16]. In the sphere of news dissemination (e.g., New York Times, Wall Street Journal), attackers manipulate the ranking of target news to affect the

effectiveness of news dissemination [17], [18]. These attack behaviors have caused severe harm to the public and shaken the trust of consumers and enterprises in the virtual market. Therefore, the security and reliability of RSs have been called into question, making it increasingly urgent to protect RSs from PAs.

A. Challenge and Motivation

To defend against these threats, existing detection methods have made considerable progress [12], [19]–[21]. As shown in Figure 1, these detection methods (e.g., statistical behavior analysis [22], hidden feature detection [23], and association graph mining [9], [24]) effectively defend against traditional PAs. However, emerging high-concealed PAs (HCPAs) have become increasingly difficult to detect by reducing the cost of the attack, simulating the behavior patterns of benign users, and carrying out the attack with less prior knowledge [25], [26]. Currently, there are still some prominent challenges that need to be urgently addressed in detecting HCPAs:

1) In scenarios of HCPA, the injection cost (i.e., the proportion of attack users and the scale of filler interactions) available to the attacker is extremely limited, resulting in a significant class imbalance in the categorical data [27].
2) The HCPA achieves concealed infiltration through behavioral disguise (i.e., mimicking benign users in terms of statistical features such as rating distribution and popularity preference), thereby making the attack behavior difficult to characterize [28]–[32];
3) Although detection methods based on hidden features can improve detection performance to a certain extent, they usually do so at the cost of an increase in the false alarm rate (FAR) [22], [33], [34];
4) Detection via association graph mining represents a promising countermeasure. However, accidental co-occurrences among benign users introduce redundant rating connections (i.e., noise edges) [5], [21], [23].
5) In real-world scenarios, conducting anomaly forensics driven by prior knowledge also encounters difficulties [14], [24]. To sum up, accurately detecting HCPAs remains a pressing and unresolved challenge.

B. Solution and Contribution

To tackle these challenges, we propose a novel detection method, named STOP (i***S*** ***T***here a b***O***ttom line for ***P***oisoning). STOP aims to explore the effectiveness boundary of HCPAs and analyze the feasibility boundary of detection. First, we construct node behavior representations by quantifying the distribution preference of ratings and build edge structures through co-occurrence association. Second, to address the noise interference caused by HCPAs, we design a distributional-consensus importance screening (DCIS) method. The DCIS employs the Wasserstein distance to measure the differences between user rating distributions and uses user activity to adjust the importance scores, thereby effectively filtering out benign users with low association to potential malicious users and solving the problem of data imbalance caused by low attack cost. Third, to eliminate the noise introduced by coarse-grained co-occurrence association and address the difficulty in characterizing HCPAs' features, we develop a multi-view relational disentanglement and information-consistent fusion (MRIF) method. The MRIF mitigates the impact of trivial relationships through a probabilistic key relation estimation (PKE) module. Furthermore, it separates dynamic and static relationships in attack patterns via the dynamic–static relation separation (DRS) module. To preserve valuable structural and semantic information, MRIF also incorporates a cross-graph information-consistent fusion (CIF) module, which retains unique subgraph features and task-relevant mutual information. Finally, to reduce the high FAR common in traditional methods used to detect HCPAs, we propose an orthogonal projection bi-hypersphere boundary learning method. Inspired by the "convergence theorem", this method introduces two concentric hyperspheres to constrain the anomaly score distribution, thereby significantly improving the detection rate (DR) and reducing the FAR.

In summary, this paper makes the following contributions:

1) Unlike static pre-filtering or sampling, we design a distributional-consensus importance screening method to address the data imbalance issue caused by HCPAs, effectively suppressing the disturbance of low association benign users in the process of detecting malicious users;
2) Compared with static graph detection based on reconstruction or density, we develop a multi-view relational disentanglement and information-consistent fusion method to address the noise interference and feature representation challenges caused by HCPAs. This method eliminates the interference of trivial relationships in the graph structure and successfully characterizes the disguised attack behaviors;
3) Compared with single boundary open set detection, we develop an orthogonal projection bi-hypersphere boundary learning method to mitigate high FAR caused by HCPAs, achieving precise demarcation of the ambiguous boundaries of attack behaviors;
4) We conducted extensive comparative experiments to verify the effectiveness of the proposed STOP, involving three real-world datasets, seven representative attack methods, three attack sizes, five filler sizes, and two target item selection strategies. In addition, we compare STOP with five benchmark detection methods for comprehensive evaluation;
5) Additionally, through forensic analysis of three real unlabeled datasets, we explored interesting findings on real-world data, including extreme ratings, self-contradictory review styles, and analysis of target items, and discussed the potential presence of related injection behaviors.

II. Background and Threat Analysis

A. Related Work

*1) **Poisoning Attacks against Recommender Systems**:* Traditional attack methods are simple and easy to implement.

However, their attack effectiveness is limited, and they are easily detected because of abnormal rating patterns. For example, Yang *et al.* [35] injected fake co-visit records into RSs. Li *et al.* [36] proposed a PA method against CFRSs. Seminario *et al.* [37] conducted nuke attacks against common RSs. Currently, PAs have evolved towards intelligence. The most notable feature of intelligent PAs is the low cost and behavioral disguise, making the attack behavior increasingly concealed [38]. For example, Leg-UP [31] uses the generative adversarial network (GAN) model to produce undetectable fake user profiles. Lin *et al.* [32] proposed the AUSH model based on the GAN. Furthermore, Wang *et al.* [28] proposed an uplift-guided budget allocation (UBA) framework to maximize attack performance. CLeaR [29] is a PA method against contrastive learning (CL) RSs. Additionally, to address the issue that PAs cannot be applied to real-world environments, Yuan *et al.* [39] utilized synthetic malicious users to upload data with poisoned gradients. Zhang *et al.* [40] directly transferred attacks to the original RS by constructing a surrogate model. These advances indicate that PAs are no longer limited to simple manipulations but are evolving toward highly intelligent and concealed strategies. In particular, HCPAs have emerged, characterized by their ability to generate malicious samples that are difficult to detect at an extremely low cost and effectively deploy them in real-world environments.

*2) **Detection and Abnormality for Recommender Systems**:* Detection methods against PAs in RSs can be broadly categorized into two classes: feature characterization-based and association graph-based methods. Traditional detection methods based on feature characterization mainly rely on user feedback in RSs to detect attack behaviors in user interaction data. For example, Aktukmak *et al.* [33] proposed a detection method that combines a latent variable model with a sequential detection algorithm. Yang *et al.* [41] comprehensively analyzed the distribution of user activity, item popularity, and special ratings. Moreover, DegreeSAD [42] extracts features from the attribute of item popularity. CNN-BAG [43] is a detection method that integrates the convolutional neural network (CNN) and the bagging algorithm. However, feature characterization-based detection methods are inadequate in dealing with HCPAs. Subsequently, anomaly detection methods based on association graph mining can learn hidden user behavior preferences and item distribution features from large-scale user-item interaction data. For example, FAP [44] relies on recursive bipartite graph propagation. CoDetector [45] utilizes user latent factors with network embedding information as detection features. GAD [46] is a detection method based on the graph neural network (GNN) model. Yang *et al.* [24] designed a unified detection framework based on co-visit and co-rating graphs constructed from association rules. However, with the continuous evolution of HCPAs, the detection methods relying on association graph mining have become inadequate in characterizing attack behaviors. When the scale of HCPAs is extremely small, the FAR even surges sharply.

It is necessary to emphasize that the ultimate goal of detection algorithms should always return to real scenarios, providing reliable guarantees for unlabeled data. Specifically, discovering abnormal rating behaviors in real-world datasets without a factual basis becomes a huge challenge. For example, Lai *et al.* [20] considered the uncertainty of labels. Liang *et al.* [47] evaluated the effectiveness of the proposed method on two real datasets of movies and music. Shang *et al.* [48] confirmed the robustness of the proposed method on two real-world datasets of baby and clothing. Wu *et al.* [49] successfully discovered hidden spammers in real Amazon data. This work, differing from existing works, aims to: (1) explore an effective solution to deal with HCPAs; and (2) investigate a novel framework to detect attacks or anomalies on both synthetic data and real data.

B. Threat Model

*1) **Attacker's Goals**:* Typically, the attacker u_a injects non-zero fake ratings $\mathcal{R}_a = \{r_{u_a i} | u_a \in \mathcal{U}_a\}$ into RSs. The u_a gives high ratings (i.e., promotion) or low ratings (i.e., demotion) to the target item $\mathcal{I}_t$ and carefully designed ratings to other selected items, to make $\mathcal{I}_t$ as many or as few as possible be recommended to real users $\mathcal{U}_b$. In this paper, we take maximizing Hit Rate@K (HR@K) as an example. HR@K denotes the proportion of items within a user's TOP-K recommendation list $\mathcal{V}_u^{\text{rec}}$ that the user has actually interacted with (e.g., clicks, purchases, and likes). The goal of u_a is to find an optimal rating vector,

$$\text{HR@}K = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} I[\mathcal{I}_t \in \mathcal{V}_u^{\text{rec}}], \text{NDCG@}K = \frac{\text{DCG@}K}{\text{IDCG@}K}, \tag{1}$$

where I is an indicator function. If $\mathcal{I}_t$ appears in $\mathcal{V}_u^{rec}$, $I[\mathcal{I}_t \in \mathcal{V}_u^{rec}]$ is 1; otherwise, it is 0. Furthermore, u_a also disrupts the ranking effect of $\mathcal{V}_u^{rec}$, and its evaluation indicator $\text{NDCG@}K$ is also shown as in Eq. (1). Here, $\text{DCG@}K$ is discounted cumulative gain, and $\text{IDCG@}K$ is ideal $\text{DCG@}K$. The defense goal of STOP is the single-target attack. The purpose of the single-target attack is to make $\mathcal{I}_t$ appear in the TOP-K recommendation list of benign users $\mathcal{U}_b$ as much as possible.

Furthermore, we categorize PAs by their primary intent into availability attacks [40], transferability attacks [30], [31], and concealed attacks [50]. These objectives are mutually constraining, pushing one (e.g., availability) typically compromises another (e.g., concealment), and practical attacks often operate at a chosen trade-off. Within the attack methods considered in this paper, AIA [30] and CLeaR [29] lean toward availability; AUSH [32] and UBA [28] target concealment; and Leg-UP [31] provides a budget-allocation mechanism to navigate the availability, concealment, and transferability trade-offs.

*2) **Attacker's Capability**:* The attacker cannot interfere with benign users but can fully control malicious users. However, due to limited resources, the attacker can only register a limited number of malicious users with a limited number of interactions. Based on the MovieLens (ML)-1M dataset, we first selected 20, 60, and 100 users as malicious attackers. The ML-1M dataset contains a total of 5950 users, so the attack sizes (ASs) are 0.35%, 1%, and 1.65% respectively, which can be expressed as,

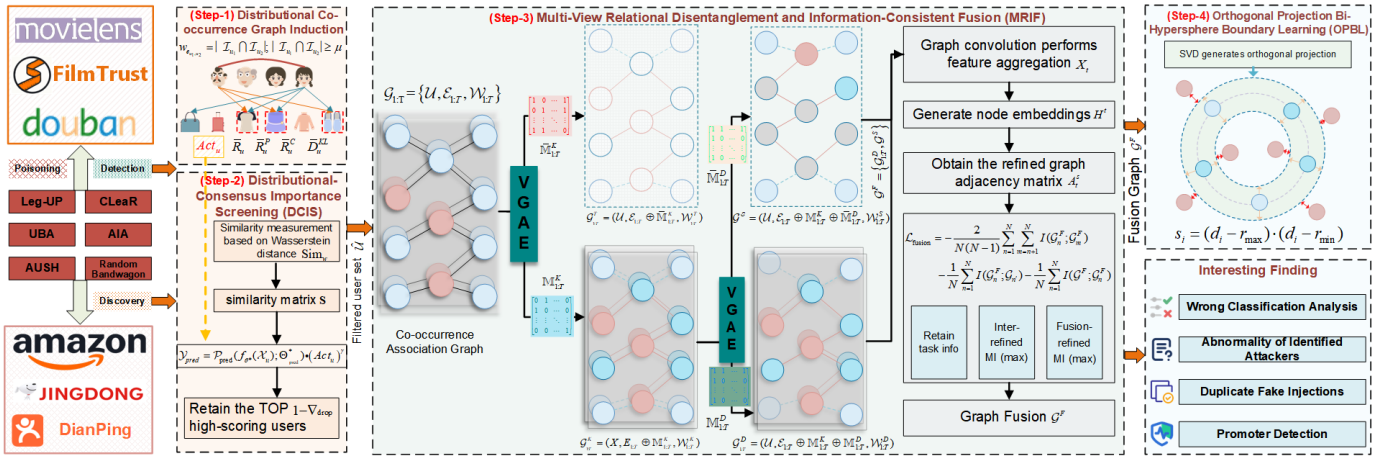


Fig. 2. The basic framework of our proposed STOP, consisting of four steps: distributional co-occurrence graph induction, distributional-consensus importance screening (DCIS), multi-view relational disentanglement and information-consistent fusion (MRIF), and orthogonal projection bi-hypersphere boundary learning (OPBL).

$$AS = \frac{|\mathcal{U}_a|}{|\mathcal{U}_a| + |\mathcal{U}_b|}, FS = \frac{|\mathcal{I}_a|}{|\mathcal{I}_a| + |\mathcal{I}_b|}, \tag{2}$$

where $\mathcal{U}_a$ and $\mathcal{U}_b$ are the sets of all fake users and all benign users. Second, we generated 20, 30, 40, and 50 rating records for each malicious user. The ML-1M dataset contains a total of 3,659 items, so the filling sizes (FSs) are 0.55%, 0.82%, 1.09%, and 1.36%, respectively, as also shown in Eq. (2). Here, $|\mathcal{I}_a|$ and $|\mathcal{I}_b|$ denote the number of ratings by each malicious user and the total number of items. We apply ASs (0.35%, 1%, and 1.65%) and FSs (0.55%, 0.82%, 1.09%, and 1.36%) to other datasets to construct the attackers' capabilities.

Meanwhile, the attacker has the ability to understand the victim's RS. This includes two scenarios: (1) in the white-box environment, the attacker can fully access the victim's RS; (2) in the black-box environment, the attacker can only obtain interaction information and has no prior knowledge of the victim's RS. In the attack methods addressed in this paper, CLeaR [29] focuses on the white-box environment; AIA [30], AUSH [32], Leg-UP [31], and UBA [28] assume that the attacker can only obtain part of the interaction information between real users.

III. The Proposed Method

A. Problem Formalization

Definition 1. The rating behavior dataset $\mathcal{D}_b$ is a set of tuples $< u_i, p_j, r_{ij}, t_{ij}, re_{ij}, c_{ij} >$, where u_i represents a user, p_j represents an item, r_{ij} is the rating given by user u_i to item p_j, t_{ij} is the timestamp, re_{ij} is the review given by user u_i to item p_j, and c_{ij} is the evaluation image of user u_i to item p_j. Let $\mathcal{U}_b$ and $\mathcal{I}$ denote the sets of all users (benign users) and items, respectively. For convenience, in this paper, r_{ij}, re_{ij}, and c_{ij} are collectively referred to as rating behaviors r_{ij}.

Definition 2. The synthetic attack dataset $\mathcal{D}_a$ is a set of tuples $< u_i^a, p_j, r_{ij} >$, where u_i^a is the attacker and r_{ij} is the rating given by u_i^a to item p_j. Additionally, r_{ij} can be further divided into r_{ij}^f and r_{ij}^a, where r_{ij}^f represents the user's filling behavior, that is, imitating the rating behavior of benign users, and r_{ij}^a represents the user's rating for the target item $\mathcal{I}_t$. Let $\mathcal{U}_a$ denote the set of all attackers. Attackers need to carefully select an appropriate attack target item $\mathcal{I}_t$ to make it appear as much as possible in the user's recommendation list $\mathcal{V}_u^{\text{rec}}$. The attack dataset $\mathcal{D}_a$ and the benign dataset $\mathcal{D}_b$ are combined to form the final detection dataset $\mathcal{D}$.

Definition 3. To effectively detect malicious users, we transform the rating behavior of users and items into a user-user association graph $\mathcal{G} = \{\mathcal{U}, \mathcal{E}, \mathcal{W}\}$, where $\mathcal{U}$ represents the set of users, $\mathcal{E}$ and $\mathcal{W}$ are the set of edges $< u_i, u_j >$ and their corresponding weights, respectively. Specifically, whether an edge is constructed between users depends on whether they have rated at least μ items together, and the edge weight is the number of co-rated items.

Based on $\mathcal{D}$ and $\mathcal{G}$, this paper aims to design a detection model $\mathcal{M}$ to identify as many $\mathcal{U}_a$ as possible from $\mathcal{D}$ and analyze the distribution characteristics of $\mathcal{I}_t$ to identify $\mathcal{I}_t$. For the real unlabeled data $\mathcal{D}_u$ (similar to $\mathcal{D}_b$), we use $\mathcal{M}$ to discover behavior data similar to $\mathcal{D}_a$ from $\mathcal{D}_u$ based on the prior knowledge learned from $\mathcal{D}$. Additionally, by analyzing its previous distribution attributes, we discover suspicious target items $\mathcal{I}_s$ from $\mathcal{D}_u$ and analyze the possibility of fake rating behavior through counterfactual evidence.

B. Method Overview

In this section, we will briefly introduce the basic process of the proposed detection method. As shown in Figure 2, STOP is divided into four key steps:

1) **Distributional Co-occurrence Graph Induction (Step-1)**: We construct node behavior representations by quantifying the distribution preference of ratings and build edge structures through co-occurrence association.
2) **Distributional-Consensus Importance Screening (DCIS, Step-2)**: DCIS uses distributional similarity and importance weight to pre-screen high-value training data on the input side, improving the signal-to-noise ratio and reducing data imbalance in advance.

TABLE I
THE BEHAVIOR REPRESENTATION OF USER NODES.

Feature	Name	Explanation
Act_u	User activity level	$Act_u = \sum_{i \in \mathcal{I}_u} I_{r_{ui} \neq \varnothing}$
$\bar{R}_u$	Average rating point of the user	$\bar{R}_u = \frac{1}{\lvert\mathcal{I}_u\rvert} \sum_{i \in \mathcal{I}_u} r_{ui}$
$\bar{R}_u^P$	Preference for popular items	$\bar{R}_u^P = \frac{1}{\lvert\mathcal{I}_u^P\rvert} \sum_{i \in \mathcal{I}_u^P} r_{ui}$
$\bar{R}_u^C$	Preference for unpopular items	$\bar{R}_u^C = \frac{1}{\lvert\mathcal{I}_u^C\rvert} \sum_{i \in \mathcal{I}_u^C} r_{ui}$
$\bar{D}_u^{KL}$	Degree of deviation in ratings	$\bar{D}_u^{KL} = \frac{1}{\lvert\mathcal{I}_u\rvert} \sum_{i \in \mathcal{I}_u} D_{KL}(r_{ui}; Q_i)$

3) **Multi-View Relational Disentanglement and Information-Consistent Fusion (MRIF, Step-3)**: MRIF removes trivial relationships; divides key relationships into static and dynamic relationships; and simplifies and retains the relationships crucial to the classification task. The three-stage purification of abnormal evidence can effectively control common coupling phenomena and characterize HCPAs.
4) **Orthogonal Projection Bi-Hypersphere Boundary Learning (OPBL. Step-4)**: OPBL employs two concentric hyperspheres to define a closed normal region and makes judgments based on distance, which can simultaneously eliminate both collapsed anomalies that are too close and outlier anomalies that are too far away.

C. Distributional Co-occurrence Graph Induction

We define the user node features as $\mathcal{F}_u = \{Act_u, \bar{R}_u, \bar{R}_u^P, \bar{R}_u^C, \bar{D}_u^{KL}\}$. The specific feature definitions are shown in Table I. Here, $\mathcal{I}_u$ denotes the set of items that the user u has rated, $\mathcal{I}_u^P$ denotes the item vector of u rated for popular items, $\mathcal{I}_u^C$ is the item vector of u rated for unpopular items, and $\bar{D}_u^{KL}$ is the Kullback-Leibler (KL) divergence of the rating r_{ui} of u and the historical distribution Q_i of item i. To define popular items and unpopular items, we select items with the top 10% of the number of ratings as popular items, and items with the bottom 10% of the number of ratings as unpopular items.

Meanwhile, to construct the relationship between user nodes, we propose a coarse-grained edge construction method based on the co-occurrence association. If users u_1 and u_2 rate at least μ items in common (μ is a preset threshold, $|r_{u_1} \cap r_{u_2}| \geq \mu$), then an edge $e_{u_1 u_2}$ is created in the co-occurrence association graph $\mathcal{G}$. In this paper, μ is set to 10, aiming to build symbiotic edges to the greatest extent while eliminating weakly associated relationships. The weight of $e_{u_1 u_2}$ is represented as $w_{u_1,u_2} = \mid r_{u_1} \cap r_{u_2} \mid$. To mine the dynamic and static behaviors in real data, we divide $\mathcal{G} = \{\mathcal{U}, \mathcal{E}, \mathcal{W}\}$ into $\mathcal{G}_{1:T}$ according to the rating sequence. However, the method may introduce noise, such as accidental rating overlaps. Therefore, in the design of subsequent DCIS methodology (see § III-D), these noises must be eliminated to enhance the accuracy of the malicious user discovery process.

D. Distributional-Consensus Importance Screening (DCIS)

To promote or demote the target item, attackers often simulate the rating behavior of benign users, thereby making malicious users show high similarity with representative benign users. Therefore, by eliminating benign users that have relatively weak associations with malicious users, such as new users, inactive users, and niche users, the problem of data imbalance can be alleviated. Based on this, we propose DCIS, which filters out benign users with low correlation to malicious users. For each user's rating sequence $\mathcal{R}(u) = \{r_1, r_2, ..., r_n\}$, DCIS generates the user's rating distribution based on Kernel Density Estimation (KDE),

$$\hat{f}(u) = \frac{1}{nh\sqrt{2\pi}} \sum_{i=1}^{n} \exp\left(-\frac{(u-u_i)^2}{2h^2}\right), \tag{3}$$

where n represents the sample size, i.e., the number of ratings given by user u, and h is the bandwidth parameter. For the rating distributions of users i and j, the Wasserstein distance is used to measure the similarity between two distributions,

$$\mathrm{Sim}_w(i,j) = \frac{1}{1 + \left(\inf_{\gamma \in \Gamma(i,j)} \int_{X \times Y} d(x,y)\, d\gamma(x,y)\right)}, \tag{4}$$

where $\Gamma(i,j)$ is the set of all joint distributions with i and j as marginal distributions, and $d(x,y)$ is the distance metric between points x and y in space. Through similarity measurement, DCIS constructs the relationships between users as a similarity matrix $\mathbf{S} \in \mathrm{Sim}_w^{|\mathcal{U}| \times |\mathcal{U}|}$. At the same time, DCIS generates similarity features, $\mathcal{X}_u = [s_{\text{avg}}(u), s_{\text{min}}(u), s_{\text{max}}(u)]$. These features are the average similarity score, the highest similarity score, and the lowest similarity score with u, respectively. Based on $\mathcal{X}_u$, DCIS defines a pre-filter $\mathcal{P}_{\text{pred}}$, which is trained using a joint strategy of the Autoencoder (AE) and group distribution alignment. The process can be defined as,

$$\begin{aligned} \mathcal{Y}_{\text{pred}}(u) &= \mathcal{P}_{\text{pred}}\Big(f_{\theta^\star}(\mathcal{X}_u);\, \Theta_{\text{pred}}^\star\Big), \\ (\theta^\star, \phi^\star, \mu^\star) &= \arg\min_{\theta,\phi,\mu} \Bigg[\frac{1}{n}\sum_{i=1}^{n} \Big\| \mathcal{X}_i - g_\phi\big(f_\theta(\mathcal{X}_i)\big)\Big\|_2^2 \\ &\qquad + \lambda_{\text{align}} \sum_g W_1\Big(\mathbb{P}_g^z,\, \mu\Big)\Bigg], \end{aligned} \tag{5}$$

where $\Theta_{\text{pred}}^\star$ is the parameter of $\mathcal{P}_{\text{pred}}$, $f_\theta(\cdot)$ is the encoder, $g_\phi(\cdot)$ is the decoder, $\frac{1}{n}\sum_{i=1}^{n} \big\| \mathcal{X}_i - g_\phi\big(f_\theta(\mathcal{X}_i)\big)\big\|_2^2$ is reconstruction loss of the AE, $\lambda_{\text{align}} \sum_g W_1\Big(\mathbb{P}_g^z,\, \mu\Big)$ is the group distribution alignment term, and $\mathcal{Y}_{\text{pred}} \in [0,1]$. Meanwhile, based on the consistency of ratings between the attacker and representative benign users, DCIS uses the mean of the latent space as the "group center", i.e., $\bar{z} = \frac{1}{n}\sum_{i=1}^{n} z_i$. The smaller the distance of the user, the higher the importance score $\mathcal{Y}_{\text{pred}}$. Therefore, DCIS optimizes $\mathcal{P}_{\text{pred}}$ by minimizing the group consistency loss $\mathcal{L}_{\text{consist}}$,

$$\mathcal{L}_{\text{consist}} = \frac{1}{n}\sum_{i=1}^{n} \|z_i - \bar{z}\|_2^2 \cdot \big(1 - \mathcal{Y}_{\text{pred}}(i)\big). \tag{6}$$

Furthermore, low-activity malicious users may be zombie users or pose no attack threat. High-activity malicious users are not only costly but also prone to exposing the attack behavior. Therefore, we adjust $\mathcal{Y}_{\text{pred}}$ based on user activity Act_u,

$$\mathcal{Y}_{\text{pred}}(u) = \mathcal{P}_{\text{pred}}\big(f_{\theta^\star}(\mathcal{X}_u); \Theta_{\text{pred}}^\star\big) \cdot (Act_u)^\gamma, \tag{7}$$

Algorithm 1 Distributional-Consensus Importance Screening.

Require: User set $\mathcal{U}$, Rating sequences $\{\mathcal{R}(u) \mid u \in \mathcal{U}\}$, Pre-filter $\mathcal{P}_{\text{pred}}$, Parameter Θ_{pred}, User activity Act_u, Drop ratio ∇_{drop}, Hyperparameter γ.

Ensure: Importance scores $\mathcal{Y}_{\text{pred}}$, Filtered user set $\widehat{\mathcal{U}}$.

1: **for** each user $u \in \mathcal{U}$ **do**
2: KDE for user's rating distribution $\hat{f}(u)$; (Eq. 3)
3: **end for**
4: **for** users $u_i, u_j \in \mathcal{U}$ **do**
5: Compute Wasserstein similarity $\text{Sim}_w(u_i, u_j)$; (Eq. 4)
6: **end for**
7: Build similarity matrix $\mathbf{S} \in \text{Sim}_w^{|\mathcal{U}|\times|\mathcal{U}|}$;
8: **for** each user $u \in \mathcal{U}$ **do**
9: $\mathcal{X}_u = [s_{\text{avg}}(u), s_{\text{min}}(u), s_{\text{max}}(u)]$;
10: $\mathcal{Y}_{\text{pred}}(u) = \mathcal{P}_{\text{pred}}\big(f_{\theta^\star}(\mathcal{X}_u); \Theta^\star_{\text{pred}}\big) \cdot (Act_u)^\gamma$; (Eq. 7)
11: **end for**
12: Sort $\mathcal{U}$ by $\mathcal{Y}_{\text{pred}}$ in descending order;
13: $\widehat{\mathcal{U}} = \text{TOP}_{(1-\nabla_{\text{drop}})}(\mathcal{Y}_{\text{pred}})$;
14: **return** $\mathcal{Y}_{\text{pred}}$, $\widehat{\mathcal{U}}$.

where γ is a hyperparameter used to adjust the density distribution. Users with excessively high or low activity will be subject to a constraint. Ultimately, we sort $\mathcal{Y}_{\text{pred}}$ and set a drop ratio ∇_{drop}, retaining the top $1-\nabla_{\text{drop}}$ users. The specific process of DCIS is shown in Algorithm 1. Steps 1-3 generate the rating distribution $\hat{f}(u)$ of users. Steps 4-7 generate the user similarity matrix $\mathbf{S} \in \text{Sim}_w^{|\mathcal{U}|\times|\mathcal{U}|}$ based on Wasserstein similarity. Steps 8-11 generate the importance score $\mathcal{Y}_{\text{pred}}$ of users based on the similarity matrix $\mathbf{S}$. Finally, steps 12-13 retain the TOP $1 - \nabla_{\text{drop}}$ users.

E. Multi-View Relational Disentanglement and Information-Consistent Fusion (MRIF)

To eliminate the influence of accidental co-occurrence in graph structures and effectively describe the characteristics of attack behaviors, we propose a multi-view relational disentanglement and information-consistent fusion method (MRIF). This method eliminates trivial accidental co-occurrences and weak interaction relationships, divides key relationships into static and dynamic relationships to characterize the features of attack behaviors, and ultimately simplifies and retains the relationships that are crucial for the anomaly classification task. Specifically:

1) ***Probabilistic Key Relation Estimation (PKE)****:* Based on probabilistic edge recovery and contrastive denoising, PKE filters out trivial relationships such as rating coincidences and weak interactions to obtain an association subgraph of high confidence. Leveraging the variational graph AE (VGAE), we calculate the existence probability of the key relationship [51]. For the $1:T$ rating sequences, key soft mask matrixs $\mathbb{M}^K_{1:T}$ are,

$$\mathbb{M}^K_{1:T} = f_v\left(\mathcal{U}, \mathcal{E}_{1:T}, \mathcal{W}_{1:T}, \Theta_K\right) = p\left(H^K_{1:T} \middle| \mathcal{G}_{1:T}\right) q(\mathbb{M}^K_{1:T} \middle| H^K_{1:T}), \tag{8}$$

where f_v is an encoder $p(\cdot)$ - decoder $q(\cdot)$ structure parameterized by Θ_K. The $p(\cdot)$ captures the latent distribution of $\mathcal{G}_{1:T}$, thereby supporting the generation of new graph structures. Then, the $q(\cdot)$ uses the latent representation to generate the interpretable key edge probability pb_{ij} between nodes i and j,

$$\begin{aligned} p(H \mid \mathcal{U}, \mathcal{E}, \mathcal{W}) &= \mathcal{N}\big(H \mid \mu,\, \text{diag}(\sigma^2)\big) \\ pb_{ij} &= \upsilon\big(W^\top [\, H_i \parallel H_j \,] + b\big)\,, \end{aligned} \tag{9}$$

where H is the latent matrix, μ and σ are the mean and variance of the node latent embeddings learned by the graph convolution neural (GCN) model with different parameters, $\upsilon(\cdot)$ is sigmoid function, $W \in 2^{d_{\text{latent}}\times 1}$ is the learnable weight vector, and $b \in \mathrm{R}$ is the learnable bias.

Using key soft masks $\mathbb{M}^K_{1:T}$, we extract key relationships $\mathcal{G}^K_{1:T}$ in the coarse-grained graph and remove redundant trivial relationships $\mathcal{G}^T_{1:T}$, i.e.,

$$\mathcal{G}_{1:T} = \begin{cases} \mathcal{G}^T_{1:T} = (\mathcal{U}, \mathcal{E}_{1:T} \oplus \bar{\mathbb{M}}^K_{1:T}, \mathcal{W}^T_{1:T}) \\ \mathcal{G}^K_{1:T} = (\mathcal{U}, \mathcal{E}_{1:T} \oplus \mathbb{M}^K_{1:T}, \mathcal{W}^K_{1:T}) \end{cases}, \tag{10}$$

where $\mathbb{M}^K_{1:T}$ are key soft masks at rating sequences $1:T$, and $\bar{\mathbb{M}}^K_{1:T}$ are complementary trivial soft masks. Since the key subgraph $\mathcal{G}^K$ is the target subgraph, it contains important relationships. The trivial subgraph $\mathcal{G}^T$, on the other hand, can be treated as noise. Therefore, key relationships should be close to the original relationships. In contrast, trivial relationships should be regarded as negative relationships. Based on this, we extract key relationships through CL,

$$\mathcal{L}_{\text{contrast}} = \max\left(\frac{\langle \mathcal{G}, \mathcal{G}^T \rangle}{\|\mathcal{G}\|\,\|\mathcal{G}^T\|} - \frac{\langle \mathcal{G}, \mathcal{G}^K \rangle}{\|\mathcal{G}\|\,\|\mathcal{G}^K\|}, 0\right). \tag{11}$$

Through CL, the $\mathcal{G}^K$ is made to approach the original graph $\mathcal{G}$, while $\mathcal{G}^T$ is made to move away from $\mathcal{G}$. Ultimately, we extract $\mathcal{G}^K$ through an empirical threshold ϱ.

PKE recovers key edges in a probabilistic manner and combines a contrastive denoising mechanism to bring the graph structure related to the task closer while pushing away irrelevant trivial associations, thereby generating high-confidence key subgraphs $\mathcal{G}^K_{1:T}$. These subgraphs effectively suppress the rating coupling phenomenon (i.e., the behavior where a large number of benign users give similar ratings to the same item due to exogenous factors such as item popularity, marketing activities, or simultaneous exposure) and weak interactions (referring to the low-reliability similarities and low-confidence edges caused by the sparsity of user-item interactions), and thus has stronger robustness.

2) ***Dynamic–Static Relation Separation (DRS)****:* On the rating series graph, key relationships are further decomposed into static and dynamic relationships, the former representing long-term consistency and the latter focusing on short-term attack to the target item. Built upon key relationship subgraphs $\mathcal{G}^K_{1:T}$, we again rely on VGAE [51] to separate dynamic and static relationships in $\mathcal{G}^K_{1:T}$. The dynamic soft masks $\mathbb{M}^D_{1:T}$ of rating sequences $1:T$ can be represented as,

$$\begin{aligned} \mathbb{M}^D_{1:T} &= f_v\left(\mathcal{U}, \mathcal{E}_{1:T} \oplus \mathbb{M}^K_{1:T}, \mathcal{W}^K_{1:T}, \Theta_D\right) \\ &= p\left(H^D_{1:T} \middle| \mathcal{G}^K_{1:T}\right) q(\mathbb{M}^D_{1:T} \middle| H^D_{1:T}). \end{aligned} \tag{12}$$

The $\mathcal{G}^K_{1:T}$ can be represented as,

$$\mathcal{G}_{1:T}^{K} = \begin{cases} \mathcal{G}_{1:T}^{D} & = (\mathcal{U}, \mathcal{E}_{1:T} \oplus \mathbb{M}_{1:T}^{K} \oplus \mathbb{M}_{1:T}^{D}, \mathcal{W}_{1:T}^{D}) \\ \mathcal{G}^{S} & = (\mathcal{U}, \mathcal{E}_{1:T} \oplus \mathbb{M}_{1:T}^{K} \oplus \bar{\mathbb{M}}_{1:T}^{D}, \mathcal{W}_{1:T}^{S}) \end{cases}, \quad (13)$$

where $\bar{\mathbb{M}}_{1:T}^{D}$ are denoted as static soft masks complementary to $\mathbb{M}_{1:T}^{D}$. Specifically, dynamic relationship $\mathcal{G}_{T}^{D}$ can be inferred from the historical dynamic relationships $\mathcal{G}_{1:(T-1)}^{D}$, while static relationship $\mathcal{G}_{T}^{S}$ is independent of the historical static relationships $\mathcal{G}_{1:(T-1)}^{s}$. Formally, we have $H_{1:(T-1)}^{D} \rightarrow H_{\mathrm{T}}^{D}, H_{1:(T-1)}^{s} \perp H_{\mathrm{T}}^{s}$. Based on this, we use the dynamic consistency loss $\mathcal{L}_{\text{dynamic}}$ to force the model to maintain dynamic consistency in the rating sequences, i.e., the hidden states at adjacent rating sequences should be as similar as possible,

$$\mathcal{L}_{\text{dynamic}} = \frac{1}{T-1} \sum_{t=2}^{T} \|H_t - H_{t-1}\|_2^2, \quad (14)$$

where $\|\cdot\|_{2}$ is the ℓ_2 regularization. The $\mathcal{L}_{\text{dynamic}}$ ensures that dynamic relationship subgraphs $\mathcal{G}_{1:T}^{D}$ is highly relevant. The remaining key relationships are independent, thereby determining the static relationship subgraph $\mathcal{G}^{S}$. Ultimately, we separate $\mathcal{G}_{1:T}^{D}$ and $\mathcal{G}^{S}$ through an empirical threshold ς.

DRS disentangles the key subgraph into static (long-term consistent) and dynamic (short-term collaborative attacks) components, jointly regularized by temporal coherence as well as sparsity and gating mechanisms, thereby enabling the separation of attacks without excessive smoothing.

*3) **Cross-Graph Information-Consistent Fusion (CIF)**:* The representations learned from the dynamic and static perspectives are fused through information consistency and redundancy suppression strategies in a task-oriented manner, outputting a highly discriminative relationship summary representation for classification decisions. After PKE and DRS, the graph can be defined as $\{\mathcal{G}_{1:T}^{D}, \mathcal{G}^{S}\}$. To learn a fused high-quality graph $\mathcal{G}^{F}$ from multiple subgraphs, we propose a cross-graph information-consistent fusion method (CIF) [52]. The CIF retains the shared task-related information and the unique key information of each subgraph. For each subgraph, CIF utilizes the GCN model for feature aggregation. Meanwhile, node embeddings are generated via a two-layer attention mechanism (AM). Subsequently, to obtain the refined graph structure, CIF performs sparse, symmetric, and normalized processing using the K-Nearest Neighbors (KNN) algorithm. The loss function $\mathcal{L}_{\text{fusion}}$ of CIF could be represented as,

$$\begin{aligned} \mathcal{L}_{\text{fusion}} = & \underbrace{-\frac{2}{N(N-1)} \sum_{n=1}^{N} \sum_{m=n+1}^{N} I(\mathcal{G}_n^F; \mathcal{G}_m^F)}_{\textbf{Inter-refined MI (max)}} \\ & \underbrace{-\frac{1}{N} \sum_{n=1}^{N} I(\mathcal{G}_n^F; \mathcal{G}_n')}_{\textbf{Retain task info}} \underbrace{-\frac{1}{N} \sum_{n=1}^{N} I(\mathcal{G}^F; \mathcal{G}_n^F)}_{\textbf{Fusion-refined MI (max)}}, \end{aligned} \quad (15)$$

where "Inter-refined MI (max)" denotes maximizing the mutual information between different refined perspectives, "Fusion-refined MI (max)" denotes maximizing the mutual information between the integrated perspective and each refined perspective, and $\mathcal{G}_n'$ is the optimal expansion graph. The $\mathcal{L}_{\text{fusion}}$ uses a lower boundary estimator $I_{lb}\left(Z^i; Z^j\right)$ based on node representations, calculated through cosine similarity and temperature parameters τ_c. To maximize the task-related information in the subgraph, first, CIF maximizes the mutual information between different subgraphs through CL. Second, to maximize the unique information in each subgraph, CIF retains task-related information through graph augmentation. Finally, CIF concatenates the node features of all subgraphs. And it generates a fusion graph $\mathcal{G}^{F}$ while maximizing the mutual information between the fusion graph and each refined graph through an AM.

Algorithm 2 Multi-View Relational Disentanglement and Information-Consistent Fusion.

Require: Co-occurrence graphs $\mathcal{G}_{1:T} = \{\mathcal{G}_1, \ldots, \mathcal{G}_T\}$, VGAE parameters Θ_K, Θ_D, Key mask threshold ϱ, Dynamic mask threshold ς.
Ensure: Fusion graph $\mathcal{G}^{F}$.
1: **for** co-occurrence graph $\mathcal{G}_t$ in $\mathcal{G}_{1:T}$ **do**
2: $p\left(H \mid \mathcal{U}, \mathcal{E}, \mathcal{W}\right) = \mathcal{N}\big(H \mid \mu, \mathrm{diag}(\sigma^2)\big)$; (Eq. 9)
3: **for** user node pair (v_i, v_j) in $\mathcal{G}_t$ **do**
4: Calculate edge probability pb_{ij}:
5: $pb_{ij} = \upsilon(W^T[H_i \parallel H_j] + b)$; (Eq. 9)
6: **end for**
7: **end for**
8: Generate key soft masks $\mathbb{M}_{1:T}^{K}$:
9: $\mathbb{M}_{1:T}^{K} = p(H_{1:T}^{K}|\mathcal{G}_{1:T})q(\mathbb{M}_{1:T}^{K}|H_{1:T}^{K})$; (Eq. 8)
10: Extract key subgraphs: $\mathcal{G}_{1:T}^{K} = \mathcal{G}_{1:T} \odot \mathbb{M}_{1:T}^{K}$;
11: Generate dynamic masks:
12: $\mathbb{M}_{1:T}^{D} = p(H_{1:T}^{D}|\mathcal{G}_{1:T}^{K})q(\mathbb{M}_{1:T}^{D}|H_{1:T}^{D})$; (Eq. 12)
13: Extract dynamic subgraphs: $\mathcal{G}_{1:T}^{D} = \mathcal{G}_{1:T}^{K} \odot \mathbb{M}_{1:T}^{D}$;
14: Extract static subgraph: $\mathcal{G}^{S} = \mathcal{G}_{1:T}^{K} \odot \bar{\mathbb{M}}_{1:T}^{D}$;
15: Fusion graph $\mathcal{G}^{F} = \mathrm{GCN}\big(\{\mathcal{G}_{1:T}^{D}, \mathcal{G}^{S}\}\big)$; (Eq. 15)
16: **return** $\mathcal{G}^{F}$.

CIF maximizes cross-graph consistency, fuses static and dynamic relationships, and explicitly suppresses redundant information. This method enhances the complementary features that are most valuable for downstream tasks while retaining common information. The representations learned by this method have a larger margin between normal and abnormal classes and exhibit stronger stability in detecting HCPAs.

MRIF resolves the issues of insufficient denoising ability and difficulty in feature characterization of previous graph association mining methods. This makes HCPAs more likely to be exposed in topological relationships. The specific process is shown in Algorithm 2. Step 2 generates the latent representation H of graph $\mathcal{G}$. Steps 3-6 calculate the existence probability pb_{ij} of the relationship between users. Steps 8-9 generate key soft masks $\mathbb{M}_{1:T}^{K}$ based on steps 1-7. Step 10 extracts key relationship subgraphs $\mathcal{G}_{1:T}^{K}$ based on step 9. Steps 11-12 generate dynamic soft masks $\mathbb{M}_{1:T}^{D}$ based on steps 1-7. Steps 13-14 extract dynamic relationship subgraphs $\mathcal{G}_{1:T}^{D}$ and static key relationship subgraph $\mathcal{G}^{S}$. Step 15 generates the fusion graph $\mathcal{G}^{F}$ based on steps 14-15.

F. Orthogonal Projection Bi-Hypersphere Boundary Learning (OPBL)

To identify potential abnormal user nodes from the fusion graph $\mathcal{G}^F$, we propose an orthogonal projection bi-hypersphere boundary learning method (OPBL) [53] inspired by the idea of the "convergence theorem". The OPBL constructs two concentric hyperspheres (with inner radius $r_{\min}$ and outer radius $r_{\max}$) to confine the decision region of normal data within the shell between them. First, we propose an orthogonal projection mechanism in OPBL, which generates an orthogonal projection matrix $\mathbf{W}^*$ through singular value decomposition (SVD). The SVD could project the potential representation $\tilde{Z}$ into an orthogonal space, effectively eliminating feature correlations and aligning the distribution closer to a standard hypersphere. As a result, the projected features satisfy the constraint $\tilde{Z}^\top \tilde{Z} = \mathbf{I}_{k'}$. Here, $\mathbf{I}_{k'}$ is an $k' \times k'$ unit matrix, k' is the projected dimension. Then, $r_{\max}$ is determined by selecting the $1-\nu$ quantile of the distance distribution, i.e., $r_{\max} = \text{argmin}_r P(D \leq r_{\max}) \geq \nu$; $r_{\min}$ is selected as the ν quantile of the distance distribution, i.e., $r_{\min} = \text{argmin}_r P(D \leq r_{\min}) \geq 1-\nu$. Here, P denotes the cumulative distribution function and D denotes the set of distance distributions of users. The optimization objective of OPBL is to adjust the network parameters Θ and W to distribute the normal data as much as possible within the shell,

$$\min_{\Theta,\mathcal{W}} \frac{1}{b}\sum_{i=1}^{b}(\max\{d_i, r_{\max}\} - \min\{d_i, r_{\min}\}) + \frac{\lambda}{2}\sum_{W\in\mathcal{W}} \parallel W \parallel_F^2, \quad (16)$$

where the first term forces the benign data points to be within the shell layer. If $d_i < r_{\min}$ or $d_i > r_{\max}$, the user is determined as abnormal; if $r_{\min} < d_i < r_{\max}$, the user is determined as normal. The second term is ℓ_2 regularization to prevent overfitting. During the testing phase, based on the determined inner and outer radius, the abnormal conditions of each node can be determined. The anomaly score function can be defined as $s_i = (d_i - r_{\max}) \cdot (d_i - r_{\min})$. If $s_i > 0$, the user is determined as abnormal; if $s_i < 0$, the user is determined as normal.

IV. Evaluation & Experiments

This section is dedicated to addressing the following key research questions (**RQs**):

1) How to evaluate the generalization ability and detection performance of the proposed STOP under different attacks and datasets?
2) What is the contribution of each step of STOP to the overall performance improvement?
3) How efficient is the operation of STOP? Under what conditions is STOP effective, and when does it fail?
4) How do the relevant parameters of STOP affect the detection performance?
5) How to prove the effectiveness of STOP in real-world data and verify the reliability of the detection results?

A. Dataset, Attack Methods and Settings

1) ***Real-word Data****:* We adopt three real-world datasets (i.e., ML-1M [54], FilmTrust [55], and Douban [56]) as the datasets for anomaly detection. Additionally, we choose four real unlabeled datasets (i.e., Amazon-Book [57], Amazon-Movies & TV [57], Dianping [58], and JingDong (JD) [59]) as the datasets for anomaly forensics, as detailed in Table II. In the anomaly detection phase and in the anomaly forensics phase, we set T to 5.

TABLE II
Details of the Real-World Dataset

Dataset	Users	Items	Ratings	Rating Range	Sparsity
ML-1M [54]	5,950	3,659	1,000,000	1-5	95.41%
FilmTrust [55]	1,515	2,100	35,584	0.5-5	98.88%
Douban [56]	2,848	39,586	894,887	1-5	99.21%
Amazon-Book [57]	165,531	505,651	1,000,000	1-5	99.99%
Amazon-Movies&TV [57]	177,374	211,352	1,000,000	1-5	99.99%
Dianping [59]	218,165	50,594	1,000,000	1-5	99.99%
JD [58]	512,268	20,370	1,000,000	1-5	99.99%

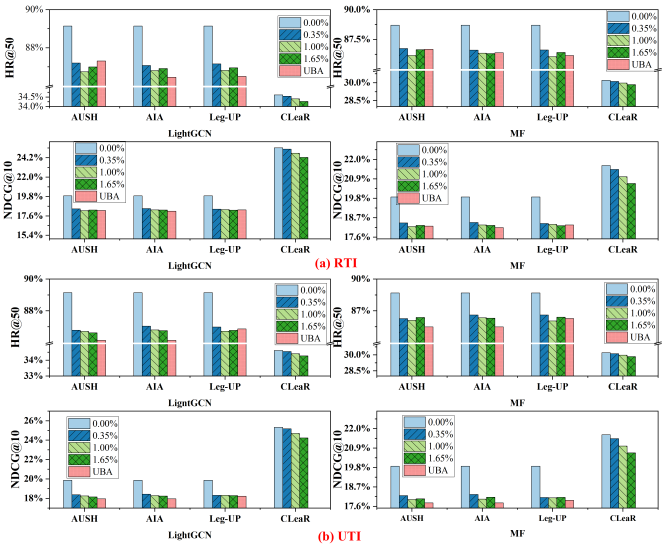


Fig. 3. The validation of attack effectiveness. We conduct experimental evaluations on two recommendation models (i.e., MF and LightGCN) and two attack scenarios (i.e., RTI and UTI) based on the ML-1M dataset. The experiments focus on the impact of attack size on the recommendation performance metrics HR@50 and NDCG@10. Among them, the results of HR@50 and NDCG@10 are the average performance under the same attack size and different filling sizes.

2) ***Attack Data Synthesis****:* Specifically, in the controllable attacks (i.e., AIA [30], AUSH [32], Bandwagon [60], CLeaR [29], Leg-UP [31], and Random [60]), we employ different attack sizes (ASs) (i.e., 0.35%, 1%, and 1.65%) and filler sizes (FSs) (i.e., 0.55%, 0.82%, 1.09%, and 1.37%). In the uncontrollable attacks (i.e., AIA, AUSH, and Leg-UP in the UBA [28] framework), we set the attack budget to 100. In all attacks, the attack target is set in two different ways: random target items (RTI) and unpopular target items (UTI). Therefore, each attack pattern is injected into the real data to construct the final experimental data. As shown in Figure 3, we verify the effectiveness of the current five popular HCPA methods on the LightGCN [61] and MF [58] recommendation models. These methods significantly reduce the recommendation performance (e.g., HR@50 and NDCG@10). Since the CLeaR [29] converts the rating into implicit feedback, i.e., like $\rightarrow$ 1; dislike $\rightarrow$ 0, HR@50 is lower than other attack methods, and NDCG@10 is higher than other attack

TABLE III
IN THE RTI SCENARIO, THE COMPARISON OF DEFENSE RESULTS OF DIFFERENT ATTACK METHODS. HERE, WE USE THE AVERAGE OF FOUR FILLER SIZES (I.E., 0.55%, 0.82%, 1.09%, AND 1.37%) AS THE EVALUATION CRITERIA FOR DR AND FAR. ADDITIONALLY, THE BEST RESULTS IN EACH DATASET ARE SHOWN IN BOLD, AND THE SECOND-BEST RESULTS ARE UNDERLINED.

Dataset	Method	Metric	AIA			AUSH			Bandwagon			CLeaR			Leg-UP			Random			UBA		
			0.35%	1.00%	1.65%	0.35%	1.00%	1.65%	0.35%	1.00%	1.65%	0.35%	1.00%	1.65%	0.35%	1.00%	1.65%	0.35%	1.00%	1.65%	AIA	AUSH	Leg-UP
ML-1M	CNN-BAG	DR	0.975	1.000	1.000	0.972	0.991	1.000	1.000	1.000	1.000	0.995	0.988	0.973	0.938	1.000	1.000	1.000	1.000	1.000	1.000	1.000	**1.000**
		FAR	0.027	0.074	0.078	0.021	0.039	0.042	0.100	0.100	0.100	0.000	0.000	0.050	0.007	0.006	0.084	0.060	0.100	0.125	0.034	0.029	0.002
	CoDetector	DR	1.000	0.871	0.233	0.804	0.809	0.708	0.854	0.864	0.976	0.923	0.895	0.835	0.875	0.892	0.828	0.375	0.642	0.892	0.714	0.833	1.000
		FAR	0.006	0.012	0.015	0.001	0.005	0.006	0.018	0.036	0.038	0.029	0.075	0.025	0.004	0.075	0.001	0.009	0.075	0.000	0.000	0.016	0.000
	DegreeSAD	DR	0.125	0.193	0.328	0.364	0.419	0.802	1.000	1.000	1.000	0.963	0.938	0.998	0.563	0.866	0.832	1.000	0.847	0.664	0.379	0.817	0.450
		FAR	0.003	0.008	0.012	0.003	0.006	0.006	0.001	0.001	0.000	0.000	0.000	0.000	0.000	0.002	0.005	0.016	0.034	0.000	0.007	0.003	0.013
	FAP	DR	0.700	0.349	0.262	1.000	1.000	0.569	1.000	0.960	0.553	0.928	0.888	0.903	1.000	0.875	0.658	0.975	0.815	0.542	0.552	0.544	0.533
		FAR	0.007	0.006	0.005	0.007	0.001	0.000	0.053	0.005	0.006	0.052	0.052	0.025	0.006	0.004	0.000	0.028	0.012	0.006	0.000	0.000	0.000
	GAD	DR	1.000	1.000	0.978	0.833	0.985	1.000	1.000	1.000	1.000	0.955	0.988	1.000	1.000	0.600	0.680	1.000	1.000	1.000	1.000	0.955	0.955
		FAR	0.033	0.017	0.022	0.034	0.026	0.026	0.015	0.050	0.000	0.050	0.088	0.075	0.000	0.038	0.050	0.010	0.068	0.050	0.000	0.032	0.024
	STOP	DR	**1.000**	**1.000**	0.998	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.985	0.998	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
		FAR	**0.000**	**0.000**	**0.001**	**0.001**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	0.002	**0.000**	**0.000**	0.001	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
FilmTrust	CNN-BAG	DR	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.980	0.970	0.978	0.938	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
		FAR	0.089	0.100	0.125	0.026	0.024	0.065	0.087	0.087	0.114	0.002	0.000	0.050	0.028	0.028	0.069	0.069	0.071	0.098	0.065	0.000	0.055
	CoDetector	DR	0.900	0.581	0.758	0.950	0.962	0.954	0.608	0.753	0.919	0.988	1.000	0.990	0.875	0.892	0.828	0.754	0.876	0.931	0.762	1.000	0.947
		FAR	0.009	0.075	0.008	0.004	0.075	0.000	0.005	0.075	0.003	0.029	0.051	0.025	0.004	0.075	0.002	0.005	0.075	0.001	0.009	0.100	0.008
	DegreeSAD	DR	0.479	0.821	0.948	0.955	1.000	1.000	0.875	1.000	1.000	1.000	0.965	1.000	0.563	0.866	0.832	1.000	1.000	0.979	1.000	1.000	1.000
		FAR	0.006	0.016	0.003	0.000	0.002	0.000	0.001	0.006	0.010	0.000	0.100	0.076	0.000	0.000	0.000	0.003	0.000	0.007	0.000	0.100	0.000
	FAP	DR	0.775	0.620	0.363	1.000	1.000	0.575	0.925	0.745	0.458	0.963	0.958	1.000	1.000	0.875	0.658	1.000	0.920	0.533	0.575	0.299	0.544
		FAR	0.018	0.012	0.010	0.017	0.004	0.000	0.017	0.010	0.005	0.028	0.052	0.025	0.017	0.002	0.000	0.017	0.004	0.001	0.000	0.000	0.002
	GAD	DR	1.000	0.972	0.974	1.000	1.000	1.000	1.000	1.000	1.000	0.980	0.983	0.980	1.000	0.600	0.680	1.000	1.000	1.000	1.000	1.000	1.000
		FAR	0.014	0.058	0.056	0.008	0.046	0.054	0.014	0.060	0.058	0.027	0.088	0.075	0.002	0.044	0.055	0.012	0.057	0.062	0.023	0.000	0.038
	STOP	DR	**1.000**	995	0.994	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
		FAR	**0.000**	**0.003**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.003**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	0.003	0.003
Douban	CNN-BAG	DR	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.940	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
		FAR	0.087	0.095	0.125	0.054	0.075	0.105	0.049	0.053	0.089	0.000	0.000	0.050	0.081	0.062	0.071	0.100	0.100	0.125	0.097	0.015	0.005
	CoDetector	DR	0.448	0.332	0.526	0.517	0.621	0.560	0.417	0.520	0.733	0.973	0.988	1.000	1.000	0.726	0.588	0.988	0.988	0.964	0.667	1.000	1.000
		FAR	0.007	0.075	0.005	0.005	0.075	0.004	0.006	0.075	0.004	0.020	0.075	0.019	0.005	0.075	0.000	0.004	0.075	0.000	0.000	0.016	0.000
	DegreeSAD	DR	0.479	0.386	0.797	0.432	0.846	0.983	0.833	0.250	0.554	0.995	1.000	1.000	0.413	0.346	0.154	1.000	1.000	1.000	0.724	1.000	0.733
		FAR	0.006	0.015	0.008	0.008	0.002	0.000	0.000	0.000	0.000	0.000	0.053	0.057	0.006	0.004	0.005	0.000	0.000	0.000	0.003	0.000	0.005
	FAP	DR	0.825	0.552	0.390	1.000	1.000	0.560	1.000	0.930	0.528	1.000	0.968	0.978	0.955	0.977	0.705	0.522	0.536	0.533	0.575	0.204	0.556
		FAR	0.009	0.009	0.002	0.009	0.003	0.000	0.009	0.003	0.000	0.013	0.021	0.024	0.011	0.010	0.004	0.045	0.034	0.019	0.000	0.002	0.000
	GAD	DR	1.000	0.865	1.000	0.850	0.981	1.000	1.000	1.000	1.000	1.000	0.960	1.000	0.833	1.000	0.833	0.908	1.000	1.000	1.000	1.000	1.000
		FAR	0.005	0.044	0.054	0.005	0.045	0.052	0.000	0.039	0.050	0.024	0.070	0.070	0.000	0.046	0.051	0.000	0.038	0.050	0.000	0.008	0.000
	STOP	DR	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.975	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
		FAR	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	0.001	0.001	**0.000**	**0.000**	**0.000**	0.002	**0.002**	**0.000**	**0.001**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**

methods. Based on this, we obtain a total of 450 attack datasets ($6\times3\times3\times4\times2 + 3\times3\times2$), where the former represents the six attack methods, three real datasets, three ASs, four FSs, and two attack targets in controllable attacks; the latter represents the three different attack methods, three real datasets, one optimization budget, and two attack targets in uncontrollable attacks. These poisoned data are respectively injected into the base datasets, ultimately forming our detection datasets.

*3) **Evaluation Metric**:* We use the detection rate (DR) and false alarm rate (FAR) to evaluate the detection performance of all methods, which are defined as follows,

$$DR = \frac{|\mathcal{U}_d \cap \mathcal{U}_a|}{|\mathcal{U}_a|}, FAR = \frac{|\mathcal{U}_d \cap \mathcal{U}_b|}{|\mathcal{U}_b|}, \tag{17}$$

where $\mathcal{U}_d$ is the set of detected malicious users.

B. Baseline Methods and Assessment of Indicators

To comprehensively verify the effectiveness of the STOP proposed in this paper, we select baselines on two mainstream technical routes. (1) Explicit feature characterization: statistical outlier detection and rating consistency detection; (2) Association graph mining: dense subgraph discovery and GNN anomaly detection. Further, we conduct a comprehensive comparison of the detection results with the following latest open-source detection methods:

1) CNN-BAG [43]: The method is based on the CNN and bagging algorithm to extract the features of PAs automatically. Specifically, we train 15 independent CNN models, with each CNN model trained for 500 epochs.
2) CoDetector [45]: The method uses user latent factors containing network embedding to detect PAs. Specifically, we set the number of latent factor dimensions to 10 and the common rating threshold to 5.
3) DegreeSAD [42]: The method extracts features from the attributes of user profiles and item popularity, and then uses machine learning classification methods to detect PAs. Specifically, we use the mean of user degree, range of user degree, and quartile of user degree as user features.
4) FAP [44]: The method relies on recursive bipartite graph propagation to estimate the probability that each user is a spam account. Specifically, we dynamically determine the TOP-K prediction number based on the probability distribution and set the number of seed users to 10.
5) GAD [46]: The method is based on the GNN to improve the robustness of RSs based on GNN. Specifically, we define the node features of the bipartite graph as degree, average rating, and standard deviation of ratings.

C. Performance Evaluation for Detection

In this section, we conduct an in-depth and comprehensive analysis of all the detection results and answer the research questions **RQ1-RQ5** in sequence.

*1) **Overall Comparison (RQ1)**:* To evaluate the generalization ability and detection performance of STOP, we conduct extensive experiments across different attacks and datasets. We compare STOP with five representative baseline methods. The model parameters of STOP will be discussed in § IV-C5 and

TABLE IV
IN THE UTI SCENARIO, THE COMPARISON OF DEFENSE RESULTS OF DIFFERENT ATTACK METHODS. HERE, WE USE THE MEAN VALUES OF FOUR FILLER SIZES (I.E., 0.55%, 0.82%, 1.09%, AND 1.37%) AS THE EVALUATION CRITERIA FOR DR AND FAR. MOREOVER, THE BEST RESULTS IN EACH DATASET ARE SHOWN IN BOLD, AND THE SECOND-BEST RESULTS ARE UNDERLINED.

Dataset	Method	Metric	AIA			AUSH			Bandwagon			CLeaR			Leg-UP			Random			UBA		
			0.35%	1.00%	1.65%	0.35%	1.00%	1.65%	0.35%	1.00%	1.65%	0.35%	1.00%	1.65%	0.35%	1.00%	1.65%	0.35%	1.00%	1.65%	AIA	AUSH	Leg-UP
ML-1M	CNN-BAG	DR	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.980	0.985	0.988	1.000	1.000	1.000	0.988	1.000	1.000	1.000	1.000	1.000
		FAR	0.040	0.073	0.099	0.000	0.001	0.050	0.006	0.014	0.064	0.060	0.039	0.102	0.007	0.008	0.067	0.010	0.017	0.067	0.000	0.000	0.004
	CoDetector	DR	0.988	0.988	0.964	0.888	0.876	0.955	0.708	0.235	0.402	1.000	1.000	0.973	0.833	0.864	0.906	0.413	0.324	0.316	1.000	1.000	1.000
		FAR	0.005	0.075	0.004	0.004	0.075	0.000	0.005	0.075	0.002	0.035	0.075	0.009	0.004	0.075	0.001	0.005	0.075	0.002	0.000	0.016	0.001
	DegreeSAD	DR	1.000	1.000	1.000	0.886	1.000	1.000	0.958	0.979	0.996	0.980	0.988	0.990	0.563	0.866	0.832	0.688	0.993	0.975	1.000	1.000	0.317
		FAR	0.003	0.008	0.012	0.000	0.000	0.000	0.000	0.000	0.000	0.047	0.070	0.053	0.000	0.002	0.005	0.000	0.000	0.001	0.000	0.000	0.016
	FAP	DR	0.522	0.536	0.533	1.000	1.000	0.575	1.000	0.940	0.536	1.000	1.000	1.000	1.000	0.889	0.658	0.950	0.720	0.517	0.196	0.196	0.544
		FAR	0.005	0.006	0.002	0.005	0.003	0.000	0.005	0.003	0.000	0.022	0.015	0.025	0.006	0.004	0.000	0.005	0.003	0.000	0.000	0.000	0.000
	GAD	DR	0.908	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.989	1.000	0.983	0.983	1.000	0.600	0.690	1.000	1.000	1.000	1.000	1.000	1.000
		FAR	0.008	0.052	0.059	0.004	0.041	0.051	0.016	0.053	0.058	0.017	0.053	0.081	0.006	0.038	0.050	0.009	0.042	0.050	0.000	0.008	0.000
	STOP	DR	**1.000**	0.996	0.993	**1.000**	**1.000**	**1.000**	**1.000**	0.996	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.992	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
		FAR	**0.000**	**0.000**	0.010	**0.000**	**0.000**	**0.000**	**0.000**	0.002	0.002	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	0.002	0.002	**0.000**	0.007	**0.000**
FilmTrust	CNN-BAG	DR	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
		FAR	0.040	0.000	0.050	0.032	0.026	0.075	0.087	0.087	0.115	0.081	0.057	0.095	0.029	0.027	0.073	0.078	0.070	0.098	0.000	0.000	0.060
	CoDetector	DR	0.446	0.628	0.657	0.896	0.930	0.963	0.713	0.787	0.844	0.973	0.973	0.980	0.972	0.946	0.951	0.875	0.702	0.914	1.000	1.000	1.000
		FAR	0.007	0.075	0.011	0.004	0.075	0.000	0.005	0.075	0.003	0.026	0.075	0.014	0.005	0.075	0.006	0.005	0.075	0.007	0.000	0.016	0.004
	DegreeSAD	DR	0.563	0.714	0.853	0.955	1.000	1.000	0.833	1.000	1.000	1.000	0.950	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.250
		FAR	0.011	0.019	0.026	0.000	0.002	0.000	0.001	0.007	0.014	0.043	0.069	0.053	0.000	0.002	0.000	0.004	0.000	0.000	0.000	0.000	0.006
	FAP	DR	0.800	0.620	0.424	1.000	1.000	0.575	0.975	0.700	0.458	0.981	0.985	0.985	0.894	1.000	0.575	1.000	0.940	0.536	0.299	0.299	1.000
		FAR	0.017	0.012	0.005	0.017	0.004	0.000	0.017	0.012	0.005	0.007	0.014	0.009	0.017	0.004	0.000	0.017	0.004	0.001	0.000	0.000	0.015
	GAD	DR	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.978	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
		FAR	0.014	0.058	0.060	0.009	0.046	0.056	0.008	0.055	0.062	0.005	0.071	0.073	0.002	0.049	0.058	0.012	0.057	0.062	0.000	0.000	0.018
	STOP	DR	0.713	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
		FAR	**0.000**	0.007	**0.007**	**0.000**	**0.000**	**0.000**	**0.000**	**0.002**	**0.003**	**0.000**	**0.002**	**0.005**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	0.007
Douban	CNN-BAG	DR	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.990	0.993	0.998	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
		FAR	0.036	0.021	0.050	0.010	0.013	0.057	0.032	0.049	0.080	0.086	0.061	0.090	0.010	0.012	0.057	0.011	0.024	0.065	0.007	0.000	0.007
	CoDetector	DR	0.792	0.323	0.448	0.720	1.000	0.988	0.688	0.652	0.709	0.996	0.943	0.983	0.896	0.959	0.962	0.667	0.959	0.962	0.667	0.196	1.000
		FAR	0.004	0.075	0.006	0.004	0.075	0.000	0.005	0.075	0.006	0.006	0.075	0.005	0.004	0.075	0.001	0.005	0.075	0.007	0.000	0.016	0.000
	DegreeSAD	DR	0.271	0.464	0.784	0.955	0.985	1.000	0.854	0.319	0.783	1.000	0.973	1.000	0.955	0.985	1.000	0.938	0.985	1.000	0.724	0.196	1.000
		FAR	0.006	0.007	0.011	0.002	0.001	0.000	0.000	0.000	0.000	0.086	0.070	0.039	0.002	0.001	0.000	0.003	0.000	0.000	0.018	0.000	0.000
	FAP	DR	0.850	0.552	0.361	1.000	1.000	0.575	1.000	0.940	0.528	0.988	0.988	0.988	1.000	1.000	0.575	1.000	1.000	0.575	1.000	1.000	0.196
		FAR	0.009	0.009	0.003	0.009	0.003	0.000	0.009	0.003	0.000	0.025	0.023	0.002	0.009	0.003	0.000	0.009	0.004	0.001	0.000	0.000	0.000
	GAD	DR	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.990	0.990	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
		FAR	0.005	0.039	0.056	0.004	0.041	0.052	0.001	0.041	0.050	0.041	0.044	0.074	0.003	0.041	0.053	0.000	0.040	0.050	0.000	0.000	0.000
	STOP	DR	0.988	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
		FAR	**0.002**	**0.002**	0.004	**0.001**	**0.000**	**0.000**	0.002	**0.000**	0.001	**0.000**	**0.000**	**0.001**	**0.001**	**0.001**	**0.000**	0.004	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**

the final parameters will be determined. Table III-IV show the DR and FAR of STOP and baseline methods under different attack methods, attack targets, ASs, and FSs.

As shown in Tables III-IV, STOP outperforms all competing baseline methods in almost all cases. In all scenarios, STOP achieves a high DR (i.e., 12.34% higher than the baseline methods) and a low FAR (i.e., 2.75% lower than the baseline methods). The experimental results confirm that STOP can effectively identify HCPAs. Another observation is that in the UTI scenario and the FilmTrust dataset, when the AS of AIA is 0.35%, the DR of STOP is only 71.3%. Although there are only 6 attack users at this time, STOP does not show a good detection effect compared with other methods. Therefore, in the § IV-C4, we will explore the detection boundary of STOP against other attack methods. From the perspective of baseline methods, CNN-BAG [43] and GAD [46] outperform other baseline methods in most cases, but have relatively high FARs. This result indicates that while CNN-BAG and GAD improve the DR, they misjudge some benign samples, leading to more severe misclassification. Additionally, the detection performance of CoDetector [45], DegreeSAD [42], and FAP [44] is slightly lower than other methods, and they show poor stability. This result suggests that these three methods fail to effectively detect HCPAs. It should be noted that we use the mean values of four FSs (i.e., 0.55%, 0.82%, 1.09%, and 1.37%) as the evaluation criteria for DR and FAR.

*2) **Module Comparison (RQ2)**:* To verify the contribution of each step in STOP to the overall performance improvement, we validated the effectiveness of each step. Specifically, we focused on verifying the effectiveness of DCIS (§ III-D), PKE (§ III-E1), DRS (§ III-E2), and CIF (§ III-E3). By adding the above four steps to our method, we evaluated the improvement effect of each step on the overall performance of STOP. As shown in Figure 4, we compared the FAR under different attack methods, attack targets, ASs, and datasets. To more clearly present the research results of each step, we use the "red line" to indicate the change trend of FAR.

The results show that after the four-step trivial relationship removal, key relationship decoupling, and information-consistent fusion, the FAR of STOP approaches 0. This

TABLE V
THE EFFICIENCY PERFORMANCE COMPARISON BETWEEN THE PROPOSED METHOD AND BASELINE METHODS (CPU TIME, MINUTES (M) AND SECONDS (S)), TAKING THE CLEAR ATTACK AS AN EXAMPLE, WHERE THE AS IS 1.00% AND THE FS IS 0.82%.

Method	Time complexity	Space complexity	Dataset	Time
CNN-BAG	$\mathcal{O}(\mathcal{C}\cdot\mathcal{U})$	$\mathcal{O}(\mathcal{U}+\mathcal{R})$	ML-1M	12m15s
			Douban	11m43s
CoDetector	$\mathcal{O}(\mathcal{U}^2)$	$\mathcal{O}(\mathcal{N}+\mathcal{R})$	ML-1M	36m14s
			Douban	26m27s
DegreeSAD	$\mathcal{O}(\mathcal{U}\log\mathcal{U})$	$\mathcal{O}(\mathcal{U}+\mathcal{R})$	ML-1M	27m45s
			Douban	20m05s
FAP	$\mathcal{O}(\mathcal{U}\log\mathcal{U})$	$\mathcal{O}(\mathcal{N}+\mathcal{R})$	ML-1M	33m14s
			Douban	22m08s
GAD	$\mathcal{O}(\mathcal{N}\cdot\mathcal{R})$	$\mathcal{O}(\mathcal{N}+\mathcal{R})$	ML-1M	43m50s
			Douban	28m05s
STOP	$\mathcal{O}(\mathcal{U}^2)$	$\mathcal{O}(\mathcal{U}^2)$	ML-1M	32m12s
			Douban	24m59s

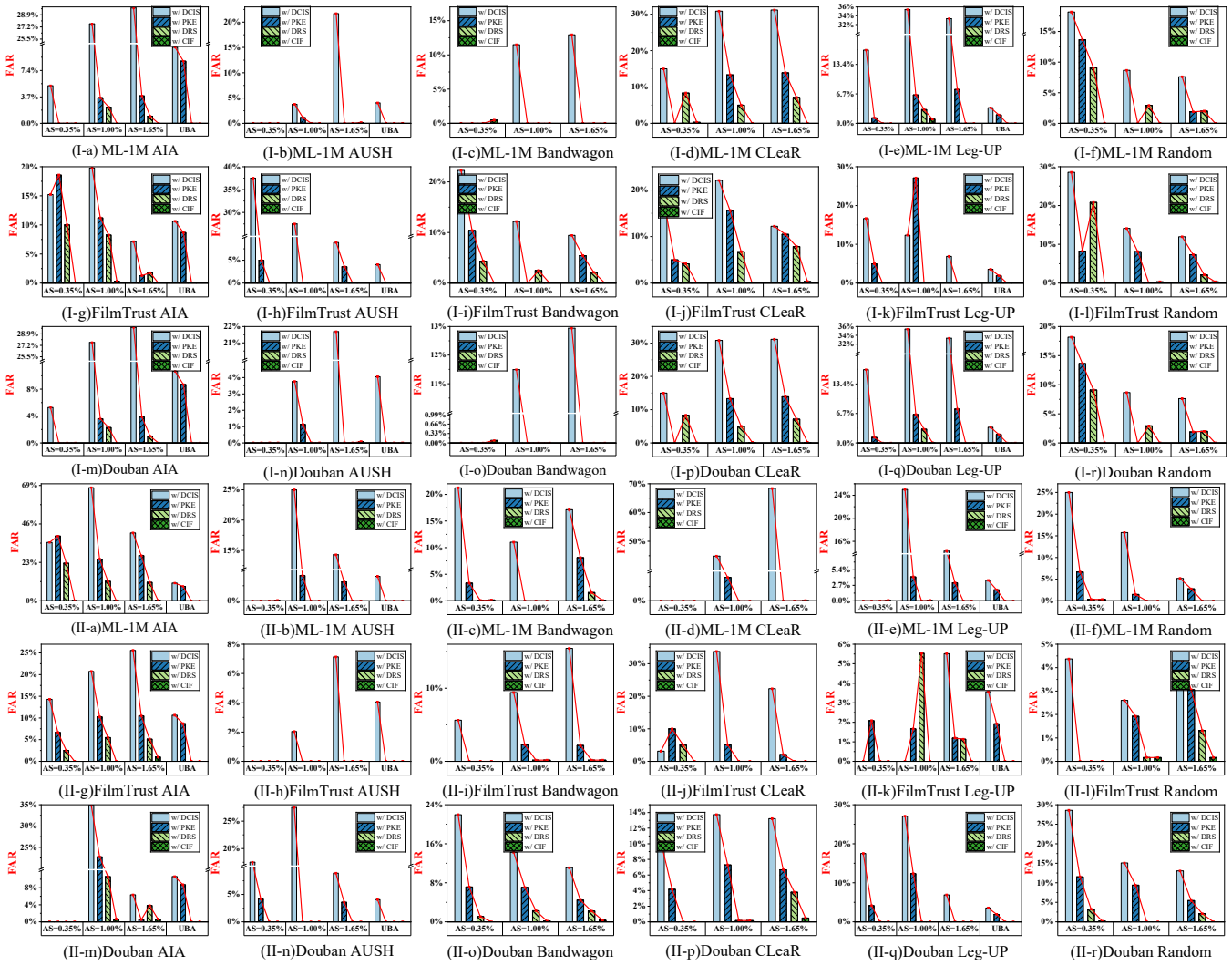


Fig. 4. The module comparison of STOP's defense performance under different attack sizes. In the RTI (subfigure I) and UTI (subfigure II) scenarios, the defense effects against different attack methods were compared. Here, we used the average of four fill sizes (i.e., 0.55%, 0.82%, 1.09%, and 1.37%) as the evaluation criterion for FAR.

indicates that in almost all cases, each step of STOP simplifies and retains the relationships crucial for the classification task. Additionally, in the AIA [30], AUSH [32], CLeaR [29], Leg-UP [31], and Random [43] attacks, from the with (w/) DCIS to w/ PKE, and then to the w/ DRS step, STOP shows significant convergence in FAR. However, in the ML-1M and Douban datasets, and the AS is 0.35%, the Bandwagon's detection result [43] begins to show an abnormal increase in the FAR after the w/ CIF step. As shown in Figure 4 (I-c) and 4 (I-o), at this point, some benign samples are judged as abnormal samples, and STOP reaches the detection boundary. Therefore, in § IV-C4, we further explore the defense boundaries of STOP against other attack methods. It should be noted that to present all experimental scenarios to the readers, this section only shows the change trend of FAR.

*3) **Running Efficiency and Complexity (RQ3)**:* To analyze the time and space complexity of the detection, we conducted additional experiments for all proposed methods and presented the results in the form of time in Table V. Here, $\mathcal{C}$ denotes the number of base learners, $\mathcal{R}$ denotes the number of ratings, and $\mathcal{N}$ denotes the total number of users and items. Please note that all experiments were carried out on a server equipped with an Intel(R) Xeon(R) Gold 6248 CPU @ 2.50GHz and 256GB of memory. We can observe that under the premise that the DR is higher than the baseline method and the FAR is lower than the baseline method, the efficiency of STOP is relatively satisfactory. The computational time and resource consumption of STOP mainly occur in the distributional co-occurrence graph induction and DCIS steps. Among them, the construction of the user similarity matrix $\mathbf{S} \in \mathrm{Sim}_w^{|\mathcal{U}|\times|\mathcal{U}|}$ in DDCF increases the time complexity and space complexity. However, STOP reduces the time and resource consumption of subsequent steps by filtering out benign users with low association with malicious users through DCIS. Therefore, compared with the baseline methods, STOP still shows certain advantages in terms of computational time. Undoubtedly, our experimental results are for reference only.

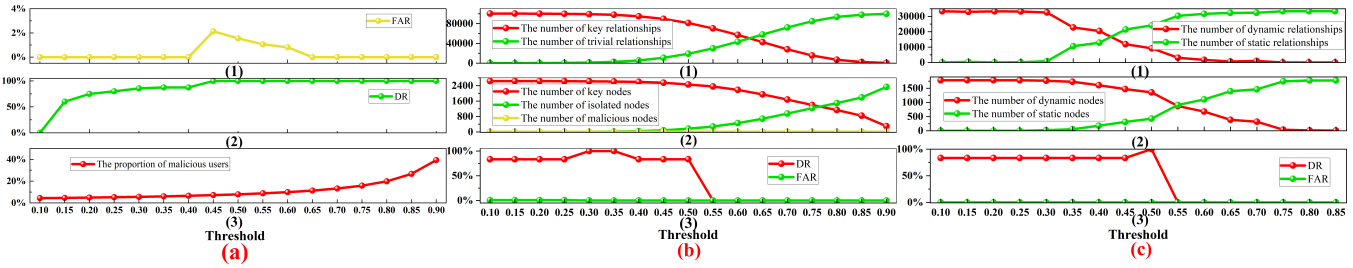


Fig. 5. The analysis of parameter sensitivity, where (a) represents the sensitivity of ∇_{drop} in DCIS; (b) represents the sensitivity of ϱ in PKE; and (c) represents the sensitivity of ς in DRS. Here, the dataset is ML-1M [54], AS = 0.1%, FS = 1.36%, and attack target is RTI.

TABLE VI
THE BOUNDARY EXPLORATION OF THE PROPOSED STOP ON THE DOUBAN DATASET. HERE, FS IS SET TO 1.36% AND THE ATTACK TARGET IS RTI.

Attack Method		Metric	Attack Size (AS)			
			0%	0.1%	0.2%	0.35%
CLeaR [29]		DR	-	0.0000	1.0000	1.0000
		FAR	-	0.0000	0.0000	0.0000
		HR@50	0.3469	0.3466	0.3453	0.3447
AUSH [32]		DR	-	1.0000	1.0000	1.0000
		FAR	-	0.0000	0.0000	0.0000
		HR@50	0.7883	0.7842	0.7747	0.7699
Leg-UP [31]		DR	-	1.0000	1.0000	1.0000
		FAR	-	0.0000	0.0000	0.0000
		HR@50	0.7883	0.7826	0.7799	0.7777
AIA [30]		DR	-	0.0000	0.0000	1.0000
		FAR	-	0.0000	0.0000	0.0000
		HR@50	0.7883	0.7789	0.7771	0.7678
UBA [28]	AUSH [32]	DR	-	1.0000	1.0000	1.0000
		FAR	-	0.0000	0.0000	0.0000
		HR@50	0.7883	0.7677	0.7677	0.7677
	Leg-UP [31]	DR	-	1.0000	1.0000	1.0000
		FAR	-	0.0000	0.0000	0.0000
		HR@50	0.7883	0.7463	0.7463	0.7463
	AIA [30]	DR	-	1.0000	1.0000	1.0000
		FAR	-	0.0000	0.0000	0.0000
		HR@50	0.7883	0.7556	0.7556	0.7556
Bandwagon [43]		DR	-	0.0000	1.0000	1.0000
		FAR	-	0.0000	0.0035	0.0035
		HR@50	0.7883	0.7867	0.7841	0.7835
Random [43]		DR	-	0.0000	1.0000	1.0000
		FAR	-	0.0000	0.0035	0.0000
		HR@50	0.7883	0.7887	0.7887	0.7870

*4) **The Boundaries of the Method (RQ3)**:* To further explore the effectiveness boundary of STOP, we investigate under what conditions STOP can be effective and when it would almost or completely fail. Due to the concealment of HCPAs, we only discuss the lower boundary for now and do not involve the upper boundary. Based on this, we conduct experiments with gradually decreasing the AS. As shown in Table VI, we take the Douban [56] dataset as an example. The AS is set to 0%, 0.1%, 0.2%, and 0.35% respectively, the FS is set to 1.36%, and the attack target is RTI. To deeply analyze the impact of STOP on the RS, we use HR@50 as the metric to evaluate the performance of the LightGCN [61] model.

From the results, when AS = 0.2%, the number of malicious users is only 6. When AS = 0.1%, the number of malicious users is only 3. The experimental results show that when facing AIA [30], Bandwagon [43], CLeaR [29], and Random [43] attacks, STOP has a DR of 0 and a FAR of 0. The reason for this phenomenon is that the number of attackers is too small, which makes STOP unable to effectively identify abnormal patterns, thus predicting all samples as benign. Therefore, under these four attack methods, STOP has reached the lower boundary of detection. In addition, we also observe that as the attack scale decreases, HR@50 shows an increasing trend (since the CLeaR [29] converts the rating into implicit feedback, i.e., like $\rightarrow$ 1; dislike $\rightarrow$ 0, HR@50 is lower than other methods). This is because the reduction in the number of attackers gradually weakens the impact of HCPAs, thus making the recommendation performance closer to the results of the original clean data. Lightweight HPCAs (e.g., AS equal to 0.1% or 0.2%) are not sufficient to cause substantial damage to the RS.

*5) **Parameter Sensitivity Analysis (RQ4)**:* This paper adopts a phased and modular design approach to ensure that each component reaches its optimal state at its corresponding level, and verifies the rationality of various parameter configurations through the performance of the overall system. To explore the impact of parameter sensitivity on model performance, we discuss the empirical values of the discard ratio ∇_{drop} of DCIS, the probabilistic key relation estimation threshold ϱ of PKE, and the dynamic–static relation separation threshold ς of DRS.

a) The Discard Ratio ∇_{drop} of DCIS (§ III-D): We set the discard ratio ∇_{drop} of DCIS as the empirical threshold. As shown in Figure 5 (a), as ∇_{drop} increases, the proportion of malicious users in the dataset increases, and the detection data gradually shows a balanced state. However, when $\nabla_{drop} < 0.45$, the detection ability of STOP is insufficient. When $\nabla_{drop} \geq 0.45$, the detection ability of STOP tends to be stable, but a small number of benign users are misjudged as malicious users. It is not until $\nabla_{drop} \geq 0.65$, the performance of STOP is truly stable. Furthermore, when the data balance exceeds 11.39% (i.e., the proportion of malicious users among all users, excluding benign users filtered by the DCIS method), STOP is capable of achieving perfect detection performance.

b) Sensitivity Analysis of ϱ in PKE (§ III-E1): As shown in Figure 5 (b), we introduce a threshold ϱ to filter out trivial relationships. Specifically, when the edge weight is below ϱ, the edge in the graph is regarded as a trivial relationship. When the edge weight is above ϱ, it is considered a key

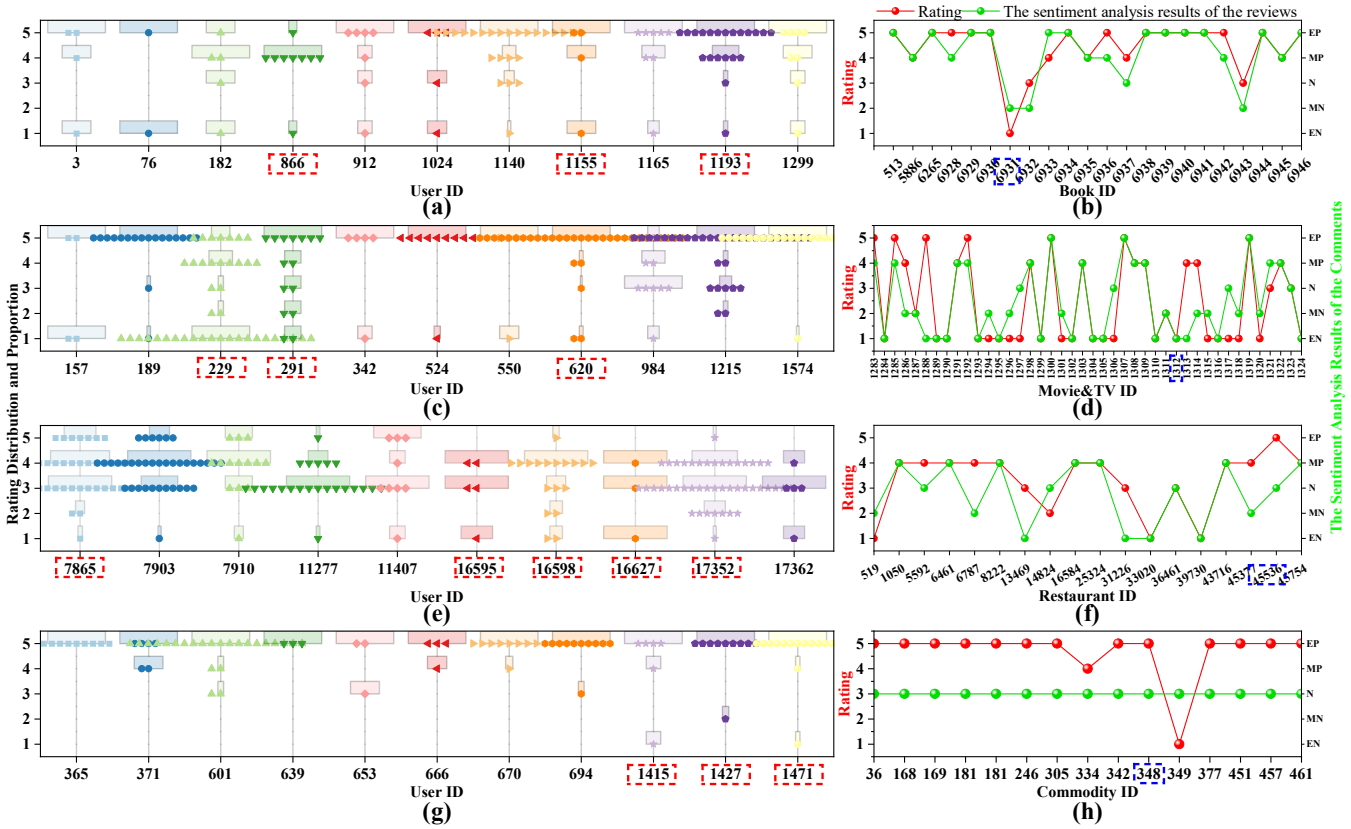


Fig. 6. Interesting findings in real unlabeled data. (a) and (b) represent interesting findings in the Amazon-Book dataset, (c) and (d) represent interesting findings in the Amazon-Movies & TV dataset, (e) and (f) represent interesting findings in the Dianping dataset, and (g) and (h) represent interesting findings in the JD dataset.

relationship. To this end, we explore the impact of ϱ on model performance within the range of 0.1 to 0.9. As ϱ increases, both the number of key relationships and key nodes gradually decrease. Conversely, the number of trivial relationships and isolated nodes increases. Since malicious nodes are closely related to the detection task, these nodes always exist in the key subgraph. Additionally, as ϱ increases, the DR of STOP initially increases, maintaining a 100% DR between 0.3 and 0.35, but then drops sharply. The experimental results indicate that as the edge structure decreases, OPBL fails to extract effective information related to the anomaly rating from key relationships, resulting in DR = 0 when $\varrho > 0.55$. To retain relevant information while allowing DRS to function effectively, we retain as many edge structures as possible under the premise of ensuring detection performance. Therefore, we set $\varrho = 0.3$.

c) Sensitivity Analysis of ς in DRS (§ III-E2): As shown in Figure 5 (c), we introduce an empirical threshold ς to set the separation ratio in DRS. Specifically, when the edge weight is lower than ς, the current edge is identified as a static relationship. When the edge weight is higher than ς, it is identified as a dynamic relationship. To this end, we explore the impact of ς on model performance within the range of 0.1 to 0.85. As ς increases, the number of static relationships and static nodes increases, while the number of dynamic relationships and dynamic nodes decreases. From the detection results, when $\varsigma \leq 0.45$, DR $\approx$ 83.33%. This is because the attacker simulates the rating behavior of benign users, and the rating behavior can be mapped to static relationships. When $\varsigma > 0.5$, DR = 0. This is because the attacker will push or nuke the target item, and this process can be mapped to dynamic relationships. Therefore, we set ς to 0.5 to balance the separation effect of DRS.

D. Interesting Finding (*RQ5*)

To verify the effectiveness of STOP in real-world data and evaluate the reliability of the detection results, we select four representative datasets (i.e., Amazon-Book dataset [57], Amazon-Movies & TV dataset [57], the Dianping review dataset [58], and the JingDong (JD) dataset [59]), and conduct a series of experiments on these datasets. We only collect one million rating records for analysis. Meanwhile, to protect user privacy, we use re-indexed user IDs and item IDs in the experiments to ensure the anonymity of the data.

Furthermore, to verify the consistency between user reviews and ratings, we utilize ChatGPT-5 Thinking [62] to conduct sentiment analysis on the user review texts. Specifically, we classify the sentiment of the review texts into five levels: extremely negative (EN), moderately negative (MN), neutral (N), moderately positive (MP), and extremely positive (EP), corresponding to the rating of 1 to 5, respectively. As shown in Figure 6, the left subgraphs (i.e., a, c, e, and g) represent the distribution of ratings by malicious users detected in the four

datasets, i.e., the proportion of each user's ratings from 1 to 5. The right subgraphs (i.e., b, d, f, and h) are the comparison results of the rating behaviors and review sentiment analyses of representative users. The following are the key findings:

*1) **Amazon-Book**:* In the Amazon-Book [57] dataset, we detect a total of 11 malicious users. As shown in Figure 6(a), the rating behaviors of these users are extreme, mainly concentrated at 1 or 5 stars. Taking a user (ID: 1193) as an example, the user (ID: 1193) rates a total of 22 books; only a low rating is given to the book (ID: 6931), while full marks are given to 14 books. This extreme rating pattern suggests that the user (ID: 1193) may carry out a nuke rating.

As shown in Figure 6(b), we further analyze the review content of the user (ID: 1193). The result indicates that when the user (ID: 1193) gives a higher rating, the sentiment analysis result of the review is relatively low; while when the rating is lower, the sentiment analysis result of the review is relatively high. Meanwhile, the review style of the user (ID: 1193) for 8 high-rating books is highly consistent, mostly simple and general positive reviews, such as "*I like this book*". However, in the only low-rating review for the book (ID: 6931), the user (ID: 1193) expresses a rather regretful attitude, with the review "*Wish for a better ending*". The contradiction between the simple and general style presented in high-rating reviews and the emotionally colored regretful attitude in low-rating reviews further confirms the abnormality of user behavior.

*2) **Amazon-Movies & TV**:* In the Amazon-Movies & TV [57] dataset, we detect 11 malicious users whose ratings exhibit obvious extreme characteristics. As shown in Figure 6(c), the ratings of these users are mainly concentrated at 1 and 5, while the number of ratings in other ranges is relatively small. Taking a user (ID: 229) as an example, as shown in Figure 6(d), the user evaluates a total of 42 movies and TV works, with 22 low ratings and 16 high ratings. This extreme distribution of ratings leads to the determination of the user (ID: 229) as an abnormal one.

In the analysis of review styles, strong negative words such as "*Awful*", "*Mess*", and "*Crap*" frequently appear in the low-rating reviews of the user (ID: 229); while in the high-rating reviews, positive words like "*Well-acted*" and "*Very worth watching*" are more common. Additionally, through the sentiment analysis results of reviews, we further discover significant differences between ratings and review behaviors of the user (ID: 229). Notably, among the 3-4 point ratings given by the user (ID: 229), the reviews of movies are extremely negative, such as "*Subpar*", "*Too boring*", and "*It's not going to win an Oscar for sure*". This mismatch between the rating and the content of the review further highlights the abnormality of user behavior.

*3) **Dianping**:* In the Dianping [58] dataset, we detect a total of 10 potential malicious users. As shown in Figure 6(e), the ratings in this dataset are mainly concentrated at 3 points, which is significantly different from the common 4-5 point rating distribution in the Amazon dataset [57]. Taking a user (ID: 16598) as an example, the user (ID: 16598) reviews 18 restaurants on the Dianping platform, among which there are 2 low-rating reviews, 10 4-point reviews, and only 1 full mark review.

As shown in Figure 6(f), among the 10 4-point reviews, the user (ID: 16598) provides relatively comprehensive and detailed reviews of the restaurants from multiple dimensions such as "Flavor", "Environment", "Service", and "Location". From this evaluation style, it seems that the user (ID: 16598) is a serious and cautious reviewer. However, the "five-star review" of the user (ID: 16598) for the restaurant (ID: 45536) stands out. In this review, the user merely uses the brief and general expression "*Very convenient*", and the sentiment analysis result of this review is "neutral". The inconsistency between this review style and the content of the highest-rated review also provides a strong clue for identifying and verifying the potentially malicious behavior of the user (ID: 16598).

*4) **JD**:* In the JD [59] dataset, we detect a total of 11 malicious users. As shown in Figure 6(g), the ratings of these malicious users are mostly concentrated at 5 points, and except for the user with ID 365, the others give very few negative reviews. Take a user (ID: 1471) as an example, as shown in Figure 6(h), the user (ID: 1471) purchases 15 commodities but only leaves one low-rating review. The sentiment analysis results are all "neutral", showing no obvious preference for the commodities. Moreover, in the high-rating reviews given by the user (ID: 1471), the reviews are incoherent. For instance, in the review for the commodity (ID: 348), the user simply repeats "*Sure*" six times. Additionally, in other reviews, the user (ID: 1471) increases the length of the reviews by filling them with punctuation marks. This behavior indicates that the user (ID: 1471) is attempting to avoid having reviews identified as "high rating but low quality" by increasing the length of the reviews.

Finally, we conduct a suspicious target analysis on the rating behaviors of suspicious users. By systematically sorting out the items commented on by suspicious users, we identify several target items that might have been manipulated. As shown in Figure 6, users who rate these suspicious target items are marked with a "blue box"; meanwhile, the rating behaviors of suspicious users towards target items are displayed with a "red box", aiming to reveal their potential abnormal patterns. Among them, we found that in the Amazon-Book and JD datasets, several suspicious users give low ratings to the target items; while in the Amazon Movies & TV and Dianping datasets, there are abnormal phenomena where high ratings are given but the review content is negative. The above behavioral patterns suggest that there may be intentional demotion manipulation targeting specific items.

These experimental results demonstrate that STOP can effectively identify the behavioral patterns of malicious users, especially in terms of extreme ratings, contradictory review styles, and analysis of target items. Although these observations are based on specific data, they provide strong clues for anomaly forensics and show the effectiveness and reliability of STOP in real-world data. It is worth noting that these signals (e.g., extreme ratings, inconsistent emotions, and suspicious target items) may have non-malicious explanations (e.g., promotional event impacts and gift card usage). Therefore, these observations are for reference only. The final judgment requires cross-validation with platform logs and multi-source evidence.

V. Conclusion and Future Work

This paper proposes an orthogonal projection bi-hypersphere detection method built on multi-view relational disentanglement and information-consistent fusion, termed STOP. We conducted comprehensive experiments on different datasets under different attack scenarios. The experiment results demonstrate that our proposed STOP outperforms competing baselines. Specifically, we also conduct anomaly forensics in four real unlabeled datasets and discover interesting findings, including extreme ratings and self-contradictory review styles. Specifically, there are several insights and limitations worth summarizing and discussing, as follows:

1) Although DCIS has a relatively good effect in handling interference data, the selection of interference data is overly dependent on the prior knowledge of abnormal data distribution. Especially for the anomaly discovery in real unlabeled data and interference filtering for large-scale data, it is often difficult or even impossible to determine what prior knowledge is available. In our next work, we will explore more intelligent adaptive interference elimination methods. Meanwhile, DCIS is confronted with the problem of high time and space complexity. In the subsequent work, we can use TOP-K similarity to replace global similarity to reduce the complexity.
2) Based on a large number of comparative experiments, this paper verifies the detection performance of STOP against HCPAs. In reality, it is difficult to comprehensively evaluate the effectiveness of STOP through limited experiments. The effectiveness of attacks and the necessity of detection for recommendations are a game process. Based on limited attack and defense experiments, it is difficult to comprehensively evaluate the detection performance. Continuously exploring the effective boundary of HCPAs and the lower boundary of detection has always been a challenge in this field.
3) The assumption that artificially synthesized raw data has no poisoning behavior (i.e., clean data) is questionable. This assumption may not fully consider the potential impact of high-concealed or weakly abnormal behaviors, and whether it is ultimately detrimental to the discovery of real unlabeled anomalies also requires further exploration.

Furthermore, STOP has not explored the detection capabilities for coordinated attacks or multi-target attacks. In our future work, we will analyze recent HCPA methods [21], [63]–[65], and study new detection mechanisms so that they can handle unlabeled data in real scenarios.

References

[1] Y. Li *et al.*, "Recent developments in recommender systems: A survey," *IEEE Computational Intelligence Magazine*, vol. 19, no. 2, pp. 78–95, 2024.

[2] R. Zheng *et al.*, "Poisoning Decentralized Collaborative Recommender System and Its Countermeasures," in *ACM SIGIR*, 2024, pp. 1712–1721.

[3] W. Xu *et al.*, "Enhancing content-based recommendation via large language model," in *ACM CIKM*, 2024, pp. 4153–4157.

[4] M. Si *et al.*, "Shilling attacks against collaborative recommender systems: A review," *Artificial Intelligence Review*, vol. 53, no. 1, pp. 291–319, 2020.

[5] F. Rezaimehr *et al.*, "A survey of attack detection approaches in collaborative filtering recommender systems," *Artificial Intelligence Review*, vol. 54, no. 3, pp. 2011–2066, 2021.

[6] Y. Xie *et al.*, "A hybrid three-way recommendation considering users variability," *EAAI*, vol. 159, no. 1, pp. 111 610–111 622, 2025.

[7] L. Meng *et al.*, "Poi recommendation for occasional groups based on hybrid graph neural networks," *ESWA*, vol. 237, no. 1, pp. 121 583–121 598, 2024.

[8] *Recommendation engine market report — industry analysis, size & forecast*, [Accessed: 2025-11-01]. [Online]. Available: www.mordorintelligence.com/industry-reports/recommendation-engine-market.

[9] Z. Yang *et al.*, "Probabilistic Inference and Trustworthiness Evaluation of Associative Links Toward Malicious Attack Detection for Online Recommendations," *IEEE TDSC*, vol. 19, no. 2, pp. 879–896, 2022.

[10] Z. Yang *et al.*, "Three Birds With One Stone: User Intention Understanding and Influential Neighbor Disclosure for Injection Attack Detection," *IEEE TIFS*, vol. 17, no. 1, pp. 531–546, 2022.

[11] C. Wu *et al.*, "Influence-Driven Data Poisoning for Robust Recommender Systems," *IEEE TPAMI*, vol. 45, no. 10, pp. 11 915–11 931, 2023.

[12] T. Baker *et al.*, "Poison-tolerant collaborative filtering against poisoning attacks on recommender systems," *IEEE TDSC*, vol. 21, no. 5, pp. 4589–4599, 2024.

[13] J. Li *et al.*, "Large-scale fake click detection for e-commerce recommendation systems," in *IEEE ICDE*, 2021, pp. 2595–2606.

[14] Z. Yang *et al.*, "Identification of malicious injection attacks in dense rating and co-visitation behaviors," *IEEE TDSC*, vol. 16, no. 11, pp. 30–40, 2020.

[15] Y. Wang *et al.*, "Revisiting item promotion in gnn-based collaborative filtering: A masked targeted topological attack perspective," in *AAAI*, 2023, pp. 15 206–15 214.

[16] M. C. Urdaneta-Ponte *et al.*, "Recommendation systems for education: Systematic review," *Electronics*, vol. 10, no. 14, pp. 1611–1632, 2021.

[17] X. Zhang *et al.*, "Targeted data poisoning attack on news recommendation system by content perturbation," *arXiv preprint arXiv:2203.03560*, 2022.

[18] J. Yi *et al.*, "Ua-fedrec: Untargeted attack on federated news recommendation," in *ACM SIGKDD*, 2023, pp. 5428–5438.

[19] K. Zhang *et al.*, "Lorec: Combating poisons with large language model for robust sequential recommendation," in *ACM SIGIR*, 2024, pp. 1733–1742.

[20] Y. Lai *et al.*, "Toward adversarially robust recommendation from adaptive fraudster detection," *IEEE TIFS*, vol. 19, no. 1, pp. 907–919, 2023.

[21] J. Zhang *et al.*, "Preventing the popular item embedding based attack in federated recommendations," in *IEEE ICDE*, 2024, pp. 2179–2191.

[22] M. Aktukmak *et al.*, "Quick and accurate attack detection in recommender systems through user attributes," in *ACM RecSys*, 2019, pp. 348–352.

[23] H. Cai *et al.*, "BS-SC: An Unsupervised Approach for Detecting Shilling Profiles in Collaborative Recommender Systems," *IEEE TKDE*, vol. 33, no. 4, pp. 1375–1388, 2021.

[24] Z. Yang *et al.*, "Inference of suspicious co-visitation and co-rating behaviors and abnormality forensics for recommender systems," *IEEE TIFS*, vol. 15, no. 1, pp. 2766–2781, 2020.

[25] C. Wu *et al.*, "FedAttack: Effective and Covert Poisoning Attack on Federated Recommendation via Hard Sampling," in *SIGKDD*, 2022, pp. 4164–4172.

[26] W. Ali *et al.*, "HidAttack: An Effective and Undetectable Model Poisoning Attack to Federated Recommenders," *IEEE TKDE*, vol. 1, no. 1, pp. 1–14, 2024.

[27] Z. Yang *et al.*, "Meet Trick With Trick: Revealing Collusion Intentions in Highly Concealed Poisoning Behavior," *IEEE TDSC*, vol. 1, no. 1, pp. 1–18, 2025.

[28] W. Wang *et al.*, "Uplift modeling for target user attacks on recommender systems," in *ACM WWW*, 2024, pp. 3343–3354.

[29] Z. Wang *et al.*, "Unveiling vulnerabilities of contrastive recommender systems to poisoning attacks," in *ACM SIGKDD*, 2024, pp. 3311–3322.

[30] J. Tang *et al.*, "Revisiting adversarially learned injection attacks against recommender systems," in *ACM RecSys*, 2020, pp. 318–327.

[31] C. Lin *et al.*, "Shilling black-box recommender systems by learning to generate fake user profiles," *IEEE TNNLS*, vol. 35, no. 1, pp. 1305–1319, 2022.

[32] C. Lin *et al.*, "Attacking recommender systems with augmented user profiles," in *ACM CIKM*, 2020, pp. 855–864.

[33] M. Aktukmak *et al.*, "Sequential attack detection in recommender systems," *IEEE TIFS*, vol. 16, no. 1, pp. 3285–3298, 2021.

[34] K. Zhang *et al.*, "Improving the shortest plank: Vulnerability-aware adversarial training for robust recommender system," in *ACM RecSys*, 2024, pp. 680–689.

[35] G. Yang *et al.*, "Fake co-visitation injection attacks to recommender systems," in *NDSS*, 2017, pp. 1–15.

[36] B. Li *et al.*, "Data poisoning attacks on factorization-based collaborative filtering," *NIPS*, vol. 29, no. 1, pp. 1885–1893, 2016.

[37] C. E. Seminario *et al.*, "Nuke'em till they go: Investigating power user attacks to disparage items in collaborative recommenders," in *RecSys*, 2015, pp. 293–296.

[38] H. Huang *et al.*, "Data poisoning attacks to deep learning based recommender systems," *arXiv preprint arXiv:2101.02644*, 2021.

[39] W. Yuan *et al.*, "Manipulating federated recommender systems: Poisoning with synthetic users and its countermeasures," in *ACM SIGIR*, 2023, pp. 1690–1699.

[40] Y. Zhang *et al.*, "Reverse attack: Black-box attacks on collaborative recommendation," in *ACM CCS*, 2021, pp. 51–68.

[41] Z. Yang *et al.*, "Uncovering anomalous rating behaviors for rating systems," *Neurocomputing*, vol. 308, no. 1, pp. 205–226, 2018.

[42] W. Li *et al.*, "Shilling attack detection in recommender systems via selecting patterns analysis," *IEICE TRANSACTIONS on Information and Systems*, vol. 99, no. 10, pp. 2600–2611, 2016.

[43] Q. Zhou *et al.*, "A recommendation attack detection approach integrating cnn with bagging," *Computers & Security*, vol. 146, no. 1, pp. 104 030–104 045, 2024.

[44] Y. Zhang *et al.*, "Catch the black sheep: Unified framework for shilling attack detection based on fraudulent action propagation," in *IJCAI*, 2015, pp. 2408–2414.

[45] T. Dou *et al.*, "Collaborative shilling detection bridging factorization and user embedding," in *CollaborateCom*, 2017, pp. 459–469.

[46] Y. Wang *et al.*, "Decoupling representation learning and classification for gnn-based anomaly detection," in *ACM SIGIR*, 2021, pp. 1239–1248.

[47] R. Liang *et al.*, "Defending federated recommender systems against untargeted attacks: A contribution-aware robust aggregation scheme," *ACM TKDD*, vol. 19, no. 1, pp. 1–28, 2025.

[48] Y. Shang *et al.*, "Enhancing adversarial robustness of multi-modal recommendation via modality balancing," in *ACM MM*, 2023, pp. 6274–6282.

[49] Z. Wu *et al.*, "Hpsd: A hybrid pu-learning-based spammer detection model for product reviews," *IEEE transactions on cybernetics*, vol. 50, no. 4, pp. 1595–1606, 2018.

[50] M. Ju *et al.*, "Let graph be the go board: Gradient-free node injection attack for graph neural networks via reinforcement learning," in *AAAI*, 2023, pp. 4383–4390.

[51] K. Zhao *et al.*, "Causality-inspired spatial-temporal explanations for dynamic graph neural networks," in *The Twelfth International Conference on Learning Representations*, 2024, pp. 1–13.

[52] Z. Shen *et al.*, "Beyond redundancy: Information-aware unsupervised multiplex graph structure learning," in *NIPS*, 2024, pp. 31 629–31 658.

[53] Y. Zhang *et al.*, "Deep orthogonal hypersphere compression for anomaly detection," *arXiv preprint arXiv:2302.06430*, 2023.

[54] F. M. Harper *et al.*, "The movielens datasets: History and context," *Acm TIIS*, vol. 5, no. 4, pp. 1–19, 2015.

[55] J. Golbeck *et al.*, "Filmtrust: Movie recommendations using trust in web-based social networks," in *IEEE CCNC*, 2006, pp. 282–286.

[56] H. Zhang *et al.*, "Adaptive graph integration for cross-domain recommendation via heterogeneous graph coordinators," in *ACM SIGIR*, 2025, pp. 1860–1869.

[57] Y. Wang *et al.*, "Amazon-kg: A knowledge graph enhanced cross-domain recommendation dataset," in *ACM SIGIR*, 2024, pp. 123–130.

[58] Y. Zhang *et al.*, "Localized matrix factorization for recommendation based on matrix block diagonal forms," in *ACM WWW*, 2013, pp. 1511–1520.

[59] Y. Zhang *et al.*, "Daily-aware personalized recommendation based on feature-level time series analysis," in *ACM WWW*, 2015, pp. 1373–1383.

[60] I. Gunes *et al.*, "Shilling attacks against recommender systems: A comprehensive survey," *Artificial Intelligence Review*, vol. 42, no. 4, pp. 767–799, 2014.

[61] X. He *et al.*, "Lightgcn: Simplifying and powering graph convolution network for recommendation," in *ACM SIGIR*, 2020, pp. 639–648.

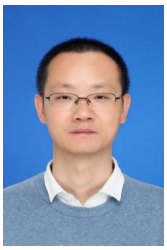
[62] *ChatGPT, A conversational AI system that listens, learns, and challenges*, [Accessed: 2025-11-01]. [Online]. Available: https://chatgpt.com.

[63] J. Zhang *et al.*, "Stealthy attack on large language model based recommendation," *arXiv preprint arXiv:2402.14836*, 2024.

[64] M. Yin *et al.*, "Poisoning federated recommender systems with fake users," in *ACM WWW*, 2024, pp. 3555–3565.

[65] Y. Wu *et al.*, "Accelerating the surrogate retraining for poisoning attacks against recommender systems," in *ACM RecSys*, 2024, pp. 701–711.

Yan Feng is currently a Ph.D. candidate in the School of Data Science and Artificial Intelligence, Chang'an University, Xi'an, China. He received his Master degree in Computer Technology from Xi'an University of Technology, Xi'an, China. His research interests include artificial intelligence security, recommender system security, recommender system attack and defense, and model security.



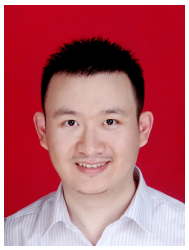
Zhihai Yang received the Ph.D. degree in Control Science and Engineering from Xi'an Jiaotong University, China, in 2016. He is currently a professor with the School of Data Science and Artificial Intelligence, Chang'an University, Xi'an, China. His research interests include artificial intelligence security, identity security, authentication, and cognitive security assessment.

Kexin Li is currently a Master candidate in the School of Data Science and Artificial Intelligence, Chang'an University, Xi'an, China. She received the bachelor degree from the College of Software, Shanxi Agricultural University, Jinzhong, China, in 2023. Her research interests include recommender system security and artificial intelligence security.

Jianhua He received the Ph.D. degree from Nanyang Technological University. He is currently a Professor with the University of Essex, U.K. He published more than 200 research articles. His research interests include data analytics and recommendation systems, computer security, mobile networking and computing, 5G/6G networks, Internet of Things, edge computing and intelligence, connected autonomous driving, intelligent transport systems, AI, and large language models. He served as an Editor for IEEE Wireless Communication Letters and The Computer Journal.

Jianxin Li (Senior Member, IEEE) received the PhD degree in computer science from the Swinburne University of Technology, Australia, in 2009. He is professor of information systems with the School of Business and Law, Edith Cowan University, Australia. His research interests include graph database query processing and optimization, social network analytics and computing, complex network representation learning, and personalized online learning analytics. He is also a grant assessor in Australia Research Council Discovery Programs and Linkage Programs, and serves as invited reviewers for many top journals and program committee members in many top conferences.

Pinghui Wang (Senior Member, IEEE) is currently a Professor with the MOE Key Laboratory for Intelligent Networks and Network Security, Xi'an Jiaotong University, Xi'an, China, and also with the Shenzhen Research Institute, Xi'an Jiaotong University, Shenzhen, China. His research interests include internet traffic measurement and modeling, traffic classification, abnormal detection, and online social network measurement.



Zhiquan Liu received the B.S. degree from the School of Science, Xidian University, Xi'an, China, in 2012, and the Ph.D. degree from the School of Computer Science and Technology, Xidian University, Xi'an, China, in 2017. He is currently a full professor, doctoral supervisor, and deputy dean with the College of Cyber Security, Jinan University, Guangzhou, China. His current research focuses on security, trust, privacy, and intelligence in vehicular networks and UAV networks. He currently serves as the area editor or associate editor of multiple SCI-index journals, such as IEEE TIFS, IEEE TDSC, IEEE TII, IEEE TVT, IEEE IOTJ, IEEE Network, Information Fusion, etc. His homepage is https://www.zqliu.com.