Robustness Analysis of Privacy-Preserving Hybrid Recommendation Algorithm

İhsan Güneş*, Hüseyin Polat*ǂ

* Anadolu University, Computer Engineering Department, 26470 Eskisehir, Turkey

ǂ Corresponding Author; Address: Anadolu University, Computer Engineering Department, 26470, Eskişehir, Turkey Tel: +90 222 321 3550, Fax: +90 222 323 9501,

e-mail: polath@anadolu.edu.tr

Abstract- In addition to memory- and model-based prediction methods, hybrid schemes are widely used due to their advantages like higher accuracy and improved online performance. Such methods should provide accurate predictions efficiently with privacy. Also, they need to be robust against profile injection or shilling attacks. These attacks insert fake profiles into user-item matrices in prediction systems. Although some privacy-preserving memory- and model-based collaborative filtering algorithms have been investigated with respect to robustness, privacy-preserving hybrid recommendation schemes have not been analyzed in terms of robustness.

In this paper, we analyze a privacy-preserving hybrid prediction scheme with respect to robustness. Four push and two nuke shilling attacks are applied to the algorithm to show how robust it is against them. Different sets of experiments are conducted using real data to show how varying control parameters affect the robustness. The hybrid scheme is compared with memory- and model-based schemes in terms of robustness. Our analysis shows that although the scheme can be marginally considered as a robust algorithm, it is less robust than memory- or model-based prediction algorithms with privacy.

Keywords- Robustness; hybrid algorithm; privacy; shilling attacks; recommendation.

1. Introduction

Recommender systems have been improved for aiding customers with selecting a product or service from the large number of product vendors or service providers. One of the frequently employed recommender systems is called collaborative filtering (CF) [1-3]. Users prefer privacy in purchasing any item or service from the Internet and may not want their product preferences and the products they rate to be known by public. Thus, for keeping the personal data and preferences secure enough, privacy-preserving collaborative filtering (PPCF) methods have been developed [4, 5].

Randomized perturbation is widely employed for protecting private data in PPCF schemes, where data are masked by adding noise data. In this way, the data collector cannot learn the actual ratings but continues to generate correct predictions. Gaussian or uniform distribution is used to produce random numbers with zero mean and a standard deviation. For disguising which products are rated, some of the uniformly randomly chosen unrated items cells are filled with random numbers.

PPCF schemes are categorized into three different schemes: memory-based, model-based, and hybrid methods. Memory-based schemes operate over entire user-item matrices for estimating predictions online. Model-based ones create a model off-line and they utilize the model to provide recommendations online. Hybrid methods can be considered as combinations of memory- and model-based methods. Memory-based techniques with privacy are the simplest heuristic methods [5]. Using such methods for producing referrals is straightforward. It is easy to add a new user or product into the collection. The mechanism scales well with commonly rated items by any two users. However, the size of data can be a disadvantage for scaling those systems. Privacy-preserving model-based prediction algorithms generate a model relying on user ratings as well as providing predictions [5-7]. They scale better in a sparse environment. They find item or user similarities off-line via the model. When a new item or user is added, a new and a fresh model should be established. However, this process is computationally expensive. Also, as useful data can be lost during a specific model production, accuracy may be reduced. Hybrid prediction scheme with privacy features a more effectively performance by utilizing advantages of memory- and model-based models [8].

Recommendation schemes without confidentiality concerns are explored in terms of shilling attacks and different approaches are proposed for handling the shilling problem [9-11]. Shilling attacks aim to manipulate estimated recommendations. Push attacks like random, average, bandwagon, and segment are used to increase the popularity of a target item. Nuke attacks such as reverse bandwagon and love/hate are designed to decrease the popularity of a target item [9-11]. Numerous PPCF schemes are recommended for solving the privacy problem [4, 5, 12]. Privacy or confidentiality in this context can be defined as hiding original rating values and which products are rated. Besides protecting confidentiality, preventive methods for PPCF schemes against shilling attacks are also claimed. However, there are few studies for scrutinizing PPCF schemes with respect to shilling attacks [13-15]. In these studies, two memory- and four model-based algorithms are studied for depicting how robust they are against profile injection attacks. The investigated memory-based schemes are *k-nn* and correlation threshold-based methods while the model-based schemes are *k*-means-, singular value decomposition (SVD)-, item-, and discrete wavelet transform (DWT)-based PPCF schemes. In the current study, a hybrid PPCF scheme is tested against six shilling attacks. These attack models are designed to manipulate private preference collections. Modified versions of random, average, bandwagon, and segment push attacks along with reverse bandwagon and love/hate nuke attack models are applied to the hybrid PPCF scheme. Its robustness against these attacks is discussed based on real data-based empirical outcomes.

The contributions of this paper, in general, are listed below:

1. Six shilling attack models are applied to the hybrid PPCF scheme for the first time.

2. Comprehensive real data-based experiments are conducted for evaluating the robustness of the hybrid PPCF algorithm against the six attack models.

3. The hybrid recommendation scheme with privacy is compared with well-known memory- and model-based PPCF schemes in terms of robustness.

The rest of the paper is structured as follows. In Section 2, related studies are reviewed and the differences between this work and the existing ones are briefly presented. Preliminary works are described in Section 3 while Section 4 presents our motivation in detail. In Section 5, real data-based experiments and their results are given. Finally, conclusions and possible future works are described in Section 6.

2. Related Work

To manipulate the output of recommender systems, malicious users might add fake profiles, referred to as shilling or profile injection attacks. These attacks usually intend to increase the popularity of a target product (referred to as the push attacks) or reduce it (called the nuke attacks). The concept of such attacks was first defined by O'Mahony et al. [16, 17]. The authors show that efficient attacks can be designed. Mobasher et al. [11] design several new push and nuke attack models. They perform several experimental evaluations to show which attacks are more successful against CF algorithms. Their experimental results show that both user- and item-based algorithms are vulnerable to these attack models, but the hybrid algorithms can be more resistant against them. Burke et al. [9] outline some of the major issues for building recommendation system. They define components of secure recommendation like attack models, algorithms, data source detection, response, and evaluation.

Privacy-preserving memory-based prediction algorithms use entire database to generate recommendations [4, 5]. Privacy-preserving model-based prediction schemes generate a model and provide predictions via this model [4, 5]. Polat and Du [4] perform SVD-based PPCF schemes to improve scalability. SVD-based method reduces the size of the user-item matrix off-line. The reduced matrix is then utilized to estimate predictions online. They use randomization in order to perturb private data. Individual user privacy is protected by masking ratings and unrated items with noise data. Bilge and Polat [6] study how to provide DWT-based recommendations without violating privacy. DWT iteratively combine adjacent items so that the result user-item matrix becomes smaller. When a new user enters the system, her data is similarly reduced using DWT. Thus, the DWT-based scheme is able to overcome scalability problem. In addition to SVD- and DWT-based methods, to overcome scalability and accuracy problems of PPCF schemes, clustering-based CF schemes are proposed by Bilge and Polat [7]. The authors apply k-means, fuzzy c-means, and self-organizing map clustering to PPCF schemes. They basically study how to offer clustering-based predictions while preserving individual user's privacy. Also, they compare such schemes in terms of efficiency, accuracy, and privacy. Renckes et al. [8] propose a hybrid PPCF algorithm to improve performance. The authors show that some works like similarity weights between any two users and creating trees for each user can be done off-line. Since off-line works are not critical to overall performance, online performance significantly improves.

Gunes et al. [14, 15] design shilling attack models against PPCF schemes and they apply them against memory-based PPCF schemes. The authors experimentally show that the schemes are vulnerable against shilling attacks. They discuss how to generate attack models from disguised data so that they are still effective for manipulating PPCF systems' outcomes. Bilge et al. [13] apply these attacks against four model-based PPCF schemes. They perform some experiments on real data and evaluate how robust these PPCF schemes are against shilling attacks. According to their results, some model-based PPCF schemes are more robust than memory-based ones. Although there are some studies focusing on the robustness analysis of memory- and model-based PPCF algorithms, there is no research about the robustness of the hybrid PPCF scheme. Hence, we scrutinize a hybrid PPCF scheme against six well-known shilling attack models and show how robust the algorithm is. We also compare it with other PPCF schemes with respect to robustness.

3. Preliminaries

3.1. Privacy Protection by Randomization

Polat and Du [5] employ randomization for accomplishing privacy in CF systems. In their proposed method, users disturb their data by adding random numbers to real ratings. These random numbers are achieved from predefined distributions like Gaussian or uniform with zero mean and a standard deviation (σ). In PPCF schemes, privacy protection process prevents the server from learning true ratings and rated/unrated items. The server defines σ_{max} (maximum permissible standard deviation for producing random numbers) and β_{max} (maximum percentage of filling unrated items to be filled with noise).

Data hiding might be shortly explained as follows:

1. Each user u calculates z-score values of their ratings. Note that z_{uj}, z-score of the user u on item j, can be estimated as follows:

$$z_{uj} = \left(v_{uj} - \overline{v_u}\right)/\sigma_u \qquad (1)$$

in which v_{uj} is the user u's rating on the item j, $\overline{v_u}$ is the average rating of the user u's ratings, and σ_u is the standard deviation of the user u's ratings.

2. Server decides σ_{max} and β_{max} values and allows each user to recognize them.

3. Each user u randomly selects β_u from the range (0, β_{max}], and β_u percent of their unrated items to be filled with random numbers.

4. Then, each user u selects σ_u of random numbers prior to performing random number distribution from the range (0, σ_{max}]. They then determine the distribution of random numbers (either uniform or Gaussian) by coin tosses.

5. In the post distribution selection phase, users form random numbers (r_{uj} values) for real ratings and unrated items. After that each user masks their z-score values through random value addition ($z'_{uj} = z_{uj} + r_{uj}$). Each user ultimately fills the selected unrated items by corresponding random numbers.

6. In the final phase, users send their hidden vectors to the server.

3.2. Hybrid Collaborative Filtering with Privacy

Renckes et al. [8] propose a novel hybrid PPCF scheme. The hybrid scheme's structure is similar to that of a tree, where each node represents a user and each link depicts similarity between two corresponding users. The root of the tree indicates the initial neighbor of a target user. The authors form trees for representing the users and the similarities between them. A tree is constructed off-line after collecting users' preferences about various items, for each user u. The root node represents the user u.

The server first constructs trees for each user u as follows:

1. Similarity weights between user u and each other user are computed. The user u is inserted into the root node. The ratings are already known and no effort is spent for finding them.

2. The most similar s users to user u are discovered and removed from the database. These s users represent the children (adjacent) of the user u and they are housed at the first level.

3. For each of the s users, the best similar s users to them among the remaining ones are found. Such users are then placed into the second level. Correspondingly, these users are the neighbors of each of the s users remaining at the first level.

4. The most related s users to each user among the remaining ones are determined until there is no one left in the records. The structure constructed for each user u is similar to a tree, where each node's children represent the most similar users to that user. Note that $n = 1 + s + s^2 + s^3 + ... + s^y$, where y is the number of levels and n is the number of users. The value of y is subjected to n and s.

For each tree, the following storage is done: initial user, her neighbors, her neighbors' neighbors, and so on. Further, similarity weights between neighbors and their preferences about a variety of items are stored. Similarities are saved for each link between users. Each user is linked to the best similar users to her. They represent her neighbors.

When an active user a asks a prediction, the first step is to decide an initial user. There are two possible ways for selecting the initial user. In the first way, the similarities between a and each user in the database are estimated online. The best similar user to a is determined as initial user. In the second way, after collecting n users' data, they can be gathered in several clusters by utilizing different clustering algorithms. Since k-means algorithm is widely employed for clustering users for CF purposes [18, 19], it is used for clustering n users into k clusters off-line. When a asks referrals, distances between a and each cluster center is computed to determine her cluster online. Then, she is inserted into the closest cluster. Similarities between a and each user in that cluster are found and the best similar user to a is selected as initial user.

The procedure for generating referrals online for a can be explained as follows.

1. a sends her ratings and a query to the server. The query consists of the target item q or items for which referrals are sought. The system first places a into a cluster. The initial user is chosen for a among the users in that cluster. The data in the tree generated for the initial user are used for finding appointments.

2. Since the tree contains n users' data, the optimum value of the number of users whose data to be used for PPCF should be decided. For improving the overall performance, the best-N neighbors can be chosen for providing recommendations and the optimum value of N can be calculated experimentally.

3. Finally, the system considers those N users' data to find referrals. The system can calculate guessing for a on item q (p_{aq}) as follows [20]. This is one of the best memory-based CF algorithms, where z_{uq} is the z-score of user u for item q and N is the number of users involved in recommendation computation:

$$p_{aq} = \overline{v_a} + \sigma_a \times \frac{\sum_{u=1}^{N} z_{uq} w_{au}}{\sum_{u=1}^{N} w_{au}} \tag{2}$$

in which $\overline{v_a}$ and σ_u represent a's mean rating and standard deviation of her ratings, respectively and w_{au} is the similarity between a and her neighbor u. Similarity weights (w_{au} values) based on z-scores can be computed as follow

$$w_{au} = \sum_{j=1}^{N} z_{aj} z_{uj} \tag{3}$$

in which j shows commonly rated items by both users a and u.

3.3. Shilling Attack Models on Disguised Data

Due to data masking, applying traditional shilling attack models against PPCF systems becomes difficult to handle. Thus, attackers need to modify conventional attack models. The attackers decide on random number distribution to produce noise data. Also, σ values are selected randomly in a uniform manner from the range (0, σ_{max}] for each attack profile prior to generating fake profiles, where σ_{max} is the privacy parameter. Gunes et al. [15] redesign well-known attack models against disguised databases as follows.

In the random attack model, the set of selected items is empty. Filler items, selected randomly, are filled with the random values. The target item is assigned with the highest possible random value. In the average attack model, the set of selected items is also empty. Randomly selected filler items are filled with the item's mean and some random value. The target item then takes the maximum random value. In the bandwagon attack model, selected items are determined from popular items, which are densely rated and have high means. The selected and randomly chosen filler items are filled with random values. The selected items are appointed the highest ones while the target item is assigned the possible maximum random value. The segment attack model is similar to the bandwagon attack. The selected items are chosen from high average products in a specific segment is the only difference.

In the reverse bandwagon attack, selected items are chosen from unpopular items, which are densely rated and have low means. The selected and randomly chosen filler items are filled with random values. The selected items get the lowest ones while the target item is assigned the minimum random value to nuke the target item. In the love/hate attack, the set of selected items is empty. Randomly determined filler items are filled with high random values. The target item is appointed the minimum random value.

4. Robustness of Hybrid Collaborative Filtering with Privacy

Recommender systems are classified into three main classes [2]: CF systems, content-based recommendation systems, and hybrid (collaborative and content-based) prediction systems. Furthermore, there are three groups of CF schemes: memory-based methods, model-based schemes, and hybrid (memory- and model-based) algorithms. Each type of algorithm has its own advantages and disadvantages. In order to combine the advantages of memory- and model-based CF algorithms, hybrid approaches are proposed.

There are some problems that should be handled by CF schemes. Examples of such problems include but not limited to accuracy, performance, privacy, robustness, sparsity, cold start, synonymy, and so on. In order to overcome accuracy and efficiency (online performance), different approaches have been proposed [1-3]. In addition, there are different schemes designed to provide accurate predictions efficiently while preserving privacy. PPCF algorithms have been proposed to overcome accuracy, efficiency, and privacy problems [4-8]. Robustness is one of the most important challenges in CF systems. Hence, PPCF schemes should be robust against shilling attacks. Like CF systems, PPCF methods might be subjected to profile injection attacks. It is vital to analyze their robustness against well-known shilling attacks.

Although robustness of memory- and model-based PPCF schemes has been analyzed [13-15], robustness of the hybrid PPCF scheme has not been analyzed. Since the hybrid scheme is popular compared to other methods, its robustness against shilling attacks should be scrutinized. Therefore, our goal is to analyze the hybrid PPCF method with respect to robustness. We consider the most popular and successful four push attack models along with two nuke attack models.

There are couple of control parameters that might affect the overall performance of shilling attacks. These are called filler size and attack size parameters. Therefore, the six attacks models and their effects on the hybrid PPCF scheme can be evaluated with varying values of filler and attack size parameters. We perform real data-based experiments to show how varying values of filler size and attack size affect the robustness of the hybrid method. In addition to these two parameters, there are also other parameters whose values might affect the overall performance of such attacks. Examples of such parameters are σ_{max}, β_{max}, and N. Their values might affect the robustness of the hybrid PPCF scheme. Hence, we also need to conduct experiments using real data while varying the values of such parameters.

Finally, since there are memory-based, model-based, and hybrid PPCF schemes, it is vital to compare them with respect to their robustness against six popular shilling attacks. Thus, we perform a comparative study to relate these three types of schemes in terms of robustness under the same attacks with the same settings.

5. Experimental Evaluation

To show the effects of the six shilling attack models on the hybrid PPCF algorithm, real data-based experiments were performed. Effects of shilling attacks were evaluated as a function of filler size and attack size. Filler size is defined as the percentage of empty cells filled in the attacker profile. Attack size represents the number of attack profiles to inject, which is proportional to the number of users in the system. For instance, five percent attack size corresponds to 50 attack profiles against a system holding initially 1,000 users. Privacy-preserving parameters β_{max} and σ_{max} were set to 25% and 2, respectively. The values are enough for providing a decent level of individual privacy [6].

MovieLens public data set was utilized in the experiments. The data was collected by GroupLens research team (http://www.grouplens.org) with 100K ratings for 943 users on 1,682 movies. Within the set, the ratings are known to be discrete from 1 to 5. At least 20 movies are rated by each user. For assessing the effects of the applied shilling attack models, prediction shift, the most commonly used metric, was measured [10]. Prediction shift is the average change in produced prediction before and after the attack is employed for the attacked item.

We divided the perturbed data into training and testing sets. We selected 150 users for testing and the rest of the users were assigned to the training set. Two distinct target item sets were formed, each consisting of 50 movies for push and nuke attacks. Items were randomly picked using the stratified sampling. Instinctively, trying to push a popular item or nuke an unpopular one is thought to be unreasonable. In this way, push and nuke attack sets consist of items with averages within ranges of 1-3 and 3-5, respectively. The statistics of the chosen target items is shown in Table 1.

During the experiments, all target items were attacked for all test users in the system and predictions were approximated pre- and post-injection of attack profiles. Then, prediction shift values were calculated for depicting relative changes on estimated recommendations for each different attack model. The empirical results for masked push and nuke attack models were given. In this section, empirical results with respect to varying control parameters are shown and significance of them are discussed.

Table 1. Statistics of target items

Ratings count	***Number of pushed items***		***Number of nuked items***	
	1-2	**2-3**	**3-4**	**4-5**
1-50	**30**	**15**	**12**	**18**
51-150	**-**	**3**	**5**	**6**
151-250	**-**	**1**	**2**	**3**
> 250	**-**	**1**	**1**	**3**

5.1. *Effect of Filler Size Parameter*

Experiments were performed for demonstrating the effects of the masked push and nuke attack models with changing filler size values on the hybrid prediction algorithm. Filler size is directly related to the success of a performed attack because filler items comprise the base for leaking into neighborhoods of genuine users in the prediction process. Since β_{max} was set to 25% at first, during the experiments, filler size was varied from 3% to 25%. Note that filler size is usually varied from 3% to 25% in the related literature [9-11]. Further, the attack size was kept constant at 15%, being the maximum value of attack size value tested. Experiments were repeated 100 times due to randomization in the perturbation process and average results were presented. Prediction shift values for push and nuke attacks are shown in Fig. 1 and Fig. 2, respectively.

As seen from Fig. 1 and Fig 2, prediction shift values show that the hybrid PPCF algorithm is not that robust against shilling attacks. In Fig. 1, the most successful attack seems to be bandwagon attack. Along all of the values of filler size parameter, prediction shift value for bandwagon attack does not show much variation and is realized in the vicinity of 1.58. It shows that when the filler size value increases, for the related item there is no change in the nearest neighbors of users. That is, for the values of filler size 25% compared with 5%, the first *n* nearest neighbors that will affect the prediction value were found not to change much. There is no much change depending on the filler size value for attacks other than average attack. Only when the filler size value is 25%, there is a marked decline in the prediction value. The other successful attack is segment attack. The reason for this phenomenon is that the bandwagon and the segment attacks are specifically designed attacks. These push attacks are advanced attacks and they are similar to each other by the way they are created. In Fig. 2, reverse bandwagon attack, which is a nuke attack, is also quite successful. In this attack, according to different filler size values, prediction values were obtained between -1.55 and -2.35. For filler size value between 3% and 15%, there has not been much of a change in the prediction shift values. The prediction shift value slightly decreases for 25% filler size. Love/hate nuke attack is quite successful as reverse bandwagon attack. According to the change of the filler size value, prediction shift values do not significantly change.

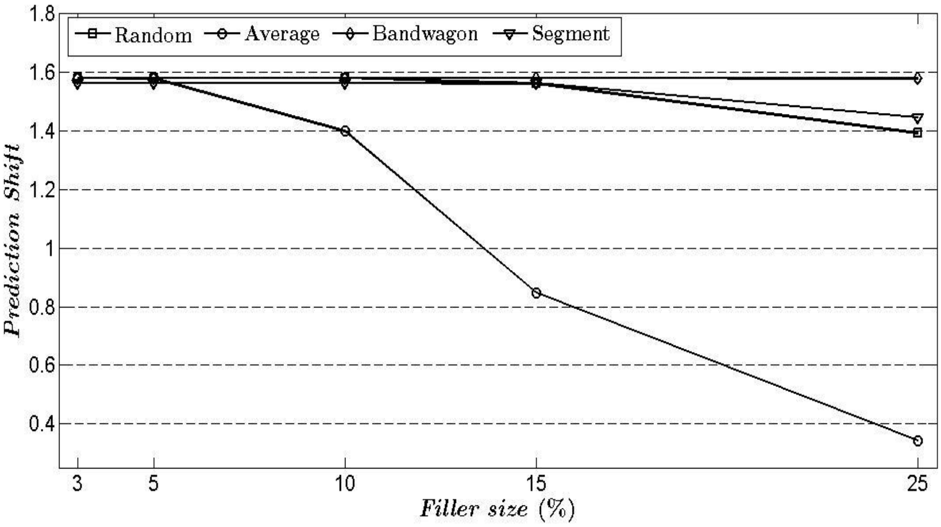

Figure 1. Prediction Shift for Varying Filler Size (Push Attacks).

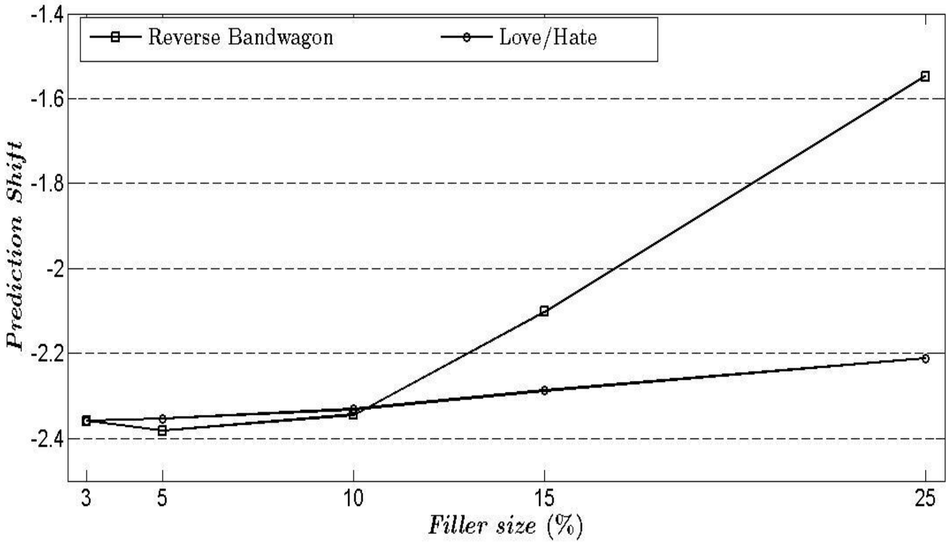

Figure 2. Prediction Shift for Varying Filler Size (Nuke Attacks).

5.2. *Effect of Attack Size Parameter*

Another set of experiments were performed for demonstrating the effects of the attacks with changing attack size values on the hybrid PPCF algorithm. Attack size is the second parameter directly affecting overall success of a shilling attack. While filler size parameter handles utility perspective of an attack, attack size focuses on impact of such utility by determining the number of bogus profiles. Although it is obvious that the more attack profiles inserted into the system, the larger the obtained shifts are; however, it constitutes a trade-off between the detectability and the impact of the applied attack model.

Therefore, in order to define varying effects of the attack size parameter, it was varied from 1% to 15% while filler size was kept constant at 15%. The experiments were repeated 100 times due to randomization. Average prediction shifts values for push and nuke attacks were presented in Fig. 3 and Fig. 4, respectively.

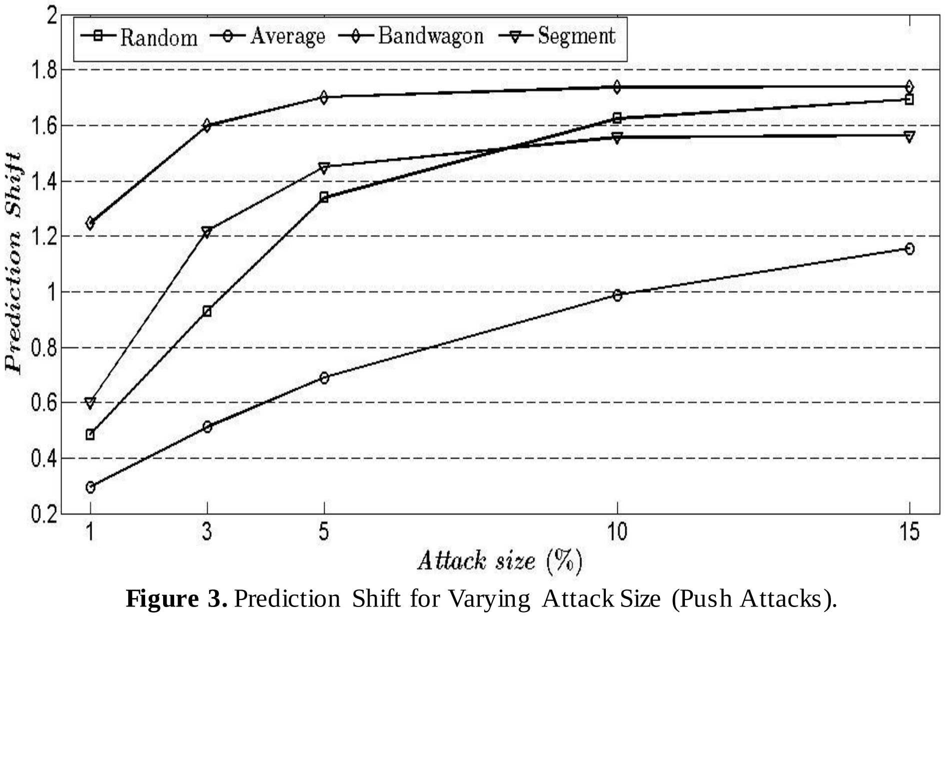

Figure 3. Prediction Shift for Varying Attack Size (Push Attacks).

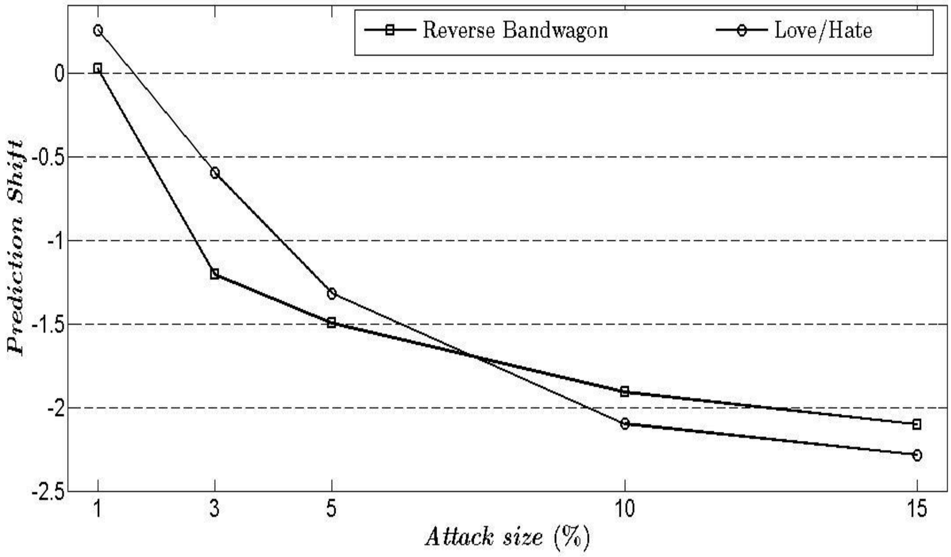

Figure 4. Prediction Shift for Varying Attack Size (Nuke Attacks).

As shown in Fig. 3, the most successful push attack models are bandwagon, segment, and random attacks. With increasing attack size, the success of attacks improves. Depending on the increase in attack size, the number of profiles added to the system also increases. Along with this increase, the probability of the users of attack profiles being nearest neighbors also increase. As a result, rise of attack size is more likely to affect the users' prediction as in the previous experiment. Similarly, reverse bandwagon attack and love/hate attack models are quite successful. As shown in the Fig. 4, for these two nuke attack models, with increased attack size value, prediction shift values also increase.

5.3. *Effect of β_{max} Parameter*

To show how changing β_{max} values affect the overall performance, another set of experiments were performed. As described before, during data disguise β_{max} value determines the rate of unrated item to be filled with random numbers. Each user u randomly selects β_u; and β_u percent of their unrated items to be filled with random numbers. At first, σ_{max} was set to 2 and during the experiments β_{max} parameter was varied from 5% to 25%. Furthermore, attack size and filler size were kept constant at 15%. The most successful attack models, two push (average and bandwagon) and one nuke (reverse bandwagon-RBW) in the previous experiments were used in this and subsequent experiments. The average prediction shift values, obtained by the changing β_{max} value, were shown in Fig. 5.

As seen from Fig.5, the values obtained by average and reverse bandwagon attacks are very close to one another. Average attack is a bit more successful. The most successful result obtained for push attacks based on changing values of β_{max} is 0.99 in the average attack. Prediction shift value has not significantly changed by varying the β_{max}. The reason for this finding is that more unrated cells are filled with increasing β_{max}; and fake profiles become inefficient due to smaller number of fake ratings compared to the filled ones. Prediction shift value increased to a limited extent parallel with the increasing β_{max} value in reverse bandwagon attack. The highest prediction shift value is obtained as -2.1 for this attack.

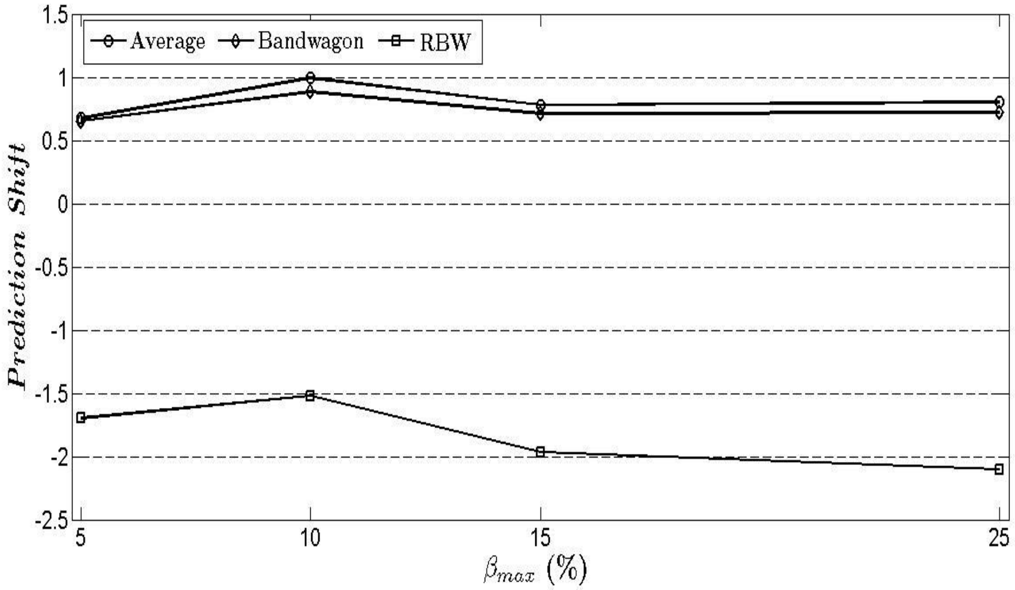

Figure 5. Prediction Shift for Varying β_{max} Parameter.

5.4. *Effect of σ_{max} Parameter*

In PPCF schemes, each user selects a standard deviation value σ_u from the range (0, σ_{max}] during data disguise. To examine the effects of σ_{max} value, its values were assigned from 0.5 to 3. While β_{max} was fixed at 25%, filler size and attack size parameters were fixed at 15%. The outcomes were displayed in Fig. 6.

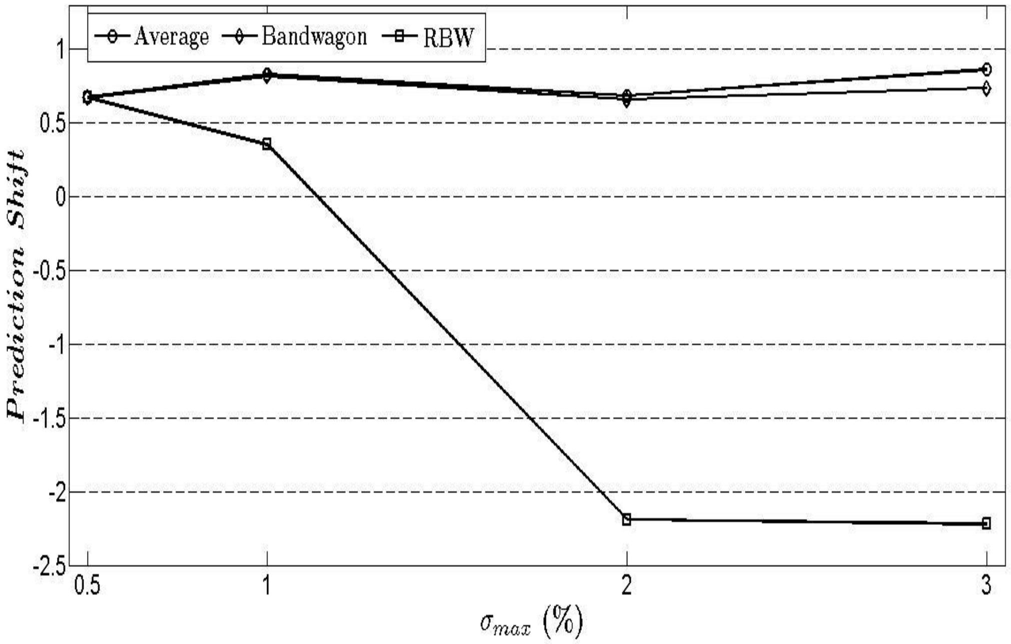

Figure 6. Prediction Shift for Varying σ_{max} Parameter.

As seen from Fig. 6, changing σ_{max} parameter in average and bandwagon attacks does not affect prediction shift values, which are between 0.6 and 0.8 according to the changing values of the parameter σ_{max}. In reverse bandwagon attack, there is a significant increase in the prediction shift value with increasing σ_{max} parameter. The prediction shift value reached -2.25 for reverse bandwagon attack. The reason for this phenomenon is that the target item is assigned to the minimum random number in this attack; and random numbers become smaller with increasing σ_{max} values. Using smaller noise data for nuking predictions causes significant manipulations.

5.5. *Effect of Neighbors Parameter*

Number of neighbors (N) determines how many of the most similar neighbors will be included when calculating prediction in the PPCF algorithm. At first, σ_{max} and β_{max} were set to 2 and 25%, respectively. Furthermore, attack size and filler size were kept constant at 15%. During the experiments, N value was varied from 10 to 100. The most successful three attacks, average, bandwagon, and reverse bandwagon, were applied in this experiment. Fig. 7 shows prediction shift values for these three attacks.

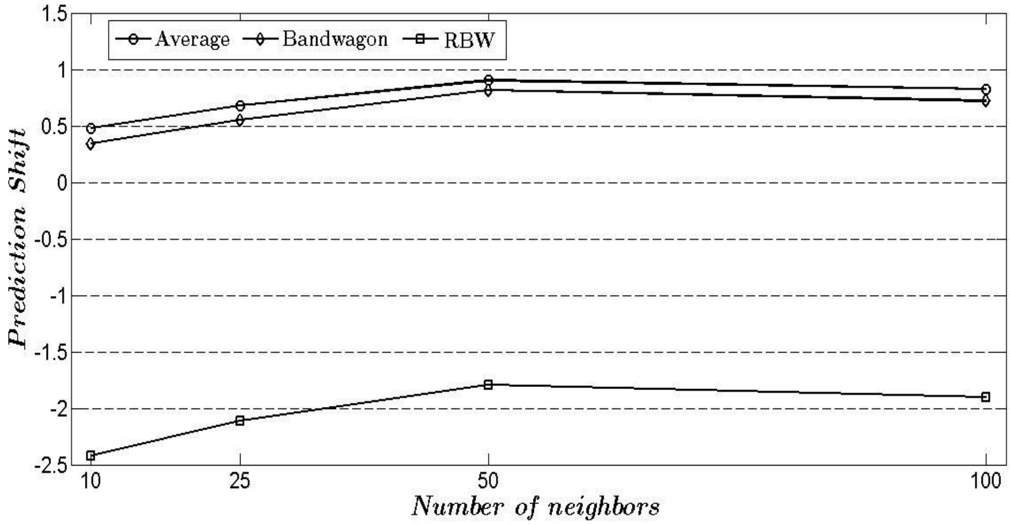

Figure 7. Prediction Shift for Varying Number of Neighbors.

As seen from Fig. 7, prediction shift values obtained via average and bandwagon push attacks increase until the value of N is 50 and later show a little change. Since more attack profiles will be included in the calculation according to the increase in the most similar number of neighbors, prediction shift values will increase. Therefore, as shown in Fig. 7, some increase in the value of N improves the success of average and bandwagon attack models. However, the increase after a certain value of N will reduce the average value because the number of users less similar will also be taken into account. The best value of N is considered as 50 for average and bandwagon attacks. It is considered as 10 for reverse bandwagon attack.

5.6. *On Robustness of Different Prediction Schemes with Privacy*

We finally compared the privacy-preserving hybrid prediction scheme with a well-known privacy-reserving memory-based and model-based recommendation schemes in terms of robustness. Recall that privacy-preserving two memory- and four model-based prediction algorithms are evaluated with respect to robustness, respectively by Gunes et al. [15] and Bilge et al. [13]. Bilge et al. [13] compare four privacy-preserving model-based methods with a well-known privacy-preserving memory-based one in terms of robustness. Now, we added the hybrid PPCF to their comparison table and showed the most significant prediction shift values caused by the six shilling attacks in Table 2.

When we look at the results in Table 2, model-based PPCF algorithms are observed more robust than memory-based and hybrid PPCF algorithms. The most robust algorithms, in general, are model-based ones against the well-known shilling attacks. The memory-based scheme is somewhat robust against such attacks. However, the hybrid method shows the worst performance in terms of robustness. Nuke attacks achieve significant success rates against the hybrid algorithm. All push attacks except average attack are also successful when they are applied to the hybrid scheme. According to the results displayed in Table 2, the most successful algorithm is SVD-based method. Notice that SVD is usually used to remove noise data. Thus, it is able to eliminate the effects of the fake profiles in a user-item matrix. It then makes it as a robust algorithm. As discussed before, recommendation algorithms should provide accurate predictions efficiently with privacy. They also need to be robust against shilling attacks. Therefore, users need to choose the most appropriate prediction schemes. If the only criterion is robustness, then the hybrid scheme is not a good choice.

6. Conclusions and Future Work

Some studies have examined the robustness of memory- and model-based prediction schemes with privacy. However, privacy-preserving hybrid prediction methods have not been evaluated in terms of robustness against shilling attacks. They might be vulnerable against such attacks. In this study, we examined a hybrid scheme with privacy exposed to shilling attacks. We applied four push (random, average, bandwagon, and segment) and two nuke (reverse bandwagon and love/hate) attacks. Empirical results show that the hybrid scheme is vulnerable to shilling attacks. Especially bandwagon and reverse bandwagon attacks are efficient attacks for manipulating referrals. Also, we conducted some experiments to show the effects of control parameters. The outcomes show that varying values of control parameters affect prediction shift values.

There are other hybrid recommendation algorithms. Such algorithms should be investigated with respect to privacy and robustness. Extensive analysis should be performed to compare different types of collaborative filtering algorithms in terms of accuracy, efficiency, privacy, and robustness.

Acknowledgements

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

Table 2. Prediction shifts for memory-based, model-based, and hybrid PPCF methods

Algorithm type	*Shilling attacks*					
	Random	**Average**	**Bandwagon**	**Segment**	**RBW**	**Love/Hate**
Memory-based PPCF						
k-nn	**1.343**	**0.545**	**1.377**	**1.523**	**-1.753**	**-0.168**
Model-based PPCF						
DWT	**0.600**	**1.032**	**0.877**	**0.601**	**-0.562**	**-0.021**
***k*-means**	**1.230**	**0.572**	**1.093**	**1.467**	**-0.298**	**-2.083**
SVD	**0.000**	**0.000**	**0.000**	**0.000**	**-0.001**	**-0.000**
Item-based	**0.018**	**0.021**	**0.018**	**0.080**	**-0.017**	**-0.018**
Hybrid PPCF						
Hybrid	**1.592**	**0.848**	**1.582**	**1.563**	**-2.102**	**-2.287**

References

[1]. C. Birtolo and D. Ronca, “Advances in Clustering Collaborative Filtering by Means of Fuzzy *C*-means and Trust,” *Expert Systems with Applications*, vol. 40, no. 17, pp. 6997-7009, 2013.

[2]. J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérez, “Recommender Systems Survey,” *Knowledge-Based Systems*, vol. 46, pp. 109-132, 2013.

[3]. J. L. Herlocker, J. A. Konstan, L. G. Terveen, J. T. Riedl, “Evaluating Collaborative Filtering Recommender Systems,” *ACM Transactions on Internet Technology*, vol. 22, no. 1, pp. 5-53, 2004.

[4]. H. Polat and W. Du, “SVD-based collaborative filtering with privacy,” in *Proceedings of the 2005 ACM Symposium on Applied Computing*, Santa Fe, NM, USA, pp. 791-795, 2005.

[5]. H. Polat and W. Du, "Privacy-Preserving Collaborative Filtering," *International Journal of Electronic Commerce*, vol. 9, no. 4, pp. 9-35, 2005.

[6]. A. Bilge and H. Polat, "An Improved Privacy-preserving DWT-based Collaborative Filtering Scheme," *Expert Systems with Applications*, vol. 39, no. 3, pp. 3841-3854, 2012.

[7]. 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.

[8]. S. Renckes, H. Polat, Y. Oysal, "A New Hybrid Recommendation Algorithm with Privacy," *Expert Systems*, vol. 29, no. 1, pp. 39-55, 2012.

[9]. R. Burke, B. Mobasher, R. Zabicki, R. Bhaumik, "Identifying attack models for secure recommendation," in *Proceedings of the Beyond Personalization: The Next Stage of Recommender Systems Research Workshop in conjuction with the International Conference on Intelligent User Interfaces*, San Diego, CA, USA, 2005.

[10]. R. Burke, M. P. O'Mahony, N. J. Hurley, "Robust Collaborative Recommendation," in *Recommender Systems Handbook*, Springer US, pp. 805-835, 2011.

[11]. B. Mobasher, R. Burke, R. Bhaumik, C. Williams, "Toward Trustworthy Recommender Systems: An Analysis of Attack Models and Algorithm Robustness," *ACM Transactions on Internet Technology*, vol. 7, no. 4, Article No 23, 38 pages, 2007.

[12]. J. Canny, "Collaborative filtering with privacy," in *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, Berkeley, CA, USA, 45-57, 2002.

[13]. A. Bilge, I. Gunes, H. Polat, "Robustness Analysis of Privacy-preserving Model-based Recommendation Schemes," *Expert Systems with Applications*, vol. 41, no. 8, pp. 3671-3681, 2014.

[14]. I. Gunes, A. Bilge, C. Kaleli, H. Polat, "Shilling Attacks against Privacy-preserving Collaborative Filtering," *Journal of Advanced Management Science*, vol. 1, no. 1, pp. 54-60, 2013.

[15]. I. Gunes, A. Bilge, H. Polat, "Shilling Attacks against Memory-based Privacy-preserving Recommendation Algorithms," *KSII Transactions on Internet & Information Systems*, vol. 7, no. 5, pp. 1272-1290, 2013.

[16]. M. P. O'Mahony, N. J. Hurley, G. C. Silvestre, "Towards Robust Collaborative Filtering," *Lecture Notes in Computer Science*, vol. 2464, pp. 87-94, 2002.

[17]. M. P. O'Mahony, N. J. Hurley, G. C. Silvestre, "Promoting recommendations: An attack on collaborative filtering," in *Proceedings of the 13th International Conference on Database and Expert Systems Applications*, Aix-en-Provence, France, pp. 494-503, 2002.

[18]. B. Marlin, "Collaborative Filtering: A Machine Learning Perspective," *MSc. Thesis*, University of Toronto, 2004.

[19]. G.-R. Xue, C. Lin, Q. Yang, W. Xi, H.-J. Zeng, Y. Yu, Z. Chen, "Scalable collaborative filtering using cluster-based smoothing," in *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Salvador, Brazil, pp. 114-121, 2005.

[20]. J. L. Herlocker, J. A. Konstan, A. Borchers, J. Riedl, "An algorithmic framework for performing collaborative filtering," in *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Berkeley, CA, USA, pp. 230-237, 1999.