IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received January 19, 2021, accepted January 21, 2021, date of publication January 25, 2021, date of current version February 1, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3054122

RMPD: Method for Enhancing the Robustness of Recommendations With Attack Environments

QI DING[1], PEIYU LIU[1], ZHENFANG ZHU[2], HUAJUAN DUAN[1], AND FUYONG XU[1]

[1]School of Information Science and Engineering, Shandong Normal University, Jinan 250014, China
[2]School of Information Science and Electrical Engineering, Shandong Jiaotong University, Jinan 250357, China

Corresponding author: Peiyu Liu (liupy@sdnu.edu.cn)

This work was supported in part by the National Social Science Fund under Grant 19BYY076, in part by the Science Foundation of Ministry of Education of China under Grant 14YJC860042, in part by the Shandong Provincial Social Science Planning Project under Grant 19BJCJ51, Grant 18CXWJ01, Grant 18BJYJ04, and Grant 16CFXJ18, and in part by the Key Research and Development project of Shandong Province under Grant 2019JZZY010129.

ABSTRACT Personalized item recommendation has become a hot topic research among academic and industry community. But lots of purposeful fraudsters maybe perform different attacks on the recommender system to insert fake ratings, which could reduce the authenticity and reliability of recommendations. For a recommender system with fraudsters, it is crucial to detect malicious ratings and reduce the proportion of fraudster's ratings. This paper presents a method Prediction and Detection of Rating Matrix(RMPD) combining rating prediction and attack detection. The detection results of the attackers are applied to the rating prediction, thereby controlling the contribution and proportion of attackers to the rating prediction component both in training and learning, and then implementing more accurate item rating projections. The method will also solve the problem of data sparsity in the recommender system to some extent. The superiority of the proposed method in predicting recommendation performance compared with other baseline methods is demonstrated on real-world datasets. The ablation experiment proves the necessity of the components.

INDEX TERMS Deep neural networks, matrix decomposition, recommender system, robustness.

I. INTRODUCTION

The generation of the recommender system is closely related to the progress and development of the times: a variety of business models are transformed from offline to online, and massive data wastes the system occupation, users cannot make accurately choices for explosive data, etc. IDC (2012) report shows that by 2020, the total number of data in various fields in the world will reach 22 times that of 2011 [1]. In recent years, the recommender system has been applied in e-commerce, social network, pushnews, information retrieval, etc., like Amazon builds personalized online stores for every customer [2], Chinese shopping e-commerce site—Tao bao, to boost sales, each user can see the recommended products ''Guess you like'' on their interface. According to Microsoft Asia Research Academy, about 30% web page browsing of Amazon comes from recommender systems [3]. Prediction accuracy and recommendation accuracy is the basic problem of the recommender system, a large number of ratings or reviews from real users can truly reflect user preferences and item levels. The interaction between users and the recommender system can be divided into explicit(e.g. user's previous ratings or reviews) and implicit(e.g.user's search browsing records, clicks and purchase records of items are implicit feedback). Interactive information is a crucial part for prediction and recommendation to learn user representation and item representation. Actually, the user-item rating matrix(URM) has a certain sparseness is due to the explicit interaction between each user and each item cannot reach 100%. For example, the MovieLens dataset, one of the most frequently used datasets in the recommender system, has a sparsity of 4.5%. The sparsity of the rating matrix will limit the ability of rating-based methods to learn accurate representation of users and items [4]. The method RMPD proposed in this work alleviates the performance degradation caused by data sparsity to some extent.

The recommender system involves some commercial interests, and most recommender systems have openness due to the simple registration or access conditions. Some users have launched attacks on the recommender systems that driven by personal or group interests, market penetration, and even for causing mischief on an underlying system [5], which has disrupted the accuracy and fairness of recommendation in the

The associate editor coordinating the review of this manuscript and approving it for publication was Yanbo Chen.

recommender systems. Such users are called fraudsters [6] or attacker, and such activities to manipulate rating results are termed shilling attack [7] or profile injection attack [8]. There is a more standardized recommended attack classification: it can be divided into push attacks and nuke attacks according to the fraudster's purpose [9], especially it has been proven that there is an imbalance between push attacks and nuke attacks [10], it could also divided into high-level knowledge attacks and low-level knowledge attacks based on the knowledge required by the fraudster to launch the attack. In [11] describe several important properties about characteristics of shilling attack: fake ratings have lower deviation than mean votes value, but have high deviation from the mean for the attacked items, and attackers have a high correlation with a significant number of real users, etc. There are three pathways to ease the shilling attack on the recommender systems. One is to increase the robustness of recommendation models with the existence of shilling attacks, other methods are efficiently detect attack or fake profiles and eliminate attacking users from the source. The first two methods have been studied separately, and the third method requires the system to have a more complete screening user system mechanism. In addition, robustness research is also crucial in other fields, such as: the large amount of data in the smart grid environment is more inclined to be either accidently damaged or maliciously attacked [12], and there are some meaningful robust state estimation method for power systems [13], [14].

The work proposed in this paper focuses on the ‘‘robustness’’ of recommendations. Our contributions in this paper are summarized as follows:

- The information sparsity of the rating matrix will affect the prediction accuracy, this paper proposes a modified Singular Value Decomposition(SVD) method to predict the vacancies in the user-item rating matrix. Considering that different users have different scoring strictness and different scoring ranges, we add user preferences, item bias, and average rating values for different items into the basic SVD prediction formula. This reduces the sparsity of the rating matrix and laying a solid foundation for subsequent work.
- We solved the task of robust recommendation and fraudster detection at the same time. We proposed a framework prediction and detection of rating matrix with Neural Network structure and Binary Tree Classifier, namely RMPD, to detect possible fraudsters and reduce the proportion of fraudsters for rating prediction.
- The comparative experimental results prove that our method has better robust performance than other baseline methods with attack environments, and the ablation experiment proves the importance of the attack detection component in the method.

II. RELATED WORK

In recent years, most research on deep learning has been carried out in the fields of image processing, natural language processing, and speech recognition, etc. [15]. Deep learning on recommender systems has also made some achievements [16], [17]. The deep learning model can learn the deep feature representation of users and items in the data set through a deep linear network structure, and perform automatic feature learning from multi-source heterogeneous data, thereby mapping different data to the same hidden space to obtain data unity characterization [18]. Over the past decade, most research in collaborative filtering of recommender systems has emphasized the use of deep learning. For example, the self-encoder-based collaborative filtering method proposed by Sedhain *et al.* [19] proposed collaborative filtering method based on autoencoder that via decoding process to output the rows or columns of the input rating matrix, and pass the minimum optimization of model parameters by reconstructing errors. In distributed representation learning, Grbovic *et al.* [20] used distributed representation technology to study the problem of advertising recommendation in mail systems. Wu *et al.* [21] proposed a recurrent recommendation network, which uses recurrent neural networks to model the evolution of user preferences and item features to predict the future behavior trajectory of users.

In addition, in some research fields that have been successful so far, the information source of the recommendation system is not only from the historical interaction data information between users and items, but also involves the collection of user desire information displayed interactively in real time, according to user social relations network information for recommendations [22]–[24], geographic location information for related recommendations [25], etc. The further study could assess the long-term effects of modeling and fusion of more heterogeneous information are needed to improve the performance of the recommender system.

Overall, most studies highlight the need for prediction and recommended performance. Due to the existence of lots of attacks and fraud in the actual recommender systems, the pure recommendation algorithm may not have strong robustness to this kind of attack, so the research on the robustness of recommender system is gradually rises. Previous robustness research [26] established a robust matrix factorization algorithm was proposed to improve the robustness by removing extreme votes, but this method may mistakenly delete real user's votes. In a newer study [5], experiments show that URM characteristics significantly affect the robustness of the collaborative filtering(CF) model in the shilling attack scenario, which is very helpful for system designers to understand the reasons why the recommender system performance changes due to the shilling attack. In [27], Yuzhe Ma *et al.* proposed a stochastic gradient descent attack, and the experiments show that private learners are vulnerable to data poisoning attacks when the adversary can poison sufficiently many examples. Due to the compulsory security performance, the detection of attacks in the recommender system is gradually increasing. In the early work, Chirita *et al.* first proposed several statisticalbased measurement methods for high-density attacked files in [28], but they could not effectively detect low-density attacked files. The attack detection

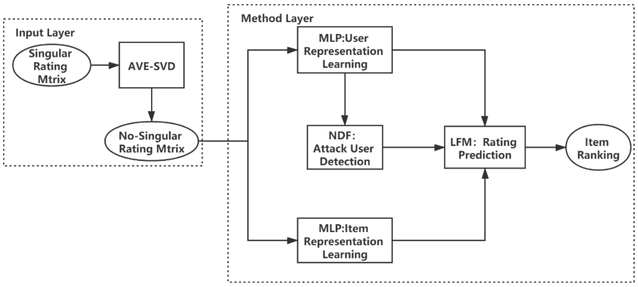


FIGURE 1. **Framework of the proposed RMPD approach.**

method based on Hidden Markov model and hierarchical clustering proposed in [29], although it has a more effective detection effect for various types of attacks, but if the detection dataset contains only a single real or attack file, the detection result poor. The detection method based on convolutional neural network(CNN) in [30], because this method only contains a convolutional layer and a pooling layer, it has a certain detection effect for smaller datasets. In the recent method [31], a classifier based on the CNN model is proposed, which can learn directly from the resized rating matrix.

Regarding the attack problem in the recommender system, most of the current researches are aimed at improving the robustness of predictive recommendation or just use the attack detection component to detect the fraudster or spam. In the work of this paper, the main innovation is to add the attack detection component to the predictive recommendation method, and use the result of the attack detection component to make a probabilistic judgment on the authenticity of the user, thereby controlling the user's contribution to the predictive recommendation component. The result of this is enhancing the robustness of recommendations with attack environments.

III. PROPOSED METHOD

In this section, we will introduce our method RMPD in detail. The overview of RMPD is shown in Fig.1, which mainly divided into input layer and method layer respectively. In the input layer, we adopt the Average- Singular Value Decomposition(AVE-SVD) component to predict the filling to reduce its sparsity and obtain a non-sparse rating matrix. The method layer consists of two key components to achieve rating prediction and fraudster detection, respectively. We use Multi-layer perceptron(MLP) neural networks [32], [33] as a rating-based encoder to learn rating-based representations of users or items, then predict ratings based on user and item representations via Latent Factor Model(LFM) [34] components. In what follows, the probability of a user being classified as an fraudster by the Neural Detection Forest(NDF) [35] component is taken as a weight to control the contribution of this fraudster's rating in the rating prediction component. In the last step of this method, the items rating ranking list is finally generated for recommendation according to the final items rating prediction. Next we will give details for illustrating each component.

A. INPUT LAYER

The Input layer framework of RMPD approach is shown in Fig.2. This layer transform rating vector to the rating matrixes by matrix generation method. In order to alleviate sparsity of rating matrixes, we use the AVE-SVD method to predict and fill the matrix rating null value, that is, an item that a user has not rated, to obtain a non-singular rating matrix as the output of this layer.

The ratings of each user are usually distributed in one-dimensional rating vector, and the distribution of ratings reflects the personalized information of users. In order to visually show the different preferences of different users, we distribute the user ratings into two-dimensional rating matrix by rating matrix method. Suppose the number of users in database is N, the number of items set M. Let I denote the number of all ratings, then the generated user-item rating matrix is N × M order matrix $A_{N,M}$. Let the n/m denote a row/column in the matrix.(nϵN, mϵM). Elements of the rating matrix are determined by using the following method.

$$A_{n,m} = \begin{cases} Rating_m & n \leq 1 \\ Rating_{(n-1)^{*}M+m} & n > 1 \end{cases} \tag{1}$$

where, $A_{n,m}$ denotes the element value of the nth row and mth column, $Rating_m$ and $Rating_{(n-1)^{*}M+m}$ denote to the initial rating vector sequence value. The null value in the rating vector also occupies the vector position and is marked as 0.

We use the matrix factorization method to predict and fill the vacancies in the $A_{N,M}$ to improve subsequent prediction performance. First, the URM $A_{N,M}$ is decomposed into user implicit vector matrix $P_{N,K}$ and item implicit vector matrix $Q_{K,M}$:

$$A_{N,M} = P_{N,K}Q_{K,M} \tag{2}$$

We refer to previous work, Koren expands and transforms the basic SVD prediction formula [34], adding user and item attribute information and global item average rating. We made a slight modification on this and named AVE-SVD: changed the global item average rating item to the average rating of each item, we think this approach is more pertinent. The formula for predicting rating $\widehat{r_{n,\ m}}$ is defined as follows. Where $\overline{r_m}$ is the average rating of the mth item, b_n and b_m are attribute value of user n and item m, respectively. p_n is the vector representation of each row of matrix $P_{N,K}$, q_m is the vector representation of each column of matrix $Q_{K,M}$.

$$\widehat{r_{n,\ m}} = \overline{r_m} + b_n + b_m + p_n^T q_m \tag{3}$$

The loss function $Loss_{AVE-SVD}$ is the sum of squared errors between the predicted value $\widehat{r_{n,\ m}}$ and the ground truth rating $R_{n,\ m}$:

$$Loss_{AVE-SVD} = \sum_{m \in N, m \in M} \left(R_{n,\ m} - \widehat{r_{n,\ m}}\right)^2 \tag{4}$$

In order to effectively relief the phenomenon of overfitting, we add a regularization to the loss function to punish the

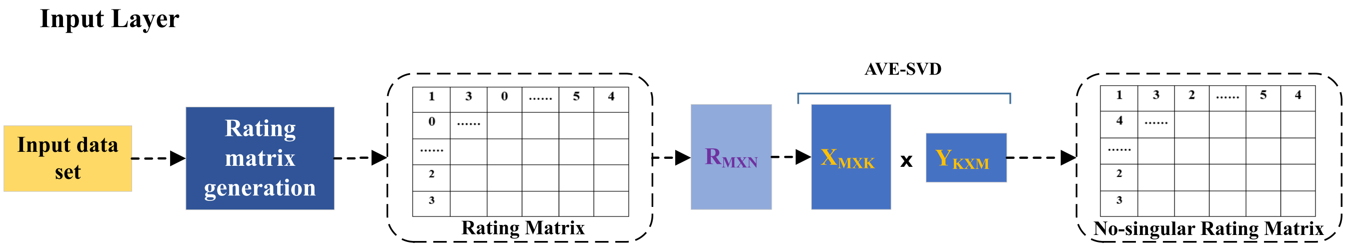


FIGURE 2. The overview of Input layer.

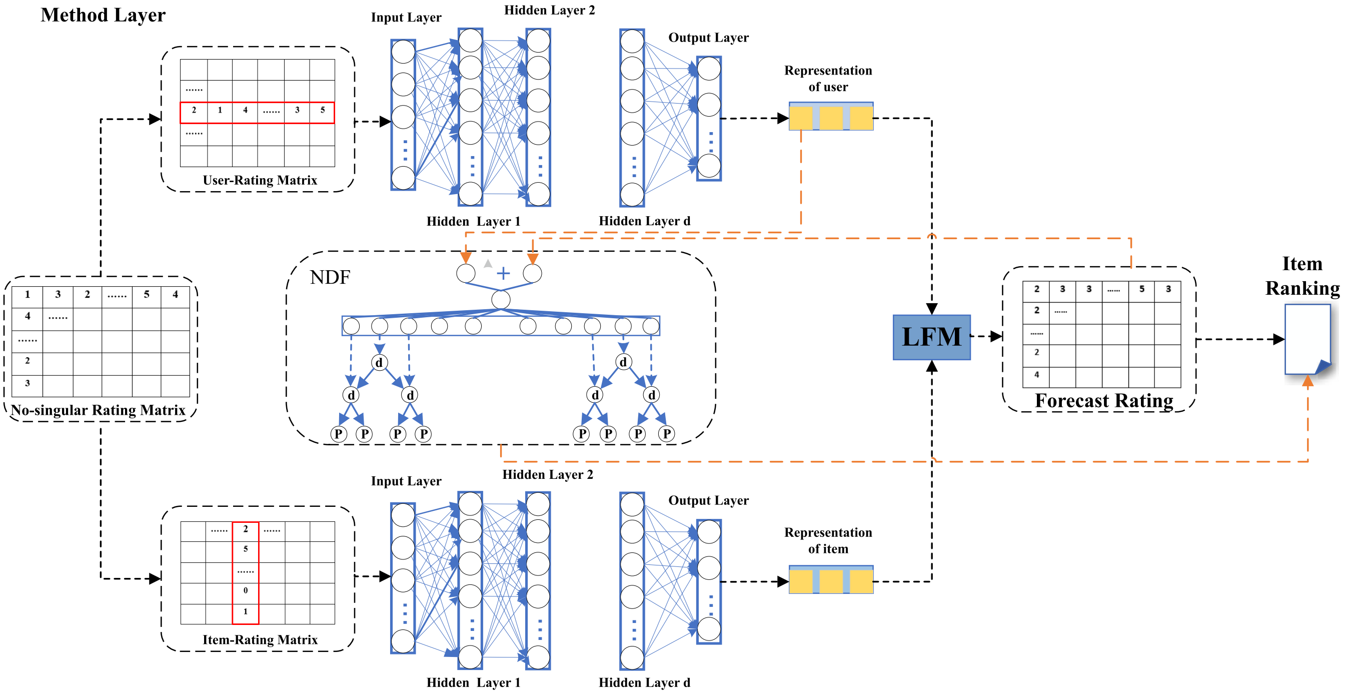


FIGURE 3. The overview of Method layer.

parameters. λ is the regularization coefficient, the optimized loss function $Loss'_{AVE-SVD}$ as follow:

$$Loss'_{AVE-SVD} = \frac{1}{2}\sum_{n,m}\left(R_{n,m} - \widehat{r_{n,\,m}}\right)^2 + \frac{1}{2}\lambda\sum_{n}|p_n|^2 + \frac{1}{2}\lambda\sum_{m}|q_m|^2 + \frac{1}{2}\lambda\sum_{n}b_n^2 + \frac{1}{2}\lambda\sum_{m}b_m^2 \quad (5)$$

B. METHOD LAYER

The method layer framework of RMPD approach is shown in Fig.3. We divided this layer into rating prediction part, recommendation part and fraudster detection part. In the predictive rating component, we divide the non-sparse rating matrix into a user-rating matrix and an item-rating matrix, and use MLP to train the two to obtain rating-based representations of users and items are used as the input of LFM to predict the rating, then generate a rating prediction information. We use NDF as the attacker detection model. The probability of a user being classified as fraudster by the NDF component is taken as a weight to control the contribution of the fraudster to rating prediction component. Each part of the method layer will be described in detail below.

1) RATING PREDICTION COMPONENT

In the initial part of the method layer, the user-item rating matrix is decomposed into a user-rating pattern matrix R_u and an item-rating pattern matrix R_i. We utilize the MLP network, which can learn the deep explicit feature E_n and E_m of the user and item from the two patterns respectively.

For the MLP multilayer perceptron, we use its multilayer hidden layer structure to mine the deep feature representation of users and items to improve its training accuracy and performance. Where the network input n' and m' are scale-normalization rating pattern of user and item (i.e., the row in R_u and the column in R_i). The output rating-based representation E_n of user and E_m of the item are obtained:

$$E_n = \sigma\left(W^d n'(d-1) + b^d\right)$$
$$E_m = \sigma\left(W^d m'(d-1) + b^d\right) \quad (6)$$

Among them, σ is the activation function, we denote the number of hidden layers to d, and W, b are the parameters in the hidden layer of MLP.

we utilize LFM to compute the rating that the user would rating the item according to their representations. LFM can describe the interaction between users and items, so integrate the personalized and individual preference for each user and item, and more accurately learn the implicit classification of users and items to predict user ratings [6]. Rating prediction $\widehat{R_{n,\,m}}$ is computed as below:

$$\widehat{R_{n,\,m}} = W_{LFM}^{T}\left(E_n \cdot E_m\right) + bis_n + bis_m + bis \tag{7}$$

LFM can further captures the potential feature of the user and the item. The rating prediction formula contains four component elements. Among them, $\cdot$ is the element-wise product, the first item is the elementwise of user feature representation and item feature representation, W_{LFM}^{T} denotes the parameter matrix representation of LFM. It is worth noting that adding the bias to LFM to indicate that the user's personalization bias can reduce the prediction error. The *bis* denotes the global average of all rating records in the training set, and represents the overall rating of the training data. User bias bis_m indicates the rating habits of a particular user and item bias bis_n indicates the rating situation of a specific item respectively.

After the final predicted rating combined with the attack detection result is generated, the rating matrix is processed: take the average rating of each item $\overline{R_m}$ represents the rating level of the item, and use it as the interaction information between the item and the user. Finally, take the average of all items sort in descending order and take the top X items to generate a recommendation list.

2) FRAUDSTER DETECTION COMPONENT

In this part, a forest composed of binary classification trees is designed as an fraudster detection component to detect all users in the dataset. In previous studies [35], the classification tree has achieved good performance by combining the end-to-end training of the classification tree with the known representation learning function from the deep convolutional network. Based on the inspiration of recent research [31] and our functional requirements for detecting fraudster components, we designed the decision forest based on the existing research about binary tree classifier. The neural detection forest is a forest composed of multiple standard binary trees based on a neural network, the detection process has a certain interpretability because it has a certain model expression ability. The probability of fraudster provides some support for the followup work of this method.

In the proposed neural detection forest, the input E_n'comes from two parts of different information, the purpose is to support the training of the binary tree classifier with sufficient information, which is reflected in formula (10):

1) Take the rating-based user representation E_n obtained by training and learning in the MLP component as a part of the NDF input.
2) We use the error $error'$ between predicted rating $\widehat{R_{n,\,m}}$ of the item output by the LFM and the real rating $R_{n,\,m}$ as the second part of the input of the NDF, which will provide a part of reference for the detection of fraudster. If a user's ratings largely deviate from the predicted ones, this user is likely to be a fraudster. We calculate the mean square of all rating prediction errors on the interacted items as follow.

$$error' = \frac{1}{\left|H_{(nm)}\right|} \sum_{n,m \in H_{(n,m)}} \left(\left|R_{n,m} - \widehat{R_{n,m}}\right|^2\right) \tag{8}$$

where $H_{(n,m)}$ is the set of item m rated by user n.

Use the concatenated feature E_n', pass it through a connection layer to densely represent information and finally get the input E_n^* of the NDF component:

$$\begin{aligned} E_n' &= E_n \oplus error' \\ E_n^* &= \sigma(W_{E'} E_n{}' + b_{E'}) \end{aligned} \tag{9}$$

σ is the activation function, $W_{E'}$ and $b_{E'}$ are the weight and bias.

A forest is an ensemble of G decision trees $F = \{T_1, T_2, \ldots, T_G\}$. The non-leaf nodes in the forest are set as decision nodes and the index is set to D, each decision node is $d \epsilon D$. Similarly, all leaf nodes are set as prediction nodes, and the index is set to P, and each prediction node is $p \epsilon P$. For each tree T_g ($g \in [1, G]$) in the forest, each prediction node has a class probability distribution π. For the classification label $y \epsilon$ [0,1], $y = 0$ and $y = 1$ respectively represent the user is detected as a real user and fraudster. The final prediction result is given by π_{py} (π_{p1} =P(y =1),π_{p0} =P(y =0)). For each tree g in the forest, the probability $P_{T_g}\left[y \mid E_n^*, \Theta, \pi\right]$ that the input is predicted to be classified as y is as follows:

$$\begin{aligned} &P_{T_g}\left[y \mid E_n^*, \Theta, \pi\right] \\ &\quad = \sum_{p \in P_g} \pi_{py} \left(\prod_{d \in D} Z_d\left(E_n^*; \Theta\right)^{T_\swarrow} \overline{Z_d}\left(E_n^*; \Theta\right)^{T_\searrow}\right) \end{aligned} \tag{10}$$

Among them, π_{py} represents the probability that the sample reaches the prediction node and is classified as y. Use the decision function $Z_d\left(E_n^*; \Theta\right)$ of the decision node d to decide whether to route the input to the left or the right, $\overline{Z_d}\left(E_n^*; \Theta\right)$ =1- $Z_d\left(E_n^*; \Theta\right)$. The explanation of the bracket part in formula (12) is: if p node is the left subtree of d node, then $T_\swarrow$ is true, that is,$T_\swarrow = 1$, if p node is the right subtree of d node, then $T_\searrow$ is true, that is, $T_\searrow$ =1. The expression of the decision function $Z_d\left(E_n^*; \Theta\right)$ is as follows:

$$Z_d\left(E_n^*; \Theta\right) = \sigma\left(w_d^T E_n^*\right) \tag{11}$$

In the formula, σ represents the sigmoid activation function, and the input E_n^* is assigned a weight w_d^T.

A forest is an ensemble of decision trees $F = \{T_1, T_2, \ldots, T_G\}$, which delivers a final prediction for a sample by averaging the output of each tree. Here we take its output as the probability value of fraudster (y =1):

$$P_T\left[y = 1 \mid E_n^*, \Theta, \pi\right] = \frac{1}{G} \sum_{g=1}^{G} P_{T_g}\left[y \mid E_n^*, \Theta, \pi\right] \tag{12}$$

Finally, we need to apply the output of the fraudster detection component to the rating prediction to generate the final prediction rating, then generate the final recommenda-tion list, thereby reflecting the recommendation prediction performance combined with the fraudster detection.

3) MODEL TRAINING

Since the ultimate goal of our proposed method is to make more accurate rating predictions for items in the environment with attack, and rating prediction task is a regres-sion problem, we utilize the square loss function to train our model, which is a kind of real number used to predict the label according to previous works [33], [36].

The sum of squared errors between the real value $R_{n,\ m}$ and the predicted value $\widehat{R_{n,\ m}}$ is used as the main section of the loss function, and the fraudster detection probability result $P_T\left[y=1 \mid E_n^*, \Theta, \pi\right]$ is added to the loss function to control the contribution of the fraudster to the rating prediction, thereby optimizing the rating prediction performance. If the probability of fraudster is high, RMPD can reduce the corresponding contribution of this suspicious user to the rating prediction task. Note that the observed ratings inevitably contain untruthful ratings. We define the loss function $Loss_{rating}$ of method layer for model is formulated as:

$$Loss_{rating} = \left|\frac{1}{NM}\right| \sum_{\forall n,\ m \in N, M} \left(1-P_T\left[y=1 \mid E_n^*, \Theta, \pi\right]\right)\left(R_{n,m}-\widehat{R_{n,\ m}}\right)^2 \tag{13}$$

In this method, there are two components that perform the task of rating prediction. The first part uses AVE-SVD in the input layer to predict the item rating, and the second part uses the predictive component and the detection component to generate the final rating prediction in the method layer. Two tasks are closely hinged with each other in our model, so that instead of training these two tasks separately, it is better to combine their losses and jointly minimize the following loss function $Loss_{RMPD}$:

$$Loss_{RMPD} = Loss'_{AVE-SVD} + \lambda' Loss_{rating} \tag{14}$$

where λ' is a tunable parameter controling the importance of each subtask.

IV. EXPERIMENTS

A. DATASETS

In our experiment, we evaluated the proposed RMPD on two real-world data sets: YelpZip dataset and Amazon dataset. The YelpZip dataset collected in [37] is a larger dataset than other Yelp datasets, which is a collection of user reviews and ratings of restaurants in consecutive regions of the United States starting from NY State (collect reviews for restaurants in that zipcode3).In the complete YelpZip data set, there are a total of 608,598 reviews/rating information, of which false reviews/rating information accounted for about 13.22%, fraudsters accounted for about 23.91% of the 260,277 users, and the number of items in the dataset was 5. 044. Here, we select a part of the YelpZip dataset, and based on Yelp's own filtering algorithm to filter and mark real and fake reviews, we consider users who post false reviews/ratings in the tags as fraudsters, accounting for about 30%, and the remaining real users account for about 70%. The data set Amazon is composed of user ratings and reviews on movies and TV series collected in [38], and it is used in [39] to vote users in the data set according to their comments related help (Ground truth is defined using Helpfulness votes) vote to mark real users and fraudsters. Part of the data is selected in the Amazon data set, of which about 30% are fraudsters, and the remaining 70% are fraudsters. Each rating is an integer between 1(the worst) to 5(the best) in two datasets. The description of the datasets we used is shown in Table 1.

TABLE 1. Statistical details of the two datasets.

Dataset	#Rating	#Users	#Items	Rating values
Yelp	293936	32393	4670	[1,5]with 1 an increment
Amazon	250423	12630	4746	[1,5]with 1 an increment

B. EXPERIMENTAL SETUP

1) HYPER-PARAMETER SETTINGS

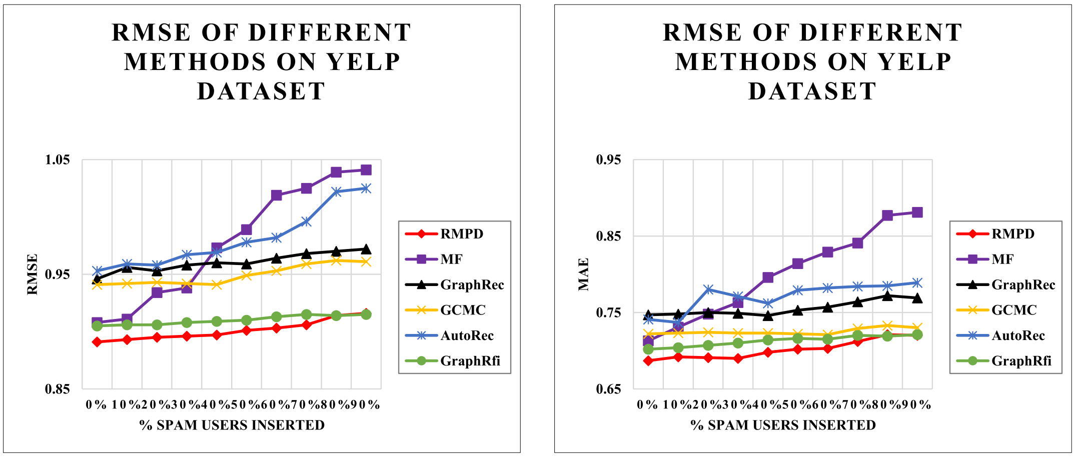
For each dataset, we used 70% as training set to learning parameters, 20% as validation set to tune hyper parameters, and 10% as testing set for the final performance comparison.

In our experiments, we adopt the Adam [40] to optimize our method and set 0.001 as the initial learning rate. The model parameters were initialized with a Gaussian distribution (The mean is 0, the standard deviation is 0.1) and the activation is function as ReLU. For the embedding size s, we tested the value of $\{50, 100, 150, 200, 250\}$. The λ value and the number of hidden layer in MLP were searched in $\{1, 3, 5, 7, 9\}$ and $\{1, 2, 3, 4, 5\}$, respectively. In addition, some parameters of the components in the model will be set based on previous work [31] and experience, here we set the depth of each tree to 3 and the ensemble size of decision trees to 5 about the NDF. In order to prevent overfitting in the model, we also explored the optimal value of dropout. The batch size for Yelp and Amazon datasets is set to 20.

2) BASELINES

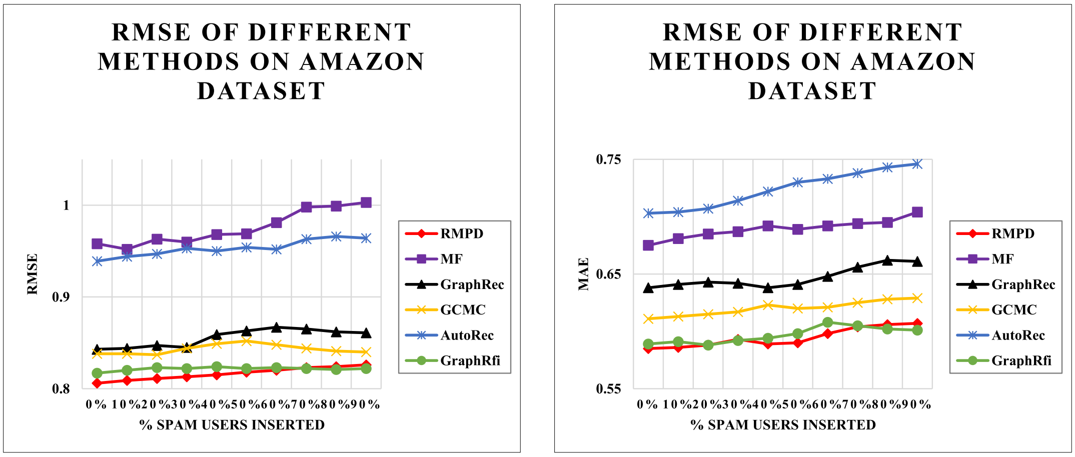
To evaluate the performance, we compared proposed RMPD with the following baselines for recommender systems. Note that not all baseline methods have fraudster detectors, but all methods have a certain degree of robustness. We will compare the robustness of these methods for rating prediction. For all baseline methods, we use the optimal hyper-parameters provided in the original papers.

- MF [34]: This is matrix factorization into the product of URM, which exploits the user-item direct interactions only as the target value of interaction function.
- GraphRec [41]: This method harnesses the power of graph neural networks (GNNs) techniques to model graph data in social recommendations by aggregating the both user-item interactions information and direct social neighbors.



(a) (b)

FIGURE 4. **(a) The RMSE w.r.t RMPD and various baseline methods with different % of spam users inserted on Yelp set; (b) The MAE w.r.t RMPD and various baseline methods with different % of spam users inserted on Yelp set.**



(a) (b)

FIGURE 5. **(a) The RMSE w.r.t RMPD and various baseline methods with different % of spam users inserted on Amazon set; (b) The MAE w.r.t RMPD and various baseline methods with different % of spam users inserted on Amazon set.**

- GCMC [42]: The framework of this method uses the weight distribution of each position in the user-item diagram, and generates the implicit features between the user and the item in the form of information transmission in the two interactive diagrams.
- AutoRec [19]: A collaborative filtering method based on AutoEncoder is proposed to solve the predictive rating method of the user-item rating matrix in the recommendation system.
- GraphRfi [31]: A user representation learning framework based on graph convolutional networks, which is mainly composed of two components: recommendation robustness and attack detection.

3) EVALUATION METRICE

In order to evaluate the recommendation performance of the model, two popular metrics are adopted to evaluate the predictive accuracy, it's Root Mean Square Error($RMSE$) and Mean Absolute Error(MAE), which are widely adopted in many related works for performance evaluation [39], denoted as:

$$RMSE = \sqrt{\frac{1}{nm}\sum_{i=1}^{nm}\left(R_{n,m} - \widehat{R_{n,m}}\right)} \tag{15}$$

$$MAE = \frac{1}{nm}\sum_{i=1}^{nm}\left|\left(R_{n,m} - \widehat{R_{n,m}}\right)\right| \tag{16}$$

C. PERFORMANCE EVALUATION

1) PERFORMANCE COMPARISON

In our experiment, we train all recommenders on our dataset, we investigate the robust performance of recommendations with the prediction accuracy, and gradually expand the percentage of inserted fraudsters from 0% to 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, we think it is unrealistic to have 100% spam fraudsters in the recommender system. Fig.4-Fig.5 show the RMSE value and MAE value of shilling attacks on the performance of all recommenders.

We compare the recommendation performance of all methods. Fig.4-Fig.5 show overall rating prediction error w.r.t. RSTM and MAE among the recommendation methods on Yelp and Amazon datasets, respectively. We have the following observations:

1) As the proportion of fraudsters increases, the performance of all methods decreases, which is in line with our expectations. Our model RMPD consistently outperforms all the baseline methods, then GraphRfi, these two are advanced models with attack detection components. The main reason can be understood as the environment created by this experiment that there are a number of fraudsters or spam, attack detection classifiers it plays a great role in the prediction robustness of the model, alleviating the recommendation bias caused by attacks, while other baselines only have cer-tain robustness to attack detection in terms of rating prediction. Taking the Yelp dataset as an example, RMPD improves over the strongest baseline GraphRfi by 1.1% and 1.2%, respectively. RMPD comperes the worst method MF in the baseline increase in average 7.1% and 9.2% respectively.
2) RMPD outperforms GraphRfi. In the rating prediction component, the GraphRfi method extends the GCN model [43], which combines node features and topological structure to model the interaction between users and items, and to predict the ratings. The topological network structure can relatively comprehensively capture user-item interactions. The rating prediction component in RMPD adopts a simpler Fully Con-volutional Networks (FCN) structure, through the multi-layer learning of the neural network and the segmentation of the user-item rating matrix for training and learning, respectively, and mining deeper feature representations to obtain better predictions recommended performance. In the Amazon dataset, when the proportion of fraudsters is greater than 60%, RMPD is slightly worse than the GraphRfi method. The reason can be explained as: the GCN model in the GraphRfi method is more sensitive to structural change information. The model will break its topology to get positive iterations. However, the deep neural network in RMPD is more sensitive to attribute information, and attacks can only imitate attribute characteristics. When there are too many injection attacks, real user attributes are easily assimilated by the fraudster attributes. Therefore, when there are too many fraudsters, the RMPD method is slightly less sensitive to fraudsters and fake ratings than GraphRfi.
3) MF and AutoRec achieve poor performance on two datasets. MF is a classic prediction method based on matrix factorization and it has fundamentally robust to prediction, and the results indicate that inner structure is insufficient to capture the complex relations between users and items, so the model's expressive ability is poor when facing spam. Taking the performance of RMSE on the Yelp dataset as example, as the spam noise increases, the performance of MF continues to decline, and the overall performance is poor. As an advanced combination model based on matrix decomposition prediction, RMPD not only uses a deep architecture, but also uses an attack detector to suppress the influence of the fraudster's configuration file on the recommendation, so its recommendation robustness is far better than MF. Although the AutoRec method can learn non-linear implicit representations compared to the classic MF method, this method is not ideal for the presence of data sparseness and the performance of the two datasets is unstable as spam increases.

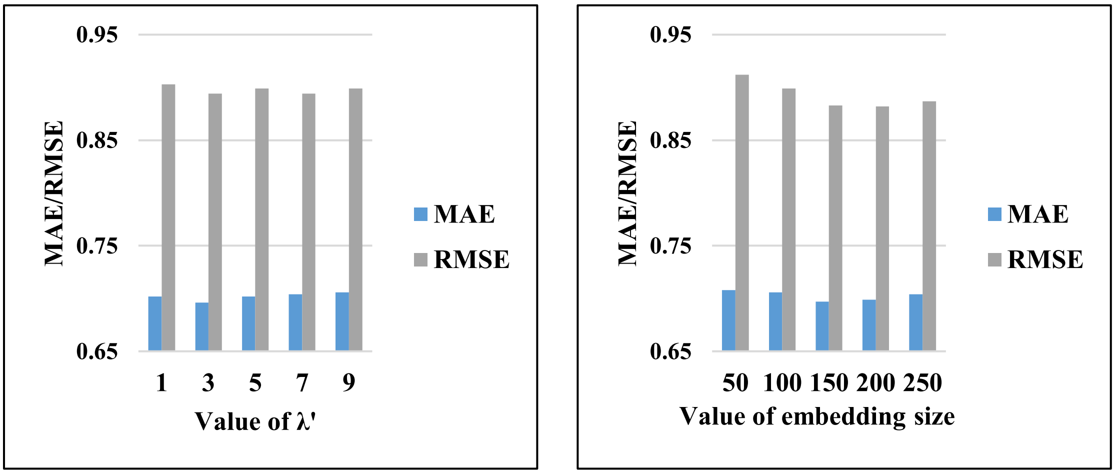
TABLE 2. The value of RMSE /MAE in different proportion of spam profiles inserted and different methods.

% ATTACK	MAE/RMSE	METHOD		
		RMPD	RMPD-ATT	RMP
0%	MAE	0.696	0.707	0.701
	RMSE	0.892	0.905	0.896
20%	MAE	0.691	0.731	0.729
	RMSE	0.895	0.926	0.928
40%	MAE	0.698	0.743	0.758
	RMSE	0.897	0.948	0.963
60%	MAE	0.707	0.751	0.792
	RMSE	0.905	0.972	0.998

2) ABLATION STUDY

In this part, we conduct ablation experiments to demonstrate the effectiveness of our RMPD model. Here we mainly verify the impact of the AVE-SVD rating prediction filling component and the fraudster detection component NDF on the final rating prediction recommendation performance. In this experiment, we change the RMPD to remove the NDF component of the fraudster detection of the original method, referred to as RMP, and the method variant that removes the AVE-SVD component is called RMPD-ATT. In the experiment, the percentage of fraudsters was continuously expanded in the form of random attacks in the Yelp dataset, and two evaluation indicators were used to compare the performance of the RMPD and other two variants of our model by removing different components. The model parameters in this experiment are consistent with those in the comparison experiment. The results are shown in Table 2.

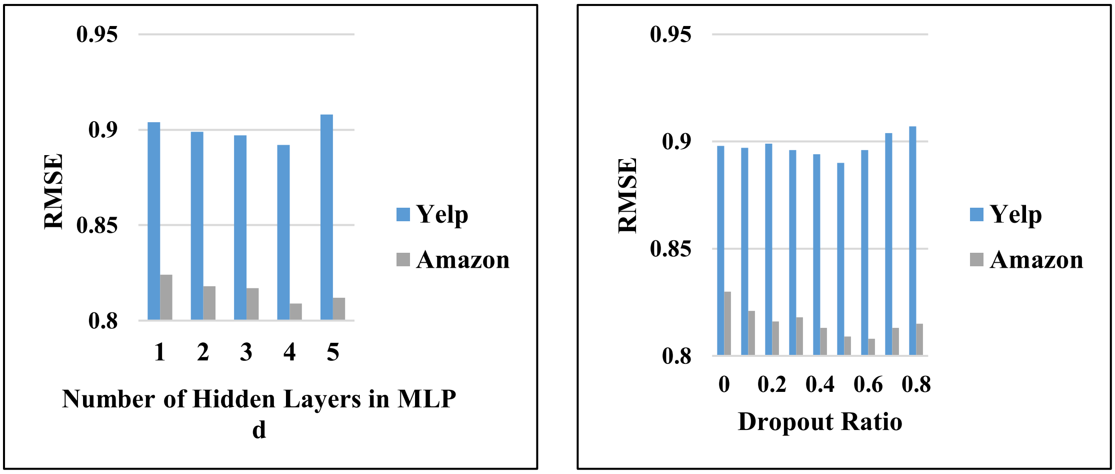
As shown in Table 2, the performance results of the proposed method and its variants are recorded when the proportion of fraudsters is 0%, 20%, 40%, and 60%. In the environment with attack, the method of removing the attack detection component degrades much faster than the RMPD recomendation performance. The ratings data of users are explicit interaction with items, and encoding the ratings by our multilayer neural network could learn deep representations of users and items from ratings. Even if the proportion of injected spam is low and the malicious ratings are less, it also has a negative impact on the deep characteristics of users and items of RMP. In contrast, the detection of fraudsters controls its proportion in the final rating prediction to improve



(a)

(b)

FIGURE 6. **(a) Performance w.r.t the different value of λ'; (b) Performance w.r.t the different value of embedding size.**



(a)

(b)

FIGURE 7. **(a) Performance w.r.t the different number of hidden layers in MLP d; (b) Performance w.r.t the different dropout ratio.**

the prediction robustness. The result indicates that combining the attack detection to our model can further improve the performance.

For the method variant RMPD-ATT that removes the AVE-SVD component, the performance of the method gradually decreases as the proportion of fraudsters increases, and the overall performance is poor compared to the RMPD we proposed. Due to the user-item own rating in the original dataset has sparseness, so after the sparsity is alleviated to some extent,the performance of the rating prediction component and the attack detection component in the method layer depends on the non-sparse rating matrix. RMPD-ATT outperforms RMP when attack more than 20%, which is not surprising. The noise generated by fraudsters on rating data has a more severe impact on the performance of rating prediction than the sparseness of the dataset itself.

Overall our complete model RMPD outperforms all the variant. This indicates that combining the fraudster detection component when attack exists can further improve the performance, and the alleviation of the information sparsity problem of the dataset itself can further improve the performance of RMPD, which all meets our exception.

3) PARAMETER ANALYSIS

In this section, we further study the effect of parameter settings on proposed RMPD performance,we conduct a set of experiments with varied parameters $\{\lambda', d, s, dropout\}$ on two dataset. Note that in this subsection, we investigate the effect of these parameters by examining how the performance changes when varying one parameter and modify others. The experimental results are shown in Fig.6-Fig.7.

Effect of the value of λ' and embedding size: Figure 6 shows the performance with the varied λ' and embedding size on Yelp dataset. We tune λ' in the range of [1,9] with a step of 2. We observe that the optimal performance is consistently achieved when λ' is around 3. We also observe that the performance becomes increasingly worse when $\lambda' \rightarrow 9$. Thus, we set $\lambda' = 3$ to achieve a balance between both rating prediction and attack detection though this may not be an optimal setting for some datasets. The embedding realizes the conversion of high-dimensional sparse feature vector to low-dimensional dense feature vector, and the value of embedding size is related to the generalization ability of the model, we choose 150 as the best embedding size according to the experimental results.

Effect of Dropout and number of hidden layers in MLP: Fig.7 displays the performance with varying value of dropout ratio and number of Hidden Layers in MLP. It is clear that performance of the RMPD first increases and then decreases with the continuous increase of d, we believe that when the number of hidden layers is too small, the model cannot better capture effective information from the rating, and when there are too many hidden layers, the model will appear overfitting. So we choose 4 as the number of hidden layers in MPL. We also report the performance of RMPD by tuning dropout ratio amongst [0,0.8] with a step of 0.1. We can see that the RMSE is the smallest when dropout is 0.5.

V. CONCLUSION AND FUTURE WORKS

The proposed RMPD model for enhancing the robustness of recommendations with attack environments. The key to our method is that we combine rating prediction process with attack detection, so that control the fraudster's contribution to rating information. Our extensive experimental results on two real world datasets demonstrate our method can effectively improve the robustness of recommendations with attack over the state-of-the-art baselines on both tasks. This findings provide the following insights for future research: we suggest adding a detection mechanism for review content in the attack detection component, because fake ratings of fraudsters are usually accompanied by fake reviews. Moreover, we will detect specific types of shilling attacks (e.g. random attack, av-erage attack, etc.) and use rich side information to refine and sublimate our methods. We will invesigate these possibilities in the future.

REFERENCES

[1] J. Gantz and D. Reinsel, "The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east," *IDC iView, IDC Analyze Future*, vol. 2007, pp. 1–16, Dec. 2012.

[2] Q. Zhou, J. Wu, and L. Duan, "Recommendation attack detection based on deep learning," *J. Inf. Secur. Appl.*, vol. 52, Jun. 2020, Art. no. 102493.

[3] Z. Fayyaz, M. Ebrahimian, D. Nawara, A. Ibrahim, and R. Kashef, "Recommendation systems: Algorithms, challenges, metrics, and business opportunities," *Appl. Sci.*, vol. 10, no. 21, p. 7748, Nov. 2020.

[4] C. Chen, M. Zhang, Y. Liu, and S. Ma, "Neural attentional rating regression with review-level explanations," in *Proc. World Wide Web Conf. World Wide Web (WWW)*, 2018, pp. 1583–1592.

[5] Y. Deldjoo, T. D. Noia, E. D. Sciascio, and F. A. Merra, "How dataset characteristics affect the robustness of collaborative recommendation models," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2020, pp. 951–960.

[6] M. Dong, L. Yao, X. Wang, B. Benatallah, C. Huang, and X. Ning, "Opinion fraud detection via neural autoencoder decision forest," 2018, *arXiv:1805.03379*. [Online]. Available: http://arxiv.org/abs/1805.03379

[7] S. K. Lam and J. Riedl, "Shilling recommender systems for fun and profit," in *Proc. 13th Conf. World Wide Web (WWW)*, 2004, pp. 393–402.

[8] B. Mobasher, R. Burke, R. Bhaumik, and C. Williams, "Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness," *ACM Trans. Internet Technol.*, vol. 7, no. 4, p. 23, Oct. 2007.

[9] M. O'Mahony, N. Hurley, N. Kushmerick, and G. Silvestre, "Collaborative recommendation: A robustness analysis," *ACM Trans. Internet Technol.*, vol. 4, no. 4, pp. 344–377, 2004.

[10] C. E. Seminario and D. C. Wilson, "Nuke 'Em till they go: Investigating power user attacks to disparage items in collaborative recommenders," in *Proc. 9th ACM Conf. Recommender Syst.*, Sep. 2015, pp. 293–296.

[11] B. Mehta and W. Nejdl, "Unsupervised strategies for shilling detection and robust collaborative filtering," *User Model. User-Adapted Interact.*, vol. 19, nos. 1–2, pp. 65–97, Feb. 2009.

[12] Y. Chen, J. Ma, P. Zhang, F. Liu, and S. Mei, "Robust state estimator based on maximum exponential absolute value," *IEEE Trans. Smart Grid*, vol. 8, no. 4, pp. 1537–1544, Jul. 2017.

[13] Y. Chen, Y. Yao, and Y. Zhang, "A robust state estimation method based on SOCP for integrated electricity-heat system," *IEEE Trans. Smart Grid*, vol. 12, no. 1, pp. 810–820, Jan. 2021.

[14] Y. Chen, F. Liu, S. Mei, and J. Ma, "A robust WLAV state estimation using optimal transformations," *IEEE Trans. Power Syst.*, vol. 30, no. 4, pp. 2190–2191, Jul. 2015.

[15] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[16] H. Chen, H. Yin, T. Chen, Q. V. H. Nguyen, W.-C. Peng, and X. Li, "Exploiting centrality information with graph convolutions for network representation learning," in *Proc. IEEE 35th Int. Conf. Data Eng. (ICDE)*, Apr. 2019, pp. 590–601.

[17] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "LightGCN: Simplifying and powering graph convolution network for recommendation," 2020, *arXiv:2002.02126*. [Online]. Available: http://arxiv.org/abs/2002.02126

[18] Y.-X. Peng, W.-W. Zhu, Y. Zhao, C.-S. Xu, Q.-M. Huang, H.-Q. Lu, Q.-H. Zheng, T.-J. Huang, and W. Gao, "Cross-media analysis and reasoning: Advances and directions," *Frontiers Inf. Technol. Electron. Eng.*, vol. 18, no. 1, pp. 44–57, Jan. 2017.

[19] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie, "AutoRec: Autoencoders meet collaborative filtering," in *Proc. 24th Int. Conf. World Wide Web*, May 2015, pp. 111–112.

[20] M. Grbovic, V. Radosavljevic, N. Djuric, N. Bhamidipati, J. Savla, V. Bhagwan, and D. Sharp, "E-commerce in your inbox: Product recommendations at scale," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2015, pp. 1809–1818.

[21] C.-Y. Wu, A. Ahmed, A. Beutel, A. J. Smola, and H. Jing, "Recurrent recommender networks," in *Proc. 10th ACM Int. Conf. Web Search Data Mining*, Feb. 2017, pp. 495–503.

[22] A. Rathod and M. Indiramma, "A survey of personalized recommendation system with user interest in social network," *Int. J. Comput. Sci. Inf. Technol.*, vol. 6, no. 1, pp. 413–415, 2015.

[23] Z. Sun, L. Han, W. Huang, X. Wang, X. Zeng, M. Wang, and H. Yan, "Recommender systems based on social networks," *J. Syst. Softw.*, vol. 99, pp. 109–119, Jan. 2015.

[24] W. Fan, Y. Ma, D. Yin, J. Wang, J. Tang, and Q. Li, "Deep social collaborative filtering," in *Proc. 13th ACM Conf. Recommender Syst.*, Sep. 2019, pp. 305–313.

[25] B. Berjani and T. Strufe, "A recommendation system for spots in location-based online social networks," in *Proc. 4th Workshop Social Netw. Syst. (SNS)*, 2011, pp. 1–6.

[26] B. Mehta, T. Hofmann, and W. Nejdl, "Robust collaborative filtering," in *Proc. ACM Conf. Recommender Syst. (RecSys)*, 2007, pp. 49–56.

[27] Y. Ma, X. Zhu, and J. Hsu, "Data poisoning against differentially-private learners: Attacks and defenses," 2019, *arXiv:1903.09860*. [Online]. Available: http://arxiv.org/abs/1903.09860

[28] P.-A. Chirita, W. Nejdl, and C. Zamfir, "Preventing shilling attacks in online recommender systems," in *Proc. 7th ACM Int. Workshop Web Inf. Data Manage. (WIDM)*, 2005, pp. 67–74.

[29] F. Zhang, Z. Zhang, P. Zhang, and S. Wang, "UD-HMM: An unsupervised method for shilling attack detection based on hidden Markov model and hierarchical clustering," *Knowl.-Based Syst.*, vol. 148, pp. 146–166, May 2018.

[30] C. Tong, X. Yin, J. Li, T. Zhu, R. Lv, L. Sun, and J. J. P. C. Rodrigues, "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, Jul. 2018.

[31] S. Zhang, H. Yin, T. Chen, Q. V. N. Hung, Z. Huang, and L. Cui, "GCN-based user representation learning for unifying robust recommendation and fraudster detection," 2020, *arXiv:2005.10150*. [Online]. Available: http://arxiv.org/abs/2005.10150

[32] J. Lian, X. Zhou, F. Zhang, Z. Chen, X. Xie, and G. Sun, ''XDeepFM: Combining explicit and implicit feature interactions for recommender systems,'' in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 1754–1763.

[33] H. Liu, Y. Wang, Q. Peng, F. Wu, L. Gan, L. Pan, and P. Jiao, ''Hybrid neural recommendation with joint deep representation learning of ratings and reviews,'' *Neurocomputing*, vol. 374, pp. 77–85, Jan. 2020.

[34] Y. Koren, R. Bell, and C. Volinsky, ''Matrix factorization techniques for recommender systems,'' *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009.

[35] P. Kontschieder, M. Fiterau, A. Criminisi, and S. R. Bulo, ''Deep neural decision forests,'' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1467–1475.

[36] L. Zheng, V. Noroozi, and P. S. Yu, ''Joint deep modeling of users and items using reviews for recommendation,'' in *Proc. 10th ACM Int. Conf. Web Search Data Mining*, Feb. 2017, pp. 425–434.

[37] S. Rayana and L. Akoglu, ''Collective opinion spam detection: Bridging review networks and metadata,'' in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2015, pp. 985–994.

[38] J. J. McAuley and J. Leskovec, ''From amateurs to connoisseurs: Modeling the evolution of user expertise through online reviews,'' in *Proc. 22nd Int. Conf. World Wide Web (WWW)*, 2013, pp. 897–908.

[39] S. Kumar, B. Hooi, D. Makhija, M. Kumar, C. Faloutsos, and V. S. Subrahmanian, ''REV2: Fraudulent user prediction in rating platforms,'' in *Proc. 11th ACM Int. Conf. Web Search Data Mining*, Feb. 2018, pp. 333–341.

[40] D. P. Kingma and J. Ba, ''Adam: A method for stochastic optimization,'' 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

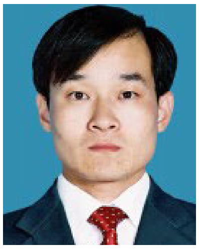
[41] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin, ''Graph neural networks for social recommendation,'' in *Proc. World Wide Web Conf. (WWW)*, 2019, pp. 417–426.

[42] R. van den Berg, T. N. Kipf, and M. Welling, ''Graph convolutional matrix completion,'' 2017, *arXiv:1706.02263*. [Online]. Available: http://arxiv.org/abs/1706.02263

[43] M. Defferrard, X. Bresson, and P. Vandergheynst, ''Convolutional neural networks on graphs with fast localized spectral filtering,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 3844–3852.

QI DING was born in 1997. She is currently pursuing the master's degree with the College of Computer Science and Engineering, Shandong Normal University, China. Her research interests include recommender system, data mining, and robustness research.

PEIYU LIU received the master's degree from East China Normal University, in 1986. He is currently a Second-level Professor, Doctoral Supervisor, with the School of Information Science and Engineering, Shandong Normal University, China. He is the national outstanding science and technology worker, middle-aged and young expert with outstanding contribution in Shandong province, and famous teacher of Shandong province. His research interests include network information security, information retrieval, natural language processing, and artificial intelligence.



ZHENFANG ZHU received the Ph.D. degree from Shandong Normal University, in 2012. He was a Postdoctoral Fellow with Shandong University, from 2012 to 2015. He is currently an Associate Professor and a Master's Supervisor with the School of Information Science and Electrical Engineering, Shandong Jiao Tong University, China. His research interests include network information security, natural language processing, and applied linguistics.

HUAJUAN DUAN was born in 1997. She is currently pursuing the Ph.D. degree with the College of Computer Science and Engineering, Shandong Normal University, China. Her research interests are deep learning and personalized recommendation.

FUYONG XU was born in 1997. He is currently pursuing the master's degree with the College of Computer Science, Shandong Normal University, China. His research focus on dialogue generation, dialogue systems, and sentiment analysis.