This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2020.3016827, IEEE Transactions on Information Forensics and Security

Identification of Malicious Injection Attacks in Dense Rating and Co-visitation Behaviors

Zhihai Yang, Qindong Sun, Yaling Zhang, and Wei Wang

***Abstract*—Personalized recommender systems are pervasive in different domains, ranging from e-commerce services, financial transaction systems to social networks. The generated ratings and reviews by users toward products are not only favourable to make targeted improvements on the products for online businesses, but also beneficial for other users to get a more insightful review of the products. In reality, recommender systems can also be deliberately manipulated by malicious users due to their fundamental vulnerabilities and openness. However, improving the detection performance for defending malicious threats including profile injection attacks and co-visitation injection attacks is constrained by the challenging issues: (1) various types of malicious attacks in real-world data coexist; (2) it is difficult to balance the commonality and speciality of rating behaviors in terms of accurate detection; and (3) rating behaviors between attackers and *anchor* users caused by the consistency of attack intentions are extremely similar. In this paper, we develop a unified detection approach named *IMIA-HCRF*, to progressively discriminate malicious injection behaviors for recommender systems. First, disturbed data are empirically eliminated by implementing both the construction of association graph and enhancement of dense behaviors, which can be adapted to different attacks. Then, the smooth boundary of dense rating (or co-visitation) behaviors is further segmented using higher order potentials, which is finally leveraged to determine the concerned injection behaviors. Extensive experiments on both synthetic data and real-world data demonstrate that the proposed *IMIA-HCRF* outperforms all baselines on various metrics. The detection performance of *IMIA-HCRF* can achieve an improvement of 7.8% for mixed profile injection attacks as well as 6% for mixed co-visitation injection attacks over the baselines in terms of FAR (false alarm rate) while keeping the highest DR (detection rate). Additional experiments on real-world data show that *IMIA-HCRF* brings an improvement with the advantage of 11.5% FAR in average compared with the baselines.**

***Index Terms*—Attack detection, recommender system, information security, behavior representation.**

I. Introduction

Personalization recommender systems have become a crucial component of various web services, such as Amazon, TripAdvisor, YouTube, Taobao, etc., which recommend a user items (e.g., clothes on Amazon, hotels on TripAdvisor) that match the user's preference [1], [2]. In particular, collaborative recommendation techniques (CRTs), including *UBCF*, *IBCF* [1], co-visitation based [3], etc., have been developed in the past two decades [1]. The underlying assumption of CRTs is that if two users have expressed similar interests in the past, they will share common interests in the future. However, CRTs are highly vulnerable to profile injection attacks (a.k.a., *shilling* attacks) [4], [5], [6], [7] and fake co-visitation injection attacks [3] due to their openness and fundamental vulnerabilities [5], [6], [7], [8]. As demonstrated in Figure 1, malicious attackers either inject a sufficient number of well-designed fake profiles (e.g., ratings and reviews) into the systems and empirically rate higher scores (termed *push* or *promotion* attacks [9]) or lower scores (called *nuke* or *demotion* attacks [9]) toward targeted items, or inject fake co-visitations to the systems to spoof CRTs [3] in order to manipulate recommendations (shaking consumers' confidence) or reduce the quality of recommendation (performance degradation) as the attackers desire. Such threats cause great damage to the public, shaking the confidence of both customers and businesses in the virtual market. Therefore, the demand for protecting users' personal benefits is becoming more pressing.

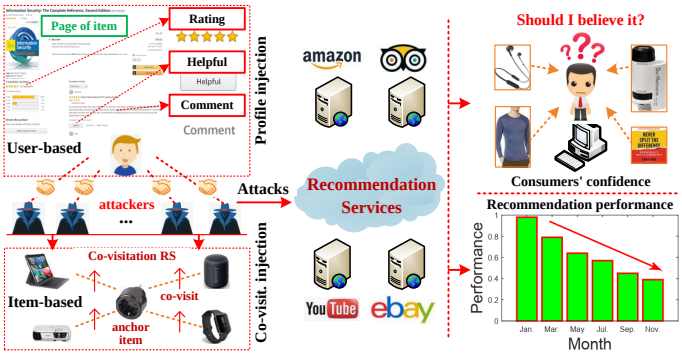


Fig. 1: A representative framework of threats concerning online recommendation services. Attackers inject a sufficient number of malicious profiles or fake co-visitations into recommender systems in order to manipulate recommendations (performance degradation or shaking consumers' confidence).

A. Problem Statement

Various potential solutions have been studied to find ways out in order to detect malicious injection profiles and reconstruct a pure land for recommender systems. Nonetheless, the improvement of detection performance for defending malicious threats such as profile injection attacks and co-visitation injection attacks is restricted due to the challenging issues: (1)

This work was supported in part by the National Natural Science Foundation of China under Grant 61702412 and Grant 61571360, in part by the Innovation Project of the Shaanxi Provincial Department of Education under Grant 17JF023, in part by the Natural Science Founds of Shaanxi under Grant 2019JM-266 and Grant 2019GY-028, in part by the Youth Innovation Team of Shaanxi Universities, and in part by the Ph.D. Research Startup Funds of the Xi'an University of Technology under Grant 112-256081704.

Z. Yang, Q. Sun, Y. Zhang, and W. Wang are currently with the school of computer science and engineering, Xi'an University of Technology, Xi'an, China. They also are with Shaanxi Key Laboratory for Network Computing and Security Technology. E-mail: zhyang_xjtu@sina.com.

different types of malicious attacks may be mixed or coexisted in reality; (2) discriminative and informative representations in terms of intrinsic attributes and global association attributes of rating and visitation behaviors are limited; and (3) it is difficult to distinguish *anchored* items (for co-visitation injection attacks) or *selected* items (for profile injection attacks) caused by the consistency of attack intentions from target items. As such, investigating how to improve the generalization ability of detection models and deeply distinguish the fuzzy boundary of dense behaviors is desirable.

B. Contributions

In this paper, we investigate a unified detection approach to identify malicious injection attacks using higher order conditional random fields (named *IMIA-HCRF*). In order to reduce the impact of disturbed data on boosting detection performance, firstly, we analyze the distribution of both rating behaviors and co-visitation behaviors and empirically filter out disturbed data by implementing both the construction of behavior association graph and enhancement of dense behaviors. To incorporate topological characteristics of behavioral association links and preserve the advantage of traditional and inherent behaviour features, we then explore *unary* and *pairwise* attributes of nodes (users or items) in the constructed association graph. Especially, the smooth boundary of dense and mixed rating behaviors or co-visitation behaviors based on weighted node and link attributes, can be further segmented using higher order conditional random fields. Finally, we can determine malicious users and items according to both the globally optimal segmentation and suspected items.

The major contributions of this paper can be briefly summarized as follows:

First, we propose to enhance dense rating (profile injection) behaviors and co-visitation injection behaviors via the elimination of disturbed data and representation of sparse behaviors, which also provides a possibility for the integrated detection of different injection attack behaviors.

Second, we explore attributes of both nodes and edges of behaviour association graph, and propose to incorporate unary potential and pairwise potential of higher order conditional random fields for informative representations of rating and co-visitation behaviours.

Third, we develop a unified detection approach to identify both profile injection attacks and co-visitation injection attacks. Additionally, mixed profile injection attacks and mixed co-visitation injection attacks with different cases are implemented. The evaluation and analyses for comparison experiments on synthetic data and real-world data demonstrate that the proposed *IMIA-HCRF* outperforms the baselines.

C. Organization

The rest of this paper is organized as follows: Section II reviews abnormality detection in recommender systems. Section III describes the background knowledge of threat models and higher order conditional random field. Section IV introduces the proposed approach. Experimental results will be analyzed and discussed in section V. Conclusions and future work will be finally provided in section VI.

II. Related Work

Investigating detection approaches and real-world application has attracted much attention in the past two decades. Previous efforts provide promising results in terms of accurate detection, the characterization of malicious behaviors, the determination of disturbed information and abnormality detection on real-world data. In this section, we briefly summarize and discuss related researches from the above aspects.

Characterizing rating behaviors of users is a crucial task in attack detection. Many previous detection methods have been designed based on representations of rating behaviors extracted from original rating data. Burke *et al.* [4] developed several rating attributes including *generic* attributes and *model-specific* attributes for detecting shilling attacks. Yang *et al.* [10] further improved the representation of both rating behaviors of users and item distribution. A careful reading of the literature suggests that the original rating data have rich information for characterizing basic rating behaviors of users. Nevertheless, boosting detection performance heavily depends on the representation of extracted features. Moreover, most of features are only effective against certain types of attacks, such as *type-specific* attributes [4], [9]. Despite the effectiveness of similarity-based features, it is still limited facing with large-scale real data due to the computational cost [11].

Despite the promising results relying on the aforementioned rating behaviors, a dense distribution between attackers and some authentic users whose rating details are mimicked by the attackers results in high false alarm rates, which makes a challenging issue for improving the detection performance. To address this issue, step-by-step detection frameworks are favorable by researchers. One group of active studies focused on target item analysis [12]. They investigated the distribution of items in order to capture suspicious target items. Another group of active studies focused on sybil detection in online social networks [13], [14], [15], [16]. They investigated the problem of fake user and fake review detection. The advantages such as structure-based attributes of networks, pairwise *Markov Random Fields* [17], synchronized and abnormal patterns, etc. in these researches are considerably significant, which can be used to design new detection models. Nevertheless, purely analyzing anomalous distributions of items and users is insufficient to distinguish target items from *popular* or *unpopular* items, especially facing with small attack sizes. Due to the fact that the existence of disturbed data (e.g., *unpopular* items or *inactive* users) increases the ambiguity between small-scale attack profiles and genuine profiles.

Discovering abnormalities on real-world data such as Amazon and Twitter has been widely investigated. To take advantages of both labeled and unlabeled rating data, Wu *et al.* [7], developed a hybrid detection approach to separate random-filler model attackers and average-filler model attackers from authentic users. Furthermore, Günnemann *et al.* [5] casted the basic behavior of users regarding a product as a latent multivariate autoregressive process and presented an efficient approach to discover interesting findings on real-world data. Günnemann *et al.* [6] also investigated a Bayesian model to spot general rating behaviors of users as well as time intervals

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2020.3016827, IEEE Transactions on Information Forensics and Security

TABLE I: Notation and description.

Notation	Description
$\|U\|$	number of users
$\|I\|$	number of items
R_d	embedding matrix
$\mathcal{T}_s$	suspicious items
x_i	random variable
$\psi_i(x_i)$	unary potential of x_i
$\psi_{ij}(x_i, x_j)$	pairwise potential of x_i and x_j
$\psi_c(\mathbf{x}_c)$	high order potential of a clique c

where the rating behaves are anomalous.

The above efforts reveal that learning prior knowledge from synthetic data can be applied to real unlabeled data. In summary, this study, different from existing studies: (1) aims to convert the original rating data into a behavior association graph using a generally acceptable way and also partly avoid the predicament of feature extraction; (2) incorporates topological characteristics of behavioral association connections while preserving the advantage of traditional features; (3) focuses on both co-visitation injection attacks and profile injection attacks in order to discover potential commonalities; and (4) evaluates the distinctiveness and permanence properties of different linkage attributes in practice, instead of assuming that the original data qualify as good behavioral trait.

III. Preliminaries

In this section, we first describe malicious threats toward CRTs. Then, we briefly introduce the background knowledge of higher order conditional random field. Note that, Table I summarizes the notation we used throughout the paper.

A. Threat Models

In this paper, two types of representative threats including fake co-visitation injection attacks [3] and profile injection attacks [4], [9] have been implemented and investigated. The goal of abnormality detection is to discriminate various malicious threats include but are not limited to these threats.

1) Profile Injection Attacks: Profile injection attacks aim to make a targeted item be recommended to more users [4], [11]. Attackers empirically insert fake rating profiles into the system in order to manipulate the recommendation [4]. The rationale behind the attacks is that a user may like an item that his neighbors who have similar preferences. The attackers rate targeted items with the highest score $\mathcal{R}_{max}$ (termed *push* attacks) or the lowest score $\mathcal{R}_{min}$ (termed *nuke* attacks) [4].

As illustrated in Figure 2, attackers carefully mimic rating details of concerned genuine users and construct well-designed attack profiles. Formally, items in attack profiles consist of selected items I_S, filler items I_F, and target items I_T. I_S can be used to mimic rating behaviors of genuine users in special cases. For instance, popular items (or power items [18]) are generally rated by most of genuine users. In this case, the influence of popular items is favored by attackers due to the cost of imitation [19]. Comparatively, filler items are used to control the length of item vector in order to make the attackers get close to similar neighbors of mimicked genuine users.

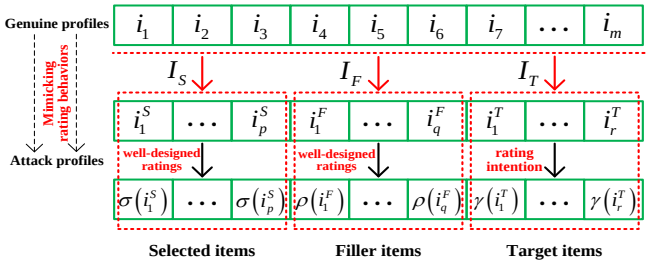


Fig. 2: The general form of attack profiles and genuine profiles in profile injection attacks, where the attack profiles contain selected items, filler items, and target items. The rating details of some genuine profiles are mimicked by attackers, which is used to manipulate recommendations that the attackers desire.

Additionally, corresponding rating functions $\sigma(\cdot)$ to I_S, $\rho(\cdot)$ to I_F, and $\gamma(\cdot)$ to I_T, are respectively assigned using the normal distribution of an item i or the global normal distribution, such as $N(\bar{\mathcal{R}}_i, \bar{\sigma}_i^2)$ and $N(\bar{\mathcal{R}}, \bar{\sigma}^2)$. It is noteworthy that the selected items and the filler items can also be determined by power items and power users [18]. In this work, we implemented 6 representative profile injection attacks. The details of these attack models are described as follows:

- Random attack: $I_S = \emptyset$ and $\rho(i) \sim N(\bar{\mathcal{R}}, \bar{\sigma}^2)$. Filler items are randomly selected from the system [19]. It is relatively easy to be implemented. The effect of random attack, however, is not very remarkable.
- Reverse bandwagon attack: $\gamma(i) = \mathcal{R}_{min}$ or $\mathcal{R}_{max}$, $\rho(i) = \bar{\mathcal{R}}$. Selected items are randomly chosen from *unpopular* items which have been rated by a small number of users [11], [18]. The more items are rated by users, the more popular they are. Filler items are randomly chosen from the system [9], [19]. Attack profiles are generated based on unpopular items and target items with $\mathcal{R}_{min}$.
- Love/Hate attack: $I_S = \emptyset$ and $\rho(i) = \mathcal{R}_{max}$. Filler items I_F are randomly selected from the system [19]. It is extremely effective for nuke attacks.
- Average over popular items attack (AOP): $I_S = \emptyset$ and $\rho(i) \sim N(\bar{\mathcal{R}}_i, \bar{\sigma}_i^2)$. Filler items are randomly selected from popular items (choose the top $x\%$ most popular items) [19], [20].
- Power user attack with the highest number of ratings (PUA-NR): It copies ratings and items from power user profiles, and $I_F = \emptyset$. Specially, power users are the users who have rated a large number of items [21].
- Power item attack with in-degree centrality (PIA-ID): $\sigma(i) \sim N(\bar{\mathcal{R}}_i\ \bar{\sigma}_i^2)$ and $I_F = \emptyset$. Power items participate in the highest number of similarity neighborhoods based on *In-Degree* centrality [18].

2) Co-visitation Injection Attacks: Co-visitation injection attacks focus on co-visitation recommender systems (e.g., recommend users videos on YouTube and products on Amazon) which are based on co-visitation graphs [3]. Attackers inject well-designed co-visitations into a co-visitation recommender system in order to spoof the system to make recommendations as the attackers desire. Due to space limit, in this paper, we exploit four co-visitation injection attacks including promotion

TABLE II: The general implementation scenarios and explanations of all presented co-visitation injection attacks.

Attack Model	Implementation Scenario	Explanation
Promotion attack with high knowledge (termed PH)	$\max \sum_{j \in V_k} a_j \cdot p_j$, $s.t. \sum_{j \in V_k} a_j \cdot m_{jk} \leq m$, $s'_{ji_t} > s'_{jk_j}, w_{i_t} + m_{jk} \geq \mu, \forall j \in V_k$	Suppose an attacker can obtain the co-visitation graph G and the popularity threshold μ. The attacker can inject m fake co-visitations using bounded resources in order to make the target item i_t appears in the recommendation list of j [3].
Promotion attack with medium knowledge (termed PM)	$\max \sum_{j \in V_k} a_j \cdot p_j$, $s.t. \sum_{j \in V_k} a_j \cdot m'_{jk} \leq m$, $s_{jx} = \frac{w_{jx}}{f(w_j, w_x)} \leq \frac{\max\{w_j, w_x\}}{f(w_j, w_x)}, \forall j \in V_k$	Attackers cannot access the number of co-visitations between items nor μ. Both an upper bound of s_{jk_j} which is the similarity between j and the k-th ranked item k_j in j's recommendation list and the number of injected co-visitations m'_{jk} can be estimated.
Demotion attack with high knowledge (termed DH)	$\max \sum_{j \in V_k} a_j \cdot p_j$, $s.t. \sum_{j \in V_k} a_j \cdot \sum_{x=u+1}^{k+1} m_{jx} \leq m$, $\min_{x=u+1}^{k+1} s'_{jx} > s'_{ji_t}, \forall j \in V_k$, $\min_{x=u+1}^{k+1} \{w_x + m_{jx}\} \geq \mu, \forall j \in V_k$	DH aims to decrease user impression of a target item i_t via removing i_t from the top-k recommendation lists of the selected anchor items. Related parameters can be determined by the same way as used in PH [3].
Demotion attack with medium knowledge (termed DM)	$\max \sum_{j \in V_k} a_j \cdot p_j$, $s.t. \sum_{j \in V_k} a_j \cdot \sum_{x=u+1}^{k+1} m'_{jx} \leq m$, $s_{jx} = \frac{w_{jx}}{f(w_j, w_x)} \leq \frac{\max\{w_j, w_x\}}{f(w_j, w_x)}, \forall j \in V_k$	DM uses medium knowledge to decrease user impression of the target items. The upper bounds of missing parameters and injected co-visitations can be estimated as exploited in PM.

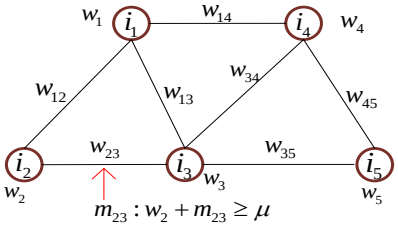


Fig. 3: A co-visitation graph for co-visitation injection attacks.

attack with high knowledge, promotion attack with medium knowledge, demotion attack with high knowledge, and demotion attack with medium knowledge as shown in Table II.

For instance, suppose i_3 is a selected *anchor* item and L_3 is the top-k recommendation list of i_3 as shown in Figure 3. Intuitively, L_3 and the relative rankings of the items in L_3 keep unchanged if we gradually add visitations to i_3 as well as the number of co-visitations between i_3 and the items in L_3 keeps unchanged. Take promotion attacks for example, the attackers need to inject m_{23} fake co-visitations between i_2 and i_3 in order to make i_2 appear in L_3 according to their background knowledge. For high knowledge (take high knowledge for example), the attackers can access the total co-visitation graph and the popularity threshold μ. To this end, m_{23} satisfies two conditions: (1) $s'_{23} > s'_{3k_3}$; and (2) $w_2 + m_{23} \geq \mu$. Therein, s'_{23} is the similarity between i_2 and i_3, and s'_{3k_3} is the similarity between i_3 and the k-th ranked item k_3 in L_3 after the attack. Formally, $s'_{23} = (w_{23} + m_{23})/f(w_3 + m_{23}, w_2 + m_{23})$, $s'_{3k_3} = w_{3k_3}/f(w_3 + m_{23}, w_{k_3})$, where w_2 is the popularity of i_2 before the attack. w_{23} denotes the number of co-visitations between i_2 and i_3. f is the normalization factor (e.g., $f(w_2, w_3) = w_2 \cdot w_3$) [3].

Ultimately, the goal of attackers is to maximize the increased probability of top-k user impression ($IUI = \sum_{i \in J_{i_t} - I_{i_t}} p_i$, for promotion attacks) or the decreased probability of top-k user impression ($DUI = \sum_{i \in I_{i_t} - J_{i_t}} p_i$, for demotion attacks), where $p_i = \frac{w_i}{w_1 + w_2 + \cdots + w_N}$, N and w_i represent the total number of items and the popularity of item i in the past, respectively. I_{i_t} denotes a set of items that a target item i_t is originally among the top-k ($k < N$) recommendation list in these items. After the attack, this set of items is enlarged or reduced to be J_{i_t} [3].

B. Higher Order Conditional Random Field

Before introducing the detail of higher order conditional random field, it is necessary to address the following issues: (a) what is the relationship between higher order conditional random field and detecting malicious attacks? and (b) why the higher order conditional random field technique can be applied to detecting malicious users? Inspired from the successful efforts of higher order conditional random fields (CRFs) in the domain of computer vision [22], [23], the smooth boundary of images can be further segmented and well represented using higher order energy fields. Due to the challenge of detecting malicious threats in dense rating (or visitation) behaviors, we exploit the basic idea used in the segmentation of image boundary to deal with our concerned task. As is known, higher order CRFs make a successful union based on unary potential (first-order), pairwise potential (second-order), and higher order (third-order) potential. In this paper, we first extract behavioral attributes for first-order, second-order, and hyper-order (third-order) nodes to represent the boundary of dense rating or visitation behaviors. We then use the framework of higher order CRF to detect malicious threats.

Minimizing an energy function has been widely investigated to deal with problems in artificial intelligence and computer vision [24]. Given a set of discrete random variables $\mathbf{x} = \{x_1, x_2, \cdots\}$, each random variable x_i takes a value from a label set $\mathcal{L} = \{\ell_1, \ell_2, \cdots\}$. A labelling or configuration is a possible assignment of labels to the random variables. Let $\psi_i(x_i)$, $\psi_{ij}(x_i, x_j)$, and $\psi_c(\mathbf{x}_c)$ respectively denote unary, pairwise, and high-order potentials (over a clique c, $\mathbf{x}_c = \{x_i, i \in c\}$) of $\mathbf{x}$, the Gibbs energy [22], [23], [24] of higher order conditional random field can be described as,

$$E(\mathbf{x}) = \sum_{i \in \mathcal{V}} \psi_i(x_i) + \sum_{i \in \mathcal{V}, j \in \mathcal{N}_i} \psi_{ij}(x_i, x_j) + \sum_{c \in \mathcal{S}} \psi_c(\mathbf{x}_c), \quad (1)$$

where $\mathcal{V}$ is a lattice. $\mathcal{N}_i, \forall i \in \mathcal{V}$ is a neighborhood system $\mathcal{N}$ of the random field. $\mathcal{S}$ refers to the set of all segments. The energy function associated with a labelling problem can be solved using the posterior probability distribution of possible configurations of $\mathbf{x}$, such as $E(\mathbf{x}) = -\log Pr(\mathbf{x}|\mathbf{D}) - \log Z$, where Z and $\mathbf{D}$ are a partition function and a set of instances, respectively. Mathematically, a conditional random field may be viewed as a Markov random field globally conditioned on the data [23]. Therefore, based on the Gibbs energy, Eq. 1

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2020.3016827, IEEE Transactions on Information Forensics and Security

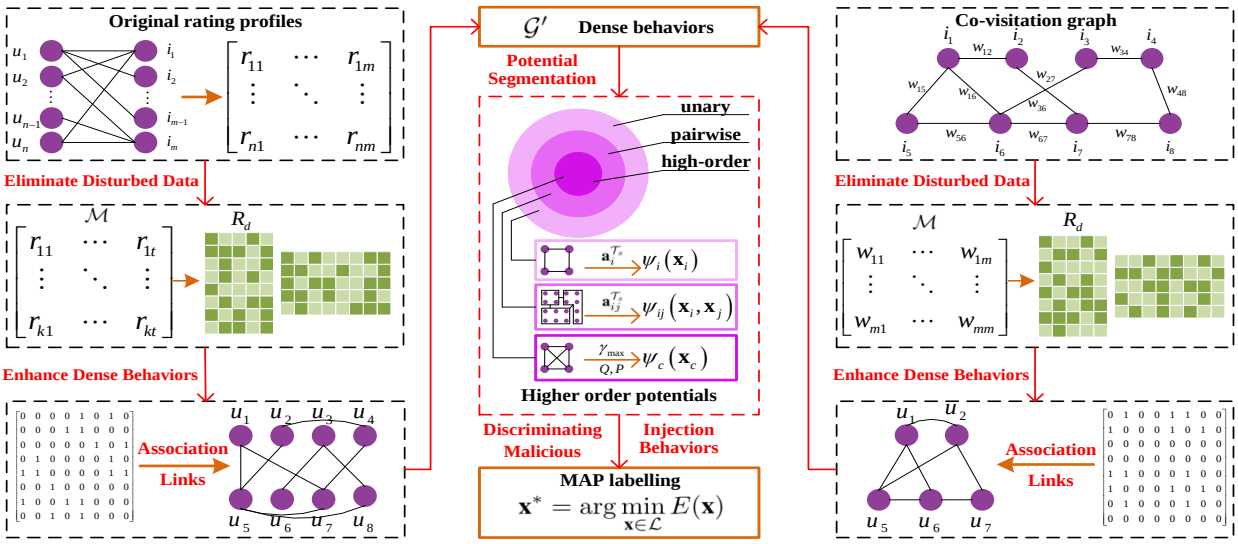


Fig. 4: The framework of our proposed detection approach consists of the elimination of disturbed data, enhancement of dense behaviors, potential segmentation of dense behaviors, and discrimination of malicious injection behaviors.

can be solved by calculating the most probable or maximum a posterior (MAP) labelling of the random field as below,

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathcal{L}} Pr(\mathbf{x}|\mathbf{D}) = \arg\min_{\mathbf{x}\in\mathcal{L}} E(\mathbf{x}), \tag{2}$$

where $\mathbf{x}^*$ represents the predicted labels on concerned nodes (variables). More details will be described in what follows.

IV. Identification of Malicious Injection Attacks

In this section, we first provide an overview of identification of malicious injection attacks. Then, we analyze disturbed data that can be empirically eliminated before detection. After that, progressively discriminating malicious injection behaviors using representations of high order potentials is investigated. Finally, we discuss how to determine the concerned attacks.

A. Overview of Threat Identification

A basic framework of the proposed approach is provided as shown in Figure 4. The main stages of the presented framework can be briefly described as follows:

1) Elimination of disturbed data: The goal of eliminating disturbed data is to (a) reduce the dimension of both rating matrix (for profile injection attacks) and *item-item* weight co-occurrence matrix (for co-visitation injection attacks) and (b) construct a *user-user* association graph based on original rating profiles (for profile injection attacks). Intuitively, *inactive* users [11] and *unpopular* items in the rating matrix can be considered as disturbed data due to their limited influence. Similarly, nodes (items) with low degrees in the co-visitation graph can also be eliminated in advance.
2) Enhancement of dense behaviors: Shilling attackers mimic rating details of *anchor* users (i.e., some genuine users), leading to high similarities (dense rating behaviors) between the attackers and *anchor* users. Likewise, injecting sufficient co-visitations between target items and *anchor* items by malicious users also leads to dense co-visitation behaviors. In this paper, both sparse rating matrix and weight matrix are decomposed to obtain a hidden space representation for each node (user or item). Additionally, dense behaviors can be further enhanced via eliminating similar behaviors and reconstructing association links.
3) Potential segmentation: To discriminate dense behaviors, the smooth boundary of local rating (or co-visitation) behaviors is further segmented using higher order potentials. Based on a dense association graph, the inherent attribute representation of unary node potential, associative representation of pairwise super-node potential, and sophisticated statistics of high-order potential are exploited to enhance the representation of dense behaviors.
4) Discrimination of malicious injection behaviors: Based on the representation of higher order potentials, we predict a possible label for each node using the maximum a posterior (MAP) labelling of conditional random field. Finally, nodes (users or items) assigned as concerned labels can be determined as malicious nodes.

B. Elimination of Disturbed Data

Generally, abnormality detection can be considered as a process of gradually filtering out authentic profiles while retaining concerned attack profiles. From the perspective of detection, in other words, the authentic profiles can also be treated as disturbed data compared with attack profiles. To this end, how to effectively determine and filter out disturbed data (authentic profiles) is a crucial task. Due to the high similarity between attack profiles and authentic profiles mimicked by the attackers [11], [25], high similarities between rating profiles (for profile injection attacks) or co-visitations (for co-visitation injection attacks) represent *dense* behaviors (i.e., rating behavior and visitation behavior). In order to gradually achieve the goal of eliminating disturbed data, in this paper, we first construct an association graph from the original data and capture dense

Algorithm 1 Elimination of disturbed data.

Require:
Original rating matrix $M \in \mathbb{R}^{|\mathcal{U}|\times|\mathcal{I}|}$;
Thresholds of parameters ϕ_u, ϕ_i, and ε.
Ensure:
A well-pruned graph $\mathcal{G}'$;
1: $\mathcal{D}^u = \emptyset$, $\mathcal{D}^i = \emptyset$;
2: **for** each user $\mathcal{U}_i \in \mathcal{U}$ in M **do**
3: Calculate the number of items rated by $\mathcal{U}_i$, $d_i = len(find(M(index(\mathcal{U}_i),\ :) \neq 0))$;
4: **if** $d_i < \phi_u$ **then**
5: $\mathcal{D}^u \leftarrow \mathcal{U}_i$;
6: **end if**
7: **end for**
8: **for** each item $\mathcal{I}_k \in \mathcal{I}$ in M **do**
9: Calculate the number of users who have rated on $\mathcal{I}_k$, $d_k = len(find(M(:,\ index(\mathcal{I}_k)) \neq 0))$;
10: **if** $d_k < \phi_i$ **then**
11: $\mathcal{D}^i \leftarrow \mathcal{I}_k$;
12: **end if**
13: **end for**
14: Remove corresponding rows and columns of users $\mathcal{D}^u$ and items $\mathcal{D}^i$ from M and obtain a reduced graph $\mathcal{G}'$;
15: Deal with $\mathcal{M} = QS_d\sum_d V_d^T$ using tSVD;
16: Calculate the final embedding matrix $R_d = QS_d\sum_d^{1/2}$;
17: **for** each pair of nodes $u_i, u_j \in \mathcal{G}'$ **do**
18: Calculate the similarity between u_i and u_j, $s_{ij} = SimiFun(R_d(index(u_i),:), R_d(index(u_j),:))$;
19: **if** $s_{ij} < \varepsilon$ **then**
20: Remove the edge (i,j) from $\mathcal{G}'$;
21: **end if**
22: **end for**
23: **return** $\mathcal{G}'$;

behaviors according to the sparse representation of behavior space. More details will be elaborated in what follows.

1) Construction of Association Graph: Investigating attack detection models according to association graphs has been received much attention in the last decade [11], [26]. Nevertheless, specially designing graph-based detection methods from the perspective of filtering out disturbed data is still a long-standing but unresolved issue. Actually, the construction of association graph is favorable to improve the detection performance due to the fact that: (1) feature extraction based on the original rating data can be partly avoidable; (2) disturbed data such as unpopular items and inactive users, can be filtered out in advance; and (3) it is also beneficial for dimension reduction of the original data. Therefore, a *user-user* undirected graph is constructed based on the original rating data, where an edge is generated if two users have jointly rated δ ($\delta \geq 1$) items. Note that, an empirical threshold of δ needs to be determined by experiments (see section V-B). Moreover, empirical thresholds for determining unpopular items and inactive users are also determined in advance, especially for large-scale real-world data. Algorithm 1 describes the basic process of determining disturbed data. More details about parameters used for determination of disturbed data will be discussed in what follows.

2) Enhancement of Dense Behaviors: According to the constructed association graph as aforementioned, how to further reduce the scope of detection and reserve the concerned targets (suspicious nodes) is crucial. To this end, we first convert the characterization of nodes into the representation of node-context pair inspired from network embedding [27], [28]. Given an undirected network, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}|$ nodes and $|\mathcal{E}|$ edges, let A and D respectively denote the adjacency matrix (binary) and diagonal degree matrix of $\mathcal{G}$, where $D_{ii} = \sum_j A_{ij}$, the goal of network embedding is to learn a mapping function $\mathbf{f} : V \rightarrow R^d$ that projects each node to a d-dimensional space ($d \ll |V|$) in order to capture the structural properties of the network. Based on embedding vector $e_i \in R^d$ and context vector $c_i \in R^d$ of node v_i, the occurrence of probability of context v_j given node v_i is defined as $\hat{p}_{i,j} = \sigma(e_i^T c_j)$, where $\sigma(\cdot)$ denotes the sigmoid function. The weighted sum of log loss over all edges can be expressed as an objective function $L = -\sum_{(i,j)\in\mathcal{D}} p_{i,j} \ln \hat{p}_{i,j}$, where $p_{i,j} = A_{ij}/D_{ii}$ represents the weight of (v_i, v_j) in $\mathcal{D}$ (a node-context pair set). The loss function can be further updated based on negative samples $P_{\mathcal{D},j}$ which associate with v_j due to the probability of the trivial solution $(e_i = c_j\ \&\ \hat{p}_{i,j} = 1)$,

$$L = -\sum_{(i,j)\in\mathcal{D}} [p_{i,j} \ln \sigma(e_i^T c_j) + \lambda_1 P_{\mathcal{D},j} \ln \sigma(-e_i^T c_j)], \quad (3)$$

where λ_1 denotes the negative sample ratio. $P_{\mathcal{D},j}$ can be defined as $P_{\mathcal{D},j} \propto (\sum_{i:(i,j)\in\mathcal{D}} p_{i,j})^\alpha$, where α is set as 1 or 0.75 [27]. Furthermore, $e_i^T c_j = \ln \hat{p}_{i,j} - \ln(\lambda_1 P_{\mathcal{D},j})$ can be obtained via partial derivative with respect to $e_i^T c_j$ of Eq. 3 (i.e., $\frac{\partial L}{\partial e_i^T c_j} = 0$). Intuitively, a proximity matrix $\mathcal{M}$ (similarity-based matrix) with each entry as $e_i^T c_j$ can be represented as $\mathcal{M}_{i,j} = \ln p_{i,j} - \ln(\lambda_1 P_{\mathcal{D},j})$ if $(v_i, v_j) \in \mathcal{D}$, $\mathcal{M}_{i,j} = 0$ otherwise.

Therefore, the objective of distributional similarity-based network embedding is converted to matrix factorization [28]. A truncated singular value decomposition (tSVD) for $\mathcal{M} \approx U_d \sum_d V_d^T$ is exploited, where U_d and V_d denote $n \times d$ orthonormal matrices corresponding to the selected singular values, respectively. $\sum_d$ represents the diagonal matrix formed from the top-d singular values. Formally, $R_d = U_d \sum_d^{1/2}$ denotes the output of network embedding, where each row represents the embedding of a node. Note that, each row of R_d can also be considered as a potential feature vector of the node. The basic process of eliminating disturbed data is described in Algorithm 1. In Algorithm 1, lines 2-7 and 8-13 represent the determination of disturbed users and items, respectively. Thereinto, $index(\cdot)$ denotes the index number of a vector. $find(\cdot)$ is used to find indices and values of nonzero elements in a matrix. $len(\cdot)$ is used to return the length of a vector. In line 18, $SimiFun(\cdot)$ denotes a function which is used to calculate the similarity between two vectors.

C. Discriminating Malicious Injection Behaviors

1) Potential Segmentation of Injection Behaviors: Promising representations of traditional attributes based on the original rating data can be obtained [4], [25]. Nevertheless, how to incorporate novel representations of nodes and keep the

TABLE III: Attributes of both nodes and edges.

Attribute	Definition	Description
User activity (UA)	$UA_u = \sum_{i \in \mathcal{I}} \mathcal{O}(r_{ui})$	User activity is used to evaluate the degree of a user u.
Item distribution (ID)	$ID_u = \frac{\sum_{i \in \mathcal{T}_s} \mathcal{O}(r_{ui} \neq \emptyset)}{\sum_{i \in \mathcal{I}} \mathcal{O}(r_{ui})}$	The number of suspected items rated by a user u_i divided by the number of all items rated by u_i.
Rating distribution (RD)	$RD_u = \frac{\sum_{i \in \mathcal{T}_s}^{N} (r_{ui} - \overline{r_i})}{N}$	Average rating deviation based on the global average rating $\overline{r_i}$ of an item i.
Time distribution (TD)	$TD_u = \frac{\sum_{i \in \mathcal{T}_s}^{N} (t_{ui} - \overline{t_i})}{N}$	Average time deviation based on the global average time $\overline{t_i}$ of an item i.
Sentimental polarity (SP)	$SP_u = \frac{\sum_{i \in \mathcal{T}_s} (l_{ui} = \ell)}{\sum_{i \in \mathcal{T}_s} (l_{ui})}$	The number of labels of comments predicted as ℓ (i.e., $\ell = 1$ and $\ell = 0$ respectively represent critical and optimistic viewpoints) divided by the number of labels of comments based on all suspected items $\mathcal{T}_s$.
Similar motive (SM)	$SM_{ij} = \frac{\lvert \mathcal{I}_i \cap \mathcal{I}_j \rvert}{\lvert \mathcal{I}_i \cup \mathcal{I}_j \rvert}$	The number of co-rated items by users u_i and u_j divided by all items rated by u_i and u_j.
Interest preference (IP)	$IP_{ij} = \max(\frac{\lvert \mathcal{I}_i \cap \mathcal{I}_j \rvert}{\lvert \mathcal{I}_i \rvert}, \frac{\lvert \mathcal{I}_i \cap \mathcal{I}_j \rvert}{\lvert \mathcal{I}_j \rvert})$	The maximum ratio between co-rated items by u_i and u_j and items rated by each one.
Common preference (CP)	$CP_{ij} = \frac{\lvert (r_{i1}, \cdots, r_{i\lvert \mathcal{T}_s \rvert}) \cap (r_{j1}, \cdots, r_{j\lvert \mathcal{T}_s \rvert}) \rvert}{\lvert (r_{i1}, \cdots, r_{i\lvert \mathcal{T}_s \rvert}) \cup (r_{j1}, \cdots, r_{j\lvert \mathcal{T}_s \rvert}) \rvert}$	The number of identical ratings on $\mathcal{T}_s$ by users u_i and u_j divided by the number of ratings on $\mathcal{T}_s$ by both of them.
Sentimental polarity (SP)	$SP_{ij} = \frac{\lvert (\ell_{i1}, \cdots, \ell_{i\lvert \mathcal{T}_s \rvert}) \cap (\ell_{j1}, \cdots, \ell_{j\lvert \mathcal{T}_s \rvert}) \rvert}{\lvert (\ell_{i1}, \cdots, \ell_{i\lvert \mathcal{T}_s \rvert}) \cup (\ell_{j1}, \cdots, \ell_{j\lvert \mathcal{T}_s \rvert}) \rvert}$	The number of identical labels on $\mathcal{T}_s$ by users u_i and u_j divided by the number of labels on $\mathcal{T}_s$ by both of them.

advantages of traditional attributes is worth investigating. In this paper, high order potential functions defined on higher order cliques, which have the capability of modelling the rich statistics of natural scenes in computer vision [22], [23], [29], are employed to deal with the combination of multi-order potentials based on the remained association graph.

To reserve the inherent attributes of node, the first-order (unary) potential of CRF $\psi_i(x_i)$, is used to define the likelihood of a label motivated from a node (user) x_i being assigned to suspicious targets $\mathcal{T}_s$, which can be written as Eq. 4,

$$\psi_i(x_i) = -\theta_1 \log \frac{P(x_i, \mathcal{T}_s)}{\sum_{x_j \in \mathcal{N}_i} P(x_j, \mathcal{T}_s)}, \tag{4}$$

and

$$P(x_i, \mathcal{T}_s) = \frac{1}{1 + \exp(-\mathbf{w}^\top \mathbf{a}_i^{\mathcal{T}_s})}, \tag{5}$$

where $\mathcal{N}_i$ denotes the neighbors of node u_i. θ_1 is a model parameter learned from training data. $\mathbf{w} = [w_0, w_1, w_2, \ldots]$ represents a vector of weights. It is noteworthy that $\mathbf{a}_i^{\mathcal{T}_s} = [UA_i, ID_i, RD_i, TD_i, SP_i]$ represents an attribute vector of a user (node) u_i as shown in Table III. All attributes of nodes except for UA_i are based on the suspected items $\mathcal{T}_s$ which are empirically determined via experiments in advance. To this end, we will analyze and discuss the effect of selecting suspicious items on detection performance in section V-B. Note that, $\mathbf{w}$ can be calculated by solving the following optimization problem,

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_{i=1}^{n} (L(x_i, P(x_i, \mathcal{T}_s)) + \lambda_2 \|\mathbf{w}\|_2^2), \tag{6}$$

where λ_2 is a regularization parameter, n is the number of instances, and $L(\cdot)$ is a loss function. The above objective function can be solved by gradient descent algorithm. Thereinto, the update process of $\mathbf{w}$ can be written as $\mathbf{w}_j := \mathbf{w}_j - \xi \frac{1}{m} \sum_{i=1}^{m} (h_w(x_i) - y_i) x_i^j$, where ξ is a parameter for learning rate. $h_w(\cdot)$ is a constructed prediction function [30].

With regard to pairwise potential ψ_{ij}, the contrast sensitive *Potts* model [22] is exploited to characterize link attributes between nodes. Concretely, $\psi(x_i, x_j) = g(i, j)$ if $x_i \neq x_j$, $\psi(x_i, x_j) = 0$ otherwise. Thereinto, $g(i, j)$ is used to evaluate the edge feature, which can be typically defined as Eq. 7,

$$g(i, j) = \theta_2 \exp(-\theta_p \|I_i - I_j\|^2), \tag{7}$$

where θ_2 is a model parameter learned from training data. I_i and I_j denote the vector of nodes (users) i and j, respectively. $\theta_p = P_{\mathcal{T}_s}(u_i, u_j)$, which can be calculated by Eq. 8,

$$P_{\mathcal{T}_s}(u_i, u_j) = \frac{1}{1 + \exp(-\mathbf{w}^\top \mathbf{a}_{ij}^{\mathcal{T}_s})}, \tag{8}$$

where $\mathbf{a}_{ij}^{\mathcal{T}_s} = [SM_{ij}, IP_{ij}, CP_{ij}, SP_{ij}]$ is a feature vector of an edge (i, j) as shown in Table III. It is worth emphasizing that Eq. 8 can be solved via the similar way used in Eq. 5.

In addition to the unary and pairwise potentials, we also incorporate a high order potential to encode sophisticated statistics of natural situations which can not be expressed using unary and pairwise potentials. Moreover, segmentations obtained using pairwise CRFs tend to be oversmooth [23]. In reality, malicious attackers mimic rating and review details of *anchor* users in order to mix in the neighbors of the anchored users [4], [3], [11], naturally leading to high similarities between attackers and anchor users. Thus, dense link behaviors between nodes including the attackers and anchor users make a challenging issue for abnormality detection. Notably, dense or similar link behaviors suggest that some nodes constituting a particular region (or segment) belong to the same class (malicious or authentic). It also means that the existence of smooth object boundary of dense link behaviors enhances the difficulty of further identifying abnormal behaviors.

To address this issue, the high-order potential inspired from [22], [23], [24] defined on the region segments generated using unsupervised segmentation techniques is incorporated. Note that, the higher-order potential only provides an optimal mechanism for the smooth segmentation of dense behaviors, such as an optimal combination. In order to deal with high runtime complexity of energy function minimization, a family of high order potentials such as P^n Potts model and its robust model has been well studied. Given a clique c, concretely, the P^n Potts model $\psi_c(\mathbf{x}_c) = \gamma_k$ if $x_i = \ell_k$, $(\ell_k \in \mathcal{L})$,

Algorithm 2 Determination of malicious injection behaviors.

Require:
Remained graph $\mathcal{G}'$ and attributes of nodes and edges, $\mathbf{a}^{\mathcal{T}_s}$.
Ensure:
Detected result $\mathcal{D}_u$;
1: Initialize potentials $Dc = \emptyset$, $sG = \emptyset$, and $hop = \emptyset$;
2: $\mathcal{D}_u = \emptyset$;
3: **for** each node $u_i \in \mathcal{V}'$ in $\mathcal{G}'$ **do**
4: Calculate the unary potential of u_i, $Dc(index(u_i), :) = -\log \frac{P(u_i, \mathcal{T}_s)}{\sum_{u_j \in \mathcal{N}_i} P(u_j, \mathcal{T}_s)}$ using Eq. 5 and Eq. 6;
5: **end for**
6: **for** each edge $(u_i, u_j) \in \mathcal{E}'$ in $\mathcal{G}'$ **do**
7: Calculate the adjacency matrix defining the graph structure for the pairwise potential of association link (u_i, u_j), $sG(index(u_i), index(u_j)) = \exp(-\frac{1}{1+\exp(-\mathbf{w}^\top \mathbf{a}_{ij}^{\mathcal{T}_s})}\|I_i - I_j\|^2)$;
8: **end for**
9: Generate two clusters $c_i \subset \mathcal{V}'$, $i \in \{1, 2\}$ using an unsupervised segmentation algorithm and create a higher order potential array of structures with two entries $hop(i)$ of c_i, $i \in \{1, 2\}$;
10: **for** each entry $hop(i)$ **do**
11: Create indices of nodes belonging to $hop(i)$, calculate weights of participating nodes and deal with the potential function parameters used in Eq. 10;
12: **end for**
13: Based on Eq. 4, Eq. 7, and Eq. 10, calculate the most probable or maximum a posterior (MAP) labelling of the random field using Eq. 2 and obtain the globally optimal labels $\mathcal{L}^*$;
14: **for** each user $u \in \mathcal{U}$ **do**
15: **if** u's label $\mathcal{L}^*(u) = \ell_a$ and u rated $i \in \mathcal{T}_s$ **then**
16: $\mathcal{D}_u \leftarrow u$;
17: **end if**
18: **end for**
19: **return** $\mathcal{D}_u$;

$\forall i \in c$, $\psi_c(\mathbf{x}_c) = \gamma_{max}$ otherwise, where γ_k and γ_{max} ($\gamma_k \leq \gamma_{max}$) are potential function parameters. To handle potential functions defined over very large cliques, a robust P^n Potts model potentials has been developed,

$$\psi_c(\mathbf{x}_c) = \min\{\underbrace{\min_{k \in \mathcal{L}}((|c| - n_k(\mathbf{x}_c))\theta_k + \gamma_k)}_{\textbf{reduce label inconsistency}}, \gamma_{max}\} \quad (9)$$

where θ_k represents a potential function parameter. $|c|$ denotes the number of variables in c. $n_k(\mathbf{x}_c)$ is the number of variables in c which take label $k \in \mathcal{L}$ in labelling $\mathbf{x}_c$ [22]. Precisely, encouraging all the variables in c to take the same label k is involved in the higher order potential. To reduce label inconsistency, the potential tries to reduce the number of variables in c not taking the *dominant* label [23].

It is worthy emphasizing that the remained nodes in $\mathcal{G}'$ are roughly divided into two clusters using an unsupervised segmentation method. In our experiments, we utilized k-means algorithm [31]. We calculated within-cluster sums of point-to-centroid distances to determine the detected cluster c_d, where labels ℓ_a and ℓ_g denote anomalous and genuine nodes, respectively (see Algorithm 2). The goal of smooth boundary segmentation in dense node (user or visitation) behaviors is to remove all authentic nodes (predicted as ℓ_a) as far as possible and retain all anomalous nodes with ℓ_a in c_d.

For instance, a possible assignment of labels to the detected nodes (variables) can be represented as a configuration $x = (\ell_a, \ell_a, \ell_g, \ell_a, \ell_g)$ (take 5 variables (nodes) $\mathbf{x} = \{x_1, x_2, \cdots, x_5\}$ in c_d for example). For the P^n Potts model, all labels of the variables will be assigned a cost γ_{max}. In contrast, the robust P^n Potts model assigns the cost: $\gamma_i + \frac{\gamma_{max} - \gamma_i}{3} \times 2$, $\gamma_{max} \geq \gamma_i$ to the same configuration, where the truncation parameter [22] of the potential is 3, $|c| - n_k(\mathbf{x}_c) = 2$ denotes the number of variables which are assigned labels different from ℓ_a. Specially, two variables (x_3 and x_5) which are assigned labels different from the dominant label ℓ_a are discriminatively considered. To produce the globally optimal segmentation and reduce the inconsistency cost for the smooth boundary in polynomial time [23], the generalized form of the robust P^n Potts model can be rewritten as,

$$\begin{aligned}
\psi_c(\mathbf{x}_c) &= \min\{\min_{k \in \mathcal{L}}((P - f_k(\mathbf{x}_c))\theta_k + \gamma_k), \gamma_{max}\} \\
&= \min\{\min_{k \in \mathcal{L}}((\sum_{i \in c} w_i^k - \sum_{i \in c} w_i^k \delta_k(x_i))\frac{\gamma_{max} - \gamma_k}{Q_k} \\
&\quad + \gamma_k), \gamma_{max}\} \\
&= \min\{\min_{k \in \mathcal{L}}(\sum_{i \in c} w_i^k (1 - \delta_k(x_i))\frac{\gamma_{max} - \gamma_k}{Q_k} + \gamma_k), \\
&\quad \gamma_{max}\}
\end{aligned} \quad (10)$$

where γ_k, θ_k, and γ_{max} are potential function parameters which satisfy: $\theta_k = \frac{\gamma_{max} - \gamma_k}{Q_k}$, $\gamma_k \leq \gamma_{max}, \forall k \in \mathcal{L}$. $Q_k (k \in \mathcal{L} = \{\ell_a, \ell_g\})$ is the truncation parameter of the potential function and satisfies the constraints $Q_a + Q_b < P$, $\forall a \neq b \in \mathcal{L}$. $f_k(\mathbf{x}_c) = \sum_{i \in c} w_i^k \delta_k(\mathbf{x}_i)$ and $\sum_{i \in c} w_i^k = P, \forall k \in \mathcal{L}$, where $\delta_k(x_i)$ is 1 if $\mathbf{x}_i = k$, 0 otherwise. The weight $w_i^k \geq 0, i \in c, k \in \mathcal{L}$ can be used to encode the relative importance of different variables.

Based on the unary potential in Eq. 4, the pairwise potential in Eq. 7, and the higher order potential in Eq. 10, the globally optimal configuration for random variables (the remained nodes in $\mathcal{G}'$) can be obtained by calculating the most probable or maximum a posterior (MAP) labelling of the random field as shown in Eq. 2. It is worthy emphasizing that the higher order clique potentials can be computed using optimal *expansion* and *swap* moves [22] in polynomial time by solving a *st-mincut* problem [24]. The basic process of dealing with the smoothness segmentation is described in Algorithm 2.

2) Determination of Malicious Behaviors: As above mentioned, the dense region of link behaviors based on the remained graph $\mathcal{G}'$ needs to be further discriminated. In the light of representing sophisticated statistics of natural scenes via high order potentials, we incorporate a class of higher order clique potentials to characterize relevance and interaction between link behaviors in the concerned dense region. Algorithm 2 shows a basic process of determining malicious behaviors. Note that, line 3-12 is used to construct each potential. The globally optimal labels $\mathcal{L}^*$ are calculated

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2020.3016827, IEEE Transactions on Information Forensics and Security

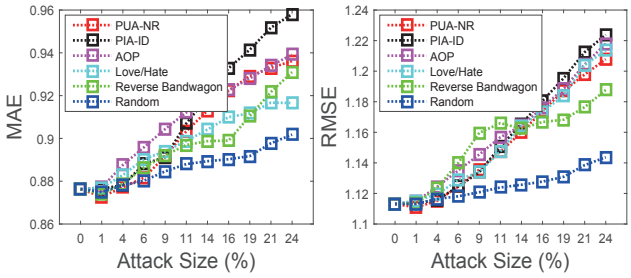
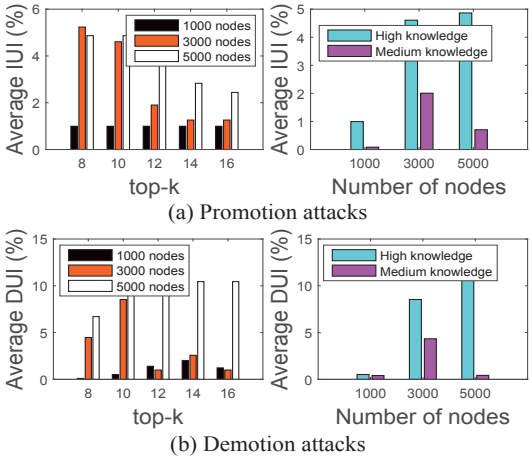


Fig. 5: The effect of profile injection attacks in terms of both mean square error (MSE) and root mean square error (RMSE), where the length of recommendation list and filler size are 200 and 7.3%, respectively.



(a) Promotion attacks

(b) Demotion attacks

Fig. 6: The effect of promotion and demotion co-visitation injection attacks with both high and medium knowledge in terms of different numbers of nodes and recommendation lists.

according to Eq. 4, Eq. 7, and Eq. 10, where sG denotes a sparse adjacency matrix defining the graph structure and the pairwise potential. $sG(i,j) \neq 0$ means nodes i and j share a pairwise potential with value $sG(i,j)$. Dc is the unary potential, which represents a $|\mathcal{V}'| \times |\mathcal{L}|$ matrix. hop denotes a higher order potential array of structures with two entries. For each entry $hop(i), i \in \{1,2\}$, $hop(i).ind$ is the indices of nodes belonging to this hop, $hop(i).w$ is the weights for each participating node, $hop(i).\gamma$ contains $|\mathcal{L}|+1$ entries for $\gamma_1, \cdots, \gamma_{max}$, and $hop(i).Q$ is the truncation value for this potential (assumes one Q for all labels), which controls the rigidity of the higher order clique potential [22], [23]. More details of parameters will be analyzed in the experiments (see Section V-B). Ultimately, based on the optimal labels $\mathcal{L}^*$, each user (node) is further determined by analyzing suspicious items $\mathcal{T}_s$. Concretely, a user u can be considered as an anomalous user if its label is ℓ_a and u has rated $i \in \mathcal{T}_s$.

V. Large-Scale Simulation

This section aims at answering the following questions: (1) how do the relative parameters of the proposed method impact the detection performance? (2) how to evaluate the detection performance of the proposed method in both co-visitation injection attacks and profile injection attacks compared with benchmarks? and (3) how to demonstrate the detection performance of the presented detection approach on real data?

A. Experimental Setting

1) Datasets: All datasets utilized in the experiments can be categorized into synthetic data and real-world data. Table IV summarizes basic statistics of the adopted datasets, including ML-100K, Amazon, LibraryThing, and TripAdvisor [32].

To generate synthetic data, firstly, the MovieLens-100K (ML-100K) dataset is considered as authentic data [4] for profiles injection attacks. In our experiments, we exploited 6 representative shilling attack models as aforementioned to generate attack profiles with diverse attack sizes {0.5%, 1.1%, 6.4%, 11.7%, 17.0%} and filler sizes {7.3%, 16.4%} [4]. The effect of each attack has been confirmed using MSE and RMSE [1] as shown in Figure 5. After that, each set of conducted attack profiles is combined with the authentic data to construct the final synthetic dataset. Therefore, for

TABLE IV: Basic statistics of experimental datasets.

Dataset	#Users	#Items	#Reviews	Rating
ML-100K	943	1,682	——	[1, 5]
Amazon	21,000,000	9,350,000	142,800,000	[1, 5]
LibraryThing	73,882	337,561	979,053	[0.5, 5]
TripAdvisor	12,773	1,759	235,793	[1, 5]

analyzing profile injection attack detection, 60 ($6 \times 5 \times 2$) datasets including 6 different attacks, 5 diverse attack sizes, and 2 different filler sizes can be obtained. Note that, the authentic users and injected users created by attacks were labeled as authentic and anomalous nodes in the experiments, respectively. For profile injection attacks, we only detect nuke attacks and push attacks can be detected in the analogous manner. For co-visitation injection attacks, each co-visitation graph is similarly constructed by inserting each corresponding co-visitation attack data into an original *Erdos-Renyi* (ER) random graph [3]. Additionally, we implemented each co-visitation injection attack (i.e., PH, PM, DH, and DM) with different numbers of nodes as illustrated in Figure 6. Hence, we have obtained 40 ($4 \times 5 \times 2$) graphs including 4 diverse attack models, 5 different numbers of nodes, and 2 different attack intentions for co-visitation injection attack detection. In particular, nodes (items) used to construct malicious co-visitations were labeled as anomalous nodes. The remained nodes were labeled as genuine nodes. Comparative experiments on the above data will be analyzed in section V-C2.

For the real-world data, as argued in [26], we used 3 real-world datasets including Amazon, TripAdvisor, and LibraryThing (see Table IV) to discover spam users. Concretely, to remove the cold users and items, take the Amazon dataset for example, we empirically selected those users and items with at least 40 historical reviews and at least 50 raters, respectively. By considering the features of Amazon dataset, including the number of times that a review has been rated as helpfulness or unhelpfulness and the number of readers by other online users, we selected those users whose percentages of helpfulness ratings are less than 0.1 as candidate spam users

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2020.3016827, IEEE Transactions on Information Forensics and Security

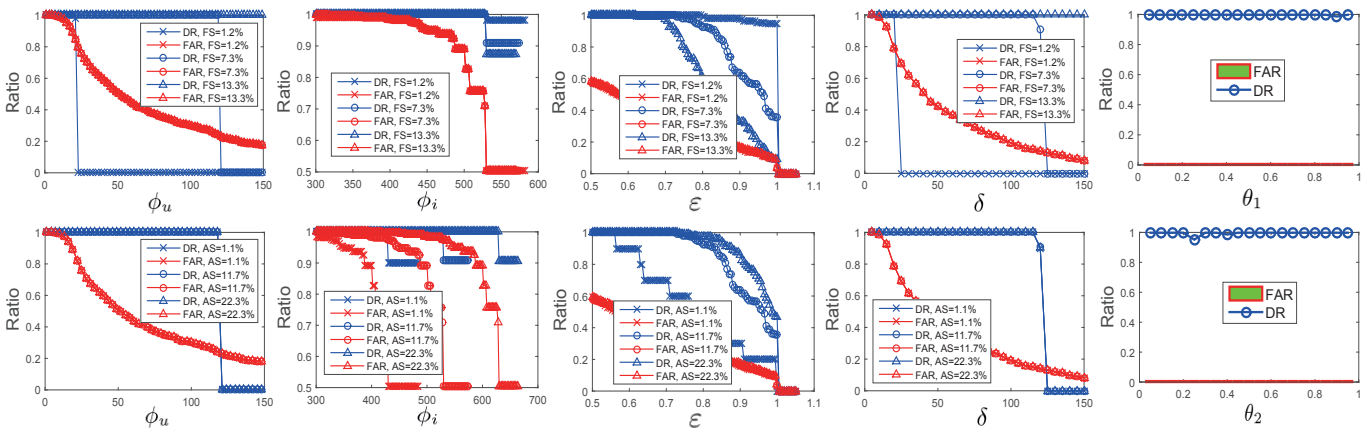


Fig. 7: Sensitivity analysis of model parameters (take the AOP attack for example), where for parameters ϕ_u, ϕ_i, ε, and δ, the attack size is 11.7% and filer size varies in the first line of the figure, the attack size varies and filler size is 7.3% in the second line of the figure. For parameters θ_1 and θ_2, the attack size and filler size are 11.7% and 7.3%, respectively.

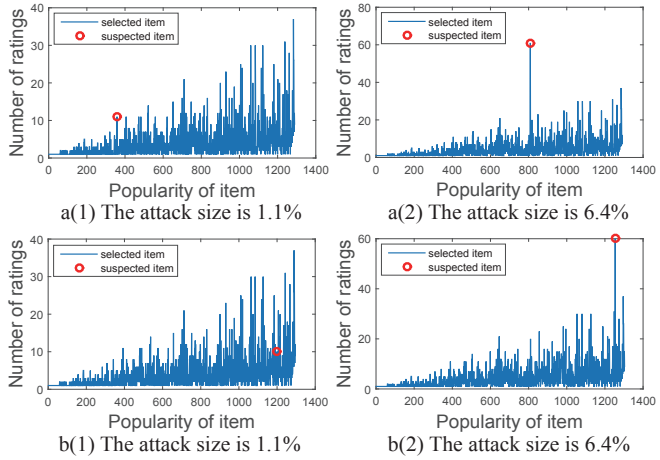


Fig. 8: Distribution of suspicious items for determining target items in two different attacks, where the filler size is 7.3% and the attack size varies. a(1)-(2) and b(1)-(2) denote reverse bandwagon and PIA-ID attacks, respectively.

[26]. Moreover, these selected spam users have been further determined by three human annotators via observing the review content of users. The user is considered as a spam user if he reviews all concerned items using the same review content and gives a rating to the items with significant intentions (e.g., $\mathcal{R}_{min}$, $\mathcal{R}_{max}$). In reality, identical and repeated comments may reflect attack behaviors (e.g., machine behavior). The consistent rating ($\mathcal{R}_{min}$, take the nuke attack for example) may also reflect the intention of attack. Finally, we obtained 161 spam users on Amazon. In the same way, we also got 5 spam users on TripAdvisor and 583 spam users on LibraryThing.

Regarding review data used in Amazon, LibraryThing, and TripAdvisor datasets, it is worth emphasizing that the sentimental polarity SP between two users as shown in Table III was obtained in the following steps:

1) Preprocess review data and remove useless symbols (e.g., special characters) from reviews;
2) Extract TF-IDF [33] features in the review data and train a logistic regression (LR) model [34];
3) Assign a label to each user (node) according to the predicted probabilities of LR;
4) Construct a label vector $(\ell_{i1}, \cdots, \ell_{i|\mathcal{T}_s|})$ for each user based on $\mathcal{T}_s$.

2) Evaluation Metrics: To measure detection performance of the presented methods, the detection rate (DR) and false alarm rate (FAR) are adopted [4], [12]. DR is defined as the number of detected attackers divided by the number of all attackers in the system. FAR is defined as the number of genuine users that are predicted as attackers divided by the number of genuine users [35]. The evaluation measures are defined in Eq. 11,

$$DR = \frac{|\mathcal{U}^d \cap \mathcal{U}^a|}{|\mathcal{U}^a|}, FAR = \frac{|\mathcal{U}^d \cap \mathcal{U}^g|}{|\mathcal{U}^g|} \tag{11}$$

where $\mathcal{U}^d$, $\mathcal{U}^a$, and $\mathcal{U}^g$ are the set of detected users, all attackers, and all genuine users, respectively. Note that, DRs and FARs with *bullets* in all tables represent the best results in what follows. Additionally, to further evaluate the discrimination of label prediction, we use PANT (labels are predicted to be anomalous and have not rated target items) as shown in Eq. 12, which is defined as the number of users whose labels are predicted to be l_a (anomalous) and those users have not rated suspected items $\mathcal{T}_s$ divided by the number of genuine users,

$$PANT = \frac{|\mathcal{U}^d_{\overline{\mathcal{T}_s}} \cap \mathcal{U}^g|}{|\mathcal{U}^g|} \tag{12}$$

where $\mathcal{U}^d_{\overline{\mathcal{T}_s}}$ is the set of users whose labels are predicted to be l_a and have not rated $\mathcal{T}_s$. All detected results in co-visitation injection attacks have been averaged using the above metrics.

B. Parameter Analysis

To reasonably determine empirical thresholds of parameters used in the proposed approach, a series of experiments have been implemented. Due to space limit, we only investigate how these parameters affect the detection results on synthetic data (take AOP attack for example) as shown in Figure 7. We can see that the false alarm rates gradually decrease with the

TABLE V: Detection results of the presented methods in five different profile injection attacks with different cases.

Attack model	Filler size	Attack size ($\lvert\mathcal{N}^a\rvert/\lvert\mathcal{N}^g\rvert$)	Metric	Benchmarks						
				DeR-TIA	$\beta\mathcal{P}$-based	PCA-based	CBS	TIA-UnaryPot	TIA-PairCRF	IMIA-HCRF
PIA-ID	7.3%	1.1% (10/943)	DR	0	0.8750	0.6000	1.0	1.0	1.0	1.0 •
			FAR	0.3181	0.3735	0.2900	0.5482	0.0806	0.0498	0.0414 •
		6.4% (60/943)	DR	0.85	0.8590	0.5000	1.0	1.0	1.0	1.0 •
			FAR	0.2640	0.3619	0.2800	0.3160	0.2099	0.0498	0 •
		11.7% (110/943)	DR	1.0	0.8750	0.4333	1.0	1.0	1.0	1.0 •
			FAR	0.2014	0.3542	0.3100	0.3309	0.2132	0.0445	0 •
		17.0% (160/943)	DR	1.0	0.9080	0.4500	1.0	1.0	1.0	1.0 •
			FAR	0.1484	0.3572	0.3100	0.3510	0.2492	0.0148	0 •
	16.4%	1.1% (10/943)	DR	0	0.8210	0.6000	1.0	1.0	1.0	1.0 •
			FAR	0.3181	0.3735	0.2900	0.5472	0.3860	0.2874	0.1103 •
		6.4% (60/943)	DR	0.7166	0.7450	0.5000	1.0	1.0	1.0	1.0 •
			FAR	0.2725	0.3619	0.2800	0.3234	0.2110	0.0891	0 •
		11.7% (110/943)	DR	1.0	0.8750	0.4333	1.0	1.0	1.0	1.0 •
			FAR	0.2014	0.3542	0.3100	0.3277	0.2418	0.0615	0 •
		17.0% (160/943)	DR	1.0	0.9080	0.4500	1.0	1.0	1.0	1.0 •
			FAR	0.1484	0.3572	0.3100	0.3489	0.3733	0.0541	0 •
PUA-NR	7.3%	1.1% (10/943)	DR	0	0.8750	0.6000	1.0	1.0	1.0 •	1.0 •
			FAR	0.3177	0.3683	0.2900	0.5620	0.2068	0.0615 •	0.0615 •
		6.4% (60/943)	DR	0.0333	0.7450	0.4500	1.0	1.0	1.0	1.0 •
			FAR	0.3156	0.3529	0.2700	0.3743	0.5695	0.1156	0 •
		11.7% (110/943)	DR	0.0727	0.9570	0.4333	1.0	1.0	1.0	1.0 •
			FAR	0.3093	0.3285	0.2900	0.4062	0.5748	0.1241	0 •
		17.0% (160/943)	DR	0.1062	0.9870	0.4500	1.0	1.0	1.0	1.0 •
			FAR	0.3001	0.3309	0.3000	0.4507	0.5801	0.1273	0 •
	16.4%	1.1% (10/943)	DR	0	0.7850	0.6000	1.0	1.0	1.0	1.0 •
			FAR	0.3177	0.3683	0.2900	0.5620	0.3722	0.0562	0.0233 •
		6.4% (60/943)	DR	0.0333	0.7450	0.4500	1.0	1.0	1.0	1.0 •
			FAR	0.3156	0.3529	0.2700	0.3743	0.5546	0.1124	0.0010 •
		11.7% (110/943)	DR	0.0727	0.9570	0.4333	1.0	1.0	1.0	1.0 •
			FAR	0.3093	0.3285	0.2900	0.4062	0.5705	0.0329	0 •
		17.0% (160/943)	DR	0.1062	0.9870	0.4500	1.0	1.0	1.0	1.0 •
			FAR	0.3001	0.3309	0.3000	0.4507	0.5939	0.0944	0 •
Love/Hate	7.3%	1.1% (10/943)	DR	0	0.6250	0.1000	1.0	1.0	1.0	1.0 •
			FAR	0.3181	0.3693	0.0100	0.5472	0.1379	0.1379	0.0349 •
		6.4% (60/943)	DR	1.0	0.7950	0.4500	1.0	1.0	1.0	1.0 •
			FAR	0.2545	0.3619	0.3000	0.3160	0.1442	0.1007	0 •
		11.7% (110/943)	DR	1.0	0.8870	0.0333	1.0	1.0	1.0	1.0 •
			FAR	0.2014	0.3542	0.0100	0.3160	0.1983	0.0997	0 •
		17.0% (160/943)	DR	1.0	0.9570	0.0250	1.0	1.0	1.0	1.0 •
			FAR	0.1484	0.3417	0.0100	0.3160	0.1432	0.1432	0 •
	16.4%	1.1% (10/943)	DR	0	0.6250	0.1000	1.0	1.0	1.0	1.0 •
			FAR	0.3181	0.3693	0.0100	0.5472	0.1389	0.1389	0.0190 •
		6.4% (60/943)	DR	1.0	0.8750	0.4500	1.0	1.0	1.0	1.0 •
			FAR	0.2545	0.3619	0.3000	0.3160	0.1463	0.1007	0.0074 •
		11.7% (110/943)	DR	1.0	0.9870	0.0333	1.0	1.0	1.0	1.0 •
			FAR	0.2014	0.3542	0.0100	0.3160	0.1919	0.0944	0 •
		17.0% (160/943)	DR	1.0	1.0	0.0250	1.0	1.0	1.0	1.0 •
			FAR	0.1484	0.3417	0.0100	0.3160	0.1442	0.1007	0 •
AOP	7.3%	1.1% (10/943)	DR	0	0.8280	0.6000	1.0	1.0	1.0	1.0 •
			FAR	0.3181	0.3693	0.2900	0.5472	0.4104	0.1326	0.0996 •
		6.4% (60/943)	DR	0.5000	0.8790	0.4500	1.0	1.0	1.0	1.0 •
			FAR	0.2863	0.3589	0.2700	0.3266	0.4178	0.0742	0 •
		11.7% (110/943)	DR	1.0	0.9150	0.4333	1.0	1.0	1.0	1.0 •
			FAR	0.2014	0.3504	0.3100	0.3542	0.1082	0.0148	0 •
		17.0% (160/943)	DR	1.0	0.9150	0.4500	1.0	1.0	1.0	1.0 •
			FAR	0.1484	0.3372	0.3200	0.3659	0.3181	0.0817	0 •
	16.4%	1.1% (10/943)	DR	0	0.8280	0.6000	1.0	1.0	1.0	1.0 •
			FAR	0.3181	0.3693	0.2900	0.5483	0.4486	0.1209	0.0869 •
		6.4% (60/943)	DR	0.3000	0.8790	0.4500	1.0	1.0	1.0	1.0 •
			FAR	0.2990	0.3589	0.2700	0.3531	0.2842	0.0318	0 •
		11.7% (110/943)	DR	1.0	1.0	0.4333	1.0	1.0	1.0	1.0 •
			FAR	0.2014	0.3504	0.3100	0.3786	0.1474	0.0042	0 •
		17.0% (160/943)	DR	1.0	1.0	0.4500	1.0	1.0	1.0	1.0 •
			FAR	0.1484	0.3372	0.3200	0.3924	0.2344	0.0382	0 •
Reverse Bandwagon	7.3%	1.1% (10/943)	DR	0	0.8750	0.6000	1.0	1.0	1.0	1.0 •
			FAR	0.3181	0.3683	0.2900	0.5472	0.3616	0.2375	0.0719 •
		6.4% (60/943)	DR	0.4166	0.8250	0.4500	1.0	1.0	1.0	1.0 •
			FAR	0.2916	0.3529	0.2700	0.3160	0.2333	0.0965	0 •
		11.7% (110/943)	DR	1.0	0.9570	0.4333	1.0	1.0	1.0	1.0 •
			FAR	0.2014	0.3285	0.2900	0.3329	0.2386	0.1039	0 •
		17.0% (160/943)	DR	1.0	1.0	0.4500	1.0	1.0	1.0	1.0 •
			FAR	0.1484	0.3309	0.3200	0.3553	0.2503	0.0880	0 •
	16.4%	1.1% (10/943)	DR	0	0.7850	0.6000	1.0	1.0	1.0	1.0 •
			FAR	0.3181	0.3683	0.2900	0.5472	0.3266	0.1209	0.0960 •
		6.4% (60/943)	DR	0.1166	0.8250	0.4500	1.0	1.0	1.0	1.0 •
			FAR	0.3107	0.3529	0.2700	0.3340	0.1251	0.0201	0 •
		11.7% (110/943)	DR	1.0	1.0	0.4333	1.0	1.0	1.0	1.0 •
			FAR	0.2014	0.3285	0.2900	0.3446	0.1251	0.0212	0 •
		17.0% (160/943)	DR	1.0	1.0	0.4500	1.0	1.0	1.0	1.0 •
			FAR	0.1484	0.3309	0.3200	0.3648	0.1315	0.0127	0 •

increase of ϕ_u, ϕ_i, ε, and δ while the detection rates keep unchanged. The goal of eliminating disturbed data regarding ϕ_u, ϕ_i, ε, and δ is to keep the detection rates unchanged and choose an acceptable threshold for each of them. To this end, we conservatively set $\phi_u = 15$, $\phi_i = 400$, $\varepsilon = 0.55$, and $\delta = 20$ in order to adapt different attack sizes and filler sizes in our experiments. It is worth emphasizing that these four thresholds can be set as larger values actually. This is due to the fact that some interactions may exist between the elimination of disturbed data and construction of association graph. Intuitively, preserving a part of disturbed data is favorable to construct topological relationships of association graph. Additionally, θ_1 and θ_2 are parameters weighting the potentials obtained from Eq. 4 and Eq. 7. According to our experimental results, both θ_1 and θ_2 have little effect on the detection performance in a given range (i.e., $0.005 < \theta_1 < 0.995$, $0.005 < \theta_2 < 0.995$). The main reason may be that, most of disturbed data have been eliminated in advance, which leads to a relatively weak impact on the detection performance using the combination of unary and pairwise attributes. The

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2020.3016827, IEEE Transactions on Information Forensics and Security

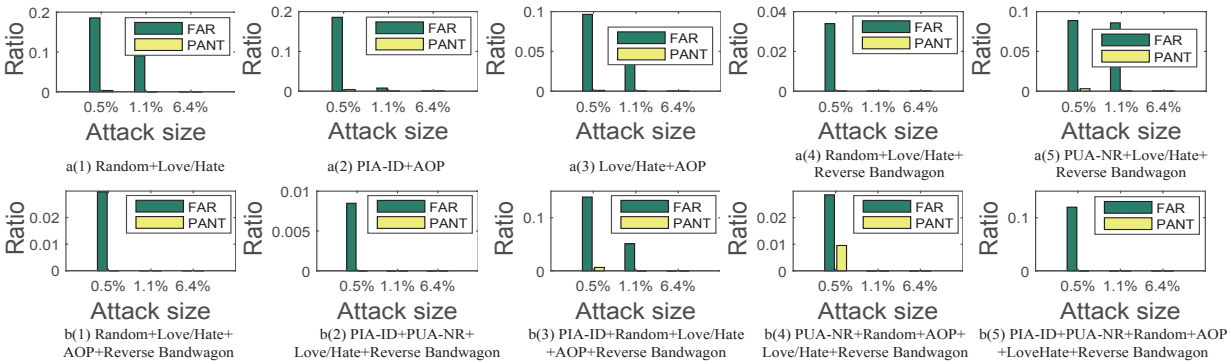


Fig. 9: Analysis of detection details for the proposed approach using FAR and PANT in ten mixed profile injection attacks with different attack sizes, where the filler size is 7.3%.

goal of analyzing the sensitivity of θ_1 and θ_2 is to determine a reasonable threshold for both of them in a given range. Here, we set $\theta_1 = 0.5$ and $\theta_2 = 0.5$ in our experiments. For parameters γ_{max} and Q, we set $\gamma_{max} = |c_i|$ and $Q = 0.1 \times |c_i|$ in the experiments inspired from [23].

To determine target items by analyzing the distribution of items, we have implemented related experiments as shown in Figure 8. Take the nuke attack for example, attackers generally give the lowest rating to target items. As a result, the special rating (i.e., $\mathcal{R}_{min}$) focused on the target items is relatively suspicious. For each item, therefore, we first calculated the number of users who have given the $\mathcal{R}_{min}$ score to the item. Then, we utilized an empirical threshold of the number to determine suspected items $\mathcal{T}_s$. Note that, the suspected items contain the concerned target items, $\mathcal{T}_t \subseteq \mathcal{T}_s$. In Figure 8, the x-axis denotes the sorted popularity of items. We can observe that the difference of rating numbers between the suspected item and normal items is significant when the attack size is 6.4%. Comparatively, it is not easy to determine suspected items using a threshold of rating number when the attack size is 1.1%. In our experiments, we empirically set the threshold of rating number to 10 in order to roughly select suspected items. Undoubtedly, the limitation of determining suspected items is evident. We will discuss the limitation and possible improvements in what follows.

C. Comparative Experiments

1) Profile Injection Attack Detection: To demonstrate the effectiveness of the proposed detection approach (IMIA-HCRF) for spotting profile injection attacks, we conducted extensive experiments in five different attacks as shown in Table V. Besides, six representative benchmarks have been implemented compared with the proposed detection approach, including DeR-TIA [12], $\beta\mathcal{P}$-based [35], PCA-based [20], CB-S [26], TIA-UnaryPot (the first-order potential based on target item analysis), and TIA-PairCRF (the second-order potential based on target item analysis). In Table V, one observation is that the detection performance of the proposed approach outperforms all baselines in almost all cases of each attack. Specifically, the detected results of the proposed approach close to a perfect level (the highest DR and the lowest FAR) in each attack when the attack sizes are relatively large. Although the highest detection rate of DeR-TIA, CBS, TIA-UnaryPot, and TIA-PairCRF can be obtained in most cases, the false alarm rates of them are not acceptable compared with IMIA-HCRF. Note that, DeR-TIA specially analyzes the distribution of target items, it is very favorable to reduce the concerned scope of detection (we set its absolute count threshold θ = 6). Nevertheless, the clustering results in its early stage may include a part of genuine users. Additionally, for PCA-based, exploiting rating styles including both rating assignment and item distribution may tend to be invalid when the attackers specially choose popular items to construct attack profiles. Regarding $\beta\mathcal{P}$-based, the limitation of definition of Beta distribution may restrain performance boost of outlier detection. Note that, the quantiles used in the first and second stages, q_f and q_s, were set as 8E-13 and 0.00115 in the experiments, respectively. Finally, for CBS, the detection results are very closed under all attacks. These results may be attributed to the similar equivalence of bipartite graphs converted from the original rating matrix, or possibly due to the fact that similar attack profiles are injected into the original rating data [26]. Note that, the parameters were set as $p_u = 1$ and $p_i = 1$ for the user and product spam probability vectors, respectively.

The second observation is that smaller false alarm rates of the proposed approach are encountered in all attacks when the attack size is 1.1%. In reality, attack and detection are a game process. Intuitively, the larger the attack scale, the easier it is to be detected. Figure 5 demonstrates the effect of profile injection attacks in different attack sizes. We can see that the attack effect is insignificant when the attack size is 1.1%. A small-scale attack has a weak attack effect as well as restricts the representation of attack behaviors. Therefore, making a balance between the impact of attacks and the necessity of detection is desirable.

To evaluate the effectiveness of the proposed detection approach facing with mixed profile injection attacks, we also conducted a list of experiments as shown in Table VI. We can see that the proposed IMIA-HCRF outperforms all baselines in different cases with an advantage of 7.8% FAR in average. It is more easy to be detected in the mixture of multiple attacks (e.g., 6 different attacks) compared with the case of small-scale mixed attacks (e.g., only 2 different attacks). Note that, the distribution characteristic of suspected items can be

TABLE VI: Detection results of the presented methods in ten mixed profile injection attacks with different attack sizes, where the filler size is 7.3%.

Mixed attacks	Metric	Benchmark methods					
		DeR-TIA	PCA-based	CBS	TIA-UnaryPot	TIA-PairCRF	IMIA-HCRF
0.5% Random + 0.5% Love/Hate	DR	1.0	0	1.0	1.0	1.0	1.0 •
	FAR	0.3679	0.1792	0.5673	0.4750	0.3361	0.1855 •
1.1% Random + 1.1% Love/Hate	DR	1.0	0	1.0	1.0	1.0	1.0 •
	FAR	0.3711	0.1770	0.3266	0.1845	0.0986	0.0901 •
6.4% Random + 6.4% Love/Hate	DR	1.0	1.0	1.0	1.0	1.0	1.0 •
	FAR	0.3531	0.2099	0.2375	0.1315	0.0445	0 •
0.5% PIA-ID + 0.5% AOP	DR	0	0.2000	1.0	1.0	1.0	1.0 •
	FAR	0.3594	0.1845	0.5684	0.5068	0.3997	0.1855 •
1.1% PIA-ID + 1.1% AOP	DR	0.5500	0.4500	1.0	1.0	1.0	1.0 •
	FAR	0.3563	0.1908	0.3351	0.2623	0.1581	0.0080 •
6.4% PIA-ID + 6.4% AOP	DR	0.4916	0.5416	1.0	1.0	1.0 •	1.0 •
	FAR	0.3573	0.2025	0.2481	0.0381	0 •	0 •
0.5% AOP + 0.5% Love/Hate	DR	0.5000	0.5000	1.0	1.0	1.0	1.0 •
	FAR	0.3563	0.1834	0.5673	0.4316	0.1261	0.0965 •
1.1% AOP + 1.1% Love/Hate	DR	0.5500	0.4500	1.0	1.0	1.0	1.0 •
	FAR	0.3573	0.1834	0.3351	0.3160	0.1240	0.0487 •
6.4% AOP + 6.4% Love/Hate	DR	0.5000	0.5000	1.0	1.0	1.0	1.0 •
	FAR	0.3510	0.2046	0.2481	0.1961	0.0381	0 •
0.5% Random + 0.5% Love/Hate + 0.5% Reverse Bandwagon	DR	1.0	0	1.0	1.0	1.0	1.0 •
	FAR	0.3679	0.1792	0.5673	0.1537	0.0349	0.0339 •
1.1% Random + 1.1% Love/Hate + 1.1% Reverse Bandwagon	DR	1.0	0	1.0	1.0	1.0	1.0 •
	FAR	0.3764	0.1717	0.3266	0.1845	0.0795	0 •
0.5% PUA-NR + 0.5% Love/Hate + 0.5% Reverse Bandwagon	DR	0.6666	0.2666	1.0	1.0	1.0	1.0 •
	FAR	0.4093	0.1527	0.5832	0.2640	0.1386	0.0887•
1.1% PIA-ID + 1.1% Love/Hate + 1.1% Reverse Bandwagon	DR	0.6666	0.3333	1.0	1.0	1.0	1.0 •
	FAR	0.4199	0.1474	0.3340	0.2820	0.1474	0.0858 •
0.5% Random + 0.5% Love/Hate + 0.5% AOP + 0.5% Reverse Bandwagon	DR	0.7500	0.2500	1.0	1.0	1.0 •	1.0 •
	FAR	0.3605	0.1834	0.5673	0.1261	0.0296 •	0.0296 •
1.1% Random + 1.1% Love/Hate + 1.1% AOP + 1.1% Reverse Bandwagon	DR	0.7750	0.2250	1.0	1.0	1.0	1.0 •
	FAR	0.3637	0.1834	0.3351	0.0699	0.0360	0 •
0.5% PIA-ID + 0.5% PUA-NR + 0.5% Love/Hate + 0.5% Reverse Bandwagon	DR	0.7500	0.2500	1.0	1.0	1.0	1.0 •
	FAR	0.3891	0.1537	0.5832	0.1792	0.1707	0.0084 •
1.1% PIA-ID + 1.1% PUA-NR + 1.1% Love/Hate + 1.1% Reverse Bandwagon	DR	0.7500	0.2500	1.0	1.0	1.0	1.0 •
	FAR	0.4199	0.1474	0.3340	0.1110	0.0434	0 •
0.5% PIA-ID + 0.5% Random + 0.5% Love/Hate + 0.5% AOP + 0.5% Reverse Bandwagon	DR	0.6000	0.2000	1.0	1.0	1.0	1.0 •
	FAR	0.3605	0.1834	0.5684	0.4984	0.3966	0.1389 •
1.1% PIA-ID + 1.1% Random + 1.1% Love/Hate + 1.1% AOP + 1.1% Reverse Bandwagon	DR	0.8200	0.1800	1.0	1.0	1.0	1.0 •
	FAR	0.3637	0.1877	0.3351	0.2661	0.1739	0.0509 •
0.5% PUA-NR + 0.5% Random + 0.5% Love/Hate + 0.5% AOP + 0.5% Reverse Bandwagon	DR	0.6000	0.3600	1.0	1.0	1.0	1.0 •
	FAR	0.3891	0.1707	0.5853	0.1866	0.1601	0.0286 •
1.1% PUA-NR + 1.1% Random + 1.1% Love/Hate + 1.1% AOP + 1.1% Reverse Bandwagon	DR	0.7000	0.3000	1.0	1.0	1.0	1.0 •
	FAR	0.4199	0.1474	0.3425	0.1887	0.0159	0 •
0.5% PIA-ID + 0.5% PUA-NR + 0.5% Random + 0.5% Love/Hate + 0.5% AOP + 0.5% Reverse Bandwagon	DR	0.6666	0.3333	1.0	1.0	1.0	1.0 •
	FAR	0.3891	0.1707	0.5853	0.5228	0.1219	0.1198 •
1.1% PIA-ID + 1.1% PUA-NR + 1.1% Random + 1.1% Love/Hate + 1.1% AOP + 1.1% Reverse Bandwagon	DR	0.7500	0.2500	1.0	1.0	1.0	1.0 •
	FAR	0.4199	0.1474	0.3425	0.1548	0.0190	0 •

partly strengthened in the mixed attacks, which is favorable to determine the target items. Nevertheless, high FARs still exist when the attack size is small. It is possible to obtain the highest DR by sacrificing FARs to some extent.

In addition, we also have implemented a list of experiments for analyzing detection details as shown in Figure 9. In order to show the ratio of users whose labels are predicted to be l_a (anomalous) and those users have not rated suspected items $\mathcal{T}_s$, we exploit PANT and FAR in different attacks. We can observe that the PANTs are insignificant in almost all cases. Despite the small FARs when the attack sizes are small, there are very few detected users that have not rated $\mathcal{T}_s$. The results may be attributed to the use of the unsupervised segmentation method before implementing the higher order potential, or possibly due to the fact that the representation of node and link behaviors is limited when the attack sizes are small. Of course, our experimental results are for reference only.

2) Co-visitation Injection Attack Detection: To examine the detection performance of the proposed approach facing with co-visitation injection attacks, a list of experiments has been implemented under different co-visitation injection attacks as shown in Table VII. In the experiments, we exploited four different benchmarked methods including *K-means*-based, *HierarchicalCluster*-based, *DensityCluster*-based and *Farthest-First*-based methods[1] in order to demonstrate the effectiveness of the proposed approach. Default parameters of Weka have been only used in all baselines. For all presented baselines, note that, nodes of a co-visitation graph are divided into two clusters using each clustering-based method. Empirically, the cluster which has relatively few nodes is considered as suspicious nodes. In reality, the number of nodes which are concerned by attackers is significantly less than the number of authentic nodes due to the limited cost of attacks. In Table VII, we can observe that optimistic FARs of the proposed IMIA-HCRF compared with baselines can be obtained in different co-visitation injection attacks except for a few cases. It is noteworthy that all results of the presented approaches including DR and FAR are averaged. The other observation

[1]www.cs.waikato.ac.nz/ml/weka/

TABLE VII: Detection results of the proposed approach compared with baselines in different co-visitation injection attacks, where "#Node" denotes the number of nodes.

Attack	#Node	Metric	*Kmeans*	*Hierarchical*	*Density*	*FarthestFirst*	IMIA-HCRF
PH	1000	DR	0.9200	0.8793	0.9416	0.2573	1.0 •
		FAR	0.0978	0.0894	0.0863	0.0873	0.0789 •
	3000	DR	0.9756	0.8780	0.9512	0.1951	1.0 •
		FAR	0.2014	0.3504	0.2900	0.0098	0.0418 •
	5000	DR	1.0	1.0	1.0	0.3846	1.0 •
		FAR	0.5571	0.5571	0.5009	0.0066	0.0008 •
PM	1000	DR	0.5000	0.5000	0.5000	0	1.0 •
		FAR	0.0957	0.0937	0.0937	0.0494	0.0988 •
	3000	DR	1.0	1.0	1.0	0	1.0 •
		FAR	0.2996	0.0578	0.1807	0.0167	0.0476 •
	5000	DR	1.0	1.0	1.0	0	1.0 •
		FAR	0.5000	0.5000	0.2129	0.5833	0.0030 •
DH	1000	DR	0.8972	0.3804	0.6055	0.4523	1.0 •
		FAR	0.2379	0.0906	0.1709	0.0144	0.0907 •
	3000	DR	0.0652	0.0434	0.0652	0.0217	1.0 •
		FAR	0.2351	0.2338	0.2344	0	0.0270 •
	5000	DR	0.7142	0	0.7142	0.2857	1.0 •
		FAR	0.5298	0.1000	0.5156	0.0187	0.0034 •
DM	1000	DR	0.1011	0.5073	1.0	0.2857	1.0 •
		FAR	0.1888	0.1143	0.1878	0.0464	0.1064 •
	3000	DR	0	0.6666	1.0	0	1.0 •
		FAR	0.3044	0.0558	0.1926	0.0154	0.0539 •
	5000	DR	0	0.7894	0	0.7894	1.0 •
		FAR	0.2140	0.0235	0.2129	0.0311	0.0463 •

is that average false alarm rates of the proposed method are higher than the reverse cases when the number of nodes is small. This result may indicate that attribute representations of nodes are limited when the number of nodes is small.

Additionally, we also provide detection results for mixed co-visitation injection attacks in different cases as shown in Table VIII. We can observe that the proposed IMIA-HCRF shows a competitive detection performance with an advantage of 6% FAR compared with all baselines. This gain in the FAR is notable, due to the fact that: (1) different background knowledge in promotion or demotion attacks make a challenging issue for the evaluation of co-visitation injection behaviors, and (2) IMIA-HCRF generates integration and adaptability that better reflect the advantage of globally optimal segmentation compared with purely exploiting node attributes (TIA-UnaryPot) and link attributes (TIA-PairCRF).

3) Real-world Application: To discover interesting findings on real-world data, we implemented a series of experiments by exploiting different detection approaches as shown in Table IX. We can observe that the proposed IMIA-HCRF outperforms the baselines except for few cases. IMIA-HCRF brings an improvement with the advantage of 11.5% FAR in average. Note that, RDI-RG spots anomalous behaviors using removal disturbed information and recursive propagation [26], which is used to demonstrate the effectiveness of eliminating disturbed data. CBS exploits a label propagation mechanism and estimates the spam probability. Promising detection results can be obtained by CBS. Nevertheless, its FARs are relatively remarkable compared with IMIA-HCRF. This may be due to the fact that the disturbed data in its early stage partly affect the structure of the bipartite graph especially for promotion attacks. Comparatively, the detected results for demotion attacks are better than the results in promotion attacks. The reason may be that users for promotion attacks focus on a higher or the highest rating. This makes it is difficult to distinguish the behavior between genuine and malicious users facing with popular items. It also makes the weight of edge in the bipartite graph indistinguishable, and eventually leading to the uncertainty in the later stage of graph mining. Measuring link behaviors for promotion attacks is the difficulty of abnormality detection, which is definitely worth further investigation.

4) Discussions and Improvements: Based on the experimental results, several insights and improvements are worth discussing as follows:

First, we modeled a unified detection framework to deal with both profile injection attacks and co-visitation injection attacks, and also provided a unified interface for eliminating disturbed data. Note that, the elimination of disturbed data also provides a guarantee for the robustness of detection performance, especially for detecting small-scale attacks. More importantly, how to choose a reasonable threshold for the determination of suspected items is a crucial task. Although the difference between genuine items and target items is significant when the attack size is large, it is not easy to capture the concerned target items by purely exploiting a parameter threshold when the attack size is small. This limitation means that a few genuine items are mistakenly considered as target items, and finally leading to relatively high FARs. Exploring an adaptive and automatic mechanism for determining suspected items is worth further investigation.

Second, we developed a strategy for enhancing dense behaviors based on the remaining association behaviors, which is favorable to reduce the scope of abnormality detection. Furthermore, we extracted similar behaviors from hidden association space according to sparse matrix factorization. It is also effective to enhance the association of malicious behaviors as well as further deal with disturbed data. Actually, distinguishing malicious users (nodes) from some genuine users (mimicked by attackers) is the difficulty of detection. For the representation of rating and co-visitation behaviors, we analyzed a combination of traditional behavior representation (e.g., inherent attributes of node) and potential representations of hidden spatial behaviors (e.g., linkage behaviors and boundary segmentation of local behavior space). However, eliminating the disturbed data in the early stage may bring negative effects in terms of structured behavior representations.

Third, a few weak cases in the experimental results can be found, such as high FARs in Amazon data and TripAdvisor data, and high FARs in profile injection attacks when the attack size is small. This may be due to the fact that (1) the representation of rating and co-visitation behaviors faced with diverse data limits a performance boost in terms of generalization ability and discriminability; and (2) some authentic users (for profile injection attacks) or items (for co-visitation injection attacks) are mistaken as malicious nodes. In addition, further research is needed in terms of abnormality forensics according to original attributes of real data (e.g., ratings or reviews). Regarding the sensitive range of abnormal distribution (e.g., ratings or reviews change over time), how to incorporate multivariate time series analysis for rating behaviors toward multiple-factor forensics is also an open issue.

TABLE VIII: Detection results of the presented methods in mixed co-visitation injection attacks with diverse graph-scales, different attack intentions, and different background knowledge.

Mixed attacks	Metric	Benchmark methods					
		K-means-based	Density-based	DBScan-based	TIA-UnaryPot	TIA-PairCRF	IMIA-HCRF
PH (1000 nodes) + PM (1000 nodes)	DR	0.2000	0.3600	0.3500	1.0	1.0	1.0 •
	FAR	0.1497	0.1671	0.1022	0.2994	0.1917	0.0923 •
PH (2000 nodes) + PM (2000 nodes)	DR	0.1612	0.1612	0.1612	1.0	1.0	1.0 •
	FAR	0.0081	0.0081	0.0076	0.2056	0.1635	0.0096 •
PH (3000 nodes) + PM (3000 nodes)	DR	0.4807	0.4807	0.1612	1.0	1.0	1.0 •
	FAR	0.2177	0.2228	0.1134	0.5108	0.2242	0.0169 •
PH (4000 nodes) + PM (4000 nodes)	DR	0.6032	0.5912	0.2199	1.0	1.0	1.0 •
	FAR	0.0804	0.0872	0.0240	0.1814	0.1025	0.0120 •
PH (5000 nodes) + PM (5000 nodes)	DR	0.1578	0.1578	0.1578	1.0	1.0	1.0 •
	FAR	0.2407	0.2603	0.1362	0.5494	0.1539	0.0200 •
DH (1000 nodes) + DM (1000 nodes)	DR	0.4375	0.5625	0.4723	1.0	1.0	1.0 •
	FAR	0.1544	0.1666	0.2501	0.2063	0.1321	0.0995 •
DH (2000 nodes) + DM (2000 nodes)	DR	0.1222	0.1333	0.1103	1.0	1.0	1.0 •
	FAR	0.0178	0.0214	0.0323	0.1822	0.1774	0.0994 •
DH (3000 nodes) + DM (3000 nodes)	DR	0.2941	0.1764	0.2398	1.0	1.0	1.0 •
	FAR	0.2235	0.2329	0.2194	0.5124	0.1153	0.0316 •
DH (4000 nodes) + DM (4000 nodes)	DR	1.0	1.0	1.0	1.0	1.0	1.0 •
	FAR	0.0874	0.0834	0.1045	0.1884	0.1092	0.0180 •
DH (5000 nodes) + DM (5000 nodes)	DR	0.0952	0.1428	0.1106	1.0	1.0	1.0 •
	FAR	0.2382	0.2765	0.2358	0.3701	0.1570	0.0958 •

TABLE IX: Detection results of the presented benchmarks compared with the proposed approach on different real-world datasets, including promotion and demotion attack scenarios.

Dataset	Method	Promotion		Demotion	
		DR	FAR	DR	FAR
Amazon 2000	CBS	1.0	0.8017	1.0	0.1169
	RDI-RG	1.0	0.5044	1.0	0.1134
	Unary-Pot	0.7500	0.0955	0.8571	0.0547
	Pair-CRF	1.0	0.1123	1.0	0.0693
	IMIA-HCRF	1.0 •	0.0817 •	1.0 •	0.0577 •
Amazon 2007	CBS	1.0	0.6299	0.9259	0.0753
	RDI-RG	0.8235	0.4033	0.9629	0.0754
	Unary-Pot	0.9411	0.3258	1.0	0.1407
	Pair-CRF	1.0	0.1677	1.0	0.1023
	IMIA-HCRF	1.0 •	0.0932 •	1.0 •	0.0976 •
Amazon 2009	CBS	0.8571	0.4326	1.0	0.0748
	RDI-RG	1.0	0.4435	1.0	0.0818
	Unary-Pot	1.0	0.1003	1.0 •	0.0153 •
	Pair-CRF	1.0	0.1003	1.0 •	0.0153 •
	IMIA-HCRF	1.0 •	0.0927 •	1.0 •	0.0153 •
Amazon 2010	CBS	1.0	0.4514	0.8571	0.0466
	RDI-RG	1.0	0.1986	1.0	0.0633
	Unary-Pot	1.0	0.2164	1.0	0.0235
	Pair-CRF	1.0	0.2199	1.0	0.0235
	IMIA-HCRF	1.0 •	0.0709 •	1.0 •	0.0220 •
TripAdvisor	CBS	1.0	0.3153	1.0	0.0740
	RDI-RG	1.0	0.2939	1.0	0.0723
	Unary-Pot	1.0	0.8812	1.0	0.0640
	Pair-CRF	1.0	0.8950	1.0	0.0741
	IMIA-HCRF	1.0 •	0.0960 •	1.0 •	0 •
LibraryThing	CBS	1.0	0.3153	0.6235	0.0582
	RDI-RG	1.0	0.2939	1.0	0.6097
	Unary-Pot	0.6	0.2561	1.0	0.6648
	Pair-CRF	1.0	0.3292	1.0	0.7489
	IMIA-HCRF	1.0 •	0.0951 •	1.0 •	0.1053 •

VI. Conclusions and Future Work

This work presents a divide-and-conquer strategy to detect profile injection attacks and co-visitation injection attacks for online recommender systems. Experimental results on both synthetic data and real-world data show that the elimination of disturbed data, determination of dense behaviors, and potential segmentation exhibit considerable stability and discriminability among nodes (users or items) for detecting malicious injection behaviors. Nevertheless, it is still less than ideal for reaching the ultimate standard (DR of 100% and almost zero FAR) faced with different and mixed injection attacks. Therefore, further research is desired before we depend solely on the elimination of disturbed data and potential segmentation and representation as an online detection mechanism. One way of boosting the performance may be to construct a more robust behavior representation of nodes and links or develop a more effective algorithm with strong generalization ability to mitigate behavioral variability. The other way may be to establish a more efficient discrimination mechanism for dealing with dense behaviors.

Moreover, facing with new threats toward recommender systems, such as data poisoning attacks on factorization-based collaborative filtering [36], poisoning attacks to graph-based recommender systems [37], and adversarial attacks on an oblivious recommender [38], investigating an adaptable and selective detection framework to defend these threats is especially worth studying.

References

[1] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.

[2] X. Luo, M. Zhou, Y. Xia, Q. Zhu, A. Ammari, and A. Alabdulwahab, "Generating highly accurate predictions for missing qos-data via aggregating non-negative latent factor models," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 3, pp. 524–537, 2016.

[3] G. Yang, N. Gong, and Y. Cai, "Fake co-visitation injection attacks to recommender systems," *Network & Distributed System Security Symposium (NDSS)*, pp. 1–15, 2017.

[4] R. Burke, B. Mobasher, and C. Williams, "Classification features for attack detection in collaborative recommender systems," *International Conference on Knowledge Discovery and Data Mining*, pp. 17–20, 2006.

[5] N. Gunnemann, S. Günnemann, and C. Faloutsos, "Robust multivariate autoregression for anomaly detection in dynamic product ratings," *Proceedings of the 23rd international conference on World Wide Web*, pp. 361–372, 2014.

[6] S. Gunnemann, N. Günnemann, and C. Faloutsos, "Detecting anomalies in dynamic rating data: A robust probabilistic model for rating evolution," *In KDD'2014*, pp. 841–850, 2014.

[7] Z. Wu, J. Wu, J. Cao, and D. Tao, "HySAD: A semi-supervised hybrid shilling attack detector for trustworthy product recommendation," *In KDD'2012*, pp. 985–993, 2012.

[8] M. Fang, N. Gong, and J. Liu, "Influence function based data poisoning attacks to top-n recommender systems," *Proceedings of The Web Conference (WWW)*, pp. 3019–3025, 2020.

[9] I. Gunes, C. Kaleli, A. Bilge, and H. Polat, "Shilling attacks against recommender systems: A comprehensive survey," *Artificial Intelligence Review*, vol. 42, no. 4, pp. 1–33, 2012.

[10] Z. Yang, L. Xu, Z. Cai, and Z. Xu, "Re-scale AdaBoost for attack detection in collaborative filtering recommender systems," *Knowledge-Based Systems*, vol. 100, pp. 74–88, 2016.

[11] Z. Yang, Z. Cai, and X. Guan, "Estimating user behavior toward detecting anomalous ratings in rating systems," *Knowledge-Based Systems*, vol. 111, pp. 144–158, 2016.

[12] W. Zhou, Y. S. Koh, J. H. Wen, S. Burki, and G. Dobbie, "Detection of abnormal profiles on group attacks in recommender systems," *Proceedings of the 37th international ACM SIGIR conference on Research on development in information retrieval*, vol. 1, pp. 955–958, 2014.

[13] N. Gong, M. Frank, and P. Mittal, "Sybilbelief: A semi-supervised learning approach for structure-based sybil detection," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 6, pp. 976–987, 2014.

[14] B. Wang, L. Zhang, and N. Gong, "SybilSCAR: Sybil detection in online social networks via local rule based propagation," *In IEEE Conference on Computer Communications (INFOCOM)*, pp. 1–9, 2017.

[15] B. Wang, J. Jia, and N. Gong, "Graph-based security and privacy analytics via collective classification with joint weight learning and propagation," *In ISOC Network and Distributed System Security Symposium (NDSS)*, pp. 1–15, 2019.

[16] D. Yuan, Y. Miao, N. Gong, Z. Yang, Q. Li, D. Song, Q. Wang, and X. Liang, "Detecting fake accounts in online social networks at the time of registrations," *In ACM Conference on Computer and Communications Security (CCS)*, pp. 1423–1438, 2019.

[17] B. Wang, N. Gong, and H. Fu, "GANG: Detecting fraudulent users in online social networks via guilt-by-association on directed graphs," *IEEE International Conference on Data Mining*, pp. 465–474, 2017.

[18] C. E. Seminario and D. C. Wilson, "Attacking item-based recommender systems with power items," *ACM Conference on Recommender Systems*, pp. 57–64, 2014.

[19] I. Gunes and H. Polat, "Detecting shilling attacks in private environments," *Information Retrieval Journal*, vol. 19, no. 6, pp. 1–26, 2016.

[20] N. Hurley, Z. Cheng, and M. Zhang, "Statistical attack detection," *ACM conference on Recommender systems*, pp. 149–156, 2009.

[21] C. Seminario and D. Wilson, "Nuke'em till they go: investigating power user attacks to disparage items in collaborative recommenders," *ACM Conference on Recommender Systems (RecSys)*, pp. 293–296, 2015.

[22] P. Kohli, L. Ladicky, and P. Torr, "Graph cuts for minimizing robust higher order potentials," *In Proceedings of the International Conference Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, 2008.

[23] P. Kohli and P. Torr, "Robust higher order potentials for enforcing label consistency," *International Journal of Computer Vision*, vol. 82, no. 3, pp. 302–324, 2009.

[24] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1124–1137, 2004.

[25] B. Mobasher, R. Burke, R. Bhaumik, and C. Williams, "Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness," *ACM Transactions on Internet Technology (TOIT)*, vol. 7, no. 4, p. 38, 2007.

[26] Y. Zhang, Y. Tan, M. Zhang, Y. Liu, T. Chua, and S. Ma, "Catch the black sheep: Unified framework for shilling attack detection based on fraudulent action propagation," *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2408–2414, 2015.

[27] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in NIPS'2013*, pp. 3111–3119, 2013.

[28] J. Zhang, Y. Dong, Y. Wang, J. Tang, and M. Ding, "Prone: Fast and scalable network representation learning," *In Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 1–7, 2019.

[29] K. Alahari, P. Kohli, and P. Torr, "Dynamic hybrid algorithms for map inference in discrete mrfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 10, pp. 1846–1857, 2009.

[30] M. Collins, R. Schapire, and Y. Singer, "Logistic regression, adaboost and bregman distances," *Machine Learning*, vol. 48, no. 253, pp. 253–285, 2002.

[31] L. Du, P. Zhou, L. Shi, H. Wang, M. Fan, W. Wang, and Y. D. Shen, "Robust multiple kernel k-means clustering using l21-norm," *the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 3476–3482, 2015.

[32] J. McAuley and J. Leskovec, "Hidden factors and hidden topics: understanding rating dimensions with review text," *ACM Conference on Recommender Systems (RecSys)*, pp. 165–172, 2013.

[33] H. Wu, R. Luk, K. Wong, and K. Kwok, "Interpreting tf-idf term weights as making relevance decisions," *ACM Transactions on Information Systems*, vol. 26, no. 3, pp. 1–37, 2008.

[34] R. Xi, N. Lin, and Y. Chen, "Compression and aggregation for logistic regression analysis in data cubes," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 4, pp. 479–492, 2008.

[35] C. Chung, P. Hsu, and S. Huang, "$\beta\mathcal{P}$: A novel approach to filter out malicious rating profiles from recommender systems," *Decision Support Systems*, vol. 55, no. 1, pp. 314–325, 2013.

[36] B. Li, Y. Wang, A. Singh, and Y. Vorobeychik, "Data poisoning attacks on factorization-based collaborative filtering," *29th Conference on Neural Information Processing Systems (NIPS)*, pp. 1–13, 2016.

[37] M. Fang, G. Yang, N. Gong, and J. Liu, "Poisoning attacks to graph-based recommender systems," *ACSAC, arXiv preprint arXiv:1809.04127*, 2018.

[38] K. Christakopoulou and A. Banerjee, "Adversarial attacks on an oblivious recommender," *Proceedings of the 13th ACM Conference on Recommender Systems*, pp. 322–330, 2019.

Zhihai Yang received the Ph.D. degree in Control Science and Engineering from Xi'an Jiaotong University, in 2016. He is currently with the School of Computer Science and Engineering, Xi'an University of Technology, Xi'an, China. His research interests include information security, recommender system, and data mining.

Qindong Sun received his Ph.D. degree in School of Electronic and Information Engineering from the Xi'an Jiaotong University, China. He is currently a professor at the Department of Computer Science and Engineering of Xi'an University of Technology. His research interests include network information security, online social networks and internet of things.

Yaling Zhang received the B.S. degree in computer science in 1988 from Northwest University, Xi'an, China. She received the B.S. degree in computer science in 2001, and earned the Ph.D. degree in mechanism electron engineering in 2008, both from the Xi'an University of Technology, Xi'an, China. She is currently a professor in Xi'an University of Technology. Hers current research interests include cryptography and differential privacy protection in data mining.

Wei Wang received the M.S. degree in computer software and theory from Xi'an Jiaotong University, in 2003. He is currently with the school of computer science and engineering, Xi'an University of Technology, Xi'an, China. His research interests are data mining and machine learning.