Robustness of Bisecting *k*-means Clustering-based Collaborative Filtering Algorithm

Alper Bilge and Huseyin Polat
Computer Engineering Department
Anadolu University
Eskisehir, Turkey
emails: {abilge, polath}@anadolu.edu.tr

***Abstract*—The unprecedented popularity of e-shopping amenities provided by online retailers escalates attention to recommendation facilities. Collaborative filtering is one of the well-known recommendation techniques that helps customers choose possible products of interest by automating word-of-mouth habits. However, due to their nature, recommendation algorithms are open to shilling attacks of malicious users to promote/demote certain products. We propose bisecting *k*-means clustering-based recommendation algorithm as a robust algorithm in non-private environments against well-known shilling attacks. We investigate its robustness against shilling attacks by performing real databased experiments. We also analyze the effects of varying attacking parameters. We empirically establish that the algorithm is resilient against shilling attacks without significantly influenced by malicious profiles.**

Keywords–robustness; shilling; clustering; recommendation.

I. INTRODUCTION

With increasing amount of information available in everyday life through widespread use of the Internet, Collaborative Filtering (CF) systems have become one of the most practical tools to determine useful information. Such systems are very successful to cope with information overload problem. CF algorithms are efficient in automating word-of-mouth habits of individuals by collecting preference information about products such as movies, music CDs, books, and so on. Typically, CF systems hold a user-item matrix containing ratings of users on products and whenever a user requests for a prediction on a target product, the system produces an estimation as a weighted average of similar users' ratings on the target product.

CF systems are usually unable to strictly distinguish genuine profiles from malicious ones. Thus, they are vulnerable to potential manipulations. Either malicious users or competing companies might intrude bogus profiles into the database in order to favor or disfavor a certain product's popularity [1]. Such intrusions are called shilling attacks, which can be categorized as push or nuke attacks according to their intent [2]. Determining fake profiles and being robust against them is critical for the success of CF algorithms. Shilling attacks have been shown to be very effective against traditional memory based CF schemes [3][4]. However, clustering-based approaches are successful in distinguishing shilling profiles from genuine ones because bogus profiles expose high resemblance among themselves, which makes them to be clustered mostly together [1][5][6]. Hence, clustering-based methods are preferable over other schemes in order to achieve required level of robustness in CF systems.

There are some common requisites of CF systems such as accuracy, scalability, and robustness. A qualified CF algorithm is required to produce personalized predictions with decent accuracy to please customers and increase online sales. Moreover, due to constantly enlarging dimensions of user-item matrix, such algorithms should be resistant against scalability issues. Finally, it is expected for the algorithms not to be significantly affected by shilling attacks and be robust against them arising from their data collection nature. In the literature, there are various techniques developed to enhance quality of produced predictions by modifying similarity calculation methods [7] and handling sparse user profiles [8]. Some researchers proposed several CF algorithms to overcome scalability issues using matrix factorization [9][10], dimensionality reduction [11][12], and clustering techniques [13][14]. And finally, some model-based techniques have been shown to be resistant against shilling attacks [1][4].

Although the essential constraints of CF systems are discovered, it is hard to claim that there exists an eligible CF algorithm fulfilling all of them. Memory-based CF schemes are very successful in producing high quality referrals. However, they suffer from scalability issues and they are vulnerable to shilling attacks [15]. Model-based and hybrid CF methods are generally scalable and more resistant against shilling attacks; however, they commonly compromise from accuracy and often come with high computational cost for model update [1][16].

A scalable, low cost, and easy-to-interpret CF algorithm is proposed to produce highly accurate predictions in both non-private and privacy-preserving CF environments [17]. Its robustness against shilling attacks in private environments is also investigated [18]. However, such algorithm is not investigated with respect to robustness in non-private environments. Since clustering-based CF algorithms are successful in grouping bogus profiles together, we hypothesize that bisecting *k*-means clustering-based algorithm is robust against shilling attacks in non-private environments.

The paper is organized as follows. Section 2 discusses relevant literature and describes shilling attacks. We explain how bisecting *k*-means clustering-based CF operates on non-privately collected databases and discuss how shilling attacks can be implemented to modify its outputs in Section 3. Section 4 experimentally evaluates the robustness of the algorithm against shilling attacks in non-private environments. Finally, conclusions as a brief discussion and future research directions are presented in Section 5.

 ISBN: 978-1-61208-415-2

II. Related Work and Preliminary concepts

CF idea was first coined by the Tapestry system, which was utilized as a filtering tool for e-mails [19]. Contemporary CF technologies are integrated as recommender systems by online shopping amenities operating on preference data to produce personalized predictions [20]. Applications of CF schemes span from filtering e-mails [19] to Web service recommendations [21] and tag-based CF schemes [22].

With increasing popularity of CF systems, several attacking mechanisms arise to manipulate their outputs in favor of particular products. Dellarocas [23] inspires manipulation attacks to recommender systems, where some mechanisms are defined to avoid fraud in online reputation reporting systems. O'Mahony et al. [24] discuss vulnerabilities of automated prediction estimation process against manipulations. The authors describe the amount of information needed about the database to realize effective shilling attacks. Lam and Riedl [2][25] analyze cost of attacks and propose that there is a relation between privacy and the value of information. Several attacking methods are proposed in the literature like random, average, bandwagon, and segment attacks as push attacks [26]. Effectiveness of such attacks are investigated against memory- and model-based CF schemes [15]. Recently, Gunes et al. [27] surveyed about researches on shilling attacks and present attacks, detection methodologies, robust algorithms, and evaluation metrics.

Shilling attacks are generated by inserting fake (shilling) profiles into user-item databases. The general attack strategy is depicted in Fig. 1 [26], where I_S, I_F, and I_φ refers to selected, filler, and empty cells in the fake profile, respectively; and a unique item, i_t, is targeted. Selected items are chosen for characterizing an attack, filler items are chosen to prevent easy detection of fake profiles, and the target item is assigned either a high or a low rating value for push and nuke attacks, respectively. Shilling attacks can be used to increase the popularity of some targeted items or decrease their popularity. In order to push a prediction (increase the popularity of a target item), the target item is assigned a high rating. For decreasing the popularity of a target item, it is assigned a low rating.

Bisecting *k*-means clustering-based privacy-preserving recommendation algorithm is proposed to be easily scalable method and it produces predictions with high accuracy [17]. Notice that clustering-based CF schemes seem to be robust CF schemes without privacy concerns due to clustering nature. Hence, bisecting *k*-means clustering-based scheme might be appropriate proposal for being a robust algorithm. In our previous study [18], we investigated the robustness of privacy-preserving bisecting *k*-means clustering-based recommendation scheme against shilling attacks. In this study, we hypothesize that bisecting *k*-means clustering-based CF algorithm might be robust against shilling attacks due to its clustering nature in non-private environments, as well. Thus, we investigate its robustness against shilling attacks in non-private environments. We also provide comparisons between the proposed method and previously presented robust approaches in terms of obtained prediction shifts, algorithm interpretability, and model update costs. We focus on the robustness analysis of bisecting *k*-means clustering-based CF method against shilling attacks. As stated previously, bisecting *k*-means clustering-based recommendation algorithm is proposed as an accurate and scalable method. In this study, we want to show that it is also robust against shilling attacks in non-private environments.

III. A Robust Recommendation algortihm

Due to the reason that recommender systems are open for public usage and therefore vulnerable to manipulations, both non-private recommendation algorithms need to have robust mechanisms to estimate predictions. However, the state-of-the-art memory-based CF schemes are not resistant to such attacks and exposed to significant shifts in predicted values. In this section, we describe non-private bisecting *k*-means clustering-based recommendation scheme, designations of four push and two nuke attacking strategies against unmasked databases, and explain how the proposed algorithm is expected to perform in a robust manner.

A. Bisecting k-means Recommendation Algorithm

Bisecting *k*-means clustering-based recommendation estimation is first proposed by Bilge and Polat [17] in order to produce personalized recommendations over plain and disguised databases. In the proposed non-private scheme, the central server collects original user vectors and forms a user-item matrix $U_{n\times m}$, where *n* and *m* represent number of users and items, respectively. At the beginning, the server forms a binary decision tree off-line by utilizing bisecting *k*-means clustering algorithm on the database. Given the database and an optimal value of number of neighbors (*N*), *k*-means clustering is applied to divide the matrix *U* into two clusters at each level (hence, it is called bisecting) and cluster centers are indexed to be used as a forwarding tool for each corresponding level. If number of users in any cluster exceeds *N*, then such clusters are continued to be divided recursively into subsets via *k*-means clustering. Finally, a binary decision tree is obtained having indexed cluster centers as branch nodes and grouped neighbor users at leaf nodes. The tree, in general, continues growing in such a way so that if any leaf node population exceeds the stopping criterion, the server immediately bisects that leaf node to grow. Therefore, it is a continual process to update the decision tree, which saves the central server to form the binary decision tree each time a user included in the system. Such mechanism enhances system maintainability and reduces model generation costs.

An example binary decision tree produced by the algorithm is presented in Figure 2, where initially there are 150 users and the stopping criterion is determined as 20 users. At the beginning, the algorithm divides 150 users into two clusters having 73 ad 77 users and cluster centers are indexed at the root as C_1^L and C_1^R for the left and right subtrees, respectively. Such process continues recursively for each subtree and cluster centers are recorded to be used for forwarding purposes until the algorithm reaches leaf nodes containing at most 20 users.

When an active user (*a*) asks a prediction, instead of calculating similarities with all users, the server only forwards the active user according to her similarity to two cluster centers at each level. By doing so, the leaf node that the user belongs is determined through forwarding. While traversing, two similarity calculations are performed at each level, where higher similarity determines next hope (either right or left). Although depth of binary decision tree (*d*) is dependent on n, intuitively, it is much less than *n* in large recommender systems suffering from scalability. Therefore, after the tree is formed, at most $2 \times (d - 1) + N$ similarity computations are performed

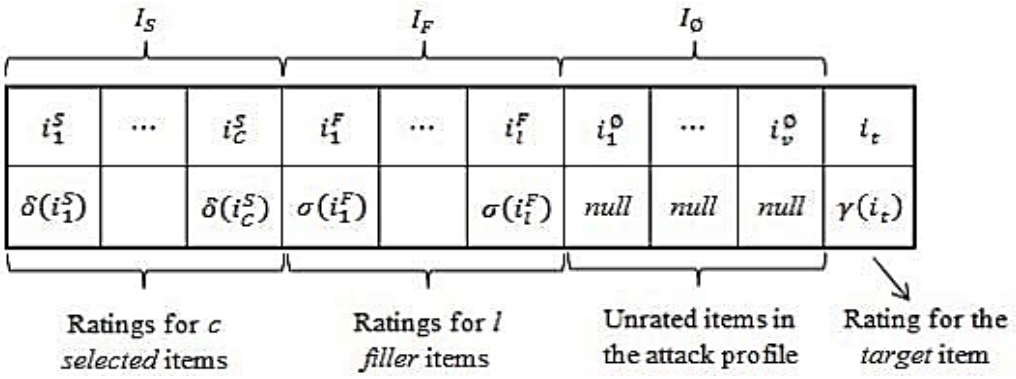

Figure 1. General form of an attack profile.

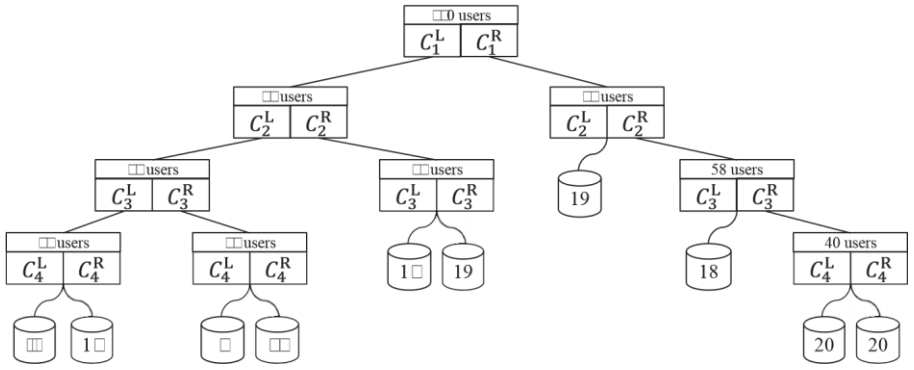

Figure 2. An example binary decision tree.

instead of *n* to form a neighborhood. Finally, the leaf node that the new user belongs is determined and all users in that corresponding node are regarded as neighbors. Then, a prediction is calculated as a weighted average of neighbors' ratings on target item as formulated in (1) and returned to *a* as a prediction.

$$p_{aq} = \overline{v_a} + \frac{\sum_{u \in N}(v_{uq} - \overline{v_u}) \times w_{au}}{\sum_{u \in N} w_{au}} \quad (1)$$

in which p_{aq} is the prediction for *a* on target item *q*, $\overline{v_a}$ and $\overline{v_u}$ are mean rating of *a*'s and *u*'s ratings, respectively, v_{uq} is the rating of *u* on item *q*, *N* is the set of neighbors, and w_{au} is the similarity weight between *a* and neighbor *u*.

B. Shilling Attack Strategies for Plain Databases

Shilling attacks have impacts on accuracy of the produced predictions. Attackers generate bogus profiles, assign their target items to maximum or minimum vote according to intends and insert them into the databases. Thus, they manipulate popularities of the target items in favor of themselves. Shilling attacks can be designed for pushing or nuking popularities of items. In order to perform manipulations, the attackers require low or high knowledge about the system. As part of their generic form depicted in Fig. 1, four push and two nuke attacks covered in this paper can be described as in the following [15]:

Random attack (RN). Random attack can be considered as a baseline *push* attack model, which requires quite minimal knowledge. Selected items set is empty and arbitrarily chosen filler items set is filled with random values drawn from a distribution with overall system mean and standard deviation for attacking non-private systems. The target item is assigned the maximum rating available in the system for non-private schemes.

Average attack (AV). Average attack is a more effective *push* attack model focusing on each item's individual mean rather than overall system's mean. Cost of this attack is related to the number of filler items in the attack profile because average votes of such items are required. Selected items set is empty and each arbitrarily chosen filler item is filled with a random value drawn from a distribution with corresponding item's ratings mean and standard deviation for attacking non-private systems. The target item is assigned the maximum rating available in the system for non-private schemes.

Bandwagon attack (BW). As a *push* attack model, bandwagon attack focuses on items that are attracting remarkable attention by many users to manipulate people who are prone to purchase such bestselling products. Selected items set consists of popular and densely-rated items having high averages. For attacking non-private systems, such selected items are given the maximum available rating, filler items are assigned random votes, and the target item is assigned the highest rating.

Segment attack (SG). Segment attack is designed as a *push* attack model for relatively robust item-based algorithms focusing on a subset (segment) of users who are likely to purchase certain kinds of products rather than attacking all users in the system. Selected items are chosen from high average items with a certain property (such as horror movies or jazz music). For non-private systems, such

selected items are assigned the maximum rating value, filler items are given the minimum rating value, and the target item is assigned the highest vote in order to push its popularity.

Reverse bandwagon attack (RBW). Reverse bandwagon *nuke* attack model is the inverted version of bandwagon push attack model. Selected items are chosen among unpopular items (having low means) rated by many users. For attacking non-private systems, such selected items are given the minimum available rating, filler items are assigned random votes, and the target item is assigned the lowest rating.

Love/hate attack (L/H). Love/hate attack is a very simple *nuke* attack model, which requires no knowledge about the system. For non-private systems, selected items set is empty and arbitrarily chosen filler items are assigned the highest available rating values while the target item is given the minimum vote.

C. Robustness Utility of the Recommendation Algorithm

Generally speaking, an attacker can attack any CF system by creating bogus profiles according to her intends as explained previously and sending them to the system. Thus, specifically, in order to attack non-private bisecting *k*-means clustering-based recommendation scheme, the attacker produces attack profiles and inserts them into the system. Since any CF scheme is vulnerable against shilling attacks, how well the scheme performs under such attacks is imperative for overall success. In other words, being robust against shilling attacks and/or able to detect bogus profiles are important.

In the previous studies [1][5][24], clustering-based CF schemes are shown to be successful in detecting fake profiles or bogus profiles. Arising from its utility of gathering similar data items together, clustering is utilized as a detection tool for shilling attacks in non-private schemes. O'Mahony et al. [24] utilize clustering as a neighborhood elimination method, where suspicious users are excluded from the system by clustering the database periodically to check if significant changes occur in memberships and cluster centers. If such significant changes occur, extreme profiles disturbing cluster centers are marked as malicious profiles. Bhaumik et al. [5] utilize *k*-means clustering with several classification attributes for attack detection. They show that shilling profiles show high resemblance to each other; therefore, when they are clustered, they tend to move together into the same and mostly small clusters. Especially, as initially determined number of clusters decrease, the likelihood of attack profiles gathering together increases.

Successful clustering-based schemes with respect to shilling attack detection inspire us to hypothesize that bisecting *k*-means clustering-based scheme can be proposed as a robust prediction algorithm. In addition to malicious profile detection, we hypothesize that clustering method can be utilized to offer robust recommendation algorithms. Relying on the results of [5], we hypothesize that elimination by clustering intuition works best for clustering into two groups to move shilling profiles together. In addition, applying such clustering repeatedly is supposed to eliminate all shilling profiles after some level of the produced binary decision tree. Therefore, we claim that malicious profiles substantially distinguishes from genuine ones after a particular level of the tree and it becomes very unlikely for any active user belonging to a leaf node consisting of shilling profiles. As a result, the proposed recommendation scheme is expected to perform robust against shilling attacks. To verify our hypothesis, we performed real data-based experiments as explained in the following section.

IV. Experimental evaluation

After explaining how shilling attacks can be implemented over non-private bisecting *k*-means recommendation algorithm, we conducted real data-based experiments to scrutinize the robustness of the scheme. We also investigated the effects of shilling attacks with respect to two control parameters. The control parameters, filler size and attack size, are defined for designing effective shilling attacks. Filler size parameter indicates the percentage of cells to be filled with fake ratings while creating the attack profiles. Attack size can be described as the pre-attack profile count proportional to the number of users in the database. We conducted various experiments for non-private bisecting *k*-means clustering-based CF scheme with varying values of the explained parameters.

A. Experimental Settings and Methodology

In the following experiments, publicly available MovieLens (ML) data set, which was collected by GroupLens [30], was utilized. It is the most widely used and well-known real collection for CF purposes. It holds 100K ratings from 943 users on 1,682 movies and the rating range allows 5-star discrete numeric values.

We used prediction shift metric in order to measure the prediction alterations due to the effects of shilling attacks. Prediction shift can be described as the average change in the prediction for the attacked item before and after the attack profiles are included.

During the experiments, we followed all-but-one experimentation methodology, which enables full utilization of the data set. This methodology considers one of the users as the active user *a* and the rest of the set as the training users at each iteration. The utilized attacks target two separate sets of 50 movies for push and nuke attacks. Those sets for push and nuke attacks were constructed selecting arbitrarily from different rating ranges to represent characteristics of the original data set. Since it is unreasonable to push a popular item with high ratings or similarly nuke an unpopular item, we principally selected items with low mean values to push and high means to nuke. Table I shows the statistics of 50 target movies for push and nuke attacks, where each value indicates how many of the movies fall into corresponding group.

In the experiments, all target items were attacked individually for all users in the system. Binary decision trees were constructed by omitting and including fake shilling profiles. Then, predictions were estimated based on the produced binary decision trees and prediction shift values were observed to show relative change on predicted values for different shilling attacks. The stopping criterion for building binary decision trees was set to 30. Although varying stopping criterion might alter obtained prediction shift values especially with varying attack sizes, we fixed such parameter due to page limitations and discuss algorithm's robustness performance relying on a constant stopping condition value. We exclusively presented the obtained results for push and nuke attacks in the following sections.

 ISBN: 978-1-61208-415-2

TABLE I. STATISTICS OF TARGETED MOVIES

Ratings	*Pushed Items* 1–2	*IItems* 2–3	*Nuked Items* 3–4	4–5
1–50	30	15	12	18
51–150	—	3	5	6
151–250	—	1	2	3
250 and up	—	1	1	3

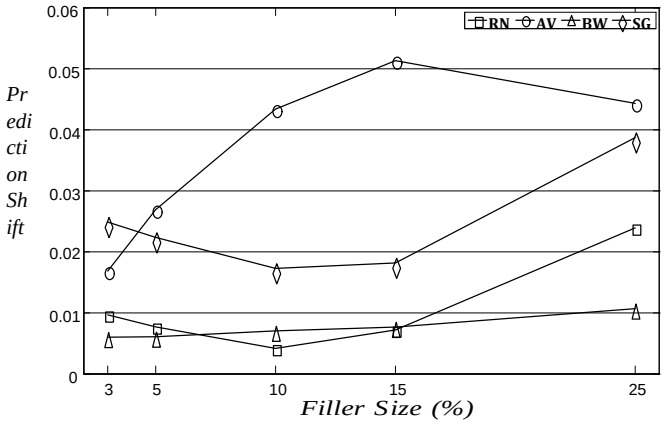

Figure 3. Prediction shift vs. filler size for push attacks.

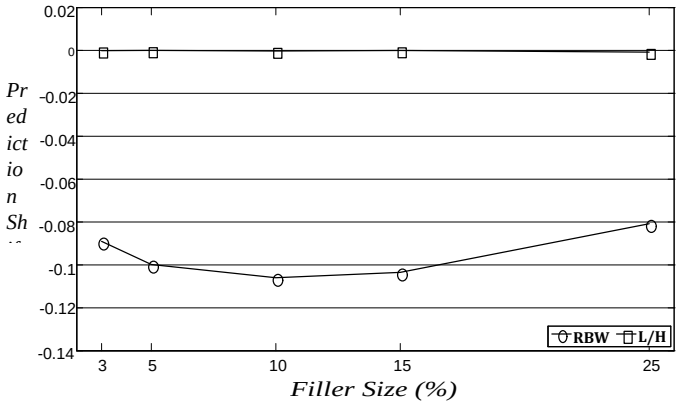

Figure 4. Prediction shift vs. filler size for nuke attacks.

B. Robustness Analysis of Non-private Scheme

1) Effects of filler size parameter: We first conducted experiments to show how varying filler size values affect the robustness of the non-private bisecting *k*-means clustering-based prediction scheme with respect to four push and two nuke attack models. Notice that filler size parameter indicates the number of fake votes for the filler items added to fill the attack profile; and thus, it is directly related to the success of the attack. To observe how varying filler size values affect robustness, we fixed attack size at 15% while we changed filler size from 3% to 25%. User-item matrix was attacked by four push and two nuke attack models. We estimated prediction shift values and displayed the overall averages for push and nuke attack models in Fig. 3 and Fig. 4, respectively.

As seen from Fig. 3, none of the four push attack models are able to achieve a significant prediction shift for varying filler size values. Generally speaking, with increasing filler size values, the effects usually become larger; however, increasing the value of filler size more is not feasible for the sake of detection of the attacks. The maximum prediction shift is observed for average attack when filler size is 15%. Compared to random and bandwagon attacks, average and segment attacks work better. However, their effects on the robustness of the scheme is still negligible because the maximum prediction shift is about 0.05 only. For bandwagon attack, changes in prediction shift values with increasing filler size values are very stable even if prediction shift values become larger. With increasing filler size values from 3% to 15%, there are notable changes in prediction shift values for average attack. As stated before, they are still insignificant assuming that the overall mean absolute error for the scheme is about 0.70. Therefore, we can conclude that bisecting *k*-means clustering-based prediction algorithm is robust against push attacks in non-private environments.

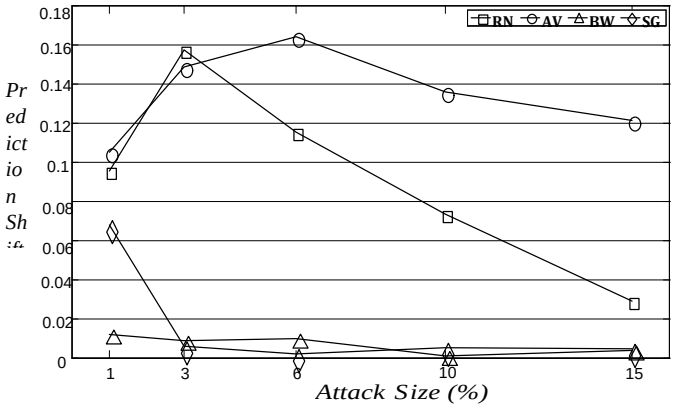

Figure 5. Prediction shift vs. attack size for push attacks.

The results in Fig. 4 show that nuke attack models are not effective against the non-private scheme with respect to varying filler size values. Changes in prediction shift values due to love/hate attack with increasing filler size values are insignificant. In other words, prediction shifts due to such attack are almost zero. Thus, love/hate attack is completely ineffective. Unlike love/hate attack, reverse bandwagon attack causes manipulations and it is more effective than love/hate attack. The maximum prediction shift value is about 0.1 for reverse bandwagon attack. Prediction shift values increase with increasing filler size values up to 10%, and then they decreases. However, such changes can be considered negligible due to the rating range. Hence, we can again conclude that bisecting k-means clustering-based prediction algorithm is resistant against nuke shilling attacks in non-private environments.

2) Effects of attack size parameter: We then performed various trials to show how varying attack size values affect the robustness of the non-private bisecting *k*-means clustering-based prediction scheme with respect to four push and two nuke attack models because in addition to filler size, attack size is another control parameter. Also note again that attack size determines the number of inserted attack profiles; thus, it is also vital in realizing significant manipulations. In order to evaluate the robustness of the non-private scheme with respect

to varying attack size values, we set filler size to 15% while we changed attack size from 1% to 15%. We again estimated prediction shift values and displayed the overall averages for push and nuke attack models in Fig. 5 and Fig. 6, respectively.

As seen from both figures, attack size is more effective than filler size parameter. The outcomes in Fig. 5 demonstrate that the most effective push attack in terms of attack size is average attack. The next most effective attack is random attack. Compared to both average and random attacks, segment and bandwagon attacks can be considered ineffective against the non-private method. Segment and bandwagon attacks cause stable changes in predictions with increasing attack size values. Almost all attack size values, prediction shifts for such attacks are very close to 0.01, which is negligible. Therefore, we can infer that our scheme is very robust against them and they do not significantly cause any manipulations. Although average and random attacks cause manipulations, the maximum shift is about 0.16 when the attack size is 6%. With increasing attack size values from 6% to 15%, prediction shift values for average and random attacks become smaller. The outcomes, in general, demonstrate that the non-private scheme is robust against push attacks in terms of varying attack size values.

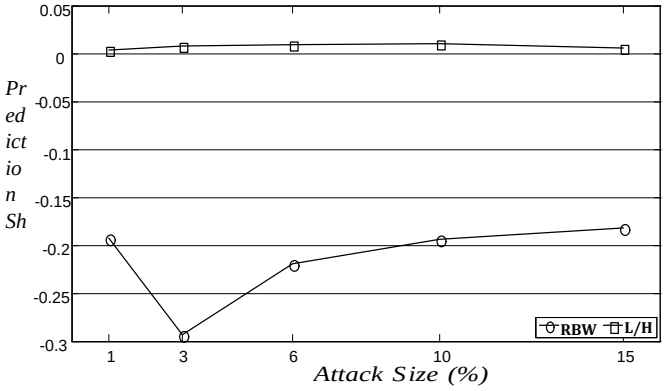

Figure 6. Prediction shift vs. attack size for nuke attacks.

Reverse bandwagon attack seems to be the most effective shilling attack, as seen from Fig. 6. When the attack size is 3%, prediction shift value reaches its maximum value, which is about 0.28. Other than 3% attack size value, predictions shift values are less than 0.20 for almost all other attack size values. Unlike reverse bandwagon attack, love/hate attack is much more ineffective. Although love/hate is used as a nuke attack and it is supposed to cause negative shifts, it causes negligible positive shifts. Moreover, changes in prediction shift values for varying attack size values for love/hate attack are stable and very close to zero. Thus, we can conclude that our scheme is very robust against love/hate attack.

C. Discussion

In addition to accuracy and scalability, robustness is also a critical requisite for recommendation algorithms. Due to its grouping nature, clustering has been used as a successful shilling attack detection method [1][5][6]. Thus, we hypothesized that bisecting *k*-means clustering-based algorithm can be proposed as a robust recommendation algorithm due to its clustering performance. We analyzed its robustness against six well-known shilling attacks (including both push and nuke attacks) in non-private environments. Bisecting *k*-means clustering-based scheme is robust in non-private environments, as shown by our real data-based trials. All of the push attacks that we scrutinize are ineffective against our scheme. Prediction shift values caused by such attacks are usually less than 0.05. In some cases, although prediction shift values reach at 0.16, they are still acceptable shifts compared to rating range. Average attack seems to be most effective push attack against our non-private method.

Like push attacks, nuke attacks can be considered ineffective against our scheme. Love/hate attack causes almost zero shifts in most of the cases. Therefore, it is not a good attack model to attack our non-private scheme. Unlike love/hate, reverse bandwagon is much more effective attack model against the non-private method. Prediction shift values due to reverse bandwagon attack reach 0.30 when attack size is set to 3%. Other than that case, prediction shifts caused by reverse bandwagon nuke attack are usually less than 0.20. Real data-based empirical outcomes demonstrate that our non-private recommendation method is robust against both push and nuke attacks. Out of six attack models, three of them (love/hate, bandwagon, and segment) are almost ineffective in many cases. Although reverse bandwagon, average, and random attacks seem to cause some prediction shifts, they are considered negligible due to the rating range.

In order to give an idea how robust our non-private scheme is, we compare it with the existing well-known recommendation algorithms with respect to robustness. According to study conducted by Mobasher et al. [28], average prediction shift values due to average attack are larger than 1.5 and 2.5 for *k*-means- and *k*-*nn*-based recommendation algorithms, respectively when attack size is 15% and filler size is 5%. For the same cases, average prediction shift values caused by segment attack are about 0.5 and 3.5 for *k*-means- and *k*-*nn*-based recommendation algorithms, respectively. Therefore, compared to them, our scheme is much more robust algorithm. Zhang et al. [29] show that prediction shift values are less than 0.003 for SVD-based prediction algorithm. Although the authors report that SVD-based scheme is a robust algorithm against shilling attacks and it is more robust than our scheme for reverse bandwagon, average, and random attacks, SVD-based model needs to be updated whenever a new user joins the system. Item-based recommendation algorithm is very susceptible against segment attack [15]. According to their empirical outcomes, average prediction shift caused by segment attack is larger than 0.9 when attack size is 15%. Similarly, bandwagon and average attacks cause more than 0.3 and 0.5 prediction shifts, respectively under the same cases. Therefore, our bisecting *k*-means clustering-based method performs better than item-based scheme with respect to shilling attacks.

V. Conclusion and Future Work

A prediction algorithm should handle various issues in order to become popular. Recommendation algorithms should provide accurate predictions, be scalable and robust, and so on. Thus, we investigated a formerly proposed accurate and scalable bisecting *k*-means clustering-based prediction algorithm's robustness against malicious shilling attacks in non-private environments. We first implemented four well-known push and two nuke attacks in non-private environments. We explained how such inserted attack profiles can affect the

recommendation scheme and why it is expected that the algorithm is robust against them. According to the obtained experimental results, the demonstrated push and nuke attack models are not able to significantly alter final predictions produced by the scheme. Thus, the algorithm is robust against shilling attacks in non-private environments. We scrutinized the effects of varying values of two control parameters like attack size and filler size. Although prediction shift values become larger as values of such parameters increase, prediction shifts are still acceptable. Our empirical results show that love/hate nuke attack is not effective against our scheme. So, even if attackers insert so many shilled profiles to our scheme, the scheme still produces accurate recommendations. Therefore, our scheme becomes more preferable than other recommendation schemes in terms of robustness in order to provide accurate predictions. Reverse bandwagon nuke attack is able to manipulate ratings; however, such manipulations are negligible. The non-private scheme performs better than item-based, *k-nn* clustering-based, and *k*-means clustering-based prediction schemes in terms of robustness against shilling attacks. Although singular value decomposition-based method is more robust than our non-private scheme, its complex model update process and model update requirement for each new user make it questionable.

It is known that clustering algorithms can be effective as a detection mechanism for shilling attacks. Hence, it warrants future work to utilize this algorithm as a detection tool of shilling profiles. Like segment attack, specific attack models can be designed as successful attacks.

ACKNOWLEDGEMENT

This work was supported by the Grant 111E218 from TUBITAK.

REFERENCES

[1] B. Mehta and T. Hofmann, “A survey of attack-resistant collaborative filtering algorithms,” IEEE Data Engineering Bulletin, vol. 31, no. 2, pp. 14–22, 2008.

[2] S. K. Lam and J. T. Riedl, “Shilling recommender systems for fun and profit,” Proc. 13th International Conference on World Wide Web, New York, NY, USA, pp. 393–402, 2004.

[3] J. Lang, M. Spear, and S. F. Wu, “Social manipulation of online recommender systems,” Lecture Notes in Computer Science, vol. 6430, pp. 125–139, 2010.

[4] B. Mobasher, R. D. Burke, R. Bhaumik, and C. A. Williams, “Effective attack models for shilling item-based collaborative filtering systems,” Proc. 2005 WebKDD Workshop, Chicago, IL, USA, 2005.

[5] R. Bhaumik, B. Mobasher, and R. D. Burke, “A clustering approach to unsupervised attack detection in collaborative recommender systems,” Proc. 7th IEEE International Conference on Data Mining, Las Vegas, NV, USA, pp. 181–187, 2011.

[6] R. D. Burke, B. Mobasher, C. A. Williams, and R. Bhaumik, “Classification features for attack detection in collaborative recommender systems,” Proc. 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, pp. 542–547, 2006.

[7] K. Choi and Y. Suh, “A new similarity function for selecting neighbors for each target item in collaborative filtering,” Knowledge-Based Systems, vol. 37, pp. 146–153, 2013.

[8] Z. Liang, X. Bo, and G. Jun, “A hybrid approach to collaborative filtering for overcoming data sparsity,” Proc. 9th International Conference on Signal Processing, Beijing, China, pp. 1595–1599, 2008.

[9] X. Luo, Y. Xia, and Q. Zhu, “Incremental collaborative filtering recommender based on regularized matrix factorization,” Knowledge-Based Systems, vol. 27, pp. 271–280, 2012.

[10] M. G. Vozalis, A. Markos, and K. G. Margaritis, “Collaborative filtering through SVD-based and hierarchical nonlinear PCA,” Lecture Notes in Computer Science, vol. 6352, pp. 395–400, 2010.

[11] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins, “Eigentaste: A constant time collaborative filtering algorithm,” Information Retrieval, vol. 4, no. 2, pp. 133–151, 2001.

[12] S. Russell and V. Yoon, “Applications of wavelet data reduction in a recommender system,” Expert Systems with Applications, vol. 34, no. 4, pp. 2316–2325, 2008.

[13] A. Bilge and H. Polat, “A comparison of clustering-based privacy-preserving collaborative filtering schemes,” Applied Soft Computing, vol. 13, no. 5, pp. 2478–2489, 2013.

[14] O. Georgiou and N. Tsapatsoulis, “Improving the scalability of recommender systems by clustering using genetic algorithms,” Lecture Notes in Computer Science, vol. 6352, pp. 442–449, 2010.

[15] B. Mobasher, R. D. Burke, R. Bhaumik, and C. A. Williams, “Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness,” ACM Transactions on Internet Technology, vol. 7, no. 4, pp. 23–60, 2007.

[16] M. G. Vozalis, A. Markos, and K. G. Margaritis, “On the performance of SVD-based algorithms for collaborative filtering,” Proc. 4th Balkan Conference in Informatics, Thessaloniki, Greece, pp. 245–250, 2009.

[17] A. Bilge and H. Polat, “A scalable privacy-preserving recommendation scheme via bisecting *k*-means clustering,” Information Processing & Management, vol. 49, no. 4, pp. 912–927, 2013.

[18] A. Bilge, I. Gunes, and H. Polat, “A robust privacy-preserving recommendation algorithm,” Proc. 2nd Asian Conference on Information Systems, Phuket, Thailand, 2013.

[19] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, “Using collaborative filtering to weave an information Tapestry,” Communications of the ACM, vol. 35, no. 12, pp. 61–70, 1992.

[20] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, “Evaluating collaborative filtering recommender systems,” ACM Transactions on Information Systems, vol. 22, no. 1, pp. 5–53, 2004.

[21] J. Cao, Z. Wu, Y. Wang, and Y. Zhuang, “Hybrid collaborative filtering algorithm for bidirectional Web service recommendation,” Knowledge and Information Systems, vol. 36, no. 3, pp. 607–627, 2013.

[22] H. Movahedian and M. R. Khayyambashi, “A tag-based recommender system using rule-based collaborative profile enrichment,” Intelligent Data Analysis, vol. 18, no. 5, pp. 953–972, 2014.

[23] C. Dellarocas, “Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior,” Proc. 2nd ACM Conference on Electronic Commerce, Minneapolis, MN, USA, pp. 150–157, 2000.

[24] M. P. O’Mahony, N. J. Hurley, and G. C. M. Silvestre, “Collaborative filtering - safe and sound?” Lecture Notes in Computer Science, vol. 2871, pp. 506–510, 2003.

[25] S. K. Lam and J. T. Riedl, “Privacy, shilling, and the value of information in recommender systems,” Proc. User Modeling Workshop on Privacy-Enhanced Personalization, Edinburgh, UK, pp. 85–92, 2005.

[26] B. Mobasher, R. D. Burke, C. A. Williams, and R. Bhaumik, “Analysis and detection of segment-focused attacks against collaborative recommendation,” Lecture Notes in Computer Science, vol. 4198, pp. 96–118, 2006.

[27] I. Gunes, C. Kaleli, A. Bilge, and H. Polat, “Shilling attacks against recommender systems: A comprehensive survey,” Artificial Intelligence Review, vol. 42, no. 4, pp. 767–799, 2014.

[28] B. Mobasher, R. Burke, and J. J. Sandvig, “Model-based collaborative filtering as a defense against profile injection attacks,” Proc. 21st National Conference on Artificial Intelligence - Volume 2, Boston, MA, USA, pp. 1388–1393, 2006.

[29] S. Zhang, Y. Ouyang, J. Ford, and F. Makedon, “Analysis of a low-dimensional linear model under recommendation attacks,” Proc. 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, WA, USA, pp. 517–524, 2006.

[30] “Non-commercial, personalized movie recommendations” MovieLens. Web. 11 Apr. 2015.

 ISBN: 978-1-61208-415-2