Low-rank Defenses Against Adversarial Attacks in Recommender Systems

1st Negin Entezari
Department of Computer Science and Engineering
University of California Riverside
Riverside, CA, USA
nente001@ucr.edu

2nd Evangelos E. Papalexakis
Department of Computer Science and Engineering
University of California Riverside
Riverside, CA, USA
epapalex@cs.ucr.edu

***Abstract*—Recommender systems are powerful tools which touch on numerous aspects of everyday life, from shopping to consuming content, and beyond. However, as other machine learning models, recommender system models are vulnerable to adversarial attacks and their performance could drop significantly with a slight modification of the input data. Most of the studies in the area of adversarial machine learning are focused on the image and vision domain. There are very few work that study adversarial attacks on recommender systems and even fewer work that study ways to make the recommender systems robust and reliable. In this study, we explore two state-of-the-art adversarial attack methods proposed by Tang et al. [1] and Christakopoulou et al. [2] and we report our proposed defenses and experimental evaluations against these attacks. In particular, we observe that low-rank reconstructions and/or transformation of the attacked data has a significant alleviating effect on the attack, and we present extensive experimental evidence to demonstrate the effectiveness of this approach. We also show that a simple classifier is able to learn to detect fake users from real users and can successfully discard them from the dataset. This observation elaborates the fact that the threat model does not generate fake users that mimic the same behavior of real users and can be easily distinguished from real users' behavior. We also examine how transforming latent factors of the matrix factorization model into a low-dimensional space impacts its performance. Furthermore, we combine fake users from both attacks to examine how our proposed defense is able to defend against multiple attacks at the same time. Local low-rank reconstruction was able to reduce the hit ratio of target items from 23.54% to 15.69% while the overall performance of the recommender system was preserved.**

Adversarial machine learning, recommender systems, low-rank reconstruction

I. Introduction

Recommender systems are powerful tools to help users better and easier choose between millions of options in different scenarios such as shopping and movie, book, or song selection. Business owners also gain benefits from appropriate recommendations that can potentially increase their revenue. Collaborative filtering techniques are widely used in recommender systems due to their simplicity and strong performance. Collaborative filtering relies on the user-item interactions in the past to predict future interactions. The idea is that users with similar choices in the past are most likely to make similar choices in the future [3]. Matrix factorization is a popular collaborative filtering technique that considers user-item interactions and tries to learn latent factors for users and items [4], [5]. Recently, neural collaborative filtering models have also been utilized to model non-linear user-item interaction [6].

Recent studies in adversarial machine learning show that machine learning algorithms are susceptible to adversarial attacks [7]–[9], and recommender system models are also susceptible to adversaries [10]. Attackers aim to fool recommender systems to recommend products (or items, in general) to users that serve their malicious intentions rather than satisfying users' needs. Attackers may achieve their objective by writing fake product reviews, creating fake user profiles, manipulating product images, etc. It is crucial to detect these adversaries and make recommender models robust against them. In this paper, we investigate characteristics of two of the adversarial attacks on recommender systems proposed by Tang et al. [1] and Christakopoulou et al. [2]. Then we report our experimental evaluations on how to make the recommender system more robust against these types of attack.

II. Related Work

A. Adversarial Attacks on Recommender Systems

Recommender systems have always been a target of attackers. Traditionally attackers tried to inject hand-engineered fake profiles to affect the recommender system to offer their target items maliciously. Injecting fake user profiles is broadly called a shilling attack. The shilling attack aims to augment some user profiles with limited item ratings. They can diverge the recommendation result and force the recommender system to recommend some target items to users. In recent years, poisoning attacks leveraged machine learning algorithms to generate fake user profiles [2], [11], [12]. Li et al. [11] proposed a data poisoning attack on factorization-based recommender systems. In a recent work by Christakopoulou et al. [2], Generative Adversarial Network (GAN) [13] is used to generate fake user profiles with similar rating distribution of real users. This ensures that the fake users generated by GAN are unnoticeable. Then by applying iterative gradient descent, a final set of fake user profiles are generated and injected into the recommender system.

B. Defense Against Adversarial Attacks on Recommender Systems

Adversarial training is the most popular approach to make recommender systems robust against adversaries [14]. In adversarial training, adversarial examples are generated using an existing attack model, and then these adversarial examples are fed to the machine learning model along with the benign examples. Machine learning models trained on the adversarial instances will have a better performance on the adversarial examples and higher generalization power. Adversarial training has been applied to recommender systems to have a more robust model [14], [15]. Adversarial training is expensive as it requires retraining of the model over benign and adversarial examples. Moreover, there are many attack models with different attack strategies and the adversarially-trained model may not work well to defend against unseen adversarial examples. Another group of defense techniques in the literature performs a preprocessing step to transform an adversarial example into a similar benign example. The goal of the transformation is to discard the adversarial artifacts. In the image domain, this can be done by removing the high-frequency noise added to the image [16], [17]. A similar idea was also applied to defend against attacks on graphs [18]. The idea behind these methods is that attackers try to generate unnoticeable perturbations, and therefore perturbations primarily affect the high-frequency domain of images and graphs. In this paper, we explore the characteristics of adversarial attacks from [1] and [2] to see how a low-rank approximation approach can help to defend against adversarial attacks.

III. Proposed Method

In Section III-A, we first briefly talk about two of the recent adversarial attacks. Next, in Section III-B, we explain our proposed methods to improve the performance of the recommender system when adversarial user profiles are present.

A. Adversarial Attacks in Recommender Systems

In this section, we investigate the characteristics of two different algorithms to generate adversarial users to attack recommender systems. Below, we explain the threat model for each method.

Attack I - RecSys20: The first adversarial model is proposed by Tang et al. [1]. This method uses publicly available data used by recommender systems to learn about user preferences and generates fake user profiles to maliciously influence the recommender system. The malicious intention is to boost the chance of a target item being recommended to users. Using a surrogate model, fake users profiles are learned as a bi-level optimization problem. The inner objective is to minimize the loss of the surrogate model with the presence of the fake users and the outer objective is to minimize the loss of the adversarial model by maximizing the chance of recommending of target items to real users.

Attack II - RecSys19: The second method is proposed by Christakopoulou et al. [2] which generates fake users in two steps. Their proposed framework first uses deep convolutional generative adversarial network (DCGAN) [19] to generate initial fake user profiles. These generated user profiles have a similar rating distribution to the real users. Next, their model iteratively updates the fake user profiles to maximize the hit ratio of a target item(s). GAN generates realistic-looking user profiles that result in an unnoticeable attack.

In [2], the assumption is that the recommender system is oblivious to the existence of an adversary, therefore it optimizes its loss over all given user profiles, including the fake profiles.

Previous study on adversarial attacks on images and graphs [16]–[18] elaborated the fact that adversaries mainly impact high frequency components of the data to remain unnoticeable. We are interested to examine if the same observation could be extended to adversarial recommender systems. In the next section, we propose different methods to diminish the negative impact of adversarial users.

B. Defense Methods

In this section, we introduce three low-rank defense methods to lessen the harm of fake users and improve the performance of the recommender system.

1) Fake User Detection:
Our goal is to train a classifier on real and fake user profiles that can successfully detect fake users. Once fake users are detected, they can be removed from the input data fed to a recommender model to improve its performance. A good choice of a classifier is a model that is simple and can run very fast during to be applicable in online recommendation. Also, it should have high recall because false negatives are very costly and we do not want to misclassify fake users as real ones, whereas classifying some of the real users as fake (false positive) does not harm the recommendation. We use Support Vector Machine (SVM) model to classify fake and real users. The training data is highly imbalanced (131 fake users vs. 13.1k real users) and extremely sparse (99.7% sparsity). For such a high-dimensional data classifiers will have a poor performance. We use Principal Component Analysis (PCA) to reduce the dimensionality of the data and then train the classifier in the lower-dimensional space. A 2D visualization of data considering the first two principal components is illustrated in Fig. 1. As shown in Fig. 1, points corresponding to fake and real users are separable in the 2D space. A prefered model should detect fake users (true positives) with high confidence while the number of false negatives is low. A fake user misclassified as a real user can harm the recommender system, but detecting a real user as a fake once is safe. Therefore, a good model should have a high recall. Once a high-performing model is found, detected fake users by the model are removed from the dataset and recommender model is trained on the rest of the data.

2) Local Low-rank Reconstruction:
Despite the successful performance of fake user detection defense technique, it has some limitations. It is a supervised model and requires training instances and examples of fake user profiles to train the classifier. Therefore, for different

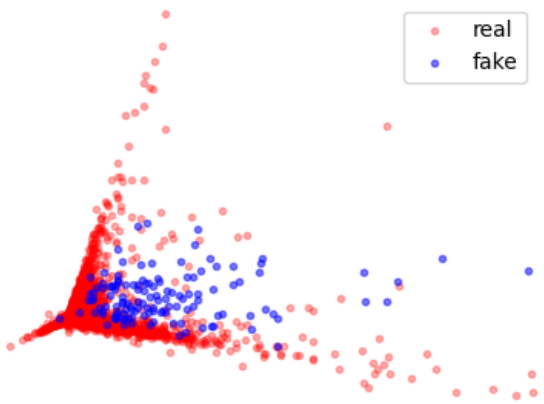


Fig. 1. 2D visualization of Gowalla dataset and learned fake users using PCA transformation.A simple SVM model is able to detect fake users, shown in blue, from real users, depicted in red

threat models, we require to obtain fake user examples and train our model against them. It is not always feasible to consider every different attack model and there are various unknown adversaries. The goal of this section is to propose an unsupervised defense model that can be applied without prior knowledge about the adversarial examples.

The number of fake users generated by adversarial models is very small compared to the number of real users in the system. Adversarial model generates fake users that mimic the same behavior of real users to avoid being easily detected and remain unnoticeable. The fake users added to the system forces the recommender model to boost the hit ratio for the target items and the overall performance of the system is preserved for unnoticeablity reasons. The impact of these fake users are very subtle compared to the impact of large number of real users and with similar intuition as [18], a low-rank solution is able to alleviate the negative impact of fake users. For a large user-rating matrix, performing a low-rank SVD requires a fairly large rank to capture main components of the data which makes the reconstruction slow. Also, reconstruction will fill the missing values and adds new values to the rating matrix. We perform local SVD low-rank reconstructions on small patches of the rating matrix and put the reconstructed patches back together to reconstruct the entire rating matrix. Reconstruction of small patches requires a very small SVD rank and can be done in parallel to speed up the process. In the experimental evaluation that follows, we consider two cases where adversarial users are absent or present and share how the performance of the recommender system is affected by the local low-rank reconstruction in both cases.

3) Low-rank Transformation:

Low-rank SVD reconstruction is able to alleviate the impact of fake users. However, we cannot infer that the performance improvement is due to the fact that the adversarial attacks are high-rank. Low-rank reconstruction introduces some additional ratings that could lead to performance improvements. To investigate this, we try to answer the following questions:

- What is the impact of low-rank SVD components?
- What is the impact of new rating values introduced from reconstruction?

To answer the first question, we transform latent factors of the matrix factorization model into SVD space. We perform the following steps:

Given X_A which is the attacked ratings matrix:

$$X_{A_k} = U_k \Sigma_k V_k^T \tag{1}$$

where $P_a = U_k U_k'$ and $P_b = V_k V_k'$ are the transformation matrices. These P_a and P_b matrices are used to transform latent factors A and B into the SVD space:

$$A' = P_a A \quad and \quad B' = P_b B \tag{2}$$

Original latent factors A and B are replaced with the transformed latent factors A' and B'. According to our experimental evaluations described in Section IV-B2, transforming MF latent factors into SVD space improves the performance of the recommender system. This shows that the performance improvement we gain from SVD low-rank reconstruction is not all because of the new ratings, and SVD components are able to discard the adversarial components.

To answer the second question regarding the impact of new rating values, we are interested in comparing the impact of transformation vs. new rating values to see which one yields higher improvements. We gradually add new rating values to the original attacked profile matrix from 0% (no new value) to 100% (all of the new values considered). The experimental evaluations reveals that recommender systems gains a higher benefit when we perform latent factor transformation along with new ratings generated after low-rank reconstruction.

IV. Experiments

We describe the dataset and experimental settings for our experiments and we share the results for each adversarial model described in Section III-A.

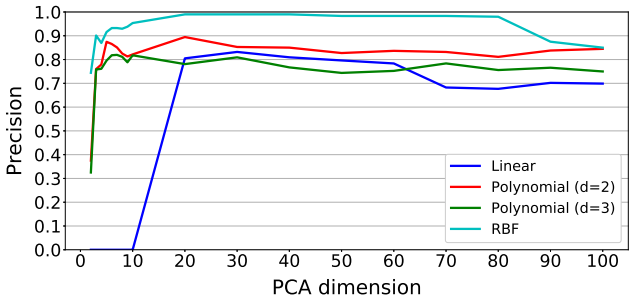
A. Attack I - RecSys20

1) Dataset and Experiment Setup:

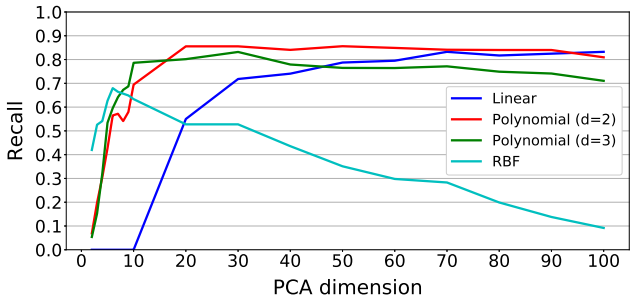
In this part of our experiments, we use Gowalla [20] dataset and we follow the same procedure as explained in [1] to preprocess the data. Gowalla dataset is an undirected location-based scocial network where users can share their locations. After preprocessing, the dataset has 13.1k users and 14.0k items. Data is randomly split into training and test set with ratio 80:20. Fake users are generated following the evaluation protocol explained in [1]. Target items are randomly selected from 4 different click percentiles:

- Head: items with total clicks greater than 95th percentile.
- Upper torso: items with total clicks between 75 and 95 percentiles.
- Lower torso: items with total clicks between 50 and 75 percentiles.
- Tail: items with total clicks less than 50th percentile.

In our experiments we only consider head target items and Weighted Regularized Matrix Factorization model [21] with



(a) Precision



(b) Recall

Fig. 2. Performance of SVM classifer on the imbalanced data. SVM with second degree polynomial kernel achieves high recall with lower number of PCA components (n=20).

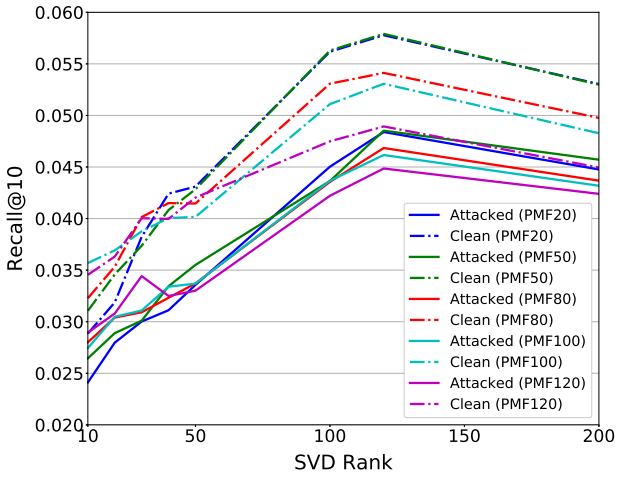


Fig. 3. Performance of PMF recommender with low-rank reconstruction of clean and attacked user profiles.

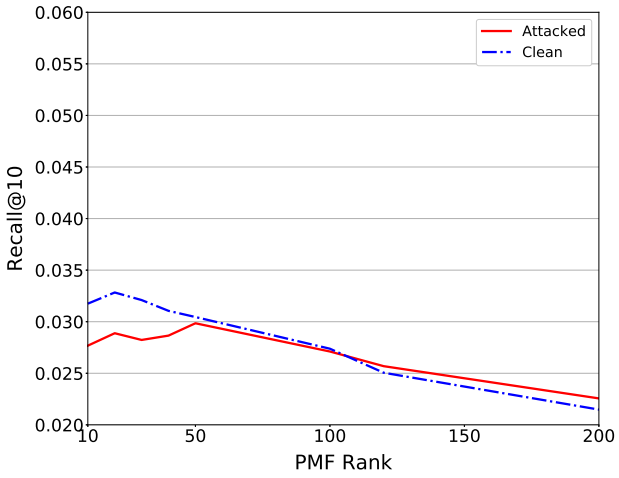


Fig. 4. Impact of different PMF ranks on the performance of recommender system for clean and attacked data.

stochastic gradient descent optimization (WRMF(SGD)) that achieves stronger adversarial performance according to the results shared in [1]. Number of fake users generated is 1% of real users, i.e. 131 fake users are generated to boost the hit ratio of the target items (HR@50).

2) Fake User Detection:

Fig. 2 shows the results of 10-fold cross-validation for the SVM model with different Kernels trained on various PCA dimensions. Also, we consider different cases where the training data is imbalanced or down-sampled. In the case of imbalanced data we use stratified cross-validation.

SVM model with second degree polynomial kernel achieved the highest recall at PCA with 20 components. Lower number of PCA components are preferable as it is faster to transform the data. We pick the SVM model with second degree polynomial to detect and remove fake users. The SVM classifier first scans all the user profiles and predicted fake users are removed from the data. Next, Recommender system recommends items to users using the sanitized data. The Performance of the WRMF(SGD) model with and without the presence of the fake user detector is reported in Table I. In our experiments, we report the performance on the target items to show how the defense method reduces the chance of target items being recommended to users. In addition, we report the overall performance of the recommender system to show that the defense method does not adversely affect the overall performance of the recommendation model.

SVM model is able to detect about 85% of fake users and causes the target items HR@50 to drop about 4% which is very close to the performance of the recommender system on the clean data.

3) Local Low-rank Reconstruction:

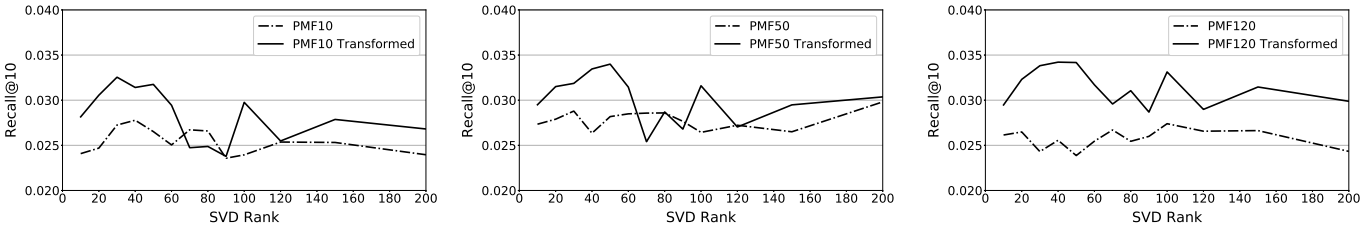
In this section, we share the result for the impact of the low-rank reconstruction on the recommender system's performance. Table II summarizes the performance of the WRMF(SGD) model with low-rank SVD reconstruction to defend against the fake users. Parameters of the local SVD reconstruction is reported in the form of [patch size, SVD rank]. Patch size = x means the entire rating matrix is considered as a single patch. We report different settings of hyperparameters that helped to reduce the hit ratio on the target items while maintaining the overall performance of the recommender model.

TABLE I
IMPACT OF THE SVM FAKE USER DETECTION ON THE PERFORMANCE OF WRMF(SGD) MODEL ON CLEAN AND ATTACKED DATA FROM GOWALLA DATASET.

Defense	Data	Overall		Target Items	
		Recall@50	HR@50	Recall@50	HR@50
No Defense	Clean	0.2895	0.7590	0.0207	0.1021
	Attacked	0.2884	0.7580	0.0261	0.1415
2nd degree polynomial SVM	Clean	0.2887	0.7580	0.0203	0.1008
	Attacked	0.2898	0.7592	0.0252	0.1095

TABLE II
IMPACT OF THE LOCAL SVD RECONSTRUCTION ON THE PERFORMANCE OF WRMF(SGD) MODEL ON CLEAN AND ATTACKED DATA FROM GOWALLA DATASET.

Defense	Data	Overall		Target Items	
		Recall@50	HR@50	Recall@50	HR@50
No Defense	Clean	0.2895	0.7590	0.0207	0.1021
	Attacked	0.2884	0.7580	0.0261	0.1415
[100, 90]	Clean	0.2893	0.7580	0.0241	0.1105
	Attacked	0.2889	0.7612	0.0260	0.1138
[300, 80]]	Clean	0.2855	0.7515	0.0216	0.0987
	Attacked	0.2838	0.7514	0.0253	0.1091
[150, 20]]	Clean	0.2609	0.7148	0.0238	0.1092
	Attacked	0.2601	0.7168	0.0236	0.1004
[x, 3500]	Clean	0.2500	0.7117	0.0209	0.0963
	Attacked	0.2468	0.7090	0.0236	0.1029
[x, 5000]	Clean	0.2866	0.7563	0.0231	0.1049
	Attacked	0.2866	0.7582	0.0278	0.1226



(a) PMF rank 10 (b) PMF rank 50 (c) PMF rank 120

Fig. 5. PMF latent factors of the attacked rating matrix are transformed into SVD latent factor space.

B. Attack II: RecSys19

1) Dataset and Experiment Setup:

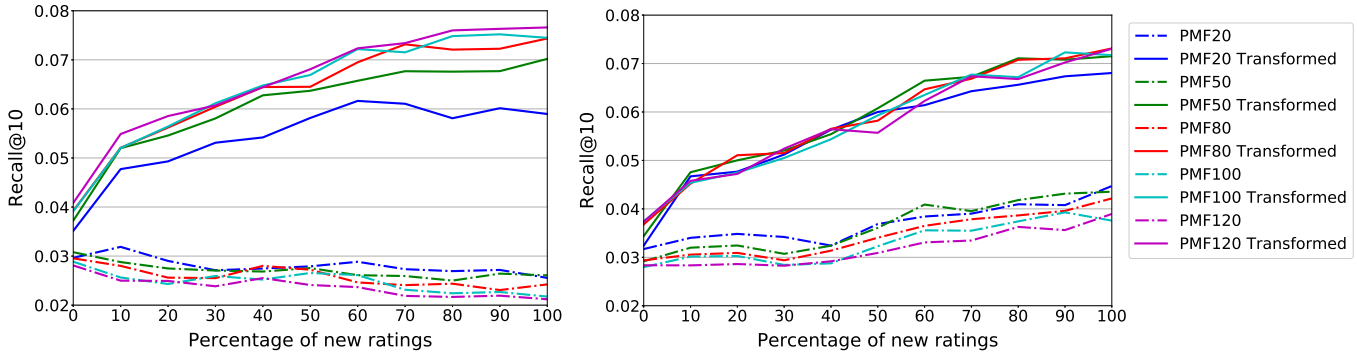
We used the MovieLens 100k dataset [22] in our experiments. MovieLens 100K dataset contains ratings for 1682 movies from 943 users. Ratings are within range 1 - 5. The recommender system model in our experiments is probabilistic matrix factorization (PMF) [23]. We use metrics such as recall@K and precision@K to report the performance of the recommender system.

2) Low-rank Transformation:

In our experiments, We performed low-rank reconstruction of the attacked user profiles and fed it to the PMF model. Fig. 3 illustrates how SVD low-rank reconstruction of the attacked profiles, improves the performance of the recommender system. We performed this experiment for different PMF ranks to evaluate the impact of PMF ranks on the performance of the recommender system and the result is shown in Fig. 4. SVD low-rank reconstruction is applied with different ranks ranging from 10 to 200. Both clean and attacked user profiles gain benefits from low-rank reconstruction up to rank 120. SVD reconstruction using Ranks greater than 120 incorporates the adversarial components and leads to a performance drop.

Moreover, to investigate if the performance improvement is because of the high-rank nature of the attack or due to the new rating values introduced after the reconstruction which where originally missing, we performed another set of experiments following the steps explained in Section III-B3. The performance of the recommender system after the transformation is shown in Fig. 5 and we can observe that the recommender model benefits from the transformation without considering the new rating values.

In another experiment, we gradually added new ratings from SVD reconstruction together with transformation of latent factors. Fig. 6 shows the results. 0% means that no new



(a) SVD rank 50

(b) SVD rank 150

Fig. 6. Transformation of PMF latent factors into SVD latent factor space vs. adding new rating values from SVD low-rank reconstruction

TABLE III
IMPACT OF THE LOCAL SVD RECONSTRUCTION ON THE PERFORMANCE OF WRMF(SGD) MODEL ON MOVIELENS 100K DATASET WHILE VARIOUS ADVERSARIAL USERS ARE PRESENT.

Defense	Attack	Overall HR@10	Target Items HR@10
No Defense	None	0.8293	0.1432
	RecSys19	0.8271	0.1505
	RecSys20	0.8346	0.1824
	RecSys19 + RecSys20	0.8452	0.2354
[200, 50]	None	0.8006	0.1697
	RecSys19	0.8165	0.1241
	RecSys20	0.8250	0.1654
	RecSys19 + RecSys20	0.8293	0.1569

rating values is added and the performance is reported on the original attacked matrix. 100% means all of the new rating values is added to the ratings matrix. Dashed lines show the performance of the recommender before transformation while gradually introducing new rating values and the solid lines are the performance after the transformation and introducing the new rating values. These plots show that the improvement gained from transformation is much higher than what is gained from new ratings added. For SVD rank 50, adding new values does not affect the performance of the recommender, but at SVD rank 150, adding more data yields higher recall@10 before transformation. On the other hand, transformation improves the performance of the recommender slightly when there are no new ratings added. With 100% new ratings added, transformation yields significant improvement in recall@10. Therefore, transformation and adding new ratings do not have a significant effect individually, but together they improve the result significantly. Svd rank 100 and rank 200 mimic the same behavior of ranks 50 and 150 respectively and we omitted their plots considering limited space.

C. Combination of Attack I and Attack II

In this section, we are interested to see if our proposed method is able to defend against multiple types of adversarial attacks. Evaluating the performance of our model on a combination of different attacks is vital as in real world recommender systems there could be various attackers with different adversarial objectives who try to inject a group of fake users to serve their malicious purposes. We performed local low-rank reconstruction on the attacked MovieLense 100k dataset with 250 fake users, half generated using attack I (RecSys20) threat model and the other half generated using attack II (RecSys19) model. A summary of results on the combined attacks is reported in Table III. The Combination of the two attacks i.e., RecSys19 + RecSys20, creates a stronger attack that achieves a higher hit ratio on target items compared to the hit ratio of each individual attack. Despite a more detrimental attack, our low-rank reconstruction defense is yet able to resist against the attack and reduce the hit ratio on the target items.

V. CONCLUSIONS

In this paper, we investigated two adversarial attacks on recommender systems and demonstrated that low-rank reconstruction and transforming into SVD space helps to reduce the impact of attack and improve the performance of the recommender system. We also demonstrated that our low-rank solution can resist adversarial users generated by different threat models.

VI. ACKNOWLEDGEMENTS

This research was sponsored by the U.S. Army Combat Capabilities Development Command Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-13-2-0045 (ARL Cyber Security CRA). The views and conclusions contained in this document are those

of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Combat Capabilities Development Command Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

REFERENCES

[1] J. Tang, H. Wen, and K. Wang, "Revisiting adversarially learned injection attacks against recommender systems," in *Fourteenth ACM Conference on Recommender Systems*, 2020, pp. 318–327.

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

[3] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*, 2001, pp. 285–295.

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

[5] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, pp. 426–434.

[6] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th international conference on world wide web*, 2017, pp. 173–182.

[7] N. Dalvi, P. Domingos, S. Sanghai, D. Verma *et al.*, "Adversarial classification," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004, pp. 99–108.

[8] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[9] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[10] Y. Deldjoo, T. D. Noia, and F. A. Merra, "A survey on adversarial recommender systems: from attack/defense strategies to generative adversarial networks," *ACM Computing Surveys (CSUR)*, vol. 54, no. 2, pp. 1–38, 2021.

[11] B. Li, Y. Wang, A. Singh, and Y. Vorobeychik, "Data poisoning attacks on factorization-based collaborative filtering," *Advances in neural information processing systems*, vol. 29, pp. 1885–1893, 2016.

[12] M. Fang, G. Yang, N. Z. Gong, and J. Liu, "Poisoning attacks to graph-based recommender systems," in *Proceedings of the 34th Annual Computer Security Applications Conference*, 2018, pp. 381–392.

[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[14] J. Tang, X. Du, X. He, F. Yuan, Q. Tian, and T.-S. Chua, "Adversarial training towards robust multimedia recommender system," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 5, pp. 855–867, 2019.

[15] Q. Wang, H. Yin, Z. Hu, D. Lian, H. Wang, and Z. Huang, "Neural memory streaming recommender networks with adversarial training," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 2467–2475.

[16] N. Das, M. Shanbhogue, S.-T. Chen, F. Hohman, S. Li, L. Chen, M. E. Kounavis, and D. H. Chau, "Shield: Fast, practical defense and vaccination for deep learning using jpeg compression," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 196–204.

[17] N. Entezari and E. E. Papalexakis, "Tensorshield: Tensor-based defense against adversarial attacks on images," *arXiv preprint arXiv:2002.10252*, 2020.

[18] N. Entezari, S. A. Al-Sayouri, A. Darvishzadeh, and E. E. Papalexakis, "All you need is low (rank) defending against adversarial attacks on graphs," in *Proceedings of the 13th International Conference on Web Search and Data Mining*, 2020, pp. 169–177.

[19] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[20] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: user movement in location-based social networks," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2011, pp. 1082–1090.

[21] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *2008 Eighth IEEE International Conference on Data Mining*. Ieee, 2008, pp. 263–272.

[22] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *Acm transactions on interactive intelligent systems (tiis)*, vol. 5, no. 4, pp. 1–19, 2015.

[23] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," *Advances in neural information processing systems*, vol. 20, pp. 1257–1264, 2007.