Detect Professional Malicious User With Metric Learning in Recommender Systems

Yuanbo Xu, Yongjian Yang, En Wang, Fuzhen Zhuang, and Hui Xiong, *Fellow, IEEE*

Abstract—In e-commerce, online retailers are usually suffering from professional malicious users (PMUs), who utilize negative reviews and low ratings to their consumed products on purpose to threaten the retailers for illegal profits. PMUs are difficult to be detected because they utilize masking strategies to disguise themselves as normal users. Specifically, there are three challenges for PMU detection: 1) professional malicious users do not conduct any abnormal or illegal interactions (they never concurrently leave too many negative reviews and low ratings at the same time), and they conduct masking strategies to disguise themselves. Therefore, conventional outlier detection methods are confused by their masking strategies. 2) the PMU detection model should take both ratings and reviews into consideration, which makes PMU detection a multi-modal problem. 3) there are no datasets with labels for professional malicious users in public, which makes PMU detection an unsupervised learning problem. To this end, we propose an unsupervised multi-modal learning model: MMD, which employs Metric learning for professional Malicious users Detection with both ratings and reviews. MMD first utilizes a modified RNN to project the informational review into a sentiment score, which jointly considers the ratings and reviews. Then professional malicious user profiling (MUP) is proposed to catch the sentiment gap between sentiment scores and ratings. MUP filters the users and builds a candidate PMU set. We apply a metric learning-based clustering to learn a proper metric matrix for PMU detection. Finally, we can utilize this metric and labeled users to detect PMUs. Specifically, we apply the attention mechanism in metric learning to improve the model's performance. The extensive experiments in four datasets demonstrate that our proposed method can solve this unsupervised detection problem. Moreover, the performance of the state-of-the-art recommender models is enhanced by taking MMD as a preprocessing stage.

Index Terms—Professional malicious users, unsupervised learning, metric learning, recommender system

1 Introduction

E-commerce giants, such as Amazon, Jingdong, and Alibaba, have been thriving with the development of Internet technology, where millions of electronic retailers produce great wealth through selling commodities on the websites [34]. For each day, billions of trades occur between retailers and consumers [27]. For the sake of improving the consumers' experience of online shopping, e-commerce websites usually allow consumers (we call them "users") to leave reviews and rank ratings on the commodities (we call them "items"). To trade off the interests between retailers and users, e-commerce websites punish the retailers who receive a high percentage of negative reviews and low ratings from users [2]. Being widely applied in almost all kinds of e-commerce websites, this feedback mechanism has been proved to be effective if all the users leave truthful and objective reviews or ratings.

However, in practice, there exist some malicious users (MU), who leverage this feedback mechanism to gain illegal profits [3], [29]. For example, these malicious users first purposefully leave negative reviews and low ratings of their consumed products without any consideration of the commodities' quality. Then they blackmail the electronic retailers to make illegal profits; otherwise, they would leave more negative feedbacks, cheating e-commerce websites to punish the electronic retailers and confuse the normal users about the items in recommendations. As a result, these malicious users undermine the fairness of e-commerce. Moreover, their negative feedbacks will confuse the recommender systems (collaborative filtering-based models [12] or content-based models [30]), leading to a chaotic recommendation for normal users, which is also named as shilling attacks [47], [55].

To address the above issues, e-commerce companies usually employ statistic outlier detection or shilling attack detection models [20], [24], [33] to detect MUs, i.e., finding objective users who always give negative reviews or low ratings. However, there are some restrictions for these detection models: first, these models only tackle this problem from a methodological perspective and ignore the real-world scenarios. For example, most detection models ignore that there are some professional malicious users (PMUs), who can utilize masking strategies to avoid detection; second, they usually focus on filtering either fake ratings to improve recommendation models, or negative reviews for

- *Yuanbo Xu, Yongjian Yang, and En Wang are with the Department of Computer Science and Technology, and Key Laboratory of Symbolic Computation and Knowledge Engineering, Ministry of Education, Jilin University, Changchun 130012, China. E-mail: {yuanbox, yyj, wangen}@jlu.edu.cn.*
- *Fuzhen Zhuang is with the Key Lab of Intelligent Information Processing, Chinese Academy of Sciences (CAS), Beijing 100190, China, and also with the University of Chinese Academy of Sciences, Beijing 100049, China. E-mail: zhuangfuzhen@ict.ac.cn.*
- *Hui Xiong is with the School of Business, Rutgers, the State University of New Jersey, New Brunswick, NJ 08901-8554 USA. E-mail: hxiong@rutgers.edu.*

Manuscript received 10 January 2020; revised 14 November 2020; accepted 20 November 2020. Date of publication 25 November 2020; date of current version 5 August 2022.
(Corresponding author: En Wang.)
Recommended for acceptance by G. Koutrika.
Digital Object Identifier no. 10.1109/TKDE.2020.3040618

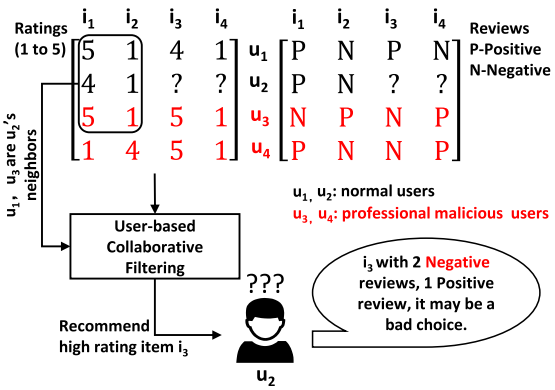


Fig. 1. An example to indicate how professional malicious users confuse normal users and undermine the fairness of online e-commerce. In this example, u_3 and u_4 are professional malicious users, who give a high rating (5) for i_3, while also a negative review (N) for i_3. Hence, normal user u_2 is confused about recommendation i_3 by the fake negative feedbacks of professional malicious users u_3, u_4. More importantly, u_3's distribution of ratings is the same as that of the normal user u_1, so traditional statistic outlier detections cannot detect this kind of professional malicious users for recommender systems.

content-based models, which do not take both ratings and reviews into account. As a result, these models may be applied in limited application scenarios in recommender systems, but not proper for professional malicious user detections.

Different from malicious users, professional malicious users (PMU) typically adopt the following two masking strategies to avoid existing detections: 1) To avoid giving too many low ratings, they provide a high rating but a negative review. In this way, they can mislead the potential consumer who is browsing this review to decide whether to buy this item. 2) To avoid giving too many negative reviews, they provide a low rating but a positive review. In this way, they can explain to the outlier detection that their interactions are "misoperations". By applying the above two strategies, alternately, professional malicious users can disguise themselves as normal users. As shown in Fig. 1, we give an example to indicate how professional malicious users confuse potential consumers and undermine the fairness of online e-commerce. In this paper, we focus on how to detect these PMUs with masking strategies in real-world scenarios by simultaneously analyzing their ratings and reviews.

To detect PMUs, there are three significant challenges in recommender systems: 1) PMUs adopt masking strategies to act like normal users, which is difficult to be detected. 2) Detecting PMUs needs to analyze both ratings and reviews, which makes it a multi-modal problem. 3) Existing public datasets do not contain the PMU label, which makes this detection an unsupervised learning problem. To this end, we propose an unsupervised multi-modal learning model: *MMD*, which applies metric learning [31], [38], [56] for professional malicious user detection with both ratings and reviews. The key to metric learning is utilizing different metrics (euclidean distance or other metrics) to represent the relationships between entities [21], [51]. MMD first utilizes Hierarchical Dual-Attention RNN (HDAN) [49] to do user profiling with reviews and ratings. By catching the sentiment gap between reviews and ratings, we build a candidate PMU set. Then we apply an unsupervised metric learning-based clustering method to this candidate set to label professional malicious users. To be specific, we apply the attention mechanism in metric learning to enhance the model. We conduct experiments on four real-world datasets: Amazon, Yelp, Taobao, and Jingdong. The results demonstrate that our proposed method can solve this unsupervised malicious user detection problem. Moreover, their performance of the state-of-the-art recommender models can be enhanced by taking MMD as a preprocessing stage.

We summarize the main contributions as follows.

- This is the first work focusing on solving the professional malicious user detection issue utilizing both users' ratings and reviews to enhance the state-of-the-art recommender systems.
- A novel multi-modal unsupervised method-MMD-is proposed to detect professional malicious users with the modified RNN and attention metric learning-based clustering.
- Extensive experiments are conducted on four real-world e-commerce datasets to verify our proposed method. Moreover, by filtering professional malicious users, some state-of-the-art models are enhanced.

The remainder of the paper is organized as follows. We first provide some preliminaries in Section 2 and then elaborate on our proposed method in Section 3. We present and discuss experimental results in Section 4 and review related work in Section 5. Finally, we conclude this paper and discuss future directions in Section 6.

2 Preliminaries

This section provides motivations, basic definitions, and background to professional malicious user detection.

2.1 Motivations

The professional malicious user (PMU) is defined by two motivations: 1) the e-commerce websites judge the credit of retailers according to the good rating/review rate among all the ratings/reviews. So the professional malicious users utilize untruthful negative reviews and low ratings to threaten retailers for illegal profits. Meanwhile, if a user always gives a high proportion of negative reviews or low ratings groundlessly, the websites will treat the user as a malicious user with traditional outlier detection and punish him/her. However, PMU can use the masking strategy to control the proportion of negative reviews/low ratings and avoid the traditional detection of websites, which makes it a challenge to detect PMUs. 2) PMUs give fake ratings or reviews, which are difficult to distinguish from truthful ratings and reviews because PMUs utilize masking strategies to disguise themselves as normal users. This makes the existing recommendation models inaccurate and inefficient and leads to a bad recommendation. If we detect PMUs and filter the fake ratings and reviews, the performance of recommendation models should be improved.

2.2 Basic Definitions

In a recommender system, let U denote a set of m users $U = \{u_1, u_2 \dots u_m\}$, and I denote a set of n items $I = \{i_1, i_2 \dots i_n\}$. r_{ui} means the rating user u marked for item i,

as the entry of user-item matrix $R_{m\times n}$. We build a review user-item matrix $V_{n\times m}$ with v_{ui} in the same way. For each user u and item i, p_u and q_i denote their latent vector learned by embedding models.

In this paper, we focus on detecting professional malicious users who utilize masking strategies:

Definition 1 (Professional Malicious Users (PMUs)). *Malicious users who give fake ratings and negative reviews to make illegal profits and utilize masking strategies to avoid detections.*

PMUs usually give fake ratings and negative reviews, alternatively, and keep them in a "safe" proportion to avoid being detected

$$\left|R_u^{\text{fa}}\right|/|R_u| \le \theta^{\text{fa}}; \left|V_u^{\text{ne}}\right|/|V_u| \le \theta^{\text{ne}}, u \in U, \tag{1}$$

where R_u^{fa} and V_u^{ne} denote the fake ratings and negative reviews sets; R_u and V_u denote the whole ratings and reviews set of user u, and θ^{ne} and θ^{fa} denote the thresholds for detection models.

In order to maximize their profits by avoiding detections, PMUs use masking strategies, which means they do not give v^{ne} and r^{fa} for an item i at the same time. Instead, they usually give high ratings with negative reviews or fake ratings with positive reviews

$$\frac{\left|(r_{ui} \in R_u^{\text{fa}}, v_{ui} \notin V_u^{\text{ne}})||(r_{ui} \notin R_u^{\text{fa}}, v_{ui} \in V_u^{\text{ne}})\right|}{|R_u \cup V_u|} \ge \theta^{\text{mu}}, \tag{2}$$

where θ^{mu} is the threshold for PMUs. Finally, we formulate professional malicious user detection as follows:

Definition 2 (Professional Malicious User Detection). *Given user set U, item set I, rating set R, and review set V as inputs, the object of professional malicious user detection is to filter the users with restrictions above, and output the PMU set U^{mu}:*

$$U^{\text{mu}} = \text{Detect}(U, I, R, V); \quad \textbf{s.t.} \quad \text{Eq}(1), \quad \text{Eq}(2). \tag{3}$$

2.3 Hierarchical Dual-Attention RNN

To utilize the review for professional malicious user detections, we employ a state-of-the-art RNN model, Hierarchical Dual-Attention RNN (HDAN) [48], [50], to project the review into a sentiment score. The structure of HDAN is shown in Fig. 2. HDAN calculates update gate ug_t, reset gate re_t and temporary state $\widetilde{h}_{t-1}$ Eqs. (4), (5), (6):

$$ug_t = \sigma(W_{ug}\widehat{y} + U_{ug}h_{t-1} + b_{ug}), \tag{4}$$

$$\widetilde{h}_{t-1} = \tanh(W_h\widehat{y} + re_t \odot (U_h h_{t-1}) + b_h), \tag{5}$$

$$re_t = \sigma(W_{re}\widehat{y} + U_{re}h_{t-1} + b_{re}), \tag{6}$$

where $\widehat{y}$=(y_{t-1}, y_t, y_{t+1}) replaces y_t in HAN for catching the sentiment in former y_{t-1} and future state y_{t+1}. Finally, HDAN updates the information as follows:

$$h_t = (1 - ug_t) \odot h_{t-1} + ug_t \odot \widetilde{h}_{t-1}. \tag{7}$$

Moreover, HDAN utilizes attention mechanisms to compute different weights for each word, as word-attention, and different weights for each sentence, as sentence-attention. In Fig. 2, it is evident that the word "*Poorer*" should take a more critical role than other words for this review. The details of attention computing are also discussed in [48].

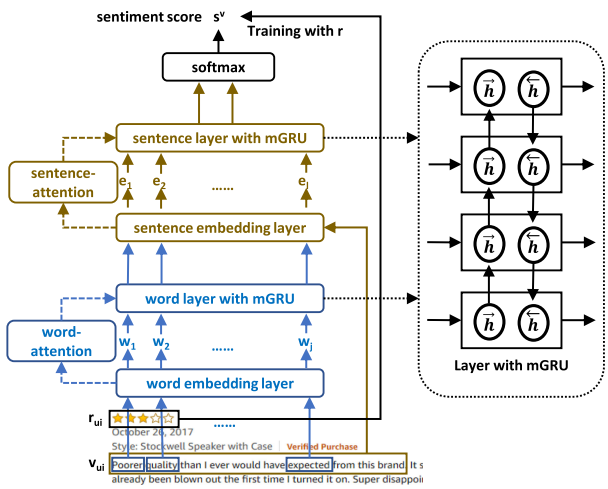


Fig. 2. An example to project a review into a sentiment score. For each review v_{ui}, HDAN inputs word embedding w and sentence embedding e and ouputs a sentiment score s^v.

In this paper, we take HDAN as a building block, which projects the review v into a sentiment score s^v. To train HDAN, we use the ratings as the ground truth, and minimize the loss L_{v-r} as follows:

$$\text{L}_{v-r} = \frac{1}{2}\sum_{U,I,V,R}(r_{ui} - s_{ui}^v)^2. \tag{8}$$

2.4 Metric Learning for clustering

The key to this detection is how to distinguish professional malicious users from normal users. The intuitive idea is to maximize the distance between malicious and normal users with clustering. In this paper, we utilize Metric Learning, a popular theory that is widely applied in clustering, embedding, and image recognition. The paradigm of metric learning is to estimate a proper "distance" between entities to measure the relationships [46]. Consider learning a distance metric matrix A as follows:

$$d_{\text{A}}(p_j, p_k) = \left\|p_j - p_k\right\|_{\text{A}} = \sqrt{(p_j - p_k)^T c^2 A (p_j - p_k)}, \tag{9}$$

where $j, k \in U$, p_j and p_k are latent vectors for j, k, and c is a regular parameter. Note that matrix A is a metric (it satisfies non-negativity and the triangle inequality in latent space) when $A \succeq 0$. To do clustering, metric learning attempts to learn a metric in which the different axes are given different "weights".

A simple idea of defining a criterion for a desired metric is to minimize the distance $d_{\text{A}}(j, k)$ if j, k are in the same user subgroup S (which means j, k should be closer under the metric matrix A), and add some constraint to ensure A does not force the user set into a point as a "metric". This gives an optimization problem

$$\min_A \sum_{j,k\in S} d_{\text{A}}(p_j, p_k); \tag{10}$$

TABLE 1
Notation List

Notation	Description
U	user set
I	item set
R, V	rating/review set
U^{mu}	PMU set
U^{cmu}	candidate PMU set
m, n	number of users/items
r_{ui}	u's rating on item i
v_{ui}	u's review on item i
p_u, q_i	user u/item i's latent vectors
s^r	rating score ($p_u \bullet q_i$)
s^v	review's sentiment score
g_{ui}	sentiment gap between s^r_{ui} and s^v_{ui}
k	the clustering number
A_{L}	distance metric matrix
α^{g}	sentiment gap threshold
θ^{mu}	detection threshold

$$\text{s.t.} \sum_{j,k \in (U-S)} d_{\text{A}}(p_j, p_k) \geq c, A \succ 0. \tag{11}$$

The method of solving this optimization is given in [46], which is very clear to follow. However, for professional malicious user detection, there are two limitations for applying metric learning directly: 1) there is no dataset with PMU labels, which means that we cannot build the user subgroup S. And 2) p_j and p_k only contain the side information of user j, k, without considering the interactions (for example, masking strategies of PMUs) in recommender systems. To the best of our knowledge, MMD is the first model that combines HDAN and metric learning for PMU detection with reviews and ratings in recommender systems. Some important notations are shown in Table 1:

3 Professional Malicious User Detection

In this section, we first present the professional malicious user profiling model (MUP), followed by the attention metric learning for clustering, MLC. Lastly, we analyze the time complexity of MMD.

3.1 Framework

To tackle the professional malicious user detection, we propose MMD, an unsupervised learning model, which applies metric learning and deep learning with both reviews and ratings. The framework of MMD is shown in Fig. 3.

At the beginning, MMD utilizes one-hot encoding to select user u and item i, then projects them into p_u and q_i. Latent factor model (LFM), which is the most widely used model in recommender area [14], is employed to get p_u and q_i, and calculates rating score s^r with them. Meanwhile, we employ HDAN to project the review v_{ui} to a sentiment score s^v. Note that we use ratings r_{ui} as training ground truth for LFM and HDAN. Then we feed s^v and s^r into the professional malicious user profiling model (MUP). This model outputs the sentiment gap vector g_{ui} and labels professional malicious users to build a candidate set U^{cmu}. We combine g_{ui} and p_u to build a profile vector z_u and utilize U^{cmu} as ground truth to learn a proper metric matrix A_{L} for professional malicious user detection. Especially, we apply attention to metric learning to enhance the model. Finally, we choose U^{cmu} as cluster centroids, A_{L} as the clustering metric to cluster the users, which achieves the professional malicious users U^{mu}. The details of MMD are introduced in the following subsections.

3.2 Professional Malicious User Profiling (MUP) Model

To profile professional malicious users, we need to analyze their masking strategies. Unlike normal users, professional malicious users always use the following two interactions as masking strategies: 1) giving a high rating r_{ui} with a negative review v^{ne}_{ui}; 2) giving a positive review v_{ui} with a fake rating r^{fa}_{ui}. By utilizing the masking strategies, professional malicious users can avoid statistic outlier detection with thresholds θ^r, θ^v, and confuse the recommender system.

Taking a deep insight, we notice that there always exist sentiment gaps between each PMU's ratings and reviews, which are the most remarkable differences from a normal user's actions. So we first utilize HDAN to project review v onto a sentiment score s^v_{ui} (Eq. (13)). Meanwhile we embed users and items onto latent space P, Q with basic LFM (Eq. (12)), and calculate a rating score s^r_{ui} with p_u, q_i (Eq. (14)).

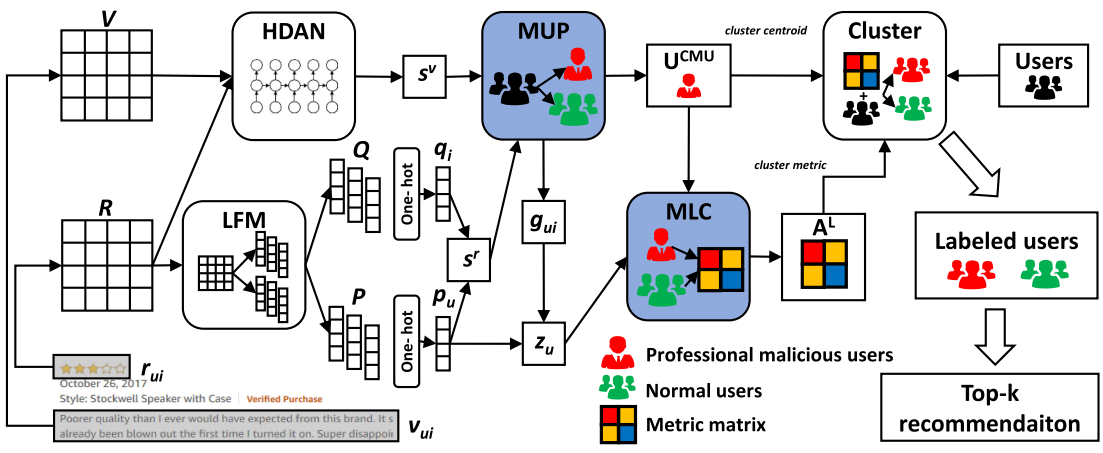


Fig. 3. An illustration of the professional malicious user detection model (MMD). MMD comprises two important modules: professional malicious user profiling (MUP), attention metric learning (MLC)), which are encompassed by grey rectangles.

Specifically, we minimize $\mathrm{L_{LFM}}$ to achieve latent vetor p_u and q_i with rating r_{ui} as input

$$\mathrm{L_{LFM}} = \sum_{u \in U, i \in I} \left(p_u{}^{\mathrm{T}} q_i - r_{ui}\right)^2. \tag{12}$$

$$s^v_{ui} = \mathrm{HDAN}(v_{ui}). \tag{13}$$

$$s^r_{ui} = p_u \odot q_i. \tag{14}$$

Note that s^v_{ui} is in the same range of s^r_{ui}, which is 1 to 5. Hence, we utilize the sentiment gap g_u to profile the users. Specifically, for each user-item pair (u, i), we calculate sentiment gap g_{ui}

$$g_{ui} = |s^r_{ui} - s^v_{ui}|_{\mathrm{abs}}, \tag{15}$$

where $|a|_{\mathrm{abs}}$ means the absolute value of a. For u as a normal user, the gap g_{ui} should be small (threshold α^{g}) because no matter how reviews and ratings are given, they all contain users' real sentiment for items. In comparison, for u as a professional malicious user, the gap should be huge for most items (threshold θ^{mu}) because of their masking strategies. We build gap vector $\hat{g_u}$ for each user u who has k items with feedbacks, which entries are g_{uk}

$$\hat{g_u} = \{g_{u1}, g_{u2} \ldots\ldots, g_{u(k-1)}, g_{uk}\}, k \in I. \tag{16}$$

To this end, we profile professional malicious user with the following practicable rule, label them and build a candidate professional malicious user set U^{cmu}:

$$u \in U^{\mathrm{cmu}}; \tag{17}$$

$$\mathbf{s.t.} \quad \frac{|\{g_{ui} | g_{ui} \geq \alpha^{\mathrm{g}}\}|}{|\hat{g_u}|} \geq \theta^{\mathrm{mu}}; g_{ui} \in \hat{g_u}, u \in U, i \in I. \tag{18}$$

With MUP, we can label professional malicious users. However, because the amount of professional malicious users only takes small portions of users, it is still challenging to use these labeled professional malicious users directly to learn more information, discover the insight, and improve the recommendation. To tackle this issue, we treat all these users in U^{cmu} as similarities, which means they can be categorized into the same cluster in a specific latent metric space. In this latent metric space, we can learn more about professional malicious users and normal users. In the next section, we will introduce how our method, attention metric learning (MLC), learns this specific latent metric space.

3.3 Attention Metric Learning for Clustering (MLC)

3.3.1 Model Construction

To learn a specific metric space, we do clustering to find the inner connections between professional malicious users. Towards a comprehensive understanding, we consider that professional malicious users are different from normal users in two perspectives: first, they have different attributes. Professional malicious users are signed up for their particular purpose, which is different from normal users. Second, there is a noticeable sentiment gap between ratings and reviews for professional malicious users, while for normal users, the sentiment gap is small. Without loss of generality, we combine users' p-dimension latent vector p_u and k-dimension gap vector $\hat{g_u}$ to construct the (p+k)-dimension profile vector z_u, which contains the information of users' attributes and sentiment gaps. Then we apply metric learning to learn the proper latent metric with U^{cmu}

$$z_u = p_u \oplus \hat{g_u}, \tag{19}$$

where $\oplus$ means a direct combination. With this representation vector, we first rewrite Eq. (9) as follows:

$$d_{\mathrm{A}}(z_j, z_k) = \left\|z_j - z_k\right\|_{\mathrm{A}} = \sqrt{(z_j - z_k)^T c^2 A (z_j - z_k)}, \tag{20}$$

where $j, k \in U$. If we utilize Eq. (20) directly, some critical information may be ignored, which leads to learning inaccurate metric. In real-world recommender systems, to cluster different users, the different attribute should take different importance and arrange different weights, which is the theory of attention mechanism [35]. The intuitive idea is that the attributes and sentiment gaps should make different contributions to metric learning. With this restriction, we can add the attention vector t into Eq. (20), using $t \otimes z$ to replace z. Note that $\otimes$ means element-wise product.

There are various ways to define attention vectors. Specifically, in our situation, it is a local optimization issue, where the general attention vector can achieve a proper performance without huge additional computing cost [23]. Hence, we utilize the general attention style to compute the attention vector t as follows:

$$\mathrm{f}(z_u, A) = z_u^T W_t A, \tag{21}$$

$$t_u = \mathrm{align}(z_u, A) = \frac{\exp(f(z_u, A))}{\sum_{\mathrm{A}} \exp(f(z_u, A))}, \tag{22}$$

where W_t is the general weights for attention t. Note that we utilize the attention vector to build a bridge between profile vector z_u and metric A. Moreover, we take a deep insight into the form of A. A is a (p+k) square metric matrix, where each entry of A stands for a weight of the metric in this dimension. In our proposed model, we restrict the metric for attributes p_u to be euclidean distance, which means the entries of A should be initialized as follows:

$$a_{i,j} = \begin{cases} 0, i \neq j; \\ 1, i = j. \end{cases} i, j \in [1, p]. \tag{23}$$

While for the sentiment gap $\hat{g}_u$, we initialize the metric weight entries in A with standard normal distribution $N(0, 1)$. Note that our original metric matrix A is partly diagonal, it can be learned quickly in the first p dimension, which is similar to [13]. Moreover, to measure the relationship between users' attributes and sentiment gap, we also learn two p×k matrices, which locate at the up-right and bottom-left of metric A, respectively.

To learn this metric matrix A_{L}, we need to build an objective function as Eqs. (10) and (11). To simplify the objective function, we jointly learn the metric A_{L} and attention vector t at the same time, with the following loss function:

$$\mathrm{L_{MLC}} = \sum_{\substack{j,k \in U^{\mathrm{cmu}} \\ j,k' \notin U^{\mathrm{cmu}}}}^{U,I,R,V} \big(\lambda d_{\mathrm{A}}(z_j, z_k) - (1-\lambda) d_{\mathrm{A}}(z_j, z_{k'}) + c\big), \tag{24}$$

where the c is the parameter for normalization, which is the same as Eq. (10). This function borrows the idea of BPR [25], which maximizes the distance between different clusters $(-(1-\lambda)d_{\mathrm{A}}(z_j, z_{k'}), j, k' \notin U^{\mathrm{cmu}})$ and minimizes the same cluster pairs $(\lambda d_{\mathrm{A}}(z_j, z_k), j, k \in U^{\mathrm{cmu}})$. λ is a parameter to tune the importance of different samples.

Without loss of generality, this loss function can be applied to learn different metrics. If we set A to be a diagonal matrix, $a_{j,k}$=1, if $j = k$ while $a_{j,k}$=0 if not. This loss function fades to a euclidean distance learning without attention vector. By arranging different forms of metric A, we can get the inner sight of the differences between professional malicious users and normal users.

3.3.2 Model Optimization

To make MLC less sensitive to the negative sampling, we also consider the restriction to the parameters in addition to minimizing the objective loss function $\mathrm{L_{MLC}}$. Let Θ be the model parameters, which includes metric matrix A and attention vector set t. Hence, we define the optimization objective for MLC as

$$\Theta^* = \arg\min_{\Theta} \mathrm{L_{MLC}} + \|\Theta\|_2, \tag{25}$$

where $\|\Theta\|_2$ is the regularization with F2-norm. In this formulation, we jointly learn metric matrix A and attention vector set t at the same time. Note that in real-world, PMU takes only small portion of users (nearly 10 percent). We need to balance the weights of labeled (PMUs) and unlabeled users (normal users), which means that λ should be larger than 0.5.

3.3.3 Learning Algorithm

This step updates model parameters by minimizing Eq. (24). By utilizing this objective function, metric A and attention weight W_t (Eq. (21)) are learned at the same time. It is a typical conventional minimization problem and can be approached with gradient descent. Specifically, we perform a gradient step for each involved parameter

$$\Theta = \Theta - \eta \frac{\partial \mathrm{L_{MLC}}}{\partial \Theta}, \tag{26}$$

where Θ=$\{A, W_t\}$. η denotes the learning rate, which is parameter-dependent if some auto-adaptive SGD models are applied. In our proposed model, we set Adagrad [6] as our SGD method. We can 1) sample the labeled professional malicious users repeatedly to build more samples (for each labeled professional malicious user, select 5 to 10 times unlabeled normal users to learn the metric); 2) enhance the importance of labeled users (a large λ). To validate the performance, we can monitor the return on a holdout validation dataset (which is the original data in Fig. 3). In this way, we can achieve a learned metric matrix A_{L}.

3.3.4 Detect Professional Malicious Users With Metric A_L

After we learn a metric A_{L}, we do simple k-means clustering (actually, it is a 2-means clustering, which puts users into normal or professional malicious user set) to detect professional malicious users in original data. Specifically, we take all the professional malicious users in U^{cmu} into the original data and do clustering in A_{L} latent data space, and label all the users in U. This k-means model can achieve convergence rapidly because we give some labeled users as heuristic information. With this step, we can get the professional malicious user set U^{mu}, which is the cluster with more labeled professional malicious users. Lastly, we conclude the processing of MMD in Algorithm 1.

Algorithm 1. Attention Metric Learning for Professional Malicious User Detection(MMD)

Input: Users U, items I, ratings R, reviews V, sentiment gap α^{g}, detection threshold θ^{mu}, learning rate η, hyper-parameter O for HDAN, MUP and MLC.
Output: Professional malicious users U^{mu}, distance metric A_{L}.

Initialize O, distance metric A with Eq. (23);
Professional Malicious User Profiling:(line 2-6)
    Calculate s^v with HDAN (Eq. (13));
    Calculate s^r with LFM (Eqs. (14) and (12));
    Calculate $\hat{g}_u$;
    Label the candidate professional malicious user U^{cmu} with $\alpha^{\mathrm{g}}, \theta^{\mathrm{mu}}$;
MLC:(line 7-17)
    Build z^u with Eq. (19);
while *not converge* **do**
  Randomly draw an example (j, k, k') from U;
  Calculate attention vector t_u with Eqs. (21) and (22);
  Calculate $\mathrm{L_{MLC}}$ with Eq. (24);
  Update A, W_t:
    $A \leftarrow A - \eta \frac{\partial \mathrm{L_{MLC}}}{\partial A}$;
    $W_t \leftarrow W_t - \eta \frac{\partial \mathrm{L_{MLC}}}{\partial W_t}$;
end
  Return A_{L};
K-means Clustering with U^{cmu}, A_{L};
Label users in U as U^{mu};
Return U^{mu}.

3.4 Time Complexity Analysis

In MMD, there are three sub-modules: MUP, MLC and K-means. Note that these models are employed sequentially in MMD, so the time complexity of MMD should be: $\mathrm{O_{MMD}}$=$\mathrm{O_{Kmeans}}$+$\mathrm{O_{MUP}}$+$\mathrm{O_{MLC}}$. For k-means, the time complexity is $\mathrm{O}(n \times k \times It) \approx \mathrm{O}(n \log n)$, where n is the data scale, k is 2 in MMD for k-means, and It is the iteration times. For MUP, which consists of HDAN and LFM, the time complexity is: O_{MUP}=O_{HDAN}+O_{LFM}. However, the output of LFM is fixed in our model, which means it could be pretrained as preprocessing. So $\mathrm{O_{MUP}} \approx \mathrm{O_{HDAN}} = \mathrm{O}(nd^2)$, where d is the dimensions of input vectors. For MLC, let $\mathrm{O_{MLC}}$ denote the time of learning A. Note that with different form settings of A, the time complexity is different. If we set A as a diagonal matrix, $\mathrm{O_{MLC}}$ should be

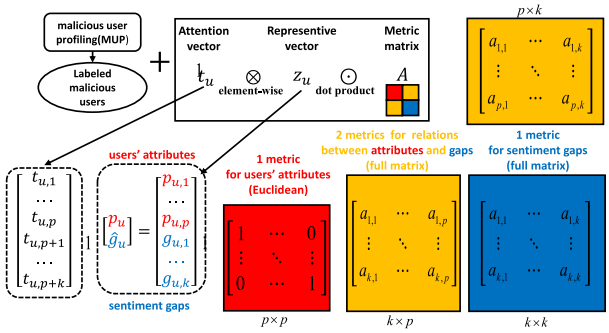


Fig. 4. An illustration of attention metric learning for clustering.

$O((k+p)log(k+p))$, where A is a $k+p$ dimension matrix. If we set A as a full matrix, it adds to $O((k+p)^2)$. Since we define the form of A as Fig. 4, O_{MLC} is $O(p \log p + k^2)$.

Because of the sequential process of MMD, the whole time complexity should be

$$O_{MMD} = O(n \log n) + O(nd^2) + O(p \log p + k^2). \quad (27)$$

Utilizing parallel processing or other computing frameworks may accelerate our model, where we leave as an important future work.

4 Experiments

In this section, we conduct extensive experiments to answer the following issues:

RQ1. Can our proposed method MMD outperform the state-of-the-art malicious user detection models?

RQ2. How is the effect of metric learning when it is applied in the detection, and can it help to relieve the lacking of labeled professional malicious users?

RQ3. How do the key parameters, such as α, θ, affect the detection performance?

RQ4. How does MMD benefit the recommender system with the detected professional malicious users?

4.1 Experimental Settings

4.1.1 Data Descriptions

We conduct abundant experiments on Amazon.com dataset[1] and Yelp for RecSys.[2] Amazon and Yelp datasets are two public datasets with abundant textual reviews and ratings. Moreover, we also collect two real-world datasets from Taobao[3] and Jindong[4] to validate MMD. These datasets all contain ratings in the range of 1 to 5. The details of datasets are shown in Table 2 (/*s*, /*r*, /*u* mean per sentence/review/user).

Since the original data of Amazon and Yelp are vast and sparse, we sample a small subset of the data to validate our method. Note that in the real world, the PMU ratio is about 10 percent. Specifically, we randomly select 450 users with more than five feedbacks (reviews and ratings). Note that Amazon and Yelp are standard datasets without professional malicious users. To validate our MMD, we add 50 artificial professional malicious users with negative feedbacks on random items (note that these artificial malicious users employ the masking strategies). For Taobao and Jindong, we select 450 normal users and 50 true professional malicious users, which are listed on a website (www.taocece.com, where the electronic retailers upload the professional malicious users' IDs to this website).

TABLE 2
Datasets' Characteristics

Dataset	Amazon	Yelp	Taobao	Jingdong
#user	30,759	45,980	10,121	8,031
#item	16,515	11,537	9,892	3,025
#review	285,644	229,900	10,791	8,310
#rating	285,644	229,900	49,053	25,152
Sparsity	0.051%	0.043%	0.049%	0.12%
PMU ratio	0%	0%	9.31%	10.71%
PMU fake ratings/ratings	0%	0%	45.5%	56.7%
PMU fake reviews/reviews	0%	0%	66.6%	54.5%
Avg words /*s*	10.1	9.9	12.7	13.2
Avg words /*r*	104	130	65	70
Avg sentences /*r*	9.7	11.9	4.9	5.1
Avg reviews /*u*	9.29	5.00	1.06	1.03

4.1.2 Performance Evaluation

Following the prominent work in malicious user detection [48] and shilling attack detection [44], we evaluate our proposed detection model with objective and subjective evaluations.

a) Objective Evaluation. we employ *specificity* and *sensitivity* as the objective metrics [4]

$$SEN = \frac{\#\text{true positive}}{\#\text{true positive} + \#\text{false negative}}, \quad (28)$$

$$SPE = \frac{\#\text{true negative}}{\#\text{true negative} + \#\text{false positive}}. \quad (29)$$

To explicitly introduce, we give definitions in Table 3, where the *specificity* (*SPE*) measures the proportion of correctly normal users, and the *sensitivity* (*SEN*) measures the proportion of correctly detected labeled professional malicious users. We also utilize $F-score = \frac{2 \times SEN \times SPE}{SEN+SPE}$ to balance *SEN* and *SPE*.

Moreover, to evaluate the enhancement of recommender systems, we measure the quality of recommendation with *Hit Ratio* (HR) and *Normalized Discounted Cumulative Gain*. Specifically, HR@N is a metric to measure whether the testing item exists in the Top-N recommendation list, where 1 for yes and 0 for no; NDCG@N measures the position of the

TABLE 3
Definitions for *Specificity* and *Sensitivity*

Actual \ Detected	Malicious Users	Normal Users
Malicious Users	true positive	false negative
Normal Users	false positive	true negative

1. https://jmcauley.ucsd.edu/data/amazon
2. https://www.kaggle.com/c/yelp-recsys-2013
3. https://www.taobao.com
4. https://www.jd.com

testing item in the top-N list, the higher, the better. The default setting of N is 5 without special mention. Both metrics are commonly applied for evaluating the recommender systems.

b) Subjective Evaluation. we employ 20 students to distinguish the detection results. After we achieve the professional malicious user set U^{mu}, we ask these students to label these users (1 for professional malicious users, 0 for normal users). We treat the students' results as the ground-truth and evaluate the detection model with the comparisons, which is also a supplement for subjective evaluation.

4.1.3 Baseline Methods

For professional malicious user detection, an unsupervised detection problem, there are few types of research available in this area. So we compare MMD with the following methods. Specifically, there are two unsupervised methods and two supervised methods.

K-means++ Clustering is a basic unsupervised method that clusters the data into k clusters. In this paper, we treat users as only normal or malicious ones, which means it is a 2-clustering issue, and we label the cluster with more labeled users as malicious users. *Statistic Outlier detection (SOD)* is a basic statistic method, which counts the negative feedbacks of each user, and labels the users with a high percentage of negative feedbacks as malicious users. This method is sensitive to the negative feedback threshold θ. *Hy-sad* [44] is a supervised hybrid shilling attack detection method, which introduces MC-Relief to select useful detection metrics. We find that using the user's embeddings p_u learned by LFM leads to better performance, so we report this specific setting. *Semi-sad* [4] is a semi-supervised learning based shilling attack detection algorithm, which tackles the labeled and unlabeled data at the same time. *CNN-sad* [33] is a novel convolutional neural network-based method, which applies a transformed network structure to exploit deep-level features from users rating profiles. CNN-SAD can detect shilling attacks more efficiently, which can be adapted for PMU detection. *SDRS* [48] utilizes a dual-attention RNN and a modified GRU to compute an opinion level for reviews, then a joint filtering method is proposed to detect malicious users.

Note that these baselines, especially two state-of-the-art methods (Hy-sad and Semi-sad), are validated only in the standard datasets with only ratings, which can not utilize the abundant information hidden in reviews to detect professional malicious users. In comparison, our proposed MMD can tackle ratings and reviews at the same time, which is an improvement in this area. Moreover, the practical application in real-world scenarios should also be explored, and we will do this valuable work in the following subsections.

4.1.4 Parameter Settings

To explore the hyper-parameter space for all methods, we randomly holdout a training interaction for each user as the validation set. First, for all the baselines, we report the best results to make a fair comparison. Specifically, for K-means++, we set k=2, and initialize original cluster centroids from labeled professional malicious users and normal users with clustering user's attributes p_u. For statistic outlier detection, we set negative feedback threshold θ=0.8. For Hy-sad and Semi-sad, we utilize labeled PMUs (50 PMUs in 500 users) to train the models because these models are supervised. Note that the labels of professional malicious users are difficult to obtain. We tune the size of labeled professional malicious user sets with an upper bound 10 percent of the dataset. Moreover, in Taobao and Jingdong datasets, we do not inject artificial professional malicious users to simulate the application scenario in the real-world.

For MMD, we initialize LFM and HDAN by [48]. Specifically, we fix the embedding dimension as 32 for users and items, then tune other parameters as follows. In MLC, we set λ=0.6. In professional malicious user profiling, we set the sentiment gap threshold α^{g}=3.5, and detection threshold θ^{mu}=0.7. We use these default settings if there are no additional instructions.

TABLE 4
PMU Detection With Labeled Artificial Professional Malicious Users in Amazon and Yelp

	Amazon			Yelp		
	SEN	*SPE*	F-score	*SEN*	*SPE*	F-score
K-means++	0.381	0.706	0.494	0.201	0.773	0.318
SOD	0.062	0.984	0.113	0.041	0.962	0.076
Hy-sad	0.371	0.853	0.516	0.542	0.889	0.671
Semi-sad	0.442	0.784	0.563	0.661	0.933	0.773
CNN-sad	0.552	0.791	0.650	0.663	0.922	0.771
SDRS	0.651	0.874	0.745	0.764	0.913	0.831
MLC	0.861	0.967	0.910	0.821	0.938	0.874
MUP	0.662	1*	0.795	0.742	1*	0.850
MMD (MUP+MLC)	**0.921***	**0.970**	**0.944***	**0.98***	**0.996**	**0.987***

bold *stands for MMD and * marks the best performance.*

4.2 Performance Comparison (RQ1)

Here we compare the performance of MMD with baselines. We explore the detection results with different datasets. The results are listed in Tables 4 and 5. Inspecting the results from top to bottom, we have the following observations.

TABLE 5
PMU Detection Without Labeled Artificial Professional Malicious Users in Taobao and Jindong

	Taobao			Jingdong		
	SEN	*SPE*	F-score	*SEN*	*SPE*	F-score
K-means++	0.141	0.728	0.234	0.44	0.667	0.530
SOD	0.022	0.978	0.039	0.14	0.896	0.242
Hy-sad	-	-	-	-	-	-
Semi-sad	-	-	-	-	-	-
CNN-sad	-	-	-	-	-	-
SDRS	0.451	0.884	0.597	0.732	0.911	0.811
MLC	-	-	-	-	-	-
MUP	0.641	0.996*	0.779	0.62	0.998*	0.764
MMD$MUP+MLC)	**0.941***	**0.987**	**0.963***	**0.96**	**0.989**	**0.974***

bold *stands for MMD and * marks the best performance.*

TABLE 6
Comparison With Students' Subjective Detections

	Taobao		
	SEN	*SPE*	F-score
K-means++	0.52	0.6	0.557
Semi-sad	0.68	0.76	0.718
MLC	0.92	0.88	0.899
MUP	0.76	1*	0.867
MMD$MUP+MLC)	**0.92**	**0.96**	**0.94**
Students	0.96*	0.96	0.96*

bold *stands for MMD and * marks the best performance.*

First, on Amazon and Yelp with artificial labeled professional malicious users, supervised baseline models (Hy-sad, Semi-sad, CNN-sad, and SDRS) largely outperform the unsupervised baselines (K-means++ and SOD) on F-score (Table 4). While on Taobao and Jingdong, where the datasets are without labeled malicious, supervised models, such as Hy-sad, Semi-sad, and MLC, do not work at all. The result demonstrates the positive effects of labeled data for detections, also the negative effects of narrow applications for supervised models.

Second, among all the supervised models, MLC consistently outperforms the other models. The enhancement of MLC demonstrates that the performance of a supervised model can be significantly improved by cooperating with metric learning. Because MLC utilizes metric learning and attention at the same time, we speculate that there exist complex relationships that can not be measured by simple metrics. Our method can catch this kind of relationship. Among all the unsupervised models, MUP consistently outperforms the other models, K-means++ and SOD. The enhancement of MUP demonstrates that MUP can obtain the characteristics of professional malicious users and achieves a better performance.

Third, on all four datasets, our proposed model MMD (MLC+MUP) outperforms all the baselines in terms of F-score. Note that MMD is a combination of MLC and MUP, which ensures its superiority over supervised and unsupervised models in general. On different application scenarios (with or without labeled professional malicious users), MMD can achieve a satisfying result, where the improvement over Semi-sad in Amazon/Yelp is 67.6/34.6 percent, over K-means++ in Taobao/Yelp is 311/302 percent. The result justifies the positive effect of our MMD on learning better metric representations for professional malicious user detection, thus leads to better detection performance.

Finally, taking a deep insight into the results, we notice some interesting phenomena. Among all the datasets on all metrics, SOD, a widely-applied simple outlier detection model, achieves the worst performance, which gives evidence that the effect of masking strategies can avoid traditional detections. Moreover, note that the improvement of MMD over MUP, MMD over MLC, is smaller than those over other baselines. Then the reason why we combine MLC and MMD to build MMD is that: first, MLC can not be applied to unlabeled datasets because it is a supervised model; second, MUP does not perform well in *SEN* among all baselines, which means that MUD may treat some professional malicious users as normal users, which leads to a high *SPE* and low *SEN*. With the combination, MMD can achieve superior performance on different application scenarios and avoid wrong detections.

Fig. 5. A real world case in Taobao.

Without the loss of generality, we randomly select 50 users in Taobao (25 professional malicious users reported by the website, 25 normal users), and employ 20 students to label the dataset (These students do not know the distribution of this dataset.). The results are reported in Table 6. Note that the students label 24 true positive and 24 true negative users on average. In comparison, MMD labels 23 true positive and 24 true negative users, which achieves the same level with students and surpasses baselines.

To be specific, we give some validations here, to reveal a real example in Taobao. We list some PMUs detected by MMD in Fig. 5. These PMUs can not be detected by Taobao, only being reported by retailers in the website (www.taocece.com):

Note that these PMUs do not usually give low ratings, so they can not be detected by traditional malicious user detection models. However, they usually give negative reviews with high ratings, which hurts retailers' profits and forms sentiment gaps between ratings and reviews. Our proposed model can catch the gap and detect this kind of PMUs.

4.3 Effect of Metric Learning (RQ2)

In this section, we explore the effect of metric learning in our model MMD, which consists of two parts: metric analysis and attention analysis.

4.3.1 Metric Analysis

To verify the effect of metric, we apply different metric matrix A on MLC: euclidean metric MMD (E-MMD), Diagonal-matrix MMD (D-MMD), and Full-matrix MMD (F-MMD) and our restricted matrix metric MMD (R-MMD). Specifically, E-MMD aims to learn the euclidean metric, where it is a scalar; D-MMD aims to learn a diagonal matrix, which assigns a weight vector for euclidean metric; F-MMD is to gain a full matrix, which is a general metric learning method. Our R-MMD is to learn a p-dimension diagonal matrix and a k-dimension full matrix, and two $p \times k$ matrices. All these metrics are shown in Fig. 6.

We show the detection performance with different metrics in Table 7. We notice that among all the datasets, E-MMD performs worst, and our R-MMD performs best. Specifically, we validate that euclidean distance (E-MMD) can not measure the relationship between users' attributes and sentiment gaps, also the weighted euclidean (D-MMD). However, if we set the metric matrix to a full matrix, it can

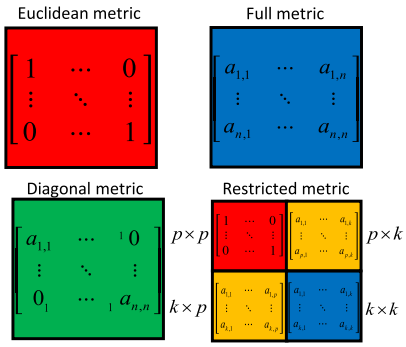


Fig. 6. An illustration of different metrics.

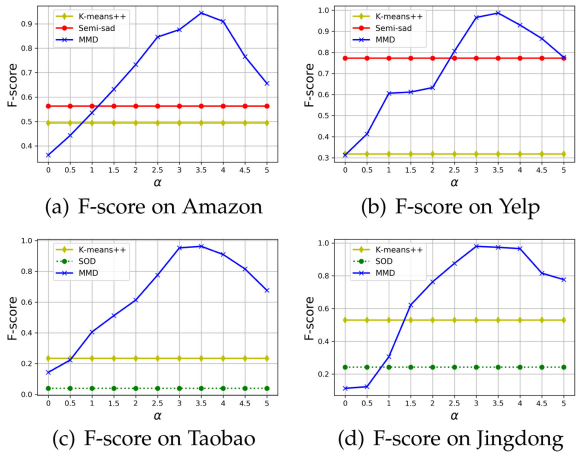
achieve the same performance as our R-MMD. However, the number of F-MMD parameters is much more substantial than R-MMD (in our experiment, F-MMD needs 4,096 parameters, while R-MMD needs 3,104 parameters), which may cost more computing resources.

4.3.2 Attention Analysis

To explore the attention mechanism for our method, we conduct MMD on all four datasets without applying attention as a reference group. We show the results in Table 8. Note that the model with attention mechanism outperforms the no-attention model on all datasets and improves the detection performance by average 7.67 percent. The attention mechanism is proper to catch the dynamic relationships, which is suitable for professional malicious user detections. Note that in our method, the attention vector can indicate the relationships not only between attributes and attributes, sentiment gap and sentiment gap, but the relationships across them, which offers the probability for explainability.

4.4 Parameter Exploration (RQ3)

In this subsection, we examine the impacts of parameters, i.e., α and θ, which control the sentiment gap to profile professional malicious users and the scale of the reviews with the sentiment gap of each user. When we explore the effect of the changing parameter, all other parameters are fixed to the initialized values.



(a) F-score on Amazon (b) F-score on Yelp

(c) F-score on Taobao (d) F-score on Jingdong

Fig. 7. Performance of MMD *w.r.t.* different values of α. MMD achieves the best F-score performance when α=3.5, 3.5, 3.5 and 3 on Amazon, Yelp, Taobao and Jingdong, respectively.

Fig. 7 illustrates the performance changes with respect to α. Note that our proposed model MMD achieves the optimal F-score performance when α=3.5, 3.5, 3.5, and 3 on Amazon, Yelp, Taobao, and Jingdong, respectively. So we set α=3.5 as default. When α is smaller than 2, increasing it leads to gradual improvement. In detail, we notice that SEN is low in this situation, which means that the model can not detect professional malicious users correctly. The result implies that the sentiment gap is always larger than 2 for PMUs, and the gap exists in normal users when it is smaller than 2. When α is larger than the optimal point (3 or 3.5), the performance drops rapidly, which means that the bigger gap threshold will affect the SPE and lead to bad detections.

Fig. 8 illustrates the performance with respect to θ. Note that our proposed model MMD achieves the optimal F-score

TABLE 7
Detection Performance With Different Metrics

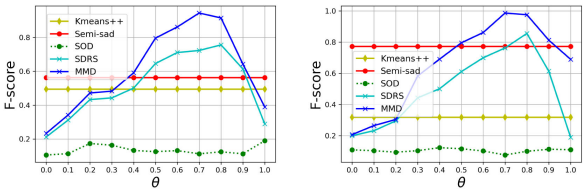
	Amazon			Yelp			Jingdong			Jingdong		
	SEN	*SPE*	F-score	*SEN*	*SPE*	F-score	*SEN*	*SPE*	F-score	*SEN*	*SPE*	F-score
E-MMD	0.183	0.261	0.215	0.113	0.137	0.123	0.143	0.222	0.173	0.133	0.141	0.136
D-MMD	0.395	0.662	0.494	0.547	0.633	0.587	0.657	0.642	0.649	0.589	0.492	0.536
F-MMD	0.902	0.93	0.914	0.96	0.977	0.968	0.945*	0.976	0.960	0.951	0.939	0.945
R-MMD	**0.92***	**0.97***	**0.944***	**0.98***	**0.996***	**0.987***	**0.94**	**0.987***	**0.963***	**0.96***	**0.989***	**0.974***

bold *stands for MMD and * marks the best performance.*

TABLE 8
Effect of Attention Mechanism on MMD for Malicious Detections

	Amazon			Yelp			Jingdong			Jingdong		
MMD	*SEN*	*SPE*	F-score	*SEN*	*SPE*	F-score	*SEN*	*SPE*	F-score	*SEN*	*SPE*	F-score
noatt	0.87	0.912	0.890	0.884	0.914	0.899	0.834	0.913	0.872	0.924	0.939	0.931
att	**0.92***	**0.97***	**0.944*(+6.0%)**	**0.98***	**0.996***	**0.987*(+9.7%)**	**0.94***	**0.987***	**0.963*+(10.4%)**	**0.96***	**0.989***	**0.974*(+4.6%)**

bold *stands for MMD and * marks the best performance.*



(a) F-score on Amazon (b) F-score on Yelp

(c) F-score on Taobao (d) F-score on Jingdong

Fig. 8. Performance of MMD *w.r.t.* different values of θ. MMD achieves the best F-score performance when θ=0.7, 0.7, 0.8 and 0.7 on Amazon, Yelp, Taobao and Jingdong, respectively.

performance when α=0.7, 0.7, 0.8, and 0.7 on Amazon, Yelp, Taobao, and Jingdong, respectively. So we set θ=0.7 as default. When θ is smaller than the threshold, the F-score improves gradually. When it adds over 0.6, MMD achieves a relatively stable performance (much better than baselines among all the datasets). Note that when θ is close to 1, it means that the model wants to detect the users with all fake feedbacks. In this situation, F-score drops rapidly because the model loses the ability to find professional malicious users. The professional malicious users do not give all fake feedbacks, which also proves our definition of professional malicious users. Specifically, the optimal points of the parameters are different for the four datasets, which indicates that for different datasets, the parameters should be separately tuned to achieve the best performance.

4.5 Recommender System Enhancement (RQ4)

In this subsection, we validate the effect of MMD on improving recommender systems. Without loss of generality, we conduct our MMD on different recommender system models, such as User-based collaborative model (UBCF) [10], Item-based collaborative model (IBCF) [28], Matrix Factorization (MF-eALS) [15], [19], Bayesian personalized ranking (MF-BPR) [26] and a state-of-the-art neural network-based model: neural collaborative filtering (NCF) [12]. The details of these models can be found in the literature. And we tune the models carefully to achieve their best performance, respectively. To validate the effect of MMD, we fix all the parameters for all the models. The only difference is in terms of the input: the models input the datasets (rating matrices), which drop the 50 professional malicious users detected by MMD (with MMD), while the control models input the original datasets, which randomly drop 50 users (original). The input of RSs is only a rating matrix without reviews, splitting datasets as 60, 20, 20 percent as training, test, and validation.

Hence, we take MMD as a preprocessing for all the recommender models and compare the results by the metric HR@N and NDCG@N. To be specific, we evaluate the performance of the recommender system concerning N=5, 15.

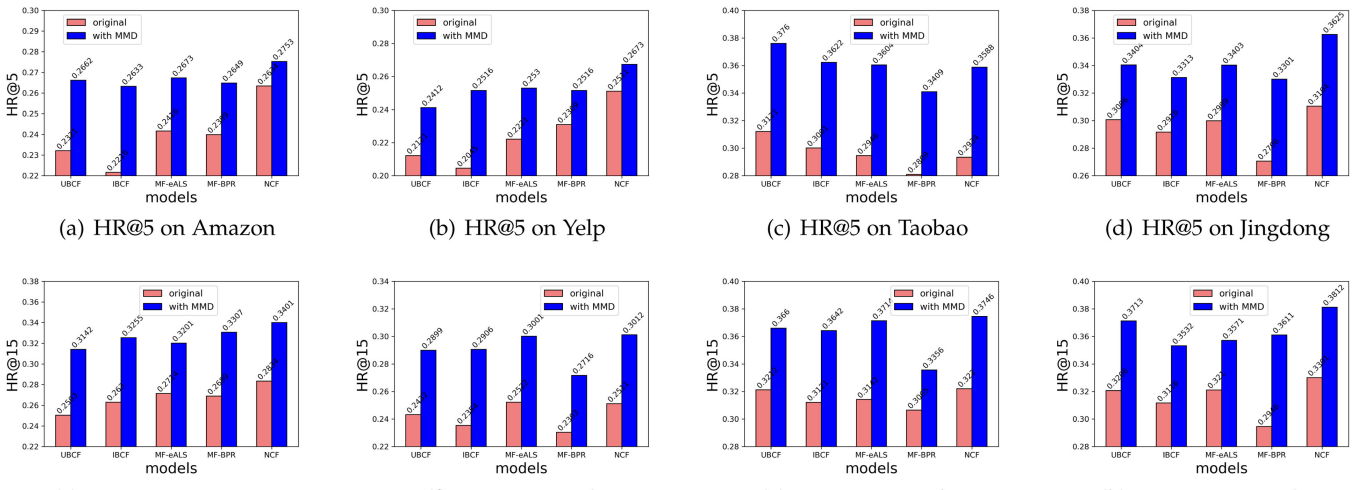
Figs. 9 and 10 show the Top-N performance of different recommendation models with or without MMD as a preprocessing. Among all four datasets, MMD improves the recommendation models significantly in terms of HR and NDCG. Specifically, MMD can enhance the performance of HR by 28.7 percent on average and HDCG by 17.3 percent on average. By deleting professional malicious users, MMD can improve the quality of datasets. Without these professional malicious users' fake feedbacks, the dataset becomes more intuitive to be understood. Because the feedbacks (reviews and ratings) are closer to the users' real opinions on items, MMD greatly benefits the CF-based models (i.e., UBCF and IBCF). Also, the neural network-based model can be enhanced by MMD to learn a proper latent space for users and items and achieve better performance (NCF).

5 Related Works

In this section, we briefly introduce related works on malicious user detection and metric learning.

5.1 Malicious User Detection

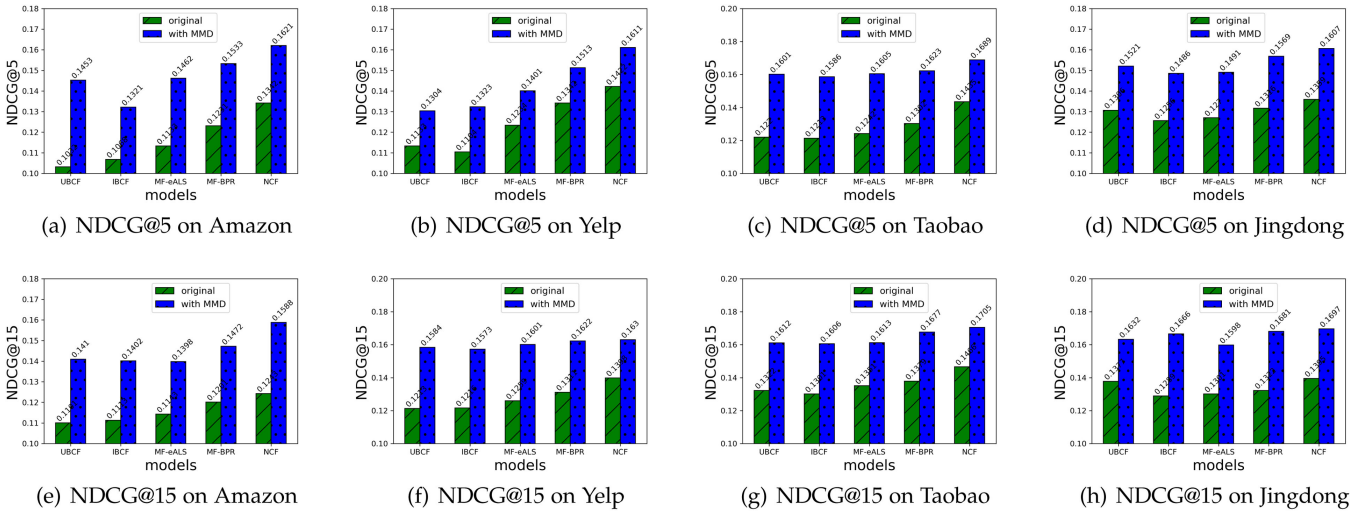
As we define professional malicious users in recommender systems, malicious user detection is a new problem, which is an issue with little attention yet. However, we can treat this detection issue as a special case of abnormal user detection,



(a) HR@5 on Amazon (b) HR@5 on Yelp (c) HR@5 on Taobao (d) HR@5 on Jingdong

(e) HR@15 on Amazon (f) HR@15 on Yelp (g) HR@15 on Taobao (h) HR@15 on Jingdong

Fig. 9. Top-N recommendation performance (HR) of different recommendation models with/without MMD as a preprocessing.



(a) NDCG@5 on Amazon (b) NDCG@5 on Yelp (c) NDCG@5 on Taobao (d) NDCG@5 on Jingdong

(e) NDCG@15 on Amazon (f) NDCG@15 on Yelp (g) NDCG@15 on Taobao (h) NDCG@15 on Jingdong

Fig. 10. Top-N recommendation performance (NDCG) of different recommendation models with/without MMD as a preprocessing.

and some existing works in this area can inspire us [36], [37]. In e-commerce, various abnormal users (spammers, shilling group, and frauds) have greatly damaged the systems, and some abnormal user detection models are proposed to tackle this issue. [43] proposed a hybrid model to detect the spammers through users' profile and relations. [11] explored spammer detection in big and sparse data. Shilling attacks harm the recommender system by injecting fake profile information of users and items. They cheat the recommendation model, such as Collaborative Filtering and Matrix Factorization [33]. [54] proposed this attack type and gave a basic supervised solution to tackle it. [33] proposed a convolutional neural network to solve shilling attacks and improved collaborative filtering. Frauds usually give fake reviews to hurt the profits of electronic retailers. [1], [42] also explored fraud detection in large-scale dataset and real scenarios. Some researches of abnormal user detection utilize the machine learning model to find fake ratings or reviews [45], [53] and achieve an effective result.

However, different from abnormal users above (spammers, shilling group, and fraud), professional malicious users are smarter and craftier. Shilling attacks inject fake ratings or reviews just before the recommendation process [7], [8], while for PMUs, all the actions that professional malicious users have taken are well-behaved by the rules of e-commerce websites (called masking strategies). They utilize the bug of abnormal detections, without leaving low ratings and negative feedback at the same time, to avoid detections. Then they can make illegal profits and hurt the electronic retailers. Basically, they are "normal" users for the existing abnormal user detection models, which makes the professional malicious user detection a critical issue in the recommender system area.

5.2 Metric Learning

To learn the complex relationships between users' attributes and sentiment gap, we employ the idea of metric learning [46]. Metric learning is a research spot for image recognition, clustering and recommendation system [22], [32], [39], [52], [57]. The key to metric learning is how to learn a proper set of metrics (such as euclidean distance or other distance metrics) to represent the relationships between different entities. As a result, some different distances are explored to understand the informational data. In [5], the authors presented an information-theoretic approach to learn a Mahalanobis distance function for complex data. While [9] improved this Mahalanobis distance function for use in classification tasks. To be specific, [41] showed how to learn a Mahanalobis distance metric for k-nearest neighbor (kNN) classification by semi-definite programming. Meanwhile, some research focuses on the constraints for metric learning with large-scale data [18]. With the predefined form of the metric matrix, metric learning can achieve a proper distance for the characterization of complex relationships.

Hence, metric learning is usually applied in the computer vision area, in which a deep transfer metric learning method for cross-domain visual recognition was proposed [17]. Because of its ability to measure the latent relations between users and items, metric learning is also widely used in recommender systems. CML [16] directly uses metric learning to embed the relationships between users and items. And IML [40] proposes a practical framework to accelerate the learning process. In a word, metric learning has shown great potential to improve the relation representation.

6 Conclusion

In this work, we first defined the professional malicious users (PMUs), who give fake feedbacks to confuse the normal users, hurt the recommender systems, and make illegal profits. We noticed that the traditional outlier detections could not be applied in the recommender system area to detect these professional malicious users because of their professional masking strategies (never give negative reviews and low ratings at the same time). Also, supervised detection models could not work well on PMU detection for the lack of labels. To address the professional malicious user detection issue, we presented a new unsupervised multi-modal learning model named MMD. By utilizing both reviews and ratings simultaneously, MMD obtained a proper metric to cluster users and detected professional malicious users. Extensive results on four real-world datasets demonstrated the effectiveness and strength of our method and the improvement by applying our method for recommender systems.

In essence, MMD is a generic solution, which can not only detect the professional malicious users that are explored in this paper but also serve as a general foundation for malicious user detections. With more data, such as image, video, or sound, the idea of MMD can be instructive to detect the sentiment gap between their title and content, which has a bright future to counter different masking strategies in different applications. Moreover, we will incorporate multimedia data into our model and consider the effect of contexts, such as consuming time, clicks, and other interactions. At last, we are very interested in building an online professional malicious user detection model that utilizes the recent advances in human-machine interactions.

Acknowledgments

This work was supported by NSFC 91746301, and the National Natural Science Foundations of China under Grant No. 61772230 and No. 61972450, Natural Science Foundation of China for Young Scholars No. 61702215 and No. 62002132, China Postdoctoral Science Foundation No. 2017M611322 and No. 2018T110247, and Changchun Science and Technology Development Project No.18DY005.

References

[1] L. Akoglu, R. Chandy, and C. Faloutsos, "Opinion fraud detection in online reviews by network effects," in *Proc. 7th Int. Conf. Weblogs Soc. Media*, 2013, pp. 2–11.
[2] S. Akter and S. F. Wamba, "Big data analytics in E-commerce: A systematic review and agenda for future research," *Electron. Markets*, vol. 26, no. 2, pp. 173–194, 2016.
[3] Y. Cai and D. Zhu, "Trustworthy and profit: A new value-based neighbor selection method in recommender systems under shilling attacks," *Decis. Support Syst.*, vol. 124, 2019, Art. no. 113112.
[4] J. Cao, Z. Wu, B. Mao, and Y. Zhang, "Shilling attack detection utilizing semi-supervised learning method for collaborative recommender system," *World Wide Web*, vol. 16, no. 5/6, pp. 729–748, 2013.
[5] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *Proc. 24th Int. Conf. Mach. Learn.*, 2007, pp. 209–216.
[6] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, no. Jul., pp. 2121–2159, 2011.
[7] M. Fang, N. Z. Gong, and J. Liu, "Influence function based data poisoning attacks to top-N recommender systems," in *Proc. Web Conf.*, 2020, pp. 3019–3025.
[8] M. Fang, G. Yang, N. Z. Gong, and J. Liu, "Poisoning attacks to graph-based recommender systems," in *Proc. 34th Annu. Comput. Secur. Appl. Conf.*, 2018, pp. 381–392.
[9] A. Globerson and S. T. Roweis, "Metric learning by collapsing classes," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2006, pp. 451–458.
[10] S. Gong, "A collaborative filtering recommendation algorithm based on user clustering and item clustering," *J. Softw.*, vol. 5, no. 7, pp. 745–752, 2010.
[11] B. Guo, H. Wang, Z. Yu, and Y. Sun, "Detecting spammers in e-commerce website via spectrum features of user relation graph," in *Proc. 5th Int. Conf. Adv. Cloud Big Data*, 2017, pp. 324–330.
[12] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 173–182.
[13] X. He, J. Tang, X. Du, R. Hong, T. Ren, and T.-S. Chua, "Fast matrix factorization with nonuniform weights on missing data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 8, pp. 2791–2804, Aug. 2020.
[14] X. He, H. Zhang, M.-Y. Kan, and T.-S. Chua, "Fast matrix factorization for online recommendation with implicit feedback," in *Proc. 39th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2016, pp. 549–558.
[15] X. He, H. Zhang, M.-Y. Kan, and T.-S. Chua, "Fast matrix factorization for online recommendation with implicit feedback," in *Proc. 39th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2016, pp. 549–558.
[16] C.-K. Hsieh, L. Yang, Y. Cui, T.-Y. Lin, S. Belongie, and D. Estrin, "Collaborative metric learning," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 193–201.
[17] J. Hu, J. Lu, and Y.-P. Tan, "Deep transfer metric learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 325–333.
[18] M. Köstinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof, "Large scale metric learning from equivalence constraints," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 2288–2295.
[19] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.
[20] S. K. Kwak and J. H. Kim, "Statistical data preparation: Management of missing values and outliers," *Korean J. Anesthesiol.*, vol. 70, no. 4, 2017, Art. no. 407.
[21] J. Li, A. J. Ma, and P. C. Yuen, "Semi-supervised region metric learning for person re-identification," *Int. J. Comput. Vis.*, vol. 126, no. 8, pp. 855–874, 2018.
[22] J. Li, A. J. Ma, and P. C. Yuen, "Semi-supervised region metric learning for person re-identification," *Int. J. Comput. Vis.*, vol. 126, no. 8, pp. 855–874, 2018.
[23] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1412–1421, doi: 10.18653/v1/d15-1166.
[24] R. A. Maronna, R. D. Martin, V. J. Yohai, and M. Salibián-Barrera, *Robust Statistics: Theory and Methods (With R)*. Hoboken, NJ, USA: Wiley, 2019.
[25] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *Proc. 25th Conf. Uncertainty Artif. Intell.*, 2009, pp. 452–461.
[26] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *Proc. 25th Conf. Uncertainty Artif. Intell.*, 2009, pp. 452–461.
[27] B. Rutherford *et al.*, "Customer authentication in e-commerce transactions," U.S. Patent 9 514 458, Dec. 6, 2016.
[28] B. M. Sarwar *et al.*, "Item-based collaborative filtering recommendation algorithms," in *Proc. 10th Int. Conf. World Wide Web*, 2001, pp. 285–295.
[29] M. Si and Q. Li, "Shilling attacks against collaborative recommender systems: A review," *Artif. Intell. Rev.*, vol. 53, pp. 291–319, 2020.
[30] J. Su, "Content based recommendation system," U.S. Patent 9 230 212, Jan. 5, 2016.
[31] X. Sui, E. L. Xu, X. Qian, and T. Liu, "Convex clustering with metric learning," *Pattern Recognit.*, vol. 81, pp. 575–584, 2018.
[32] X. Sui, E. L. Xu, X. Qian, and T. Liu, "Convex clustering with metric learning," *Pattern Recognit.*, vol. 81, pp. 575–584, 2018.
[33] C. Tong *et al.*, "A shilling attack detector based on convolutional neural network for collaborative recommender system in social aware network," *Comput. J.*, vol. 61, no. 7, pp. 949–958, 2018.
[34] C. G. Traver and K. C. Laudon, *E-Commerce: Business, Technology, Society*. Englewood Cliffs, NJ, USA/London, U.K.: Pearson Prentice Hall/Pearson Education, 2008.
[35] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
[36] B. Wang, N. Z. Gong, and H. Fu, "GANG: Detecting fraudulent users in online social networks via guilt-by-association on directed graphs," in *Proc. IEEE Int. Conf. Data Mining*, 2017, pp. 465–474.
[37] B. Wang, J. Jia, and N. Z. Gong, "Graph-based security and privacy analytics via collective classification with joint weight learning and propagation," in *Proc. 26th Annu. Netw. Distrib. Syst. Secur. Symp.*, 2019, pp. 1–15.
[38] D. Wang and X. Tan, "Robust distance metric learning via Bayesian inference," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1542–1553, Mar. 2018.
[39] D. Wang and X. Tan, "Robust distance metric learning via Bayesian inference," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1542–1553, Mar. 2018.
[40] N. Wang, X. Zhao, Y. Jiang, and Y. Gao, "Iterative metric learning for imbalance data classification," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, pp. 2805–2811.
[41] K. Q. Weinberger, J. Blitzer, and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2006, pp. 1473–1480.
[42] H. Weng *et al.*, "Online E-commerce fraud: A large-scale detection and analysis," in *Proc. IEEE 34th Int. Conf. Data Eng.*, 2018, pp. 1435–1440.
[43] Z. Wu, Y. Wang, Y. Wang, J. Wu, J. Cao, and L. Zhang, "Spammers detection from product reviews: A hybrid model," in *Proc. IEEE Int. Conf. Data Mining*, 2015, pp. 1039–1044.

[44] Z. Wu, J. Wu, J. Cao, and D. Tao, "HySAD: A semi-supervised hybrid shilling attack detector for trustworthy product recommendation," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2012, pp. 985–993.
[45] Z. Xia, C. Liu, N. Z. Gong, Q. Li, Y. Cui, and D. Song, "Characterizing and detecting malicious accounts in privacy-centric mobile social networks: A case study," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 2012–2022.
[46] E. P. Xing, M. I. Jordan, S. J. Russell, and A. Y. Ng, "Distance metric learning with application to clustering with side-information," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2003, pp. 521–528.
[47] Y. Xu and F. Zhang, "Detecting shilling attacks in social recommender systems based on time series analysis and trust features," *Knowl.-Based Syst.*, vol. 178, pp. 25–47, 2019.
[48] Y. Xu, Y. Yang, J. Han, E. Wang, J. Ming, and H. Xiong, "Slanderous user detection with modified recurrent neural networks in recommender system," *Inf. Sci.*, vol. 505, pp. 265–281, 2019.
[49] Y. Xu *et al.*, "NeuO: Exploiting the sentimental bias between ratings and reviews with neural networks," *Neural Netw.*, vol. 111, pp. 77–88, 2019.
[50] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2016, pp. 1480–1489.
[51] H. J. Ye, D. C. Zhan, and Y. Jiang, "Fast generalization rates for distance metric learning," *Mach. Learn.*, vol. 108, pp. 267–295, 2019.
[52] H. J. Ye, D. C. Zhan, and Y. Jiang, "Fast generalization rates for distance metric learning," *Mach. Learn.*, vol. 108, pp. 267–295, 2019.
[53] D. Yuan *et al.* "Detecting fake accounts in online social networks at the time of registrations," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2019, pp. 1423–1438.
[54] W. Zhou *et al.* "Shilling attacks detection in recommender systems based on target item analysis," *PLoS One*, vol. 10, no. 7, 2015, Art. no. e0130968.
[55] W. Zhou, J. Wen, Q. Xiong, M. Gao, and J. Zeng, "SVM-TIA a shilling attack detection method based on SVM and target item analysis in recommender systems," *Neurocomputing*, vol. 210, pp. 197–205, 2016.
[56] W. Zuo *et al.* "Distance metric learning via iterated support vector machines," *IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 4937–4950, Oct. 2017.
[57] W. Zuo *et al.*, "Distance metric learning via iterated support vector machines," *IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 4937–4950, Oct. 2017.

Yuanbo Xu received the BE, ME, and PhD degrees in computer science and technology from Jilin University, Changchun, China, in 2012, 2015, and 2019, respectively. He is currently a postdoc with the Department of Artificial Intelligence, Jilin University, Changchun. His research interests include applications of data mining, recommender system, and mobile computing. He has published some research results on journals such as the *IEEE Transactions on Multimedia*, *IEEE Transactions on Neural Networks and Learning Systems*, and conference as ICDM.

Yongjian Yang received the BE degree in automatization from the Jilin University of Technology, Changchun, Jilin, China, in 1983, the ME degree in computer communication from the Beijing University of Post and Telecommunications, Beijing, China, in 1991, and the PhD degree in software and theory of computer from Jilin University, Changchun, Jilin, China, in 2005. He is currently a professor and a PhD supervisor with Jilin University, director of Key Lab under the Ministry of Information Industry, standing director of Communication Academy, member of the Computer Science Academy of Jilin Province. His research interests include theory and software technology of network intelligence management and key technology research of wireless mobile communication and services. He participated three projects of NSFC, 863 and funded by National Education Ministry for Doctoral Base Foundation. He has authored 12 projects of NSFC, key projects of Ministry of Information Industry, Middle and Young Science and Technology Developing Funds, Jilin provincial programs, ShenZhen, ZhuHai, and Changchun.

En Wang received the BE degree in software engineering from Jilin University, Changchun, China, in 2011, and the ME and PhD degrees in computer science and technology from Jilin University, Changchun, China, in 2013 and 2016, respectively. He is currently an associate professor with the Department of Computer Science and Technology, Jilin University, Changchun. He is also a visiting scholar with the Department of Computer and Information Sciences, Temple University in Philadelphia. His current research focuses on the efficient utilization of network resources, scheduling and drop strategy in terms of buffer-management, energy-efficient communication between human-carried devices, and mobile crowdsensing.

Fuzhen Zhuang is an associate professor with the Institute of Computing Technology, Chinese Academy of Sciences. His research interests include transfer learning, machine learning, data mining, multi-task learning, and recommendation systems. He has published more than 100 papers in some prestigious refereed journals and conference proceedings, such as the *IEEE Transactions on Knowledge and Data Engineering*, *IEEE Transactions on Cybernetics*, the *IEEE Transactions on Neural Networks and Learning Systems*, *ACM Transactions on Intelligent Systems and Technology*, the *Information Sciences*, *Neural Networks*, SIGKDD, IJCAI, AAAI, WWW, ICDE, ACM CIKM, ACM WSDM, SIAM SDM, and IEEE ICDM.

Hui Xiong (Fellow, IEEE) received the PhD degree from the University of Minnesota (UMN), Minneapolis, Minnesota. He is currently a full professor with the Rutgers, the State University of New Jersey, where he received the 2018 Ram Charan Management Practice Award as the Grand Prix winner from the Harvard Business Review, RBS Dean's Research Professorship (2016), the Rutgers University Board of Trustees Research Fellowship for Scholarly Excellence (2009), the ICDM Best Research Paper Award (2011), and the IEEE ICDM Outstanding Service Award (2017). He is a co-editor-in-chief of *Encyclopedia of GIS*, an associate editor of the *IEEE Transactions on Big Data* (TBD), *ACM Transactions on Knowledge Discovery from Data* (TKDD), and *ACM Transactions on Management Information Systems* (TMIS). He has served regularly on the organization and program committees of numerous conferences, including as a program co-chair of the Industrial and Government Track for the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), a program co-chair for the IEEE 2013 International Conference on Data Mining (ICDM), a general co-chair for the IEEE 2015 International Conference on Data Mining (ICDM), and a program co-chair of the Research Track for the 2018 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. He is an ACM distinguished scientist.