Quick and Accurate Attack Detection in Recommender Systems through User Attributes

Mehmet Aktukmak
University of South Florida
Tampa, FL
maktukmak@mail.usf.edu

Yasin Yilmaz
University of South Florida
Tampa, FL
yasiny@usf.edu

Ismail Uysal
University of South Florida
Tampa, FL
iuysal@usf.edu

ABSTRACT

Malicious profiles have been a credible threat to collaborative recommender systems. Attackers provide fake item ratings to systematically manipulate the platform. Attack detection algorithms can identify and remove such users by observing rating distributions. In this study, we aim to use the user attributes as an additional information source to improve the accuracy and speed of attack detection. We propose a probabilistic factorization model which can embed mixed data type user attributes and observed ratings into a latent space to generate anomaly statistics for new users. To identify the persistent outliers in the system, we also propose a sequential attack detection algorithm to enable quick and accurate detection based on the probabilistic model learned from genuine users. The proposed model demonstrates significant improvements in both accuracy and speed when compared to baseline algorithms on a popular benchmark dataset.

CCS CONCEPTS

• **Security and privacy → Intrusion/anomaly detection**.

KEYWORDS

Sequential attack detection; probabilistic matrix factorization; attribute embedding

ACM Reference Format:
Mehmet Aktukmak, Yasin Yilmaz, and Ismail Uysal. 2019. Quick and Accurate Attack Detection in Recommender Systems through User Attributes. In *Thirteenth ACM Conference on Recommender Systems (RecSys '19), September 16–20, 2019, Copenhagen, Denmark.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3298689.3347050

1 INTRODUCTION

Recommender systems are important building blocks of successful and popular commercial systems, which gave rise to malicious motivations for some participants, called the attackers, to manipulate the outcomes. The main purposes for such activities could be to promote an item by pushing its ratings, to nuke a rival item with malicious feedback or just to disrupt the platform to reduce its efficiency. These activities are considered mostly profit oriented and mitigated by attack detection algorithms.

In collaborative filtering models, the interactions of target users and their neighbors are exploited to make better recommendations. To this end, in a trained model including many genuine users, items and interactions, it is assumed that the neighbors share similar preferences. Most of the attack detection models leverage this similarity to detect attack profiles by processing the rating distributions to form distinctive features [1, 20]. However, some newly registered users may exhibit different preferences, which make them prone to false alarms. As it was in the case of improving cold-start recommendation performance of collaborative filtering models when not enough user interaction has yet been entered, user attributes can provide an additional source of information to improve attack detection performance. Since most of the systems require entering basic demographic data for each user during registration, this information is available only to the system operator while being inaccessible to outside attackers. In this study, we aim to exploit this observation by utilizing user attributes as an additional information source. The assumption, in this case, is that the users who have similar attributes will have similar preferences. For an attacker, it requires in-depth knowledge to obtain this affinity of genuine user profiles and ratings. Hence, it can be assumed that the attacker will choose random attributes for newly created fake profiles, which can be used as another statistical metric to detect anomalous behavior, in addition to the anomalies in the ratings.

For an effective attack, the attackers register many fake profiles. In this study, we assume such fake profiles are registered sequentially in a short time period to obtain effective results. In this scenario, the fake user profiles generally appear as *persistent outliers*, which we call an anomaly, under the model trained with genuine users. With this definition of temporal anomaly, the genuine users, who might have more diverse preferences than the general population in the system, are no longer detected as attack profiles, i.e., such false alarms are avoided, due to their non-persistent (i.e., low frequency) registration rate. Our objective is to detect the attacks as soon as possible while controlling the false alarm rate.

In this study, our motivation is to exploit user attributes as an additional source of evidence for sequential fake profile detection to quickly and accurately raise a system-wide alarm and to detect anomalous profiles and mitigate the attack by dropping the ratings entered by them. The main contributions of this paper are two-fold:

i) A probabilistic matrix factorization model that effectively embeds observed ratings and mixed data type attributes of the genuine users into a low dimensional latent manifold and provides decision statistics from both ratings and attributes and;

RecSys '19, September 16–20, 2019, Copenhagen, Denmark
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6243-6/19/09…$15.00
https://doi.org/10.1145/3298689.3347050

ii) A sequential detection framework that uses statistics produced under this multimodal matrix factorization model to quickly and accurately detect fake profiles.

2 RELATED WORK

Many unsupervised attack detection algorithms have been proposed in the literature. Most of them use rating statistics to distinguish attack profiles considering them as individuals or groups of users. An early method [19] explored clustering the neighborhood into two clusters during a particular item's prediction and filtered the cluster based on mean and standard deviations. In [9], rating deviation from mean agreement (RDMA) was introduced and used as evidence of attacking along with similarity to the closest neighbors. UnRAP [7] algorithm used sum of squared deviations from the user, item and global means as a statistic, which was called the H_v score, to classify the users with a sliding window method to refine the results. In [17], highly correlated users were defined as the potential attack users and PCA of the user covariance matrix was exploited to identify the attackers by choosing a few principal components associated with the smallest eigenvalues. In [11], a statistical test called Neyman-Pearson (NP), based on the likelihood ratio test, was proposed to detect random, average and bandwagon attacks. The probability that a new user is an attack profile was determined based on the overlap of the item selections of the new user and the genuine users in the training set. In [3], detection attributes were combined to form discriminatory features by using previously proposed metrics including RDMA, Weighted Degree of Agreement (WDA), Weighted Degree of Mean Agreement (WDMA), Length Variance [8] and H_v score. Afterward, k-means clustering was applied to these features to identify attack clusters with small standard deviations. In [14], a latent variable model was proposed. The inferred latent variables were used to distinguish the item selection strategy of genuine and attack users by identifying the type that maximizes the entropy of the rating distribution. In [13], a multidimensional scaling approach was adopted to identify distinct behaviors. Subsequently, clustering was performed to discriminate attack users by assuming the attack profiles were at the center of the genuine users' distributions. In [5], hierarchical clustering was used by identifying the cluster with the highest average similarity weight as the attack cluster. Some recent studies introduced exploiting the graph structures. A graph-based method was employed to find a maximum submatrix in the similarity matrix by finding the largest component that corresponds to the most highly correlated group in the graph [24, 25]. In [22], an undirected user-user graph was constructed and the similarity between vertices was learned by using the graph mining method. By analyzing the similarity and target items, attack users were identified. In [23], the Hidden Markov Model was used to model user's rating behaviors with hierarchical clustering to group users according to the suspicion degree obtained from the model.

From the anomaly detection point of view, [4] proposed a detector that observes the mean ratings of each item in predefined time intervals. Mean detector was also used within the sequential detection framework to detect dishonest ratings in reputation systems [16]. In this work, the cumulative sum (CUSUM) algorithm was used to detect the mean changes in observed ratings for each item. Recently, a better approach than the mean detector was proposed in [15]. In this work, the ratings were considered as categorically distributed and a generalized likelihood ratio (GLR) based detector was proposed, which performed better than the mean detector.

In this paper, we show that the attack detection performance can be improved in terms of (i) quick and accurate detection of system-wide attacks compared to the GLR detector [15] by exploiting the additional evidence provided by the user attributes (see Fig. 2), and (ii) detection of attack profiles compared to several existing baseline detectors (see Fig. 1).

3 METHODOLOGY

3.1 Generative Factor Model

We design a probabilistic generative model for observed data that consists of a sparse rating matrix and mixed data type user attributes. The latent variables are assumed to underlie the observed variables within a linear framework. Let's assume we have M real and N categorical valued attributes for each user. It is convenient to model the mth real-valued attribute $\boldsymbol{x}_{i,m}$ of user i, where $m \in \{1, ..., M\}$ and $i \in \{1, ..., I\}$, as a Gaussian random variable with the following conditional distribution given user latent variable $\boldsymbol{u}_i$:

$$p(\boldsymbol{x}_{i,m}|\boldsymbol{u}_i) = \mathcal{N}(\boldsymbol{x}_{i,m}|\boldsymbol{W}_m\boldsymbol{u}_i, \Sigma_{\boldsymbol{x},m}), \tag{1}$$

where $\boldsymbol{W}_m$ is the weight matrix associated with the mth observed real-valued attribute and $\Sigma_{\boldsymbol{x},m}$ is the covariance matrix. The data is assumed to be centered. Distinctively, the categorical valued attributes are modeled using the following conditional distribution:

$$p(\boldsymbol{y}_{i,n}|\boldsymbol{u}_i) = Cat(\boldsymbol{y}_{i,n}|\mathcal{S}(\boldsymbol{H}_n\boldsymbol{u}_i)), \tag{2}$$

where $\boldsymbol{H}_n$ is the weight matrix associated with the nth observed categorical valued variable and $\mathcal{S}$ is the softmax function. The ordinal ratings are modeled as Gaussian due to its simplicity and extensive use,

$$p(r_{ij}|\boldsymbol{u}_i, \boldsymbol{v}_j) = \mathcal{N}(r_{ij}|\boldsymbol{u}_i^T\boldsymbol{v}_j, c^{-1}), \tag{3}$$

where c corresponds to the rating confidence parameter and $\boldsymbol{v}_j$ is the latent variable of item j where $j \in \{1, ..., J\}$. Combining the aforementioned conditional distributions of attributes and zero mean spherical prior distributions of latent variables, we obtain the following model likelihood for user i and item j, that should be maximized during model training,

$$L_{ij} = p(\boldsymbol{u}_i)p(\boldsymbol{v}_j)p(r_{ij}|\boldsymbol{u}_i, \boldsymbol{v}_j)\prod_{m=1}^{M} p(\boldsymbol{x}_{i,m}|\boldsymbol{u}_i)\prod_{n=1}^{N} p(\boldsymbol{y}_{i,m}|\boldsymbol{u}_i). \tag{4}$$

3.2 Inference and Parameter Estimation

The model defined in Section 3.1 has latent variables $\boldsymbol{u}_i$ for each user i and $\boldsymbol{v}_j$ for each item j, whose distributions are to be inferred. The model also has the parameter set $\Theta = \{\boldsymbol{W}_m, \boldsymbol{H}_n, \Sigma_{\boldsymbol{x},m}\}$ for each attribute m and n, that should be optimized given the observed data. Expectation Maximization (EM) is a convenient algorithm to train latent variable models. However, due to the categorical likelihoods, exact inference of posterior distributions is not tractable. In order to solve this problem, a variational inference method is used for fast and accurate approximation. Following [12, 18] by using the local variational Bohning bound [6] to the log-sum-exp (lse) term, categorical log-likelihoods are approximated using the following quadratic form:

$$\log p(\boldsymbol{y}_{i,n}|\boldsymbol{u}_i) \geq \boldsymbol{y}_{i,n}^T \boldsymbol{H}_m \boldsymbol{u}_i - \frac{1}{2}\boldsymbol{u}_i^T \boldsymbol{H}_m^T \boldsymbol{F}_{u,n} \boldsymbol{H}_m \boldsymbol{u}_i + \boldsymbol{g}_{i,n}^T \boldsymbol{H}_m \boldsymbol{u}_i - \boldsymbol{e}_{i,n}. \tag{5}$$

Intermediate parameters $\boldsymbol{F}_{\boldsymbol{u},n}$, $\boldsymbol{g}_{i,n}$ and $\boldsymbol{e}_{i,n}$ are given as

$$\boldsymbol{F}_{\boldsymbol{u},n} = \frac{1}{2}(\boldsymbol{I}_{M_{u,n}} - \frac{1}{M_{u,n}+1}\boldsymbol{1}_{M_{u,n}}\boldsymbol{1}_{M_{u,n}}^T), \tag{6}$$

$$\boldsymbol{g}_{i,n} = \boldsymbol{F}_{\boldsymbol{u},n}\boldsymbol{\psi}_{i,n} - \mathcal{S}(\boldsymbol{\psi}_{i,n}), \tag{7}$$

$$\boldsymbol{e}_{i,n} = \frac{1}{2}\boldsymbol{\psi}_{i,n}^T \boldsymbol{F}_{\boldsymbol{u},n}\boldsymbol{\psi}_{i,n} - \mathcal{S}(\boldsymbol{\psi}_{i,n})^T \boldsymbol{\psi}_{i,n} + \mathrm{lse}(\boldsymbol{\psi}_{i,n}), \tag{8}$$

where $\boldsymbol{\psi}_{i,n}$ is the free variational parameter around which the likelihood is approximated. This parameter is updated as $\boldsymbol{\psi}_{i,n} = \boldsymbol{H}_n \boldsymbol{m}_{\boldsymbol{u},i}$ (where $\boldsymbol{m}_{\boldsymbol{u},i}$ is the mean of $\boldsymbol{u}_i$) until convergence at each iteration in E-step to form a tight bound. Consequently, the posterior distributions of latent variables are approximated using the Gaussian distributions $q(\boldsymbol{u}_i) = \mathcal{N}(\boldsymbol{u}_i|\boldsymbol{m}_{\boldsymbol{u},i}, \Sigma_{\boldsymbol{u},i})$ and $q(\boldsymbol{v}_j) = \mathcal{N}(\boldsymbol{v}_j|\boldsymbol{m}_{\boldsymbol{v},j}, \Sigma_{\boldsymbol{v},j})$. E-step updates for user i can be derived as in [18] as follows:

$$\Sigma_{\boldsymbol{u},i} = (\lambda_u \boldsymbol{I}_K + \sum_{n=1}^{N} \boldsymbol{H}_n^T \boldsymbol{F}_{\boldsymbol{u},n} \boldsymbol{H}_n + \sum_{m=1}^{M} \boldsymbol{W}_m^T \Sigma_{\boldsymbol{x},m}^{-1} \boldsymbol{W}_m + c(E[\boldsymbol{V}\boldsymbol{O}_i\boldsymbol{V}^T]))^{-1}, \tag{9}$$

$$\boldsymbol{m}_{\boldsymbol{u},i} = \Sigma_{\boldsymbol{u},i}(c(E[V]\boldsymbol{O}_i\boldsymbol{r}_i) + \sum_{n=1}^{N} \boldsymbol{H}_n^T(\boldsymbol{y}_{i,n} + \boldsymbol{g}_{i,n}) + \sum_{m=1}^{M} \boldsymbol{W}_m^T \Sigma_{\boldsymbol{x},m}^{-1} \boldsymbol{x}_{i,m}), \tag{10}$$

where the term $\lambda_u \boldsymbol{I}_K$ comes from the zero mean spherical prior given for each latent variable. $\boldsymbol{V}$ is the matrix of item latent variables ordered as $\boldsymbol{V} = [\boldsymbol{v}_1, ..., \boldsymbol{v}_J]$. $\boldsymbol{O}_i$ is a diagonal matrix whose elements are binary indicators of observed ratings for each item and r_i is the observed rating vector of user i. The updates for $q(\boldsymbol{v}_j)$ are performed analogously excluding the side information terms. In M-step, model parameters are updated using the maximum likelihood solutions. Similar to the factor analysis models [10], the updates are given as follows:

$$\boldsymbol{H}_n = \Big[\sum_i \boldsymbol{F}_{\boldsymbol{u},n}^{-1}(\boldsymbol{y}_{i,n} + \boldsymbol{g}_{i,n})E[\boldsymbol{u}_i]^T\Big]\Big[\sum_i E[\boldsymbol{u}_i\boldsymbol{u}_i^T]\Big]^{-1}, \tag{11}$$

$$\boldsymbol{W}_m = \Big[\sum_i \boldsymbol{x}_{i,m}E[\boldsymbol{u}_i]^T\Big]\Big[\sum_i E[\boldsymbol{u}_i\boldsymbol{u}_i^T]\Big]^{-1}, \tag{12}$$

$$\Sigma_{\boldsymbol{x},m} = \mathrm{diag}\Big\{\frac{1}{I}\sum_i \boldsymbol{x}_{i,m}\boldsymbol{x}_{i,m}^T - \boldsymbol{x}_{i,m}E[\boldsymbol{u}_i]^T\boldsymbol{W}_m^T\Big\}, \tag{13}$$

where $E[\boldsymbol{u}_i] = \boldsymbol{m}_{\boldsymbol{u},i}$ and $E[\boldsymbol{u}_i\boldsymbol{u}_i^T] = \Sigma_{\boldsymbol{u},i} + \boldsymbol{m}_{\boldsymbol{u},i}\boldsymbol{m}_{\boldsymbol{u},i}^T$. EM steps are repeated until convergence of model parameters. After convergence, the parameter set Θ of the factor model provides a rich latent space for genuine users which is used in the proposed sequential detection algorithm.

3.3 Sequential Attack Detection

The factor model consists of latent affinities of the users toward the items. Since the model is trained using both ratings and attributes of the users, it groups the users with similar ratings, as well as similar attributes in the latent space. We leverage this property of the model to detect anomalies.

Let us assume a new user t is observed with complete attributes and a sparse rating vector. First, we use the attributes of the new user to project it to the latent space under the trained model parameters. This corresponds to inferring the mean of latent variables of that user given the model parameters and attributes:

$$\boldsymbol{m}_{\boldsymbol{u},t} = p(\boldsymbol{u}_t|\boldsymbol{x}_{t,1}, ..., \boldsymbol{x}_{t,m}, \boldsymbol{y}_{t,1}, ..., \boldsymbol{y}_{t,n}, \Theta, q(\boldsymbol{V})). \tag{14}$$

This can be easily evaluated since it corresponds to the E-step of the model and has a closed form expression given in Eq. (10). After inferring $\boldsymbol{m}_{\boldsymbol{u},t}$, we evaluate the likelihood of ratings for each item as:

$$p(r_{tj}|\boldsymbol{u}_t, \boldsymbol{v}_j) = \mathcal{N}(r_{tj}|\boldsymbol{m}_{\boldsymbol{u},t}^T \boldsymbol{m}_{\boldsymbol{v},j}, c^{-1}). \tag{15}$$

For a genuine user, it is expected that the rating likelihoods based on the user profile would be higher than those of an attack user due to a mismatch between the randomly generated user profile and the ratings of attack user. Motivated by this assumption, we evaluate an anomaly score for each user t as follows:

$$d_t = \frac{1}{|\Omega_t|}\sum_{j\in\Omega_t} -\log p(r_{tj}|\boldsymbol{u}_t, \boldsymbol{v}_j), \tag{16}$$

where Ω_t is the set of items user t has a rating for. In order to statistically compare d_t with the scores for genuine users, we compute $\{d_i\}$ offline for all users in the training set using Eq. (16). In the online testing for a new user t, the tail probability p_t is evaluated by calculating the fraction of genuine users whose scores are higher than d_t, i.e., $p_t = \frac{1}{I}\sum_{i=1}^{I} 1_{\{d_i \geq d_t\}}$, where $1_{\{\cdot\}}$ is the indicator function. Instead of deciding user t is attacking if $p_t \leq \alpha$ (p-value approach), where α is a significance level, we apply a sequential test to make that decision by accumulating an anomaly statistic over time since the p-value approach, and in general the single-instance outlier detection techniques, are known to be prone to frequent false alarms [21]. Specifically, by computing the anomaly statistic for user t

$$s_t = \log\frac{\alpha}{p_t}, \tag{17}$$

and updating the cumulative anomaly statistic

$$g_t = \max\{g_{t-1} + s_t, 0\},\ g_0 = 0, \tag{18}$$

we avoid raising an alarm for a non-persistent (nominal) outlier, and raise an alarm at time

$$t_d = \min\{t : g_t \geq h\} \tag{19}$$

only when persistent outliers result in a sufficiently large cumulative anomaly statistic. The sequential detector proposed in (17)-(19) is a CUSUM-like test, which is minimax-optimum in terms of minimizing the average detection delay for a given false alarm constraint when both the nominal and anomalous distributions are completely known [2]. After the system-wide alarm is set at time t_d, the beginning and end of the attack are determined as the times $t_b = \max\{t < t_d : g_t = 0\}$ and $t_e = \min\{t > t_d : g_t = 0\}$ when g_t gets its last/first zero value before/after the detection time, respectively. The malicious users are then detected by examining the anomaly statistics: identify user t as attacker if $s_t \geq 0$ for $t \in (t_b, t_e)$.

4 EXPERIMENTS

4.1 Setup

Dataset: The experiments are performed on the Movielens 100K dataset, which is a popular benchmark dataset in attack detection studies. The dataset includes 943 users and 1682 items and provides mixed data type attributes for each user and item. The user attributes that are included in this study are age (real-valued), occupation (categorical) and gender (categorical). The data sparsity of the dataset is around 0.94.

Attack: The ratings of the attack users are generated by mixing three well-studied attack types; random, average and bandwagon. The mixing is performed randomly at each experiment with even probabilities. The attack size is selected as 10%, which corresponds to 94 user profiles. The filler size is selected according to the overall data density, which is 5%, corresponding to 85 items. The attributes of attack users are generated as follows: for each attribute, a random genuine user is selected and her corresponding attribute is copied. This procedure results in a random but realistic attribute selection for attack users. To form the sequential attack, 100 genuine users are randomly selected and held out from the training set. 5 of them are then randomly selected and distributed to random locations among the attack users. Finally, a sequence of the remaining genuine users followed by the attack users constitute the test set, which is called the mixed sequential attack.

4.2 Detection Accuracy

Algorithms: Four baseline algorithms NP [11], PCA [17], RDMA [9] and UnRAP [7] are selected to assess the detection accuracy of the proposed algorithm. For PCA, the mean and standard deviation of each user are evaluated by imputing the missing values with 0 following [11], and the number of principal components are selected as 3. For UnRAP, r is selected as 10 [7].

Metric: ROC curves are evaluated to compare the performances. For each experiment, a mixed sequential attack was generated randomly and true-false positive rates were evaluated for each algorithm. The results in Figure 1 are obtained by conducting 50 experiments for robust average performance.

Results: Figure 1 demonstrates that the proposed algorithm has a better ROC curve compared to the baseline algorithms suggesting that the user attributes are utilized as an additional information source for detecting attacks. The intractability of designing attack user profiles aligned with the produced ratings, in general, causes a mismatch between the profile and the ratings, which provides extra statistical anomaly evidence. The proposed detector can then draw upon this extra evidence to improve the detection performance against the compared methods.

4.3 Detection Speed

Algorithms: We compare the proposed detection algorithm with the GLR detector defined in [15]. In this model, for each item in the system, the changes on the parameters of the categorical distributions of ratings are observed and an alarm is set when the decision statistic of any item exceeds a certain threshold.

Metric: The objective of sequential detectors is to minimize the mean detection delay (MDD) while controlling the false alarm rate [2]. Hence, we plot the mean detection delay of algorithms against the logarithm of their false alarm periods. The decision threshold (see h in Eq. (19)) is changed to obtain different trade-offs between the mean detection delay and the false alarm period. Similarly, 50 experiments are performed to find a robust average performance.

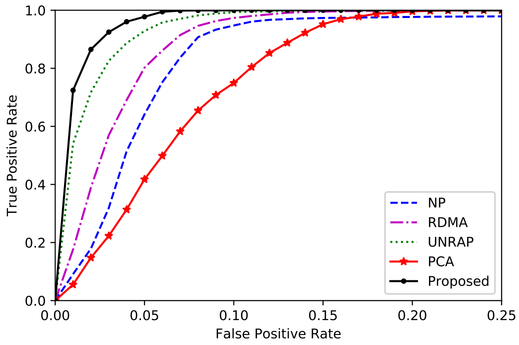

Figure 1: Comparison of ROC curves for detection accuracy

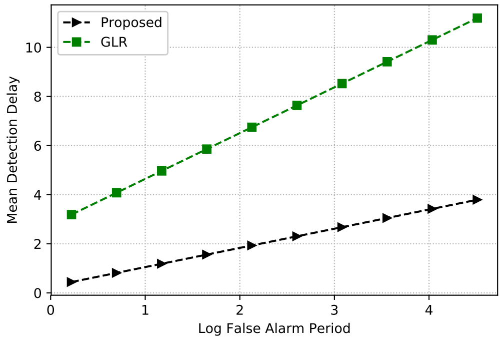

Figure 2: Comparison of MDDs for detection speed

Results: Figure 2 shows that the proposed algorithm achieves quicker detection for mixed sequential attack than the GLR detector. Since the GLR detector uses only the ratings in the system, the results indicate that the proposed detector is able to exploit the user attributes as an additional information source to increase its detection performance.

5 CONCLUSION

We proposed an algorithm that exploits user attributes in a probabilistic model to detect sequential attacks on recommender systems. We developed a probabilistic generative model to embed mixed-data type user attributes along with ratings into a low dimensional latent space by optimizing the model parameters via EM algorithm. New users are projected into the latent space learned from training with genuine user attributes and ratings, and an anomaly statistic is computed in a sequential framework to detect persistent outliers. To show the effectiveness of the algorithm, we designed a sequential attack scenario on a real dataset in which the malicious profiles are associated with realistic but random attributes. Initial experiments on the popular benchmark Movielens dataset demonstrate that the proposed algorithm outperforms the baseline algorithms in both detection accuracy and speed which will be further confirmed with future extensive experiments with more sophisticated attack models.

REFERENCES

[1] Charu C Aggarwal. 2016. *Recommender Systems: The Textbook*. Springer.

[2] Michèle Basseville, Igor V Nikiforov, et al. 1993. *Detection of abrupt changes: theory and application*. Vol. 104. Prentice Hall Englewood Cliffs.

[3] Runa Bhaumik, Bamshad Mobasher, and Robin Burke. 2011. A clustering approach to unsupervised attack detection in collaborative recommender systems. In *Proceedings of the International Conference on Data Mining (DMIN)*. Citeseer, 181–187.

[4] Runa Bhaumik, Chad Williams, Bamshad Mobasher, and Robin Burke. 2006. Securing collaborative filtering against malicious attacks through anomaly detection. In *Proceedings of the 4th Workshop on Intelligent Techniques for Web Personalization (ITWP06), Boston*, Vol. 6. 10.

[5] Alper Bilge, Zeynep Ozdemir, and Huseyin Polat. 2014. A novel shilling attack detection method. *Procedia Computer Science* 31 (2014), 165–174.

[6] Dankmar Böhning. 1992. Multinomial logistic regression algorithm. *Annals of the institute of Statistical Mathematics* 44, 1 (1992), 197–200.

[7] Kenneth Bryan, Michael O'Mahony, and Pádraig Cunningham. 2008. Unsupervised retrieval of attack profiles in collaborative recommender systems. In *Proceedings of the 2008 ACM conference on Recommender systems*. ACM, 155–162.

[8] Robin Burke, Bamshad Mobasher, Chad Williams, and Runa Bhaumik. 2006. Classification features for attack detection in collaborative recommender systems. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 542–547.

[9] Paul-Alexandru Chirita, Wolfgang Nejdl, and Cristian Zamfir. 2005. Preventing shilling attacks in online recommender systems. In *Proceedings of the 7th annual ACM international workshop on Web information and data management*. ACM, 67–74.

[10] Zoubin Ghahramani, Geoffrey E Hinton, et al. 1996. *The EM algorithm for mixtures of factor analyzers*. Technical Report CRG-TR-96-1.

[11] Neil Hurley, Zunping Cheng, and Mi Zhang. 2009. Statistical attack detection. In *Proceedings of the third ACM conference on Recommender systems*. ACM, 149–156.

[12] Mohammad E Khan, Guillaume Bouchard, Kevin P Murphy, and Benjamin M Marlin. 2010. Variational bounds for mixed-data factor analysis. In *Advances in Neural Information Processing Systems*. 1108–1116.

[13] Jong-Seok Lee and Dan Zhu. 2012. Shilling attack detection–A new approach for a trustworthy recommender system. *INFORMS Journal on Computing* 24, 1 (2012), 117–131.

[14] Cong Li and Zhigang Luo. 2011. Detection of shilling attacks in collaborative filtering recommender systems. In *2011 International Conference of Soft Computing and Pattern Recognition (SoCPaR)*. IEEE, 190–193.

[15] Shang Li and Xiaodong Wang. 2014. Quickest attack detection in multi-agent reputation systems. *IEEE Journal of Selected Topics in Signal Processing* 8, 4 (2014), 653–666.

[16] Yuhong Liu and Yan Sun. 2010. Anomaly detection in feedback-based reputation systems through temporal and correlation analysis. In *2010 IEEE Second International Conference on Social Computing*. IEEE, 65–72.

[17] Bhaskar Mehta and Wolfgang Nejdl. 2009. Unsupervised strategies for shilling detection and robust collaborative filtering. *User Modeling and User-Adapted Interaction* 19, 1-2 (2009), 65–97.

[18] Kevin P Murphy. 2012. *Machine learning: a probabilistic perspective*.

[19] Michael P O'mahony, Neil J Hurley, and Guenole CM Silvestre. 2004. An evaluation of neighbourhood formation on the performance of collaborative filtering. *Artificial Intelligence Review* 21, 3-4 (2004), 215–228.

[20] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2015. *Recommender Systems Handbook*. Springer.

[21] Ronald L Wasserstein, Nicole A Lazar, et al. 2016. The ASAâĂŹs statement on p-values: context, process, and purpose. *The American Statistician* 70, 2 (2016), 129–133.

[22] Zhihai Yang, Zhongmin Cai, and Xiaohong Guan. 2016. Estimating user behavior toward detecting anomalous ratings in rating systems. *Knowledge-Based Systems* 111 (2016), 144–158.

[23] Fuzhi Zhang, Zening Zhang, Peng Zhang, and Shilei Wang. 2018. UD-HMM: An unsupervised method for shilling attack detection based on hidden Markov model and hierarchical clustering. *Knowledge-Based Systems* 148 (2018), 146–166.

[24] Zhuo Zhang and Sanjeev R Kulkarni. 2013. Graph-based detection of shilling attacks in recommender systems. In *2013 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 1–6.

[25] Zhuo Zhang and Sanjeev R Kulkarni. 2014. Detection of shilling attacks in recommender systems via spectral clustering. In *17th International Conference on Information Fusion (FUSION)*. IEEE, 1–8.