Strategic Attacks on Recommender Systems: An Obfuscation Scenario

Wissam Al Jurdi, Jacques Bou Abdo, Jacques Demerjian, Abdallah Makhoul

Strategic Attacks on Recommender Systems: An Obfuscation Scenario

Wissam Al Jurdi
FEMTO-ST Institute, CNRS — LaRRIS, Faculty of Sciences
University Bourgogne Franche-Comté — Lebanese University
Montbéliard, France
wissam.al_jurdi@univ-fcomte.fr, 0000-0001-9497-0515

Jacques Bou Abdo
University of Nebraska at Kearney
Kearney, USA
bouabdoj@unk.edu, 0000-0002-3482-9154

Jacques Demerjian
LaRRIS, Faculty of Sciences
Lebanese University
Fanar, Lebanon
jacques.demerjian@ul.edu.lb, 0000-0001-9798-8390

Abdallah Makhoul
FEMTO-ST Institute, CNRS
University Bourgogne Franche-Comté
Montbéliard, France
abdallah.makhoul@univ-fcomte.fr, 0000-0003-0485-097X

***Abstract*—Understanding user behavior in the context of recommender systems remains challenging for researchers and practitioners. Inconsistent and misleading user information, which is often concealed in datasets, can inevitably shape the recommendation results in certain distorted ways despite utilizing recommender models with enhanced personalizing capabilities. Naturally, the quality of data that fuels those recommenders should be extremely reliable and free of any biases that might be invisible to a model, irrespective of its type. In this article, we introduce two modern forms of noise that are intrinsically hard to detect and eliminate; one is malicious in nature and will be termed *Burst* while the other is unique in that it forms its own category and will be referred to as *Opt-out*. Additionally, with the aim of segregating the nature of noise behind such threats, we present a distinct case study on Burst and Opt-out to illustrate how the detection of those threats can be challenging compared to that of traditional noise and with the current detection methods. Finally, we expound on the ability of such threats to bias the output of recommenders in their own unique way while primarily retaining data that is not fundamentally erroneous.**

***Index Terms*—recommender systems, natural noise, dataset attacks, privacy, trust.**

I. Introduction

Throughout the years, Recommender Systems (RSs) have become increasingly essential to online businesses, irrespective of their size, especially in the e-commerce field [1], [2]. Recently, talks about methods of scaling the e-commerce sector have dominated the majority of webinars and articles especially with the increased shift towards online-based services [3]–[6]. Comprising a varied range of implementation methods that extend from the prominent collaborative filtering techniques to advanced latent factor models, recommenders partake in most top-ranked commercial platforms like Amazon, Netflix, Spotify, Last.fm, etc. [7] and enormously contribute to their success. This originates from the substantial problem such approaches try to tackle through attempting to provide highly personalized services, the information overload. As a result, it isn't surprising when the demand for employing recommenders profoundly increases as the shift to online platforms registers a sharp advance.

The primary power of the personalized recommendations generated by various types of RSs is extremely dependent on the presence of abundant user contributions in the forms of ratings, reviews, tags, likes, etc. Researchers studying and enhancing RSs and their algorithms have merely focused on algorithmic improvements paying nominal attention to the quality of the underlying data. The involvement of the human factor in processes such as the rating elicitation renders it immensely prone to errors that might occur deliberately or naturally. Ratings, reviews, and other details recommender algorithms depend on hold critical information that might not always be genuine or reliable. This is recognized as noise in the datasets used by RSs, and naturally, if RSs train on inaccurate data to learn and predict user behavior, they will inevitably have inconsistent results.

Previous research shows there are two primary types of noise in RSs, malicious and natural. Briefly put, the general definition of noise in datasets is the *rating feedback that does not reflect a user's true preference or intention*. This anomaly might be purposely set by outsiders in the form of attacks on a system to bias the output (malicious noise) [9]. It could also occur naturally as a result of users inconsistent rating behavior (natural noise) [10], [11]. Malicious noise results from numerous forms of attacks carried out on online applications that are typically powered by diverse types of RSs and it has witnessed much of the research attention in the past few years [9]; conversely, the natural noise domain has not yet received a lot of focus from researchers, and lately, it has become an interesting topic in the study of anomalies in datasets [7]. In contrast to malicious noise, natural noise occurs inherently due to certain user-specific behavior which makes it special and unusually complex to model. It's completely arbitrary and user-dependent: Users can be miserable or feeling down on a certain day and rate all recommendations they encounter on

their favorite platform as bad, even the genres they tend to prefer – Natural noise due to an emotional state. Significant improvements are still required to develop a generic noise-aware layer that is compatible with all RSs and capable of overcoming natural and malicious inconsistencies in datasets [8]. In addition, such a unified system would be able to deal with noise irrespective of its type and independent from the deployed recommendation engine.

In this article, we introduce and discuss a new noisy user behavior that is obfuscation-based (the act of opting-out from a system). Further, we demonstrate how this mechanism could be segmented into two main types, one of which does not belong to any of the two noise categories presented above and will maintain its own class called Obfuscation, while we will call the noise itself Opt-out. The other type, which we will name Burst, retains a malicious component and accordingly belongs to the malicious noise class. This new behavioral noise is purely user-intended, a behavioral form affecting the authenticity of item feedback, and can be indirectly harmful to a recommender's output. Using the neighborhood-based assessment method [28], our study shows how the effect of obfuscation is very harmful to the local group of users such as a user's neighborhood in a K-Nearest Neighbour. This implies that such noise is generally unnoticed when using conventional evaluation metrics, and only when evaluating the performance on a granular level, we are able to detect its actual effect.

The rest of the work is organized as follows: In Section 2 we discuss the obfuscation noise background while in Section 3 we introduce the two newly discovered noise types in RSs. In Section 4 we present and discuss the simulation experiments and their results, and in Section 5 we conclude the work.

II. Background and Related Work

In this section, we will discuss the obfuscation mechanism and how it affected recommender-related applications in the past. We will cover two major categories of obfuscation and touch on their relationship to RSs.

A. Obfuscation as Twitter Phenomenon

On several occasions such as the elections in Russia (2011) and Mexico (2012), Twitter became the court for pivotal attacks that ultimately deviated the targeted public opinion; they came to be known as obfuscation attacks [20]. During those two incidents, people relied on Twitter to convey a certain public message and plan movements for government-targeted protests, an effective way that has become quite mainstream in many countries nowadays (Twitter revolutions [16]). Entities wanting to oppose the information flow fell short of initiating attacks (such as Distributed Denial of Service – DDoS) on highly secured platforms such as Twitter, and instead resorted to a unique way of tampering with the system's algorithm. To trick the system, the attack was aimed at highly-relied-on hashtag trends and timeline recommendations and worked through injecting random and false posts under said hashtags. The plot in Figure 1 shows the average combined behavior from three very famous hashtags where we can observe how

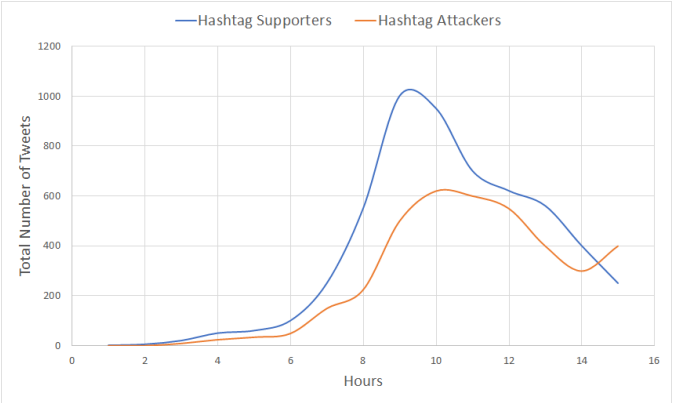

Fig. 1. Burst attack on a Twitter hashtag.

the number of relevant hashtag tweets, in a relatively short period of time (an average of 9 hours), decreased quickly after the attacks flooded the timelines. Ultimately, the extraneous injected tweets dominated the stream for the target topic to an extent where those relevant to it were completely overshadowed.

B. Obfuscation as User Weapon

Naturally, account privacy in a recommender-powered system, or any other social platform, is important and it's becoming much normalized as awareness about personalization protocols continues to increase [13]–[15]. Those online platforms, albeit continuously re-assuring about personal privacy protection often conveyed in exaggerated ad campaigns that mask indiscriminately agreed-on privacy policies, cause users to voluntarily stay connected and use them. Deloitte's 2017 study proves it: 91% of people in the US consent to legal terms and services conditions without reading them [17]. Furthermore, a great deal of those applications that range from online payment solutions to massive social networking platforms have become an integral part of our routine. In this case, a different form of obfuscation emerges [20] as a free and elementary attack which users could leverage to opt-out from those systems; it is different than the obfuscation-powered Twitter attacks in that it is primarily utilized by normal individuals who find themselves having shared, whether implicitly or explicitly, countless personal preferences with online systems and simply want to quit. That said, users who choose to do this don't just disable their accounts or refrain from logging in again. Instead, they tend to initiate a self-destructive profile behavior mechanism through introducing loads of information (in the form of ratings, likes, posts, etc. – that depends on the platform) that are not erroneous, however, not very consistent with their predilections. As a result, this tricks the system and conveys false information about the kind of content that genuinely engages this user, further amplifying a hidden form noise in the dataset. To the system, such users maintain normal profiles and merely refined or changed their tastes over a certain period of time, but what actually happened was that those users subtly leveraged an opt-out obfuscation

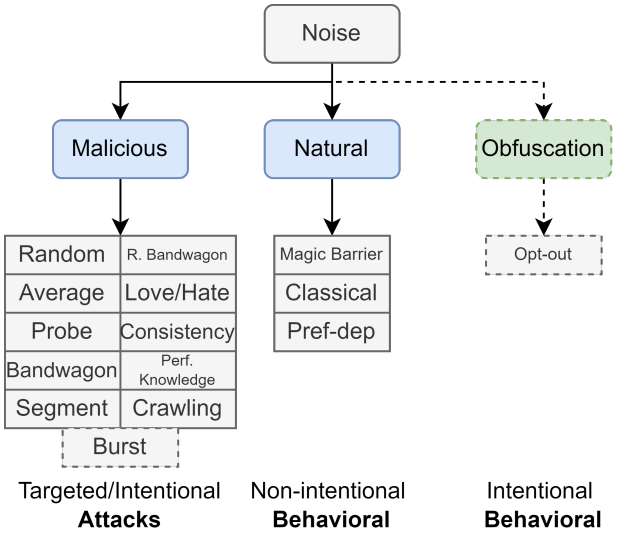

Fig. 2. Noise branches in RSs including the new obfuscation forms.

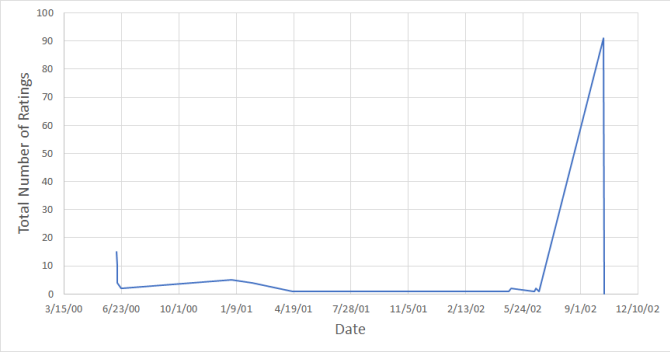

Fig. 3. Rating activity of a user from ML-1m.

attack masking their interests and concealing their preferences [20].

III. A New Type of Noise in Recommenders

The above introduced notions about obfuscation allow the formation of two types of attacks on recommenders. The first is Burst and it is used by attackers utilizing fake/inactive profiles to deviate a certain opinion and tamper with the system to specifically target a group of users. The second type is called Opt-out and it is utilized for personal reasons should users decide to eliminate any data that might constitute their profile preferences. Those types of attacks weren't mentioned in previous research about anomalies and noisy user behavior in recommenders [8], [9], [12], [19]. Figure 2 shows the categories of noise with the new obfuscation types.

Burst in RSs is very similar to that in the Twitter case (section II). It's even present in traditional recommender datasets that are extensively used in research studies. Figure 3 shows one example of such users, and like the Twitter bots, this user was inactive for a lengthy period of time before suddenly registering large activities and then going dormant. Typically, in an online setting, Burst can be leveraged to target a specific trending item (resembling the Twitter scenario) and the recommendation system needs to curb such behavior as it could negatively influence the general opinion. Therefore, we define the following two obfuscation-based attacks in recommenders:

- Burst attack: A RS attack strategy that targets a group of users, mainly to deviate their opinion. It utilizes fake or inactive profiles and basically tampers with the whole system.
- Opt-out attack: A RS attack that's mainly used by a single individual as a means of eliminating any data that might constitute his personal profile status.

1) Noise algorithm: Natural noise in datasets can be uncovered through several approaches [8] and for that purpose, we selected the most famous natural noise management and user/item clustering methodology [18] from the natural noise path that's thoroughly discussed in [8]. After that, we combined it with the work done in [22], [23] for serendipity detection and analysis as well as overall parameter tuning for a more optimal noise detection output. This strategy allows us to detect if a certain rating by an individual truly deviates from his usual predilections. The principal clustering method that runs as a pre-recommender step (completely independent of the recommendation system employed) on the dataset itself basically classifies all users and items into distinct groups; every rating a user has will be examined against their unique overall profile, if it accommodates their type then it's likely a correct rating and if doesn't, then most probably it comprises noise [18].

In addition to that, we ensured, with the application of a serendipity-oriented approach, that the actual noise is not confused with serendipity since there's a very fine line between the two [22], [23]. In the real world, user tastes undergo alterations as they explore new items in the huge inventories, and it is important that a recommender is powerful yet flexible enough to suitably handle those variations employing acceptable ranges of churn, responsiveness and serendipity without overdoing it - user interests, likes, dislikes, and fashions inevitably evolve with time [26]. Those substantial factors have been surprisingly overlooked in almost all the studies in the natural noise field where the research path became predominantly fixated on accuracy-related metrics such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) [8], [27].

2) Obfuscation detection mechanism: Natural noise algorithms can help us detect noisy behaviors [8], however, exploring the dataset for opt-out scenarios is our main aim. It might be enough to analyze the natural noise percentage in the last couple of days for the users who abandoned the system but we hypothesis that it is much more accurate to examine a full retrospect of the profile of an opt-out candidate. As the last experiment will show, opt-out obfuscation in datasets can be in different forms and rating peaks aren't just located in the last couple of days. The opt-out attack case can be equivocal and very easily overlooked by the system as it's similar to a normal activity that might even be the outcome of

serendipitous discoveries (as touched on in the introduction of Obfuscation). In an attempt to generalize an opt-out detection strategy, we propose the following equation:

$$u_{(opt-out)} = \frac{|d_{(n,u)}|}{|N_u|} > 0.5, |N_u| > 0 \quad (1)$$

Where $u_{(opt-out)}$ is a potential opt-out candidate, $|d_{(n,u)}|$ is the total noise in the last day of the user activities and $|N_u|$ is the total number of noise for user u. The measure for abandoning the system can be easily achieved through ensuring that the day in $|d_{(n,u)}|$ is much older than today's date, or in case of offline datasets such as the one we are using for this example, the last day it was published online. To test the impact of opt-out malicious behaviors in the dataset and their hidden effect on the performance, we define the following characteristics of the ratings that were considered as to be eliminated from the dataset for the experiments in the next section:

- A large number of ratings in a very short period of time (e.g. 1 or 2 days at most)
- A significant variation of taste between peak rating days and other normal days
- A significant noise score on the peak day (Equation 1)

IV. Simulation and Results

In this section, we will introduce the experiment setup used to simulate the obfuscation noise in RSs and then discuss the results and the effect of such noise on the users of our system.

A. Experimental Setup

1) Datasets and Algorithms: In our experiments, the ml-latest-small dataset as well as the ml-1m [21] are used to do the testing of the opt-out introduced in the previous sections. They consist of 610 users, 9,742 movies, 100,836 ratings, and 6,040 users, 3,900 movies, 1,000,209 ratings respectively. The recommender algorithm that is employed in the experiment is a collaborative filtering recommender [1] that takes into account the mean ratings of each user with the parameter k set to 40 neighbors and a pearson correlation similarity measure. Natural noise in datasets can be uncovered through several approaches [8], in our experiments, we selected a famous natural noise management and user/item clustering methodology that was proposed by Toledo et al. [18]. The evaluation process used for the opt-out obfuscation experiments follows the neighborhood evaluation process presented in [28] with MAE, RMSE and the Normalized Discounted Cumulative Gain (NDCG).

2) Target User Profiles: To evaluate the performance using the neighborhood-based method, we select two profiles from each dataset that satisfy the above conditions. Both exhibit malicious behaviors and resemble a mild obfuscation attack where a major part of their ratings contain malicious or natural noise. These profiles have ratings that abruptly shift in taste and also don't have a reasonable amount of ratings on certain days, which is distinctly counter-intuitive to the case of a normal user in a real-world scenario. The natural

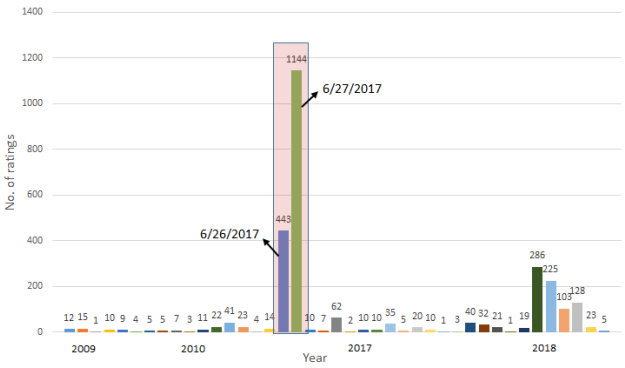

Fig. 4. User profile attack case.

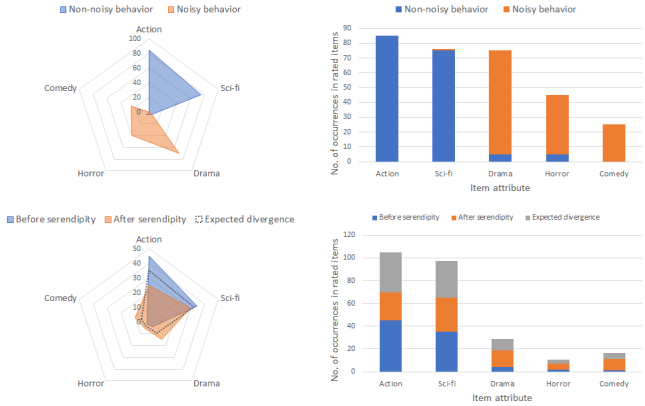

Fig. 5. Item attribute sudden variation example (top) and a case of serendipitous discovery (bottom).

noise algorithm signaled that most of those users' ratings in their peak rating days were actually noise and do not fit their genuine overall profile predilection.

In addition to the two profiles selected for the tests, we run the experiments in parallel on the case of legitimate ratings to test if the neighborhood evaluation reports different results. Legitimate ratings are selected from other users who do not meet the above critical conditions while the total number of ratings is selected to be equal to that of the malicious case in both datasets. In the two cases, the total number of ratings eliminated from the dataset is around 1,500, i.e. 1% of ml-latest-small and 0.15% of ml-1m).

B. Case Study - User Rating Noise and Sudden Taste Variation

Figure 4 shows the malicious ratings of a user from the ml-latest-small dataset that meets the target profile conditions presented in the previous section. There is an increased activity in two specific days where the natural noise factor in them registers 82%. This goes hand-in-hand with the user taste variation on those days as the item attributes are significantly different from the profile preferences. The abrupt change in taste is also demonstrated in the examples of Figure 5 (top). For brevity, only the most affected attributes are displayed. The Figure shows how items of new genres such as drama,

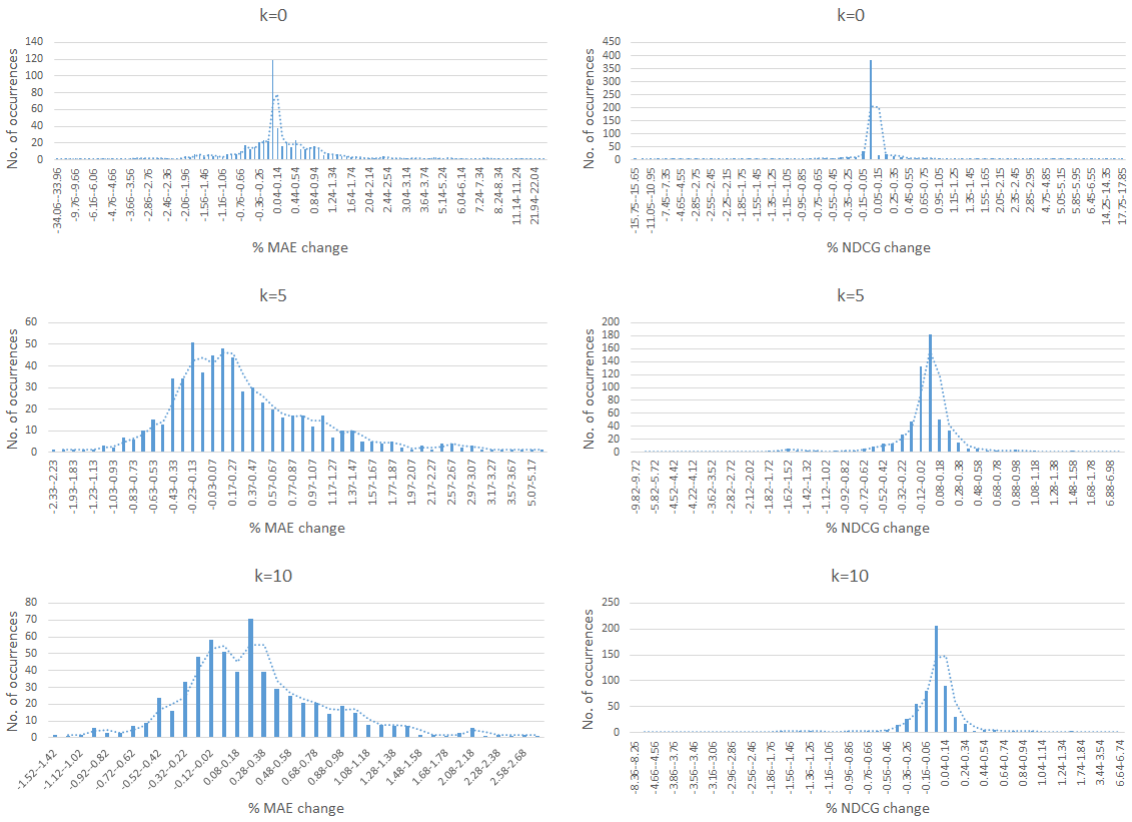

Fig. 6. MAE (left) and NDCG (right) results on ml-latest-small using the neighborhood-based mechanism with different neighborhood sizes.

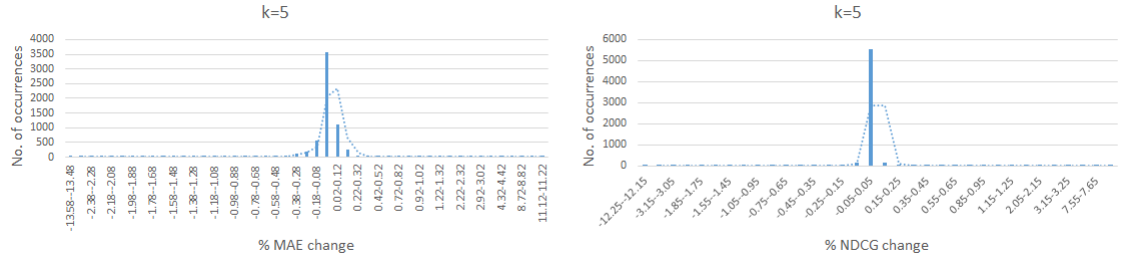

Fig. 7. MAE (left) and NDCG (right) results on ml-1m using the neighborhood-based mechanism with $k = 5$.

horror and comedy are given very high ratings before going back to normal. A very similar user is selected from the ml-1m dataset. Finally, we note that Badran et al. [22] and Al Jurdi et al. [23] had similar observations pertaining serendipity and the item discoveries that users undergo in RSs. Based on their discussions on serendipity and the difference between it and noise, we can predict how a normal user profile might generally vary due to certain serendipitous discovery. This is shown in Figure 5 (bottom).

C. Effect on the System - Experiment results

After the target malicious ratings have been identified for two users in both datasets, we test the effect on the system with two methods, the first is the conventional method is normal offline evaluation metrics. Such methods are used to evaluate the effectiveness of new proposals in the natural noise research [8] and data poisoning [24], [25]. The second test is the neighborhood-based evaluation mechanism, which allows evaluating at a granular level based on neighborhood clusters.

1) Impact on the System: Tables I and II summarize the results of the percentage change in the metrics before and after eliminating the identified malicious user behavior for both the conventional method and the neighborhood-based one (k in the neighborhood-based case stands for the neighborhood size). For clarity, the results for every neighborhood, Figures 6 (ml-latest-small) and 7 (ml-1m) show the percentage MAE and NDCG change for several values of k and for the case of eliminating the target malicious ratings only. First, it's clear from the data in Table I that the normal method does not

TABLE I
THE EFFECT OF MALICIOUS RATING REMOVAL USING THE OVERALL SYSTEM METRIC RESULTS.

Dataset	Metric	Test case	
		Legitimate target ratings (% change)	Attack/Noise target ratings (% change)
ml-latest-small	MAE	0.01	-0.01
	RMSE	0.03	-0.01
	NDCG@10	0	0.02
ml-1m	MAE	-0.01	0.31
	RMSE	0.01	0.3
	NDCG@10	0.1	-0.04

report any significant impact on the system before or after the removal of the target malicious ratings in the two datasets. The change is nominal and registers a mere 0.01% decrease in MAE and RMSE after the attack in the case of ml-latest-small with a 0.02% increase in NDCG@10. In the case of ml-1m, the removal of the attack caused a somewhat opposite effect from the ml-latest-small dataset with around 0.3% increase in MAE and RMSE as opposed to a 0,04% decrease in NDCG@10. Finally, and as expected, the results of the malicious target ratings case are very close to that of the legitimate ratings, which also resulted in minor variations after the rating removal (Table I), and one cannot even know that something might be wrong with the data due to such marginal effects. The malicious case cannot be spotted and therefore nothing appears to be wrong with the dataset in both cases.

Conversely, the neighborhood-based mechanism reported different results for the same cases on both datasets while it similarly registered the same findings for the legitimate target users case. Table II shows a relatively large fluctuation in both MAE and NDCG@10 (especially for $k = 0$) in the case of malicious rating removal. With the ml-latest-small dataset, MAE scored a significant 5% increase for several neighborhoods $k = 5$ while also registering a lower 2% decrease for others. This generally means that the ratings removal causes a slightly more oriented shift towards more accurate recommendations in many aspects of the dataset. On the other hand, MAE resulted in a more significant decrease in the case of ml-1m for the selected malicious ratings of the user. The ratings removal in this case was negatively affecting some neighborhoods rather than being more oriented towards a positive accuracy change. We speculate that there are many profiles in the ml-1m that exhibit natural and malicious noise and that could be affecting the results since they weren't eliminated. We are only evaluating the performance using the local neighborhoods of users after a very small malicious data has been eliminated. NDCG@10 registers a significant increase for $k = 0$ after the attack and a slightly less decrease for the others. This has to do with the nature of the ranking-based evaluation method and the way Discounted Cumulative Gain (DCG) measures the relevance of ratings in the dataset. Lastly, it is worth noting that the percentage change decreases as the value of k increases and gradually diverges towards the percentage change for $k = N$ (where N is the total number of users in a dataset).

2) Impact on the Neighborhood Recommendations: In the second part of the experiment, we convey a closer look at effect on the neighbors of the target users. Table III shows the impact on the neighborhood of the users after the malicious ratings were eliminated. In both cases, the neighbors of the users markedly changed after the malicious rating removal. As shown in the Table, the first user (case of ml-latest-small) shows a very small similarity (5.26%) between his two neighborhoods while the second registers no similarity at all. We can safely say that both target users ended up with a totally new neighborhood after the malicious ratings were eliminated. As for the recommendations for the target users before and after correction, the order and the content varied significantly as clearly shown in the same Table. The new neighborhood yielded four new items in the top-10 of the first case while six new items in the second case. Those new recommendations are now more convenient to the user's authentic profile after the malicious data has been eliminated.

As signaled by the neighborhood results in the previous section, the major effect of malicious ratings of the users in both datasets lies in the recommendations of the neighborhood of the target users and not just their own recommendations. For that, we analyzed the recommendations of the neighborhood before and after the malicious data and we found that they were indeed affected. Table IV shows the most affected neighbor for both users and it can be seen that the similarity between the items has considerably changed in both cases. The first was presented with five new unique items in his top-10 after the correction while the second six new items. The order of the items in the top-10 list has also changed drastically proving the results of the neighborhood evaluation in the previous section. Digging further, we find that those recommendations differ a lot in terms of content. Figure 8 displays the types of the items (genres) for the most effected users before and after the correction. It is eminent how the recommendations of one of the top neighbors of the user in the ml-1m case completely shifted from Drama, Romance and Comedy to Mystery and Adventure when his profile was corrected. The most affected neighbor in the ml-latest-small case also registered a difference in the recommended content where the new top-10 items are more oriented towards Drama, Action and Western as opposed to Romance, Comedy and Thriller. It's important to note that in both cases, there were

TABLE II
THE EFFECT OF MALICIOUS RATING REMOVAL USING THE NEIGHBORHOOD-BASED MECHANISM WITH DIFFERENT NEIGHBORHOOD SIZES.

Dataset	Metric	Measure	Test Case					
			Legitimate target ratings (% change)			Attack/Noise target ratings (% change)		
			0	5	10	0	5	10
ml-latest-small	MAE	Avg	-0.0028	-0.0158	-0.0049	0.2104	0.302	0.2349
		Std	0.243	0.194	0.178	3.143	0.828	0.609
		Max	1.05	0	0	22.01	5.13	2.66
		Min	-1.78	-1.02	-0.8	-21.73	-2.33	-1.52
	NDCG@10	Avg	0.0038	-0.0145	-0.0042	0.0739	-0.1884	-0.1701
		Std	0.067	0.167	0.081	2.138	1.007	0.868
		Max	1.16	0.52	0.39	17.81	2.09	3.47
		Min	-0.27	-2.3	-1.27	-15.75	-9.82	-8.36
ml-1m	MAE	Avg	0.002	0.0023	0.0017	-0.0041	-0.033	-0.038
		Std	0.105	0.041	0.034	0.422	0.388	0.288
		Max	1.42	0.6	0.68	8.62	11.2	6.47
		Min	-1.17	-0.3	-0.19	-15.19	-13.58	-5.92
	NDCG@10	Avg	0.0005	-0.0002	-0.0007	0.0036	-0.0014	0.0039
		Std	0.046	0.042	0.039	0.5	0.288	0.155
		Max	1.27	0.68	0.57	12.03	9.12	5.04
		Min	-1.13	-1.38	-1.27	-17.2	-12.25	-3

TABLE III
EFFECT OF MALICIOUS NOISE REMOVAL ON THE NEIGHBORHOOD AND THE RECOMMENDATIONS.

User Case	Correction Status	Ttop-10 Neighborhood	Recommendations (item order)
ml-latest-small	Before	{53,175,154,496,366,87,319,214,25,138}	{1,2,3,4,5,6,7,8,9,10}
	After	{2,8,11,12,26,31,35,37,44,53}	{1,2,4,11,6,9,10,12,13,14}
	% Similarity	**5.26**	**46.7**
ml-1m	Before	{556,88,276,25,595,72,550,515,53,511}	{1,2,3,4,5,6,7,8,9,10}
	After	{2,7,10,11,12,13,14,26,29,31}	{11,6,7,12,4,5,13,14,15,16}
	% Similarity	**0**	**25**

TABLE IV
EFFECT OF MALICIOUS NOISE REMOVAL ON THE TOP-10 RECOMMENDATIONS OF THE NEIGHBORHOOD OF THE TWO TEST USERS.

User Case	Most Affected Neighbor	Malicious Ratings Status	Recommendations (item order)
ml-latest-small	276	Before	{1,2,3,4,5,6,7,8,9,10}
		After	{1,2,4,5,11,12,13,14,15,6}
		% Similarity	**33.34**
ml-1m	154	Before	{1,2,3,4,5,6,7,8,9,10}
		After	{11,1,12,13,2,3,14,5,15,16}
		% Similarity	**26.7**

new genres that popped up heavily in the recommendations after the correction was made and they are Mystery and Crime for that in the first user's neighborhood and Western and Action for that in the second user's neighborhood. This shows how minor malicious variations that generally go undetected can affect the true preferences of the neighborhoods.

V. CONCLUSION AND FUTURE WORK

Implementing an effective and agile natural noise management algorithm for recommenders is challenging due to numerous parameters that ought to be taken into consideration, especially in the evaluation process. As demonstrated in this study, the obfuscation phenomenon created yet another challenge to the evaluation process. We have introduced two modern forms of noise that are hard to detect with current evaluation strategies and showed how the data appears to be perfectly normal. The impact was only visible when we evaluated the performance in data subgroups using the new proposed group validation process in the evaluation ecosystem of recommenders. Additionally, there has been no attempt to synthesize what is known about the various categories of noise in RSs, nor to systematically devise a unified protocol that would be able to deal with noise irrespective of its type and independent of the deployed recommendation engine. Whether it's user-induced for the purpose of simply opting-out for certain security concerns, or publicly injected by authorities such as the case of Russia, Mexico, and Lebanon, Obfuscation is a challenge that RSs should be aware of.

Opt-out attacks pave the way for multiple discussion paths that cover numerous topics; for instance, identifying a user's opt-out behavior can permit tracing back to the primary user

Fig. 8. Top-10 list genres before and after the malicious ratings removal for the most affected neighbors (276 - left and 154 - right).

tastes. Additionally, data owners can develop data mining methods to discover the general trends of users opting out of the online platform.

VI. Acknowledgement

This work has been supported by the EIPHI Graduate School (contract "ANR-17-EURE-0002") and the Lebanese University Research Program.

References

[1] C. Aggarwal "Recommender systems (Vol. 1)", Cham: Springer International Publishing, 2016.

[2] M. Scholz, V. Dorner, G. Schryen, and A. Benlian, "A configuration-based recommender system for supporting e-commerce decisions", European Journal of Operational Research, 259(1), 205-215, 2017.

[3] S. Meyer, "Understanding the COVID-19 Effect on Online Shopping Behavior", https://www.bigcommerce.com/blog/covid-19-ecommerce/, bigcommerce, April 2022.

[4] C. Schoenauer, "Buying behavior after COVID-19: e-commerce boom will remain", https://www.the-future-of-commerce.com. The Future of Customer Engagement and Experience, April 2022.

[5] L. Columbus, "How COVID-19 Is Transforming E-Commerce", https://www.forbes.com/sites/louiscolumbus/2020/04/28/how-covid-19-is-transforming-e-commerce/#610c38c23544. The Future of Customer Engagement and Experience, April 2022.

[6] J. Smaros and M. Falck, "The New Normal in Ecommerce after COVID-19", https://www.relexsolutions.com/resources/the-new-normal-in-ecommerce-after-covid-19/. The Future of Customer Engagement and Experience, April 2022.

[7] F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook", Recommender systems handbook - 978-0-387-85820-3, Springer, 2011.

[8] W. Al Jurdi, J. Bou abdo, J. Demerjian, and A. Makhoul, "Critique on Natual Noise in Recommender Systems", ACM Transactions on Knowledge Discovery from Data (TKDD), 2021.

[9] I. Gunes, C. Kaleli, A. Bilge, and H. Polat, "Shilling Attacks Against Recommender Systems: A Comprehensive Survey", Kluwer Academic Publishers, Norwell, MA, USA, V42-N4, December 2014.

[10] X. Amatriain, J. Pujol, and N. Oliver, "I Like It... I Like It Not: Evaluating User Ratings Noise in Recommender Systems", User Modeling, Adaptation, and Personalization, Springer Berlin Heidelberg, p:247-258, ISBN:978-3-642-02247-0, 2009.

[11] X. Amatriain, J. Pujol, N. Tintarev, and N. Nuria, "Rate it again: Increasing recommendation accuracy by user re-rating", Human-Computer Interaction, Springer Berlin Heidelberg, p:173-180, DOI:10.1145/1639714.1639744, December 2009.

[12] S. Mingdan and Q. Li, "Shilling attacks against collaborative recommender systems: a review", Artificial Intelligence Review, DOI:10.1007/s10462-018-9655-x, September 2018.

[13] S. Badsha, X. Yi, and I. Khalil,"A Practical Privacy-Preserving Recommender System", Data Science and Engineering, DOI:10.1007/s41019-016-0020-2, September 2016.

[14] H. Cai and S. Gambs, "Detecting shilling attacks in recommender systems based on analysis of user rating behavior", PLoS ONE, p:22-43, 2019.

[15] F. Zhang, V. Lee, R. Jin, S. Garg, and K. Choo, "Privacy-aware smart city: A case study in collaborative filtering recommender systems", Journal of Parallel and Distributed Computing, DOI:10.1016/j.jpdc.2017.12.015, 2018.

[16] A. Comninos, "E-revolutions and cyber crackdowns: User-generated content and social networking in protests in MENA and beyond", Association for Progressive Communications, Chapter 2, 2011.

[17] Deloitte, "Global Mobile Consumer Survey, US Edition", https://www2.deloitte.com/tr/en/pages/technology-media-and-telecommunications/articles/global-mobile-consumer-survey-us-edition.html, Deloitte, 2017.

[18] R. Toledo, M. Yera, Y. Caballero, and L. Martínez, "Correcting noisy ratings in collaborative recommender systems", Knowledge-Based Systems, Elsevier, p:96-108, 2015.

[19] M. Badran, W. Al Jurdi, and J. Abou Abdo, "Survey on shilling attacks and their detection algorithms in Recommender Systems", Proceedings of the International Conference on Security (SAM), ACM, p:141-146, 2019.

[20] F. Brunton and H. Nissenbaum, "Obfuscation A User's Guide for Privacy and Protest", The MIT Press, ISBN:9780262029735-9780262529860, 2015.

[21] F. M. Harper and J. Konstan, "The movielens datasets: History and context", Acm transactions on interactive intelligent systems (tiis), p:1-19, 2015.

[22] M. Badran, J. Bou Abdo, W. Al Jurdi, and J. Demerjian, "Adaptive Serendipity for Recommender Systems: Let It Find You", Proceedings of the 11th International Conference on Agents and Artificial Intelligence, ICAART 2019, Volume 2, Prague, Czech Republic, p:739-745, DOI:10.5220/0007409507390745.

[23] W. Al Jurdi, M. Badran, C. Abou Jaoude, J. Bou Abdo, and J. Demerjian, "Serendipity-Aware Noise Detection System for Recommender Systems", Int'l Conf. Information and Knowledge Engineering, 2019.

[24] B. Li, Y. Wang, A. Singh and Y. Vorobeychik, "Data poisoning attacks on factorization-based collaborative filtering.", arXiv preprint arXiv:1608.08182. 2016.

[25] M. Fang, G. Yang, N. Gong, and J. Liu, "Poisoning attacks to graph-based recommender systems", Proceedings of the 34th Annual Computer Security Applications Conference. 2018.

[26] C. Aggarwal, "Recommender Systems: The Textbook", Springer Publishing Company Incorporated, ISBN:3319296574-9783319296579, 2016.

[27] J. Herlocker, J. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating Collaborative Filtering Recommender Systems", ACM Trans. Inf. Syst., DOI:10.1145/963770.963772, January 2004.

[28] W. Al Jurdi, J. Bou Abdo, J. Demerjian and A. Makhoul, "Group Validation in Recommender Systems: Framework for Multi-layer Performance Evaluation", DOI:10.48550/ARXIV.2207.09320, 2022