Data Poisoning Attacks against Differentially Private Recommender Systems

Soumya Wadhwa
soumya.wadhwa@walmartlabs.com
Walmart Labs

Saurabh Agrawal
sagrawal@walmartlabs.com
Walmart Labs

Harsh Chaudhari[*]
chaudharim@iisc.ac.in
Indian Institute of Science

Deepthi Sharma
deepthi.sharma@walmartlabs.com
Walmart Labs

Kannan Achan
kachan@walmartlabs.com
Walmart Labs

ABSTRACT

Recommender systems based on collaborative filtering are highly vulnerable to data poisoning attacks, where a determined attacker injects fake users with false user-item feedback, with an objective to either corrupt the recommender system or promote/demote a target set of items. Recently, differential privacy was explored as a defense technique against data poisoning attacks in the typical machine learning setting. In this paper, we study the effectiveness of differential privacy against such attacks on matrix factorization based collaborative filtering systems. Concretely, we conduct extensive experiments for evaluating robustness to injection of malicious user profiles by simulating common types of shilling attacks on real-world data and comparing the predictions of typical matrix factorization with differentially private matrix factorization.

KEYWORDS

Data Poisoning, Shilling Attacks, Differential Privacy, Matrix Factorization, Collaborative Filtering, Recommender Systems

ACM Reference Format:
Soumya Wadhwa, Saurabh Agrawal, Harsh Chaudhari, Deepthi Sharma, and Kannan Achan. 2020. Data Poisoning Attacks against Differentially Private Recommender Systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR ’20), July 25–30, 2020, Virtual Event, China.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3397271.3401301

1 INTRODUCTION

Collaborative Filtering (CF) recommender systems have been shown to be prone to data poisoning in which fake users along with their feedback are injected into the system [4]. The attacker can construct the preferences of these fake users so as to fool the recommender system into behaving in a way desired by the attacker. The attacker may have an objective to promote a certain set of items, or may try to compromise the overall quality of the recommendations. While such attacks are possible on all kinds collaborative filtering systems, we focus on matrix factorization based CF in this paper.

[8] studied defense against data poisoning attacks, focusing on classification algorithms. However, their technique of outlier removal doesn’t naturally lend itself to the matrix factorization setting. Recently, [6] proposed differential privacy as a defense technique against data poisoning attacks on machine learning systems, while formally defining the attacker’s cost and proving bounds on the minimization of this cost against the proposed defense mechanism. We extend this work to our CF setting. Concretely,

- We define attacker utility for a given data poisoning attack objective, and derive a finite upper bound on this utility for a differentially private matrix factorization algorithm.
- We simulate different types of shilling attacks (for promoting specific movies) on a real world dataset (MovieLens) to compare the difference in impact of data poisoning between typical matrix factorization and differentially private matrix factorization (DPMF). We observe empirically that DPMF is more robust to such attacks and leads to lower values of attack utility up to a reasonable level of injection.

2 BACKGROUND

2.1 Differential Privacy

Given data space $\mathbb{Z}$, let $\mathcal{M}$ be a randomized learner and $\mathcal{D} = \bigcup_{i=0}^{\infty} \mathbb{Z}^i$ be the space of all training data with $D \in \mathcal{D}$ being a particular data set. We define Differential Privacy [1] as follows:

Definition 2.1. (Differential Privacy) We call a randomized learner, $\mathcal{M}$, (ϵ, δ)-differentially private if $\forall D, D' \in \mathcal{D}$ that differ by one item and for all measurable sets $\mathcal{S} \subset Range(\mathcal{M})$

$$\mathbf{P}(\mathcal{M}(D) \in \mathcal{S}) \leq e^{\epsilon}\mathbf{P}(\mathcal{M}(D') \in \mathcal{S}) + \delta$$

If $\delta = 0$, we call $\mathcal{M}$ ϵ-differentially private. Informally, the above definition states that if any one point in the database is modified, the output of the randomized learner will not change by much. In the above equation, ϵ is positive, and the smaller the value of ϵ, the stronger is the privacy guarantee.

2.2 Collaborative Filtering

We assume the standard setting of collaborative filtering, where m users rate a subset of n items. We denote the full rating matrix by $\mathbf{R} = [r_{ij}]_{m \times n}$ and $\mathcal{R} \subset [m] \times [n]$ as the r_{ij} entries in $\mathbf{R}$ where user i has rated item j ("seen" or "observed" ratings). Our goal is to predict

[*] Work was done as an intern at Walmart Labs, Bangalore.

SIGIR ’20, July 25–30, 2020, Virtual Event, China
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-8016-4/20/07…$15.00
https://doi.org/10.1145/3397271.3401301

the ratings for the remaining blank user-item entries $\mathbf{R}/\mathcal{R}$ of the rating matrix. For this, a popular method is Matrix Factorization (MF) which approximates $\mathcal{R}$ using low rank factorization. Each user i is represented using a low dimensional vector $u_i \in \mathbb{R}^d$ and similarly each item j is represented using vector $v_j \in \mathbb{R}^d$. Each $r_{ij} \in \mathcal{R}$ is then approximated using the inner product $< u_i, v_j >$. Usually, the dimension d is set to a small value ($\approx 10 - 200$). Matrix $\mathbf{U} = [u_i]_{i\in[m]}$ is called the user matrix, where each row is the vector representation u_i of user i. Similarly, $\mathbf{V} = [v_j]_{j\in[n]}$ is the item matrix, where each row is the vector representation v_j of item j. The goal is to compute matrices $\mathbf{U}$ and $\mathbf{V}$ by minimizing the square of the difference between all observed ratings and predictions:

$$\min_{\mathbf{U},\mathbf{V}} \sum_{i,j\in\mathcal{R}} (r_{ij} - < u_i, v_j >)^2 + \lambda(\|\mathbf{U}\|_2^2 + \|\mathbf{V}\|_2^2) \tag{1}$$

where the regularization parameter (λ) is a positive constant. We use Stochastic Gradient Descent (SGD) to minimize Equation (1). In SGD, an update step for seen rating r_{ij} (learning rate η) is:

$$\begin{aligned} e_{ij} &= r_{ij} - < u_i, v_j > \\ u_i &= u_i + \eta(v_j e_{ij} - \lambda u_i) \\ v_j &= v_j + \eta(u_i e_{ij} - \lambda v_j) \end{aligned} \tag{2}$$

2.3 Differentially Private Matrix Factorization (DPMF)

We adopt the method proposed by [5] to make matrix factorization differentially private, which uses the result that posterior sampling preserves differential privacy, given log-likelihood of each user is uniformly bounded [9]. Formally, if the max sum of errors per user, $\max_{\mathbf{U},\mathbf{V},\mathcal{R},i} \sum_{j\in\mathcal{R}_i} (r_{ij} - < u_i, v_j >)^2 \leq B$ then matrices U and V sampled from the distribution

$$\mathbf{P}(\mathbf{U},\mathbf{V}) \propto \exp(\frac{-\epsilon}{4B} F(\mathbf{U},\mathbf{V})) \tag{3}$$

where $F(\mathbf{U},\mathbf{V}) = \sum_{i,j\in\mathcal{R}} (r_{ij} - < u_i, v_j >)^2 + \lambda(\|\mathbf{U}\|_2^2 + \|\mathbf{V}\|_2^2)$, preserve ϵ-differential privacy. Since the ratings are bounded ($1 \leq r_{ij} \leq 5$) and there exists a reasonable sublevel set

$$\{\mathbf{U},\mathbf{V} : (1-\kappa) \leq < u_i, v_j > \leq (5+\kappa)\} \tag{4}$$

every summand is bounded by $(5 - 1 + \kappa)^2$. To ensure that their sum, the bound B, is not too large, [5] proposes methods such as trimming and re-weighting. In this paper, we achieve the same using the trimming mechanism and fix the maximum number of ratings (τ) that each user has. For users with more than τ ratings, we randomly sample τ ratings and delete the others. A tractable approach called Stochastic Gradient Langevin Dynamics (SGLD) is used to sample from the distribution (3). The basic update rule is:

$$(u_i, v_j) = (u_i, v_j) - \eta_s \hat{\nabla}_{(u_i,v_j)} F(\mathbf{U},\mathbf{V}) + \mathcal{N}(0, \eta_s I)$$

where η_s is the learning rate at iteration step s, $\hat{\nabla}$ is the gradient, $\mathcal{N}$ is Gaussian noise. We encourage readers to refer to [5] for details.

3 DATA POISONING ATTACKS ON DPMF

3.1 Threat Model

Attacker's Knowledge: The attacker has knowledge about the learning algorithm being used for recommendation (i.e. matrix factorization) as well as some aggregate statistics (mean, standard deviation) about ratings per item and across all items, hence the attack methodology is based on intuition about the algorithm and might not be optimal given full knowledge, as derived in [4].

Attacker's Power: The attacker can modify the clean rating matrix $\mathcal{R}$ by injecting fake users along with their ratings to obtain a poisoned matrix $\tilde{\mathcal{R}}$. We consider the case where the attacker can alter at most k ratings. Formally, $\tilde{\mathcal{R}}$ should lie in ball $\mathcal{B}(\mathcal{R}, k)$, where matrix $\mathcal{R}$ represents the center of the ball and radius k is the maximum number of added or modified ratings.

Attacker's Goal: The goal of the attacker is to force $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{V}}$, learnt from the poisoned rating matrix $\tilde{\mathcal{R}}$, to achieve certain targets. Formally, we define a utility function ϕ, which measures the extent to which the aim of the attack is attained. Let the randomized (differentially private) learner be $\mathcal{M}$. Then, $\tilde{\mathbf{U}}, \tilde{\mathbf{V}} = \mathcal{M}(\tilde{\mathcal{R}})$ and we formulate the problem as maximization of the expected utility:

$$\max_{\tilde{\mathcal{R}}\in\mathcal{B}(\mathcal{R},k)} \Phi(\tilde{\mathcal{R}}) := \mathbf{E}_{\mathcal{M}(\tilde{\mathcal{R}})}[\phi(\mathcal{M}(\tilde{\mathcal{R}}))] \tag{5}$$

For the scope of this paper, we focus on integrity attacks, inspired from [4], which are aimed at promoting specific target items and define the attacker's utility as:

$$\phi(\mathcal{M}(\tilde{\mathcal{R}})) = \phi(\tilde{U}, \tilde{V}) = \frac{1}{m} \sum_{i\in P} \sum_{t\in Q_T} < \tilde{u}_i, \tilde{v}_t > \tag{6}$$

where m is the number of real users, P is the set of all real users, and Q_T is the set of target items for promotion. However, our theoretical analysis is applicable to all non-negative utility functions.

3.2 Upper Bound for Attacks

Similar to [6], we provide a finite upper bound on the extent to which $\Phi(\tilde{\mathcal{R}})$ can be maximized, demonstrating that differentially private matrix factorization is, in theory, resistant to data poisoning attacks. We first prove an upper bound on utility for the case where the attacker can modify / add / delete exactly one rating and then trivially extend it for the case of k ratings. Note that $\phi \geq 0$.

Lemma 3.1. *Let $\mathcal{M}$ be an ϵ-differentially private matrix factorization algorithm. Then, $\bar{U}, \bar{V} = \mathcal{M}(\mathcal{R})$ and $U', V' = \mathcal{M}(\mathcal{R}')$. Let $\Phi(\mathcal{R}')$ be the objective function to maximize, where $\mathcal{R}' \in \mathcal{B}(\mathcal{R}, 1)$, then*

$$\Phi(\mathcal{R}') \leq e^\epsilon \Phi(\mathcal{R}) \tag{7}$$

Proof. Given $\phi \geq 0$, let $\Omega(x) = \{U, V : \phi(U, V) > x\}, \forall x \geq 0$. Since, $\mathcal{R}' \in \mathcal{B}(\mathcal{R}, 1)$, i.e., $\mathcal{R}$ and $\mathcal{R}'$ differ by one rating, differential privacy provides the guarantee that $\forall x \geq 0$,

$$\mathbf{P}(\mathcal{M}(\mathcal{R}') \in \Omega(x)) \leq e^\epsilon \mathbf{P}(\mathcal{M}(\mathcal{R}) \in \Omega(x))$$

Since $\Phi(\mathcal{R}) \geq 0$, using integral identity, we have

$$\begin{aligned} \Phi(\mathcal{R}') &= \mathbf{E}_{\mathcal{M}(\mathcal{R}')}[\phi(\mathcal{M}(\mathcal{R}'))] = \int_0^\infty \mathbf{P}(\phi(\mathcal{M}(\mathcal{R}')) > x)\, dx \\ &= \int_0^\infty \mathbf{P}(\mathcal{M}(\mathcal{R}') \in \Omega(x))\, dx \leq e^\epsilon \int_0^\infty \mathbf{P}(\mathcal{M}(\mathcal{R}) \in \Omega(x))\, dx \\ &= e^\epsilon \int_0^\infty \mathbf{P}(\phi(\mathcal{M}(\mathcal{R})) > x)\, dx = e^\epsilon \Phi(\mathcal{R}) \end{aligned}$$

Thus, we prove that $\Phi(\mathcal{R}') \leq e^\epsilon \Phi(\mathcal{R})$. □

Theorem 3.2. *Let $\mathcal{M}$ be an ϵ-differentially private learner. Let $\Phi(\tilde{\mathcal{R}})$ be the attack utility and n_T be the number of target items, where $\tilde{\mathcal{R}} \in \mathcal{B}(\mathcal{R}, k)$, then*

$$\Phi(\tilde{\mathcal{R}}) \leq n_T(5 + \kappa)e^{k\epsilon} \tag{8}$$

Proof. We can view $\tilde{\mathcal{R}} \in \mathcal{B}(\mathcal{R}, k)$ as changing ratings in the clean rating matrix, $\mathcal{R}$, k times. So, we can apply Lemma 3.1 k times to obtain that $\Phi(\tilde{\mathcal{R}}) \leq e^{k\epsilon}\Phi(\mathcal{R})$. From (4) and (6), we observe

$$\phi(\mathcal{M}(\mathcal{R})) \leq \frac{1}{m}\sum_{i \in P}\sum_{t \in Q_T}(5 + \kappa) \leq n_T(5 + \kappa)$$

Thus, $\Phi(\mathcal{R}) \leq n_T(5 + \kappa)$, and substituting this value back into the equation, we can obtain (8). □

3.3 Common Attacks on MF

Since the focus of this paper is on the defense capability of differentially private MF and not on the attack methodology, we do not directly optimize the attack objective. Rather, we cater to common attacks on collaborative filtering systems in literature ([10] [3] [7] [11]) which lead to high utility values for the attacker. Items under consideration per injected user for the attacks are categorized as:

- *Null Items (I_N):* Items which do not have any fake injection.
- *Filler Items (I_F):* Randomly sampled items for which feedback is injected to make the malicious users seem real, and ensure that some correlation is established with other existing items.
- *Selected Items (I_S):* Items, usually popular, for which injecting ratings makes the attack more effective (at the cost of increased risk of detection) since the ratings for the selected and target items become highly correlated for the fake users.
- *Target Items (I_T):* Items that the attacker wants to promote or demote using data poisoning.

We consider the following common shilling/data injection attacks:

- *Random Attack:* Filler items are chosen randomly from all items and their ratings are sampled from $\mathcal{N}(\mu_{all}, \sigma_{all}^2)$. There are no selected items and the target items are given the max rating (push) or min rating (nuke).
- *Average Attack:* Filler items are chosen randomly from all items and ratings are sampled from $\mathcal{N}(\mu_{item}, \sigma_{item}^2)$ for each item. There are no selected items and the target items are given the max rating (push) or min rating (nuke).
- *Average over Popular (AoP) Attack:* Filler items are chosen randomly from a fixed percentage of popular items to make injections more realistic and ratings sampled from $\mathcal{N}(\mu_{item}, \sigma_{item}^2)$ for each item. There are no selected items and the target items are given the max rating (push) or min rating (nuke).
- *Bandwagon (Random / Average) Attack:* Equivalent to a random / average attack along with a fixed number of selected (usually popular) items which are given fake max ratings.

4 EXPERIMENTS

We conduct experiments on the MovieLens-100k dataset [2] which consists of 100,000 ratings (1-5) collected via the MovieLens website, from 943 users on 1682 movies.

We split up our dataset into training (80%) and test (20%) data, and compute the U and V matrices for both non-private (optimization using SGD) and private (sampling using SGLD) matrix factorization. The number of latent factors (dimensions) learnt is 10 in both cases, with batch-size = 100, learning rate decaying as per $\eta_s = \frac{\eta_0}{s^\gamma}$ and L2 regularization weight (λ) = 0.1. The parameters used for SGD to minimize sum of squared errors are initial learning rate (η_0) = 0.016, with $\gamma = 1.0$ and this converges after around 10 epochs, overfitting thereafter. For SGLD, η_0 was set to 5×10^{-4}, with $\gamma = 0.5$ and this almost converges after around 200 epochs. The sampling is allowed to burn-in for around 10% of the total number of steps and a temperature parameter of $\zeta = 0.002$ is multiplied with the variance of the inserted Gaussian noise to speed up the burn-in process. Additionally, there are also parameters τ, which is the maximum number of ratings per user (trimmed otherwise), and κ, which bounds the error value for each predicted rating (each prediction should be in the range $[1 - \kappa, 5 + \kappa]$) for the SGLD sampling mechanism, which determine the privacy guarantees that it provides [5]. We set $\kappa = 1$ and experiment with different values of τ. Higher the value of τ, weaker the privacy guarantee and better the accuracy for differentially private matrix factorization. Table 1 shows the root mean squared error (RMSE) results for test data, which increases as the value of τ decreases, since a higher fraction of ratings data which can be used for training gets discarded as a part of the trimming process, to ensure more privacy.

Model Type	τ	**rmse**
SGD	-	0.98
SGLD	50	1.11
SGLD	100	1.05
SGLD	200	1.02
SGLD	400	1.01

Table 1: RMSE values for variations of MF

Based on our observations, we chose $\tau = 200$ for our attack experiments (almost 90% of users rated less than 200 movies). We keep the filler item set (I_F) size fixed at 3% of all movies, the popularity percentage for sampling filler items in AoP attacks at 15% and the selected item set (I_S) size for Bandwagon attacks at 0.5%. For simplicity, we report results (in Table 2) for the most unpopular movie in the dataset, but the analysis can easily be extended to other items as well. Note that the mean predicted rating of the target item (target mean) is equivalent to the attacker's utility defined for an integrity attack in Equation (6).

We can see that as the attack percentage increases, there is a considerable increase in the mean predicted rating for the target item. We suspect that the sharp increase, even for lower injection rates, is due to two factors: the target item is unpopular and has very few ratings (only one), and the dataset size is fairly small. However, especially for reasonable levels of malicious injections, the differentially private MF method is significantly more robust, slightly less so for the bandwagon attacks. At 5% injection, SGD leads to a slightly lower predicted value of the target mean. We also compute hit rate at 40 as the fraction of users for whom the target item appeared within their top 40 recommended movies. This is a commonly used metric for gauging the final impact of the attack. Here, DPMF consistently outperforms typical MF, and we observe much lower hit rates indicating that the attacks are less successful at boosting the target item in the differentially private setting. AoP seems to be the most effective type of attack, followed closely by

Attack Type	Attack %age	Hit Rate@40		Target Mean	
		SGD	SGLD	SGD	SGLD
None	None	0.00	0.00	1.44	1.62
Random	0.5%	0.01	**0.00**	3.61	**2.12**
	1.0%	0.25	**0.01**	4.21	**3.50**
	3.0%	0.48	**0.36**	4.39	**4.38**
	5.0%	0.63	**0.53**	**4.50**	4.54
Average	0.5%	0.02	**0.00**	3.69	**2.12**
	1.0%	0.51	**0.01**	4.43	**3.55**
	3.0%	0.75	**0.60**	4.64	**4.59**
	5.0%	0.84	**0.76**	**4.72**	4.75
AoP	0.5%	0.01	**0.00**	3.41	**2.12**
	1.0%	0.58	**0.01**	4.48	**3.57**
	3.0%	0.86	**0.70**	4.76	**4.71**
	5.0%	0.91	**0.81**	**4.79**	4.82
Bandwagon Random	0.5%	**0.00**	**0.00**	3.54	**2.41**
	1.0%	0.21	**0.01**	4.18	**3.46**
	3.0%	0.26	**0.24**	**4.27**	**4.27**
	5.0%	0.44	**0.40**	**4.38**	4.41
Bandwagon Average	0.5%	0.01	**0.00**	3.62	**2.40**
	1.0%	0.43	**0.01**	4.35	**3.51**
	3.0%	0.50	**0.42**	**4.42**	4.43
	5.0%	0.65	**0.61**	**4.52**	4.59

Table 2: Attack Metrics for Typical and DP MF Algorithms

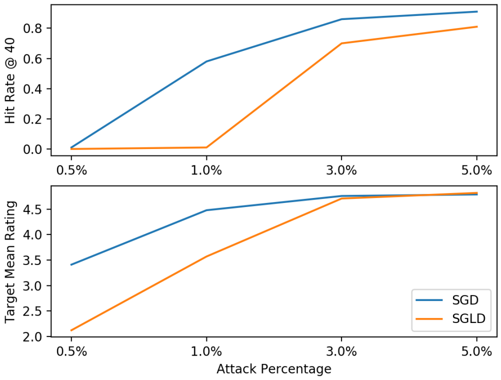

Figure 1: Comparison of Attack Metrics for AoP Attacks

the average attack. The shift in the overall mean predicted rating is more subtle, but still seems to be lower for DPMF versus MF for attacks such as random and bandwagon random attacks.

Next, we conduct some experiments (with attack percentage 3%) to check how the defense capabilities of differentially private matrix factorization vary when we vary τ which reflects the degree of privacy offered by the algorithm. Results are shown in Table 3.

We can clearly observe an increasing trend in the value of the target mean as τ increases and the privacy guarantee becomes weaker. This can be because of two reasons. One, lower values of τ increase the randomness component of the differentially private learner, $\mathcal{M}$, and hence lead to higher robustness to attacks. Two, fake user profiles have a higher likelihood of getting their ratings trimmed for lower values of τ, which can directly reduce impact.

Attack Type	τ value	Hit Rate@40		Target Mean	
		SGD	SGLD	SGD	SGLD
None	50	0.00	0.00	1.44	1.23
	100		0.00		1.34
	200		0.00		1.62
Random	50	0.48	0.06	4.39	4.07
	100		0.23		4.29
	200		0.36		4.38
Average	50	0.75	0.25	4.64	4.32
	100		0.47		4.52
	200		0.60		4.59
AoP	50	0.86	0.49	4.76	4.60
	100		0.64		4.68
	200		0.70		4.71
Bandwagon Random	50	0.26	0.07	4.27	4.12
	100		0.15		4.19
	200		0.24		4.27
Bandwagon Average	50	0.50	0.26	4.42	4.34
	100		0.33		4.36
	200		0.42		4.43

Table 3: Attack Metrics for Varying Privacy Guarantees

5 CONCLUSION

In this work, we derived an upper bound on attack utility for a differentially private matrix factorization algorithm. We experimented with different types of data poisoning attacks on real world user-item feedback data, and observed that differentially private MF is more robust to such attacks than typical MF and leads to lower values of attack utility in most cases. While we demonstrated results for integrity attacks, in the future, this work can be extended to other types of attacks, like availability attacks. Also, we leave evaluation on larger datasets, and comparison of the obtained results with existing baselines to future work.

REFERENCES

[1] C.Dwork and A.Roth. 2014. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends in Theoretical Computer Science* (2014).
[2] F. M. Harper and J. A. Konstan. 2015. The MovieLens Datasets: History and Context. In *ACM Transactions on Interactive Intelligent Systems (TiiS'15).*
[3] N. Hurley, Z. Cheng, and M. Zhang. 2009. Statistical attack detection. In *ACM Conference on Recommender Systems (RecSys'09).*
[4] B. Li, Y. Wang, A. Singh, and Y. Vorobeychik. 2016. Data Poisoning Attacks on Factorization-Based Collaborative Filtering. In *NeurIPS'16.*
[5] Z. Liu, Y.X. Wang, and A. Smola. 2015. Fast Differentially Private Matrix Factorization. In *RecSys'15.*
[6] Y. Ma, X. Zhu, and J. Hsu. 2019. Data Poisoning against Differentially-Private Learners: Attacks and Defenses. In *IJCAI'19.*
[7] C. E. Seminario and D. C. Wilson. 2014. Attacking item-based recommender systems with power items. In *ACM Conference on Recommender systems (RecSys'14).*
[8] J. Steinhardt, P.W. Koh, and P. Liang. 2017. Certified Defenses for Data Poisoning Attacks. In *Conference on Neural Information Processing Systems (NeurIPS'17).*
[9] Y. X. Wang, S. Fienberg, and A. Smola. 2015. Privacy for free: Posterior sampling and stochastic gradient monte carlo. In *ICML'15.*
[10] Z. Yang and Z. Cai. 2017. Detecting abnormal profiles in collaborative filtering recommender systems. In *Journal of Intelligent Information Systems.*
[11] Z. Zhang and S. R. Kulkarni. 2013. Graph-based detection of shilling attacks in recommender systems. In *IEEE Workshop Machine Learning for Signal Processing.*