SHILLING ATTACK DESIGN AND DETECTION ON MASKED BINARY DATA

Zeynep Batmaz

Master of Science Thesis

Computer Engineering Program

June, 2015

This thesis is partially supported by the Grant 111E218 from TUBITAK.

JÜRİ VE ENSTİTÜ ONAYI

Zeynep Batmaz'ın **"Shilling Attack Design and Detection on Masked Binary Data"** başlıklı **Bilgisayar Mühendisliği** Anabilim Dalındaki, Yüksek Lisans Tezi 25.06.2015 tarihinde, aşağıdaki jüri tarafından Anadolu Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

		Adı Soyadı	**İmza**
Üye (Tez Danışmanı) :		**Doç. Dr. Hüseyin POLAT**	**……..**
Üye	**:**	**Doç. Dr. Serkan GÜNAL**	**……..**
Üye	**:**	**Yrd. Doç. Dr. Mehmet KOÇ**	**……..**

Anadolu Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun …….. tarih ve …….. sayılı kararıyla onaylanmıştır.

Enstitü Müdürü

ABSTRACT

Master of Science Thesis

SHILLING ATTACK DESIGN AND DETECTION ON MASKED BINARY DATA

Zeynep BATMAZ

Anadolu University

Graduate School of Sciences

Computer Engineering Program

Supervisor: Assoc. Prof. Dr. Huseyin POLAT

2015, 83 pages

Privacy-preserving collaborative filtering methods are effectual ways of coping with information overload problem while protecting confidential data. Their success depends mainly on the quality of the collected data for filtering purposes. Malicious entities might create fake profiles (noise data) and insert user-item matrices of such filtering schemes. Hence, shilling attacks play an important role on the quality of data. Designing effective shilling attacks, developing methods to detect them, and performing robustness analysis of privacy-preserving collaborative filtering methods are receiving increasing attention.

In this thesis, six well-known shilling attack models are modified in order to attack binary masked databases in privacy-preserving collaborative filtering methods. Three attack design approaches are proposed. The attack profiles, generated by such schemes, are applied to naïve Bayesian classifier-based collaborating filtering scheme with privacy. A novel shilling attack detection scheme based on classification is proposed to detect fake profiles. Attributes derived from user profiles are utilized for detecting shill profiles. Empirical results show that designing effective shilling attacks is still possible on binary masked data. The proposed detection method is able to successfully detect fake profiles.

Keywords: Shilling Attack, Collaborative Filtering, Privacy, Binary Data, Detection, Robustness

ÖZET

Yüksek Lisans Tezi

GİZLENMİŞ İKİLİ VERİLER ÜZERİNDE ŞİLİN ATAK TASARIMI VE TESPİTİ

Zeynep BATMAZ

Anadolu Üniversitesi

Fen Bilimleri Enstitüsü

Bilgisayar Yazılımı Anabilim Dalı

Danışman: Doç. Dr. Hüseyin POLAT

2015, 83 sayfa

Gizliliği koruyan işbirlikçi filtreleme yöntemleri gizli verileri koruyarak aşırı enformasyon problemi ile başa çıkmanın etkili yoludur. Bu yöntemlerin başarısı filtreleme amacıyla toplanan verilerin kalitesine bağlıdır. Kötü amaçlı kullanıcılar bu filtreleme sistemlerinin veri tabanına sahte profil (gürültülü veri) ekleyebilir. Bu nedenle şilin ataklar veri kalitesi için önemli rol oynarlar. Etkili şilin atak tasarımı, bunların tespiti için yöntemler geliştirilmesi ve gizliliği koruyan işbirlikçi filtreleme algoritmalarının gürbüzlüğünün analizleri artan ilgi görmektedir.

Bu tezde en çok bilinen altı şilin atak modeli gizlenmiş ikili veri tabanlarına saldırmak amacıyla değiştirilerek geliştirilmiştir. Bu amaçla üç şilin atak oluşturma tasarısı önerilmiştir. Bu ataklar basit Bayes sınıflandırıcı tabanlı gizliliği koruyan işbirlikçi filtreleme algoritmasına uygulanmıştır. Sahte profillerin tespit edilmesi için sınıflandırma tabanlı yeni bir atak tespit yöntemi geliştirilmiştir. Atak profillerini tespit etmek amacıyla, sınıflandırma için kullanıcı profillerinden türetilen özniteliklerden yararlanılmıştır. Deneysel sonuçlar ikili saklanmış veri üzerinde etkili şilin atak profilleri oluşturmanın mümkün olduğunu göstermiştir. Önerilen tespit algoritmasının başarılı bir şekilde sahte profilleri tespit ettiği görülmüştür.

Anahtar Kelimeler: Şilin Atak, İşbirlikçi Filtreleme, Gizlilik, İkili Veri, Tespit, Gürbüzlük

ACKNOWLEDGEMENTS

I would like to thank my advisor Assoc. Prof. Dr. Huseyin POLAT for his guidance and support during my study. It was my pleasure to work with him during this study.

I also would like to thank my committee members Assoc. Prof. Dr. Serkan GUNAL and Assist. Prof. Dr. Mehmet KOC for their guidance. I also would like to thank Assoc. Prof. Dr. Cihan KALELI and Assist. Prof. Dr. Alper BILGE for their support during my study.

I also would like to thank to Scientific and Technical Research Council of Turkey (TUBITAK) for giving me financial support through its scolarship program during this study.

Finally, I would like to express my eternal gratitude to my husband and my parents for their everlasting love and support.

Zeynep Batmaz

June, 2015

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

AA	:	Average Attack
aou	:	Agreement with other Users
avgSim	:	Similarity with top-*N* Neighbors
BA	:	Bandwagon Attack
CF	:	Collaborative Filtering
dti	:	Disagreement with Possible Target Items
dup	:	Dissimilarity in User's Profile
LH	:	Love/Hate Attack
MLP	:	MovieLens Public
NBC	:	Naïve Bayesian Classifier
PCA	:	Principal Component Analysis
PLSA	:	Probabilistic Latent Semantic Analysis
PPCF	:	Privacy-Preserving Collaborative Filtering
RA	:	Random Attack
RB	:	Reverse Bandwagon Attack
RRT	:	Randomized Response Technique
SA	:	Segment Attack

1. INTRODUCTION

With increasing amount of data used in everyday life, importance of recommender systems swells by the day. Such systems make users to reach items they are interested in without losing time. Thanks to produced predictions by recommender systems, right products are matched up with the right users. Recommender systems produce recommendations using three approaches as collaborative filtering (CF), content-based filtering, and hybrid systems. CF techniques fabricate predictions utilizing users' past habits. Content-based filtering schemes produce predictions by utilizing contents of the items with user profiles. Hybrid systems are developed to utilize positive aspects of both recommendation systems.

CF is one of the widely used recommendation methods, which provides highly accurate predictions. Those users who have vicinal experiences on past items are tend to agree on new items. CF schemes can be grouped as memory-based, model-based, and hybrid systems. Memory-based methods generally work on entire data in order to produce predictions. Due to the highly increasing number of users and items especially in online systems, memory-based schemes have some challenges like scalability (Sarwar et al., 2001). Model-based methods generate a model from original user-item matrix by utilizing some data mining approaches like Bayesian classifier (Miyahara and Pazzani, 2000) and dimension reduction (Vozalis and Margaritis, 2007). Hybrid CF schemes utilize benefits of memory and model-based CF algorithms (Rashid et al., 2006).

Even though many e-commerce sites use numeric data, some of them may work with binary data. Some e-commerce sites that use recommender systems for any purpose may prefer to know who likes or dislikes products rather than how much users or customers like. Thus, binary data oriented recommender systems are developed to produce referrals from binary data. Binary attribute represents an attribute whose value either only 1 or 0. Providing recommendations from binary data is also extremely important for market basket data analysis. Binary data analyzing schemes can be grouped as similarity, classification, and clustering algorithms (Han et al., 2011). Similarity metrics such as Anderberg, Jaccard,

Yule, Pearson's Correlation, etc. can be used for calculating similarity even if data is binary (Senyurek and Polat, 2013).

Classification is a form of data analysis that extracts models describing classes. Classification is a two-step process. In the first stage, a classification model is built on previously collected data called as learning phase. In the second stage, data is classified based on the models' accuracy. Since class labels are known, classification is supervised learning. There are several approaches for classification like decision tree induction algorithms and the algorithms based on Bayes' theorem (Anderson et al., 1986). Naïve Bayesian classification is one of the classification methods based on Bayes' theorem. Assuming that there are g classes like C_1, C_2, …, C_g, the classifier predicts that a given tuple X belongs to the class having the highest posterior probability. $P(C_i|X)$ determines the probability of given tuple X belongs to class C_i (Han et al., 2011).

Clustering is the process of partitioning a set of data objects into subsets (Han et al., 2011): Each subset is called as a cluster. Objects in the same cluster have high intra-correlation whereas objects in different clusters have low inter-correlation. Clustering methods are grouped as partitioning methods, hierarchical methods, density-based, and grid-based methods. In partitioning methods, data is partitioned into k groups. Each group contains at least one object and an object can belong to only one group. In most partitioning methods, a partitioning method creates an initial state. Then, it uses an iterative relocation technique for partitioning by moving the objects from one cluster to another one. The most popular partitioning algorithms are k-means and k-medoids algorithms. In hierarchical methods, bottom-up and top-down approaches are used. In bottom-up approaches, each object represents a group. The method merges the objects or groups close to one another until all the groups are merged into one or a termination condition occurs. In top-down approaches, all objects are in the same group. The method divides the group into sub-clusters until each object is in one cluster or a stopping condition occurs.

Effectiveness of recommender systems depends on the quality of data. Protecting privacy of users is extremely vital in terms of scaling up the quality of data. Due to the privacy risks such as government surveillance and unsolicited

marketing, users either give false ratings or refuse to use the recommendation system (Cranor, 2004). Since the given ratings do not represent the real preferences, the accuracy of the produced predictions will decrease. In order to provide highly accurate recommendations while preserving privacy, privacy-preserving collaborative filtering (PPCF) schemes, which work with either numeric or binary data, can be used. Aiming to protecting privacy on numeric data, cryptographic (Canny, 2002), obfuscation-based (Berkovsky et al., 2012), and randomization-based (Polat and Du, 2005a) schemes are proposed. Especially, randomization-based methods are widely used in CF systems in order to preserve privacy.

Randomized response techniques (RRTs) are one of the algorithms that provide privacy by preventing the server from learning true data about users. RRT was first introduced by Warner (1965) to solve the following problem. A surveyor wants to find out the percentage of people who have the confidential data *Q*. Respondents may not give the true information. One of the solutions to solve this problem is asking two related questions to the respondents instead of asking directly whether they have *Q*. The answers to the questions are opposite to each other. Respondents choose one of the questions by a randomizing device and reply it with preventing the server to know which question is answered. The randomizing device makes the probability of choosing the first question θ and the probability of choosing the second question 1-θ. Hereby, the server estimates the percentage of the users who have *Q* without knowing the answered questions. RRT is utilized by many researchers (Mild and Reutterer, 2001; Polat and Du, 2005b; 2006; Kaleli and Polat, 2007).

CF systems provide highly accurate recommendations; however, they are vulnerable to shilling attacks. Aiming to manipulate items' popularities in favor of attackers, they generate bogus profiles and insert them into the system's database. Consequently, the attackers affect the system's reliability and fulfillment of users adversely. Thus, detecting shilling attacks is an effective way of dealing with them. In order to manipulate the produced recommendations in a numeric database, shilling attacks are designed (Burke et al., 2005; Bhaumik et al., 2006). Shilling or profile injection attacks can be designed not only for CF systems with

numeric data but also CF systems with binary data. Also, attack profiles can be redesigned to manipulate items' popularities in disguised databases.

Like in numerical data oriented CF systems, malicious users can attack the binary data oriented CF systems (Long and Hu, 2010; Kaleli and Polat, 2013). Long and Hu (2010) compare binary *k-nn* algorithm with user-based *k-nn* scheme in terms of robustness against shilling attacks. They attack both systems by implementing random, average, and bandwagon attacks. As a result, they show that binary *k-nn* scheme is more robust than user-based *k-nn* algorithm against shilling attacks. Kaleli and Polat (2013) design well-known attacks like random, average, segment, etc. based on binary ratings.

PPCF schemes are also vulnerable to shilling attacks (Gunes et al., 2013a; 2013b; Bilge et al., 2014a). Gunes et al. (2013a) design random and bandwagon attack models with privacy concerns. Bandwagon and random attack profiles are disguised by utilizing generated random values according to predefined distribution and masking parameters. Aiming to disguise a target item for a profile, a value is selected among positive or negative generated random numbers for push and nuke attack strategies, respectively (Gunes et al., 2013a). Remaining random numbers are added to *z*-scores of filler items and selected items in order to mask them. Gunes et al. (2013b) redesign six well-known shilling attack models in order to attack a disguised system. Besides attack profiles' rated items, unrated items of attack profiles are also disguised in the work. The attacker decides on β_{max} and σ_{max} values, which are known as privacy parameters. In order to disguise a random attack profile, the maximum value among generated random values according to chosen distribution is assigned to the target item and the remaining values are assigned to filler items. For masking an average attack profile, utilizing the masking parameter α, l random numbers are generated, where l is the number of filler items. Filler items are filled as item's mean plus random numbers. The target item is filled with the maximum value of the generated numbers. Aiming to disguise a bandwagon profile, the top of the generated random numbers is assigned to the target item, the remaining top m_t values are assigned to selected items, where m_t is the number of selected items and filler items are filled with remaining values. A segment attack profile is disguised as bandwagon attack

model. For masking a love/hate attack profile, filler items are filled with the highest values and the target item is filled with the minimum value of the generated random values (Gunes et al., 2013b). Bilge et al. (2014a) perform robustness analyses of privacy-preserving *k*-means-, discrete wavelet transform-, singular value decomposition-, and item-based prediction algorithms against six well-known shilling attack models. The authors show that model-based PPCF methods are more resistant than memory-based ones against shilling attacks.

Since shilling attacks have an important effect on produced predictions, dealing with attack profiles is extremely imperative. Detecting bogus profiles is one of the ways of overcoming profile injection attacks. To detect fake profiles, many detection schemes based on statistical methods (Zhang et al., 2006; Gao et al., 2014; Xia et al., 2015), clustering (O'Mahony et al., 2003; Mehta and Nejdl, 2009; Bilge et al., 2014b; Zhang and Kulkarni, 2014; Gunes and Polat, 2015), classification (Chirita et al., 2005; Williams et al., 2007; Wu et al., 2012a; Cao et al., 2013), variable selection (Mehta et al., 2007), and other techniques (Su et al., 2005) are proposed. Since shilling attacks are generated according to a certain strategy, classification-based schemes are effective methods for detecting attack profiles by utilizing generic attributes and model specific attributes derived from each separate profile.

In the literature, there is no work on designing shilling attacks against binary ratings oriented PPCF schemes or robustness analysis of binary ratings oriented PPCF schemes. To the best of our knowledge, this dissertation is the first one, which designs shilling attacks against binary ratings oriented PPCF schemes and analyze them with respect to robustness. Also, a novel detection scheme is proposed in order to determine such attack profiles on masked binary ratings.

The organization of the thesis is as follows: Related works are discussed in Section 2. Background information is given in Section 3. Redesigned forms of six well-known shilling attack models are described in Section 4. Section 5 examines the robustness of PPCF scheme based on naïve Bayesian classifier (NBC) under the proposed attack strategies. In Section 6, the proposed classification-based detection scheme is described and empirial results are displayed. Section 7 presents conclusions and explains future works.

2. RELATED WORK

Although CF approach is a successful variation of recommender systems, it is vulnerable to profile injection attacks. Since robustness analysis of CF schemes shows how much a scheme is resistant to shilling attacks, fake profiles are designed to attack the CF system. Many researchers design attack profiles to manipulate systems working with numerical undisguised data (O'Mahony et al., 2005; Mobasher et al., 2007; Yan and Van Roy, 2009; Cheng and Hurley, 2010). Also, shilling attack models are redesigned in order to manipulate predictions with binary ratings (Long and Hu, 2010; Kaleli and Polat, 2013). Long and Hu (2010) compare binary *k-nn* and user-based *k-nn* algorithms with respect to robustness against shilling attacks by attacking both systems. Kaleli and Polat (2013) redesign six well-known attack types in order to attack a binary ratings oriented system. The authors apply their attack strategies to NBC-based CF algorithm and their results show that the algorithm is not robust against profile injection attacks.

Since privacy is a substantial point for recommender systems in order to supply quality data, classical attack strategies are redesigned aiming to play an effective part on produced predictions (Gunes et al., 2013a; 2013b; Bilge et al., 2014a). Gunes et al. (2013a) generate new forms of random and bandwagon attack models for attacking a system with numeric private data by utilizing randomization techniques. In the study proposed by Gunes et al. (2013b), six well-known shilling attack models are redesigned for attacking a disguised numeric ratings oriented system. The authors attack memory-based PPCF schemes with new versions of the attack models in order to determine how much they are robust against shilling attacks. Bilge et al. (2014a) apply redesigned attack profiles to some model-based PPCF algorithms and their experiments show that model-based PPCF schemes are more robust than memory-based ones. Even though lots of study perform attacks to CF algorithms in order to measure the algorithms' robustness, our dissertation redesigns six well-known attack models for attacking a system with binary disguised data and performs robustness analysis of NBC-based PPCF scheme against redesigned attacks.

Robustness term is presented by O'Mahony et al. (2002) as a metric measuring performance of a recommender system. O'Mahony et al. (2004) show the effects of generally used neighborhood formation schemes and similarity measures on recommender system performance in terms of robustness. Robustness analyses of CF method based on probabilistic latent semantic analysis (PLSA) and *k*-means clustering approach are performed for comparing performance of model-based CF schemes under shilling attacks with the memory-based scheme's performance (Mobasher et al., 2006). Long and Hu (2010) compare binary *k-nn* algorithm with user-based *k-nn* scheme in terms of robustness against random, average, and bandwagon attack models. Their empirical results show that binary *k-nn* scheme is more robust than user-based *k-nn* algorithm against shilling attacks. Kaleli and Polat (2013) generate binary forms of previously defined well-known attacks like random, average, segment, etc. and apply them to NBC-based CF scheme in order to measure how robust the scheme is. Besides unmasked data, robustness analysis of PPCF schemes with numeric masked data are performed by Gunes et al. (2013a; 2013b) and Bilge et al. (2014a). In (Gunes et al., 2013a), privacy forms of random and bandwagon attack models are proposed to perform robustness analysis of privacy-preserving *k-nn* memory-based CF algorithm under shilling attacks. In another study, Gunes et al. (2013b) redesign six well-known shilling attack models in order to attack memory-based privacy-preserving recommendation algorithms. In the work proposed by Bilge et al. (2014a), robustness analysis of different model-based PPCF algorithms against six well-known shilling attack models are performed. There is no work measuring performances of PPCF schemes on binary masked data under shilling attacks. In this thesis, redesigned shilling attack profiles are applied to NBC-based PPCF scheme in order to show its performance on binary masked data under profile injection attacks.

Since effectiveness of a recommender system depends mostly on the quality of data, dealing with shilling attacks is extremely significant. Detecting bogus profiles is one of the influential ways of coping with profile injection attacks. In the literature, there are many detection schemes based on statistical methods, classification, clustering, variable selection, and other techniques (Gunes et al.,

2014). In (Zhang et al., 2006), detection of fake profiles is performed by utilizing time series of items' ratings. In order to observe time series, the authors group sequential ratings of items into separate windows and they calculate sample average and entropy for each window. Fake profiles can be detected by revealing anomalies caused by attack profiles by utilizing statistical anomaly detection method (Bhaumik et al., 2006). Hurley et al. (2009) point out that obfuscated forms of attacks are not detected by many detection schemes. Thus, supervised and unsupervised Neyman-Pearson detectors are designed in order to detect attack profiles even if they are obfuscated. A bottom-up discretized scheme based on time intervals and two common features for all attack types are designed to detect abnormal items by comparing ratings distribution in different time intervals (Gao et al., 2014). Xia et al. (2015) propose a technique based on a dynamic time interval segmentation, which is used for detecting shilling attacks.

Shilling attacks resemble to each other. Hence, clustering-based schemes are proposed by many researchers for detecting attacks (O'Mahony et al., 2003; Mehta and Nejdl, 2009; Bilge et al., 2014b; Zhang and Kulkarni, 2014). In (O'Mahony et al., 2003), a neighborhood selection scheme based on clustering in reputation reporting systems is proposed to detect bogus profiles. Two methods, based on PLSA and principal component analysis (PCA), are proposed for detecting shilling attacks (Mehta and Nejdl, 2009). PLSA provides to calculate probabilistic distribution over clusters and PCA, which is a linear dimensionality reduction model, provides to select dimensions. In (Zhang and Kulkarni, 2014), a spectral clustering-based detection scheme is proposed by utilizing high correlation between attack profiles. Spectral clustering algorithm is applied to find the maximum sub-matrix of user-user correlation matrix, where the sub-matrix contains shill profiles. Bilge et al. (2014b) propose a bisecting *k*-means clustering algorithm in order to detect shill profiles. A binary decision tree is derived from the given user-item matrices, and intra-cluster correlation values are calculated for each sub-cluster. Then, utilizing the intra-cluster correlation values, the clusters containing shill profiles is specified depending on the fact that shill clusters have high intra-cluster correlation values and sub-clusters of a parent cluster, which contains fake profiles cannot differ diversely from the parent in terms of intra-

cluster correlation. Also, in order to detect shill profiles on numeric masked data, Gunes and Polat (2015) propose a detection method based on hierarchical clustering. They cluster profiles utilizing their similarity weights on masked data.

Due to the certain generation strategies of shilling attack models, some generic attributes derived from profiles such as rating deviation from mean agreement, standard deviation, degree of similarity, etc. are used to detect bogus profiles. Chirita et al. (2005) utilize generic attributes derived from each individual profile to determine whether a profile is genuine or not. Besides generic attributes, model specific attributes such as filler mean variance for average attack model and filler mean target difference for segment attack model are utilized for detecting fake profiles (Williams et al., 2007). Burke et al. (2006) utilize generic attributes and model specific attributes like filler average correlation for random attack model to classify a profile. Model specific and generic attributes are also utilized in order to detect obfuscated attack profiles (Williams et al., 2006). Wu et al. (2012a) use two classification-based methods, NBC and *k-nn* classifier, to detect profile injection attack profiles. Similarly, Wu et al. (2012b) propose a hybrid shilling attack detector based on NBC for detecting random filler model and average filler model attack profiles. A semi-supervised learning-based detection scheme is proposed by utilizing two schemes like NBC for labeled users and EM-λ for unlabeled users in order to improve NBC (Cao et al., 2013).

All of the detection schemes proposed to date work with numeric undisguised ratings oriented systems. In this dissertation, six well-known shilling attack models are redesigned by utilizing three proposed shilling attack generation schemes in order to attack binary masked data. To measure the successes of the proposed attack models and the robustness of NBC-based PPCF scheme, the regenerated attack profiles are applied to the system. Also, a classification-based detection scheme is proposed to detect shilling attack profiles in databases including binary masked data. To the best of our knowledge, this work is the first one generating successful shilling attack profiles for binary masked data and detecting them.

3. PRELIMINARIES

3.1. Shilling Attacks

CF systems overcome information overload problem by making people to reach information they are looking for. For reliability and continuity of CF systems, accuracy and trusted predictions are highly important. Revealing accurate estimations for users provides that users trust the system. Especially, e-commerce sites utilize CF systems for increasing their sales thanks to accurate outcomes of CF systems. Aiming to increase or decrease popularities of target items, some malicious users or competitive firms can create fake profiles and insert them into the system's database. Thus, the malicious users or competitives manage to manipulate the outcomes on behalf of their advantages.

Profile injection or shilling attacks jeopardize reliability of CF systems. To deal with shilling attacks, probing their structures is extremely considerable. Shilling attack profile is defined by Bhaumik et al. (2006), as shown in Fig. 3.1. In such figure, I_S determines the characteristic of the attack, I_F is chosen randomly and obviates the detectability of the attack, I_Φ is the set of unrated items, i_t is the target item whose popularity is manipulated.

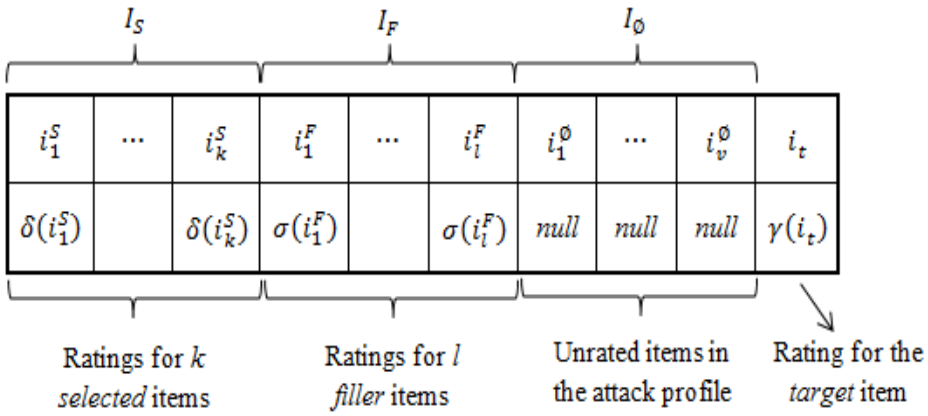

Figure 3.1. General form of an attack profile

Shilling attacks vary according to their intends, required knowledge, targets, cost, algorithm, and detectability (Burke et al., 2005). Shilling attacks are categorized as push and nuke attacks according to their intend. While push attacks aim to increase popularity of a target item, nuke attacks have goal to decrease it. Also, shilling attacks can be grouped as high knowledge required attacks and low knowledge required attacks according to required knowledge. High knowledge required attacks are efficient but need confidential information about a system, such as items' means. However, low knowledge required attacks need public information about a system like system mean. The most well-known six attack types are described as follows (Burke et al., 2005):

Random attack (RA): For random attack model, filler items are chosen randomly and filled around system mean. Rating of the target item is chosen as the maximum or minimum vote in the rating interval depending on the intend of the attacker. Remaining items are unrated. Random attacks are low knowledge required attacks.

Average attack (AA): Filler items are chosen randomly and rated around the corresponding item's mean. The target item is voted with the maximum or minimum rating in the rating interval depending on the intend. Since items' means are not public information, average attack model is high knowledge required attack.

Bandwagon attack (BA): For banwagon attack model, popular items are chosen as selected items and rated with the maximum vote in the rating interval. Filler items are chosen randomly and voted around system mean. Target item is voted as the maximum rating in the rating interval. Bandwagon attacks are low knowledge required push attacks.

Segment attack (SA): Segment attack is designed for a group of users interested with the target item. Utilizing that an item with related users will increase the sales of the item, selected items are chosen as popular items for segmented users and rated with the maximum vote in the rating interval. Filler items are chosen randomly and voted with the minimum rating in the interval. Target item is filled with the maximum rating value in the interval. Segment attacks are low knowledge required push attacks.

Reverse bandwagon attack (RB): Reverse bandwagon attack model is a variation of bandwagon attack model in order to decrease popularity of a target item. Since reverse bandwagon attacks are nuke attacks, unpopular items are chosen as selected items and rated with the minimum vote in the rating interval. Filler items are chosen randomly and filled as bandwagon attack model. Target item is rated with the mininmum vote in the rating interval. Reverse bandwagon attacks are also low knowledge required attacks.

Love/hate attack (L/H): Love/hate attacks are designed for nuking. Filler items are chosen randomly and filled with the maximum rating value in the rating interval. Target item is voted with the minimum rating in the interval. Love/hate attacks are low knowledge required attacks.

3.2. Binary Ratings Oriented Shilling Attacks

Although many e-commerce sites work with numeric data, there are some sites that prefer to know who likes or dislikes products rather than how much a user likes a product. Thus, binary data oriented CF schemes are proposed in order to provide accurate referrals from binary data.

Kaleli and Polat (2013) generate binary forms of the attacks by grouping them into two groups as attack models with or without selected items set. Besides, they propose a new evaluation metric called *ratio shift* in order to measure the success of binary shilling attacks. The proposed metric quantifies the proportion of 1s before and after the attack. Also, they perform robustness analysis of NBC-based CF algorithm under binary profile injection attacks. Their results show that NBC-based CF algorithm is not a robust algorithm against binary shilling attacks. Shilling attack models with selected items set against NBC-based CF algorithm are designed as follows (Kaleli and Polat, 2013):

- For all attack types in the group except segment attack model, each filler item is filled with 1 if generated random number is greater than 0.5, otherwise it is filled with 0. For segment attack model, each filler item is filled with 0.

- All of the selected items are filled with 1, 1, and 0 for bandwagon, segment, and reverse bandwagon attack models, respectively.
- Target item is rated as 1 for bandwagon attack, 1 for segment attack, and 0 for reverse bandwagon attack.

Shilling attack models without selected items set against NBC-based CF algorithm are designed as follows (Kaleli and Polat, 2013):

- For random attack model, each filler item is filled with 1 if generated random number is greater than 0.5, otherwise it is filled with 0. For average attack model, each filler item is filled with its mode value. For love/hate attack model, each filler item is filled with 1.
- Target item is filled with 0 for love/hate attack model. Target item for random and average attack models is filled with 1 for pushing and 0 for nuking.

3.3. Privacy-preserving Collaborative Filtering on Binary Data

Quality of data is one of the serious factors of enhancing or protecting accuracy of recommender systems. Preserving privacy is one of the efficacious ways of increasing quality of data. Due to privacy risks, users either avoid to give true ratings or give up using the recommender system (Cranor, 2004). Accordingly, the quality of data worsens and the recommender system produces low accurate predictions. Thus, users become unsatisfied. In order to produce highly accurate predictions while preserving privacy, PPCF schemes are proposed, which work with either numeric or binary data.

RRT is one of the algorithms that provides privacy on binary data by preventing the server from learning true data about users. PPCF scheme on RRTs can be described as follows (Polat and Du, 2006; Kaleli and Polat, 2007):

One-group scheme: In one-group scheme, all ratings of items are reversed together or all of them remain the same. The accuracy gained from perturbed data is as high as the accuracy gained from original one for this scheme. Even though accuracy is high, the privacy level is low. If the server learns the true rating for

any item, it can easily estimate the true ratings of other items. One-group scheme can be defined as followed:

- In order to mask a user profile
  - Each user u generates a random number r using uniform distribution over the range [0, 1].
  - If r is smaller than or equal to the masking parameter θ (θ is determined by the server), the user sends true ratings; otherwise, she reverses 1s to 0s and 0s to 1s and sends reversed ratings to the server.
- The server produces accurate predictions without learning whether ratings are true or false utilizing the fact that the user sends true ratings with probability θ and sends false ratings with probability 1-θ.

Multi-group scheme: In multi-group schemes, items are partitioned into groups and each group is processed independently. The server chooses θ and each user selects a random number for each group and sends true or false data depending on selected random numbers and θ. Thus, even if the server learns true rating for an item in a group, it can estimate only true ratings of the other items in the same group. Remaining items in the other groups continue to protect their true ratings. In multi-group scheme, number of groups is represented by M, where $1 < M < m$ (m is the number of items). With increasing M values, privacy increases but accuracy decreases due to randomness.

Full Privacy: Preventing the server from learning the rated items is as important as preventing it from learning the true ratings. Hereby, besides disguising true ratings, each user u finds number of rated items, m_{ut}, and randomly creates a number, m_{ur}, which is in the range of (1, m_{ut}). Then, the user selects m_{ur} unrated items and fills randomly $m_{ur}/2$ items' values with 1 and the remaining unrated items with 0 (Polat and Du, 2006). In (Kaleli and Polat, 2007), privacy is provided similarly; however, in order to provide full privacy, each user selects m_{ur} over the range (1, γ). The user calculates the number of unrated items will be filled f, where $f = m_{ut} \times m_{ur} / 100$. Then, the user fills randomly $f/2$ items' values with 1 and the remaining unrated items with 0. In this dissertation, the scheme proposed by Kaleli and Polat (2007) is used as disguising scheme both in detection and designing attack profiles.

4. DESIGNING SHILLING ATTACK PROFILES FOR BINARY DISGUISED DATABASES

Binary data can be disguised using RRTs (Du and Zhan, 2003; Polat and Du, 2006; Kaleli and Polat, 2007). In this dissertation, data is disguised with RRTs as described in Section 3.3. The six well-known attack models are modified in the thesis in order to manipulate produced predictions from binary masked data. For this purpose, some values like modes of items, which are required for generating attack profiles are calculated as the server does in RRT with one-group scheme. The modified attack models can be applied to binary data disguised with RRT according to one-group and multi-group schemes with/without full privacy. A general procedure for generating shilling attack profiles on binary data is described in Section 3.2. As the modified shill profiles can be applied to the system, they can be disguised with RRT according to one- or multi-group schemes in order to obstruct detection of them. This time, even though effect of disguised modified attack models on binary masked data is lower than their undisguised versions, detection of them will get harder. The general shilling attack design scheme for one- and multi-group versions of RRT is described in Section 4.1. Furthermore, to obstruct detectability of disguised attack profiles, unrated items of the shill profiles are filled as genuine users. The mentioned procedures are described in Section 4.2 and 4.3.

4.1. Generating Shilling Attacks against Binary Data Disguised with RRTs

For generating attacks against binary masked data, understanding RRT with one-group scheme is important. Items' modes are used for generating average attack model, and popular and unpopular items are specified for bandwagon and reverse bandwagon attack models, respectively. For one- and multi-group schemes, ratings of items are true with probability θ or false with probability 1-θ. Thus, the estimated mode approaches its real value depending on pre-determined value of θ. If θ is 1, true data is sent. If θ is 0, all binary ratings are reversed and thus, false data is sent. For both situations, real values of items' modes can be

calculated. For other values of θ, if θ closes to 1 or 0, estimated modes of items increasingly approach to their real values. When θ is 0.5 or closes to 0.5, the worst results are gained because ratings are either true or false with probabilities near to each other. Mode of an item, M_r, is estimated as follows:

$$M_r = \frac{((-1)*\theta*M + (1-\theta)*\bar{M})}{1-2\theta}, \quad (4.1)$$

where, M represents the mode of the item gained from the ratings' vectors in the system, and $\bar{M}$ represents the mode of the item gained from the reversed ratings.

4.1.1. Random attack model

Random attack model can be applied as proposed by Kaleli and Polat (2013). The procedure is described as follows:

- Number of filler items and number of attack profiles are determined by the attacker.
- The attacker chooses a random number r for each filler item chosen randomly and fills with 1 if r is larger than 0.5; otherwise, she fills with 0.
- The attacker fills the target item with either 1 or 0 for intend of pushing or nuking, respectively.

4.1.2. Average attack model

Average attack model can be modified in order to manipulate produced predictions from binary masked data as follows:

- Number of filler items and number of attack profiles are determined by the attacker.
- The attacker estimates modes of filler items using Eq. (4.1). She fills each filler item with its estimated mode.
- The target item is filled with either 1 or 0 for push or nuke attacks, respectively.

4.1.3. Bandwagon attack model

Bandwagon attack model is modified as follows:

- Number of filler items, number of selected items, and number of attack profiles are determined by the attacker.
- The attacker estimates modes of items using Eq. (4.1). Then, she estimates the number of ratings for each item and sorts the items in descending order depending on number of ratings. The attacker chooses top number of selected items from the ranked item list, where mode of the item is 1.
- After choosing the selected items, the attacker fills them with 1.
- The attacker chooses a random number r for each filler item chosen randomly and fills with 1 if r is larger than 0.5; otherwise, she fills with 0.
- Target item is filled with 1.

4.1.4. Segment attack model

Segment attack model can be applied as proposed by Kaleli and Polat (2013). The steps are listed in the following:

- Number of filler items, number of selected items, and number of attack profiles are determined by the attacker.
- Selected items are filled with 1.
- The attacker chooses a random number r for each filler item chosen randomly and fills with 1 if r is larger than 0.5; otherwise, she fills with 0.
- Target item is filled with 1.

4.1.5. Reverse bandwagon attack model

Reverse bandwagon attack model is redesigned as follows:

- Number of filler items, number of selected items, and number of attack profiles are determined by the attacker.

- The attacker calculates modes of items using Eq. (4.1). Then, she calculates the number of ratings for each item and sorts the items in descending order depending on the number of ratings. The attacker chooses top number of selected items from the ranked item list, where mode of the item is 0.
- After choosing selected items, the attacker fills them with 0.
- The attacker chooses a random number r for each filler item chosen randomly and fills with 1 if r is larger than 0.5; otherwise, she fills with 0.
- Target item is filled with 0.

4.1.6. Love/hate attack model

Love/hate attack model can be applied as proposed by Kaleli and Polat (2013). The procedure is described as follows:

- Number of filler items and number of attack profiles are determined by the attacker.
- The attacker fills each filler item with 1.
- The attacker fills the target item with 0.

4.2. Disguising Modified Attack Models

The modified attack models mentioned in Section 4.1 are effective on produced predictions. However, since they are not disguised, detectability of them might be high. Thus, the disguising scheme presented in (Du and Zhan, 2003; Kaleli and Polat, 2007) can be used for masking attack profiles as genuine users.

4.2.1. Disguising modified attack models with one-group RRT

The disguising procedure for one-group scheme is as follows (Du and Zhan, 2003; Kaleli and Polat, 2007):

- The server selects θ over the range [0, 1] and lets each user know.

- The attacker generates attack profiles as in Section 4.1.
- The attacker uniformly randomly chooses a number α over the range [0, 1] for each modified attack profile. If α is larger than θ, she reverses 0s to 1s and 1s to 0s; otherwise, the attacker sends true ratings.

4.2.2. Disguising modified attack models with multi-group RRT

The attack profiles can be disguised for multi-group scheme (Kaleli and Polat, 2007). The procedure is as follows:

- The server selects θ over the range [0, 1], determines number of groups M, and lets each user know them.
- The attacker generates attack profiles as in Section 4.1.
- The attacker divides items into M groups.
- The attacker uniformly randomly generates M random numbers like r_1, r_2,…, r_M for each attack profile. For each group of a profile, if the group's random number is larger than θ, she reverses 0s to 1s and 1s to 0s; otherwise, the attacker sends true ratings.

4.3. Disguising Modified Attack Models with Full Privacy

The best results with privacy-preserving NBC-based CF algorithm are gained in terms of privacy level and accuracy when the group number is three and data is masked with full privacy (Kaleli and Polat, 2007). In order to improve privacy level, the server does not learn how many or which items a user rated. Thus, the unrated items are filled using random filling (Kaleli and Polat, 2007). Hence, aiming to obstruct detection of the modified attack profiles, the unrated items can also be disguised. The procedure to disguise attack profiles for one- and multi-group schemes is as follows:

- The server selects θ over the range [0, 1], determines number of groups M, specifies the percentage d, which is used for disguising unrated items, and lets each user know them.
- The attacker generates attack profiles as in Section 4.1.

- The attacker selects m_{ur} over the range (1, f) and calculates the number of rated items m_{ut} for each profile. The attacker calculates the number of unrated items will be filled d, where $d = m_{ut} \times m_{ur} / 100$. Then, the attacker fills randomly $d/2$ items' values with 1 and the remaining unrated items with 0.
- The attacker disguises attack profiles as in Section 4.2.

5. ROBUSTNESS ANALYSIS OF NBC-BASED PPCF SCHEME

A privacy-preserving CF scheme based on NBC with binary masked data is proposed by Kaleli and Polat (2007). Their empirical results show that NBC-based PPCF scheme still provides accurate recommendations while ensuring reasonable privacy level. Although the scheme is able to preserve privacy and provide precise predictions, it might be subjected to shilling attacks. Its robustness against well-known profile injection attacks is not scrutinized. Hence, a robustness analysis is performed to show how effective the proposed modified shilling attack models on binary data against the NBC-based PPCF scheme proposed by Kaleli and Polat (2007).

To perform such analysis, six modified shilling attack models are utilized. Different sets of experiments are conducted using real data. Several experiments are performed for evaluating the effectiveness of the proposed modified attack models on NBC-based PPCF scheme. The data set and the evaluation criteria used in the experiments are described in Section 5.1. Empirical outcomes are presented in Section 5.2.

5.1. Data Set and Evaluation Criteria

Various experiments are conducted using a benchmark data set in order to measure the effectiveness of the proposed shilling attack generation schemes on NBC-based PPCF scheme. Publicly available data set MovieLens Public (MLP) is used for experiments, which includes 100,000 ratings in a 5-star rating scale from 943 users for 1,682 movies. *Ratio shift* is used as an evaluation criteria in order to measure how the proposed schemes are effective on produced predictions. The measure can be described as follows (Kaleli and Polat, 2013):

$$Ratio\ Shift = Ratio\ of\ 1s\ after\ attack - Ratio\ of\ 1s\ before\ attack \quad (5.1)$$

Eq. (5.1) is proposed for push attacks. In order to measure the success of a nuke attack, instead of ratios of 1s, ratios of 0s before and after attacks are calculated.

5.2. Experimental Results

To show how much the proposed schemes are affected by varying values of different control parameters, various sets of experiments are performed. There are some privacy parameters. Moreover, *filler size* and *attack size* are two control parameters affecting the success of shilling attacks (Gunes et al., 2014). Filler size is related to number of filled cells in the attack profiles. Attack size is the ratio of the attack profiles to number of profiles. The proposed schemes are applied to NBC-based PPCF algorithm disguised with one-group scheme, multi-group scheme and their full privacy versions. In (Kaleli and Polat, 2007), empirical outcomes show that $\theta = 0.7$ gives the best results for both one- and multi-group schemes in terms of accuracy and privacy level. According to the results, for multi-group scheme, $M = 3$ gives the best results with respect to accuracy and privacy level. Therefore, to specify how the proposed schemes perform according to varying filler size and attack size values, θ is chosen as 0.7 for both schemes and M is chosen as 3 for multi-group scheme. Also, some experiments are performed to show influences of varying M, f, and θ values on success of the proposed schemes. Each experiment is performed for 50 nuke items and 50 push items. In all of the experiments, to calculate average ratio shift, the ratio shift values for each user in the data set is calculated for 50 push or nuke items. For all of the experiments, one user is selected as an active user and the others are selected as train users. This procedure is repeated for all users in the data set.

5.2.1. One-group scheme without full privacy

Some experiments are performed in order to specify how varying filler size, attack size, and θ values affect the success of the proposed attack models on NBC-based PPCF scheme. θ is first chosen as 0.7 while varying filler size and attack

size values. Attack size is set to 15 while filler size values are varied from 1 to 15. Then, filler size is set to 15 while varying attack size values from 1 to 15. Morever, to show how varying θ values impact the performance of the proposed schemes, finally, filler size and attack size are set to 15 while varying θ from 0.51 to 1.

Table 5.1 shows the effects of varying attack size values on overall performances of shilling attack profiles generated using the scheme in Section 4.1. Empirical outcomes indicate that modified average, bandwagon, reverse bandwagon, and love/hate attack models are effective on NBC-based PPCF scheme with one-group scheme. It is obvious that increasing attack size values improve the effectiveness of attack profiles on produced predictions.

Impacts of varying filler size values on performances of shilled profiles designed in Section 4.1 are shown in Table 5.2. As it is seen from Table 5.2, incremental values of filler size enhance influences of the proposed attack profiles on produced recommendations. In Table 5.1 and 5.2, the empirical results indicate that the most successful attack model is average attack model when θ = 0.7. The only factor that affects the success of random attack model is filler size. Since filler items are filled with either 1 or 0 according to chosen α value, the success of random attack model changes according to given ratings on filler items. The random attack profiles may or may not look like the genuine profiles.

Table 5.1. Effects of varying attack sizes on performance (no masking, one-group w/o full privacy)

Attack Size (%)	**1**	**3**	**6**	**10**	**15**
RA (Push)	-1.673	-0.617	0.867	1.060	2.197
AA (Push)	22.721	43.866	53.722	57.858	59.457
BA	8.163	20.530	31.071	38.344	42.261
SA	1.775	9.315	12.940	14.747	15.678
RA (Nuke)	1.697	4.477	6.694	8.144	9.898
AA (Nuke)	25.580	41.116	47.671	50.609	51.843
RB	5.616	13.599	18.929	23.027	25.773
L/H	12.609	20.157	23.357	24.683	25.493

Table 5.2. Effects of varying filler sizes on performance (no masking, one-group w/o full privacy)

Filler Size (%)	**1**	**3**	**6**	**10**	**15**
RA (Push)	0.475	0.522	0.456	0.920	2.197
AA (Push)	18.662	35.519	46.110	54.286	59.457
BA	32.293	36.842	39.875	41.733	42.261
SA	8.473	12.723	14.672	15.287	15.678
RA (Nuke)	20.229	19.811	16.580	12.454	9.898
AA (Nuke)	34.371	43.533	48.263	50.596	51.843
RB	17.090	26.282	28.240	27.599	25.773
L/H	23.128	24.734	25.275	25.379	25.493

Also, to show how the disguising parameter θ affects performance of the proposed attack models on NBC-based PPCF masked with one-group using RRT, some experiments are performed. The results are displayed in Table 5.3.

Table 5.3. Effects of varying θ values on performance (no masking, one-group w/o full privacy)

θ	**0.51**	**0.6**	**0.7**	**0.85**	**1**
RA (Push)	2.197	2.810	2.197	2.727	2.81
AA (Push)	6.045	56.301	59.457	60.814	61.097
BA	28.185	41.597	42.261	44.117	43.792
SA	15.678	15.805	15.678	15.792	15.805
RA (Nuke)	8.774	9.022	9.898	9.022	9.9
AA (Nuke)	3.133	50.630	51.843	53.109	53.287
RB	-11.211	19.453	25.773	25.756	25.701
L/H	25.533	25.557	25.493	25.557	25.557

As it is shown in Table 5.3, average attack model is affected by θ more than other attack models. With decreasing values of θ from 1 to 0.51, the successes of the attack profiles are less except random attack model. When θ approaches to 1 or 0, the proposed attack models except some attack models become more effective on produced predictions. Since attack size and filler size values are set to 15 and enough large while varying θ values, love/hate and segment attack models are not affected so much by θ. Even if θ is 0.51, at least half of the attack profiles

look like each other, the ratio shift values of them will be similar with the values when θ is 1. Also, success of the random attack model is independent of varying values of θ. Since filler items of random attack model are randomly filled and filler items are only factor on success of the attack model, the disguised genuine profiles may resemble random attack profiles more than their original forms depending on the given ratings to the filler items. Although filler items are filled with the same strategy of random attack model for bandwagon and reverse bandwagon attack models, selected items are dependent on θ. As it is seen from Table 5.1, Table 5.2, and Table 5.3, most of ratio shift values of the proposed attack models are efficient when attack size and filler size values are small and θ is close to 0.51.

Also, shilling attack profiles are generated according to the scheme described in Section 4.2. In other words, shilling attack profiles are generated and disguised like as genuine users according to one-group RRT without full privacy. Since the disguised and undisguised forms of attack profiles give the same results, the experimental outcomes are not repeated. Even if both undisguised and disguised forms of shilling attack profiles for one-group scheme are evenly successful, disguised forms of them are detected less than their undisguised ones.

5.2.2. One-group scheme with full privacy

Some trials are performed to specify how varying filler size, attack size, θ, and f values (note that f is percentage of unrated items will be filled) affect the successful of the proposed shilling attack generation schemes described in Section 4 on NBC-based PPCF scheme with full privacy provided one-group RRT. θ and f are chosen as 0.7 and 50, respectively while varying filler size and attak size values. To show how varying θ values impact the performance of the proposed schemes, filler size and attack size are set to 15 while varying θ from 0.51 to 1. Moreover, to demonstrate effects of varying f values on the proposed scheme, filler size and attack size values are set to 15 while varying f from 100 to 10. First, attack profiles are generated utilizing the scheme described in Section 4.1 and applied to the NBC-based PPCF scheme disguised with one-group RRT with full

privacy. Then, attack profiles are generated utilizing the schemes described in Section 4.2 and 4.3. Since the attack profiles generated by the schemes in Section 4.1 and 4.2 have the same impacts on the produced predictions, the empirical results are shown in the same tables.

Table 5.4 shows effects of varying attack size values of the proposed schemes described in Section 4.1 and 4.2 on produced predictions. As it is seen in Table 5.4, even if data is disguised utilizing one-group RRT scheme with full privacy, NBC-based PPCF scheme is still vulnerable to shilling attacks. Even though regenerated attack models are disguised with one-group RRT, they are effective as much as their undisguised versions on the produced predictions. The full privacy provided for NBC-based PPCF makes ratio shift values of average attack model to decrease. The attack models whose filler items are filled randomly may increase completely depending on the randomly filled items. Table 5.4 shows that increasing attack size values improve effects of the proposed attack models.

Table 5.4. Effects of varying attack sizes on performance (no masking, one-group w/o full privacy)

Attack Size (%)	**1**	**3**	**6**	**10**	**15**
RA (Push)	0.267	0.804	2.337	3.803	4.942
AA (Push)	14.093	35.459	48.861	55.321	57.902
BA	5.972	17.926	30.889	39.268	43.902
SA	3.869	10.829	14.392	16.244	17.442
RA (Nuke)	0.838	1.979	3.033	3.440	5.525
AA (Nuke)	18.069	33.302	40.554	43.917	47.179
RB	3.268	8.747	13.413	16.969	21.196
L/H	7.676	14.598	17.644	18.825	21.253

To obstruct detection of the proposed attack profiles described in Section 4.1, they are disguised utilizing the scheme in Section 4.3 like as genuine users. Table 5.5 shows how varying attack size values influence success of the proposed generation scheme described in Section 4.3 on NBC-based PPCF masked utilizing the one-group RRT scheme with full privacy. Table 5.5 remarks that increasing attack size values enhance ratio shift values of the proposed attack models.

Table 5.5. Effects of varying attack sizes on performance (one-group with full privacy)

Attack Size (%)	**1**	**3**	**6**	**10**	**15**
RA (Push)	0.612	1.283	3.120	4.651	5.871
AA (Push)	11.985	32.316	47.126	55.003	58.199
BA	5.622	16.825	29.650	38.45	43.548
SA	1.864	8.280	13.290	15.813	17.389
RA (Nuke)	0.405	0.933	1.472	1.548	3.283
AA (Nuke)	14.282	29.690	38.085	42.664	46.848
RB	2.772	7.858	12.106	15.739	19.915
L/H	4.284	9.646	13.285	15.718	19.461

Table 5.6 shows impacts of varying filler size values on the proposed generation schemes described in Section 4.1 and 4.2. The empirical outcomes demonstrate that increasing filler size values mostly enhance ratio shift values of the attack models. As it is mentioned before, the attack models whose filler items are filled randomly, may increase or decrease with increasing filler size values depending on the votes given to the filler items. As it is seen from Table 5.7, increasing filler size values generally improve effectiveness of the proposed attack models whose filler items are not filled randomly. Since f is enough smaller for mentioned attack models, they are still effective with higher filler size values.

Table 5.6. Effects of varying filler sizes on performance (no masking, one-group w/o full privacy)

Filler Size (%)	**1**	**3**	**6**	**10**	**15**
RA (Push)	2.216	1.934	2.284	3.432	4.942
AA (Push)	17.654	34.611	45.502	52.887	57.902
BA	35.319	39.695	42.528	43.902	43.902
SA	9.703	14.431	16.271	16.961	17.442
RA (Nuke)	14.978	14.187	10.636	7.584	5.525
AA (Nuke)	26.598	35.701	40.874	43.533	47.179
RB	11.847	20.817	22.352	20.999	21.196
L/H	16.937	19.001	19.502	19.690	21.253

Table 5.7. Effects of varying filler sizes on performance (one-group with full privacy)

Filler Size (%)	**1**	**3**	**6**	**10**	**15**
RA (Push)	2.341	2.954	3.495	4.989	5.871
AA (Push)	16.295	33.960	45.843	53.285	58.199
BA	42.392	43.321	44.049	44.112	43.549
SA	15.217	16.626	17.083	16.982	17.389
RA (Nuke)	10.867	9.338	6.967	4.487	3.283
AA (Nuke)	23.707	33.966	39.432	43.001	46.684
RB	22.078	21.601	20.733	19.459	19.915
L/H	13.326	15.116	15.758	16.937	19.461

Table 5.8 demonstrates how varying θ values affect performances of the attack profiles generated by the schemes in Section 4.1 and 4.2. Table 5.9 represents how effectively the attack profiles generated as in Section 4.3 perform with varying θ values. When θ closes or equals to 1, the number of masked profiles decrease and reach to 0. Thus, the best results are gained for most of the attack models for three schemes, as it is shown in Tables 5.8 and 5.9.

Table 5.8. Effects of varying θ values on performance (no masking, one-group w/o full privacy)

θ	**0.51**	**0.6**	**0.7**	**0.85**	**1**
RA (Push)	4.861	5.692	4.942	5.277	5.277
AA (Push)	7.690	52.657	57.902	61.644	62.333
BA	28.229	42.797	43.902	45.186	45.186
SA	18.354	18.375	17.442	18.403	18.403
RA (Nuke)	3.676	3.578	5.525	3.720	3.569
AA (Nuke)	-4.598	38.299	47.179	45.533	45.569
RB	-15.909	11.648	21.196	18.424	19.975
L/H	18.348	18.420	21.253	18.269	18.330

Table 5.9. Effects of varying θ values on performance (one-group with full privacy)

θ	0.51	0.6	0.7	0.85	1
RA (Push)	5.722	6.242	5.871	6.244	6.244
AA (Push)	7.858	53.139	58.199	62.151	62.950
BA	28.314	42.592	43.548	44.944	44.944
SA	18.511	18.454	17.389	18.424	18.424
RA (Nuke)	1.359	1.466	3.283	1.510	1.729
AA (Nuke)	-5.684	37.067	46.848	45.415	45.468
RB	-16.136	10.687	19.915	17.005	18.373
L/H	16.125	16.034	19.461	15.947	15.924

Table 5.10 shows how the number of filled items impresses the effectiveness of the proposed shilling attack generation schemes. When f is varied, the ratio shift values of attack models change depending on θ, f, and the votes given to the randomly filled items. When θ is large enough, ratio shift values are supposed to decrease with incresing f values.

Table 5.10. Effects of varying f values on performance (no masking, one-group w/o full privacy)

f (%)	100	50	25	10
RA (Push)	4.399	3.616	4.276	2.717
AA (Push)	60.702	59.612	60.002	58.766
BA	44.308	43.578	43.557	42.577
SA	18.464	17.166	17.186	16.049
RA (Nuke)	4.345	5.525	7.584	7.877
AA (Nuke)	44.348	47.179	43.533	49.824
RB	18.607	21.196	20.999	23.302
L/H	18.672	21.253	19.690	23.958

Disguising both data and attack profiles with full privacy will change the effectiveness of the proposed shilling attack generation scheme in Section 4.3 on NBC-based PPCF disguised by utilizing one-group RRT scheme with full privacy. Table 5.11 shows effects of varying f values on performances of the attack profiles generated as in Section 4.3. Filling unrated items of random,

bandwagon, and reverse bandwagon attack models equals to increasing filler size of them. Therefore, as it is mentioned before, the ratio shift values of mentioned attack models may increase or decrease with incremental filler size values depending on the votes given to the filler items. Since MLP data is sparse, the number of filled items is low even if *f* is 100. If the data set is dense or filler size is much more than 15, the value of *f* will be more effective on the successes of the average, segment, and love/hate attack models.

Table 5.11. Effects of varying *f* values on performance (one-group with full privacy)

f (%)	**100**	**50**	**25**	**10**
RA (Push)	5.186	4.462	5.092	2.717
AA (Push)	61.283	60.125	60.193	58.766
BA	43.438	43.363	43.497	42.577
SA	18.354	17.173	17.071	16.049
RA (Nuke)	1.031	3.283	4.488	7.270
AA (Nuke)	43.699	46.848	43.001	49.796
RB	16.443	19.915	19.459	22.865
L/H	15.871	19.461	16.937	23.671

When NBC-based PPCF scheme is masked utilizing one-group RRT scheme with full privacy, it is still possible to manipulate produced predictions. When attack models are generated as in either Section 4.1 or 4.2, they have the same impacts on the produced predictions. However, the attack models generated as in Section 4.2 are less detectable. When the scheme described in Section 4.3 is chosen as generation algorithm, the proposed attack models are successful as nearly much as the other versions and the detectability of the scheme is much less than the others.

5.2.3. Multi-group scheme without full privacy

The proposed shilling attack generation schemes are applied to NBC-based CF algorithm with multi-group RRT scheme. The best outcomes are gained when the number of groups $M = 3$ and the masking parameter $\theta = 0.7$ for NBC-based

PPCF with multi-group scheme (Kaleli and Polat, 2007). Thus, M and θ are set to 3 and 0.7, respectively while varying filler size and attack size values from 1 to 15. In order to measure robustness of NBC-based PPCF algorithm with multi-group scheme, initially, the attack profiles are generated using the procedure in Section 4.1 and applied to the system. Then, generated attack profiles are also disguised utilizing the procedure in Section 4.2 and injected to the database for obstructing detectability.

Table 5.12 and 5.13 show how varying attack size values influence performances of the attack models. Increasing attack size values enhance ratio shift values of the attack profiles for both of the shilling attack generation schemes. As it is seen in the tables, injecting attack profiles without disguising gives more successful results but eases detactability. Also, undisguised push attacks require modes of items such as average and bandwagon attack models are much more successful than their disguised versions because of items' properties, θ, and M. To prevent these differences, choosing target push or nuke items is important.

Table 5.12. Effects of varying attack sizes on performance (no masking)

Attack Size (%)	**1**	**3**	**6**	**10**	**15**
RA (Push)	-1.917	-2.783	-2.787	-2.993	-1.266
AA (Push)	22.244	38.115	46.040	49.393	50.920
BA	9.741	20.740	28.848	34.102	36.859
SA	-0.829	3.192	6.036	6.764	7.546
RA (Nuke)	1.616	4.093	5.264	5.902	6.235
AA (Nuke)	25.031	38.450	44.208	46.373	47.133
RB	5.173	11.277	15.936	19.103	20.973
L/H	11.652	16.768	19.408	20.554	21.143

Table 5.13. Effects of varying attack sizes on performance (multi-group w/o full privacy)

Attack Size (%)	**1**	**3**	**6**	**10**	**15**
RA (Push)	-2.271	-3.077	-3.065	-2.411	-2.028
AA (Push)	3.421	13.508	18.422	23.226	25.082
BA	-0.176	5.722	9.256	10.356	12.507
SA	-2.153	0.780	3.018	4.125	4.802
RA (Nuke)	2.042	3.381	4.587	6.127	6.297
AA (Nuke)	18.590	33.421	40.524	43.692	44.821
RB	3.485	7.050	10.694	13.022	14.944
L/H	10.394	19.245	23.385	25.317	26.170

In Tables 5.14 and 5.15, the ratio shift values of average, segment, and love/hate attack models enhance with incremental filler size values for both generation schemes. However, performances of random, bandwagon, and reverse bandwagon attack models may increase or decrease with increasing filler size values depending on the votes given to the filler items for both schemes. Thus, the idea of that incremental filler size values enhance performances of all attack models is not possible.

Table 5.14. Effects of varying filler sizes on performance (no masking)

Filler Size (%)	**1**	**3**	**6**	**10**	**15**
RA (Push)	-7.601	-7.446	-5.712	-4.106	-1.266
AA (Push)	12.615	28.229	38.106	45.811	50.920
BA	24.702	29.082	31.557	35.858	36.859
SA	-0.271	4.152	5.504	7.396	7.546
RA (Nuke)	17.729	17.262	13.451	10.505	6.235
AA (Nuke)	30.244	39.26	44.148	46.927	47.133
RB	13.8119	19.529	21.718	20.891	20.973
L/H	19.667	20.744	21.213	21.608	21.143

Table 5.15. Effects of varying filler sizes on performance (multi-group w/o full privacy)

Filler Size (%)	**1**	**3**	**6**	**10**	**15**
RA (Push)	-7.843	-7.845	-5.890	-4.011	-2.028
AA (Push)	0.210	8.643	16.477	20.749	25.082
BA	-5.567	-3.022	1.962	10.155	12.507
SA	-3.960	1.406	3.701	5.287	4.802
RA (Nuke)	17.601	17.317	13.739	10.551	6.297
AA (Nuke)	26.076	35.179	40.861	44.121	44.821
RB	7.128	12.585	15.463	14.795	14.944
L/H	20.163	22.070	24.008	25.854	26.170

Tables 5.16 and 5.17 show how the proposed attack models generated as in Section 4.1 and 4.2 perform depending on varying θ values. As it is seen from the tables, the attack models generated by either the first scheme or the second scheme manipulate target items' predictions much more when θ has higher values. When θ aproaches to 0.5, the privacy level increases but effects of attack profiles on predictions produced by NBC-based PPCF with multi-group RRT scheme decrease.

Table 5.16. Effects of varying θ values on performance (no masking)

θ	**0.51**	**0.6**	**0.7**	**0.85**	**1**
RA (Push)	-4.768	-3.442	-1.266	1.663	2.772
AA (Push)	4.827	42.365	50.920	57.090	61.035
BA	17.047	32.498	36.859	43.041	43.777
SA	2.808	4.598	7.546	12.778	15.762
RA (Nuke)	5.578	6.212	6.235	8.197	9.113
AA (Nuke)	7.370	42.439	47.133	51.296	53.285
RB	-13.425	12.904	20.973	25.531	25.798
L/H	18.967	19.970	21.143	23.779	25.502

Table 5.17. Effects of varying θ values on performance (multi-group w/o full privacy)

θ	**0.51**	**0.6**	**0.7**	**0.85**	**1**
RA (Push)	-4.821	-3.442	-2.028	1.813	2.772
AA (Push)	0.808	10.216	25.082	48.309	61.035
BA	2.424	4.903	12.507	34.034	43.777
SA	-0.112	1.889	4.802	11.088	15.762
RA (Nuke)	6.458	6.187	6.297	8.021	9.113
AA (Nuke)	8.460	39.198	44.821	50.341	53.285
RB	-1.459	8.240	14.944	23.750	25.798
L/H	25.599	25.981	26.400	26.441	25.502

The number of groups, *M*, is one of the agents impresses the performances of both schemes. As it is seen from Tables 5.18 and 5.19, the effects of attack profiles generally decrease with increasing *M* values. Ratio shift values of some attack models such as random attack model for both schemes may increase with higher *M* values depending on the votes given to the filler items.

Table 5.18. Effects of varying *M* values on performance (no masking)

M	**1**	**2**	**3**	**5**
RA (Push)	2.197	0.723	-1.266	-1.758
AA (Push)	59.457	55.213	50.92	49.557
BA	42.261	39.408	36.859	35.612
SA	15.678	11.315	7.546	5.754
RA (Nuke)	9.898	5.512	6.235	7.540
AA (Nuke)	51.843	46.874	47.133	47.618
RB	25.773	17.413	20.973	21.572
L/H	25.493	20.305	21.143	21.041

As it is seen from the tables represented in Section 5.2.3, the ratio shift values of disguised average and bandwagon attack models are much smaller than their unmasked forms. However, when average attack model is used for nuking, the difference between its masked and unmasked forms decreases. The reason of

that this situation happens for only average and bandwagon attack models is that they need means of the items. Also, the cause of the difference between ratio shift values of their undisguised and disguised forms depends on some factors like θ, M, the number of votes given to the target items, etc.

Table 5.19. Effects of varying M values on performance (multi-group w/o full privacy)

M	**1**	**2**	**3**	**5**
RA (Push)	2.197	0.081	-2.028	-1.228
AA (Push)	59.457	35.109	25.082	16.013
BA	42.261	24.532	12.507	3.313
SA	15.678	9.446	4.802	2.961
RA (Nuke)	9.898	6.189	6.297	7.101
AA (Nuke)	51.843	45.86	44.821	43.994
RB	25.773	14.445	14.944	13.101
L/H	25.493	24.151	26.400	26.575

The items whose modes equal to 0 are chosen as target items for pushing. Conversely, the items which have modes as 1 are chosen as target items for nuking. Since MLP is sparse, for pushing, the percentage of the target items whose number of given votes is smaller than 40 is 50. Unlike the target items for pushing, that percentage is smaller than 20 for the target items to nuke. Besides θ and M, the number of votes given to a target item is another factor, which affects performances of the undisguised and disguised forms of the attack models. When the number of votes is large enough (more than 40), even though the calculated means of items deviate so much from their originals, the masked and unmasked attack models can mostly give similar ratio shift values.

When the calculated mean of disguised data deviates so much for the items with smaller number of votes, the ratio shift value of disguised attack models generally differs so much than their unmasked versions. The effect of θ on this situation is represented in Tables 5.16 and 5.17. The influence of M is shown in Tables 5.18 and 5.19. Also, to illustrate the effects of target items' properties, some examples are represented in Table 5.20. In Table 5.20, # represents the number of votes given to a target item and d represents the difference between

mean of an unmasked item and mean of its masked form. The ratio shift values of four items are represented. The idea is that either only θ, M, or item's property is the reason of the difference is not enough. The number of disguised items of an attack profile and the number of disguised attack profiles should be considered.

Table 5.20. Effects of target items' properties on performance of masked and unmasked attacks

Property	**# < 40**		**# > = 40**	
	$d < 0.05$	$d >= 0.05$	$d < 0.05$	$d >= 0.05$
AA (Unmasked)	62.460	60.445	62.884	69.459
AA (Masked)	61.082	-0.848	59.915	65.854
BA (Unmasked)	46.871	47.190	39.661	49.629
BA (Masked)	47.508	7.211	39.237	49.311

Generally speaking, even if NBC-based CF is masked with multi-group scheme RRT without providing full privacy, effective attack models can be generated. The performances of the proposed attack models enhance mostly with increasing θ, attack size, filler size, and decreasing M values. When attack profiles are generated as in Section 4.1 and injected to the system, they have admissible impacts on produced predictions with $\theta = 0.7$ and $M = 3$ values. Under the same conditions, the masked forms of them have also high ratio shift values with aggravated detactabilities. As it is shown in the tables in Section 5.2.3, the average attack model is the most successful attack model for the scheme discussed in Section 4.1. For the second scheme, the average attack model is also the most effective attack model when the target items are chosen carefully or with high θ and low M values. For both of the schemes, bandwagon, reverse bandwagon, and love/hate attacks are successful. When approppriate conditions are provided, the masked forms of the proposed schemes can be preferred for manipulating predictions in order to obstruct detectability.

5.2.4. Multi-group scheme with full privacy

The proposed shilling attack generation schemes are applied to NBC-based CF algorithm masked with full privacy provided multi-group RRT scheme. The best outcomes are gained when the number of groups M = 3 and the masking parameter θ is 0.7 for NBC-based PPCF with multi-group scheme (Kaleli and Polat, 2007). Thus, M and θ are set to 3 and 0.7, respectively while varying filler size and attack size values from 1 to 15. Also, to provide full privacy the masking parameter f is chosen as 50 during the experiments. In order to measure robustness of NBC-based PPCF algorithm masked with multi-group scheme and provides full privacy, the attack profiles are generated using the procedure in Section 4.1 and inserted to the system. Then, generated attack profiles are also disguised utilizing the procedure in Section 4.2 and injected to the database for obstructing detectability. Lastly, the attack profiles are generated as in Section 4.3 to provide full privacy as genuine users aiming to aggravating further detactability.

Tables 5.21, 5.22, and 5.23 show how varying attack size values affect performances of the proposed attack models on NBC-based CF masked with multi-group RRT scheme and provides full privacy.

Table 5.21. Effects of varying attack sizes on performance (no masking)

Attack Size (%)	**1**	**3**	**6**	**10**	**15**
RA (Push)	-0.229	1.003	1.606	2.770	1.737
AA (Push)	15.529	37.419	49.012	53.552	54.721
BA	6.742	20.117	32.019	39.994	41.065
SA	3.024	8.817	11.635	13.294	11.945
RA (Nuke)	0.530	1.120	2.590	2.770	3.037
AA (Nuke)	18.719	33.364	39.847	42.689	43.650
RB	2.761	7.003	10.937	13.514	15.724
L/H	7.226	13.705	16.14	17.355	18.187

Table 5.22. Effects of varying attack sizes on performance (multi-group w/o full privacy)

Attack Size (%)	**1**	**3**	**6**	**10**	**15**
RA (Push)	-0.178	0.973	1.782	3.332	1.620
AA (Push)	2.810	13.784	19.654	25.279	29.843
BA	0.823	5.826	11.116	15.525	16.197
SA	0.443	5.128	8.534	10.448	9.241
RA (Nuke)	0.829	1.497	2.233	3.300	3.820
AA (Nuke)	12.388	26.785	35.771	39.396	41.024
RB	1.506	3.998	6.062	8.600	11.230
L/H	6.611	16.195	19.726	22.418	23.224

Table 5.23. Effects of varying attack sizes on performance (multi-group with full privacy)

Attack Size (%)	**1**	**3**	**6**	**10**	**15**
RA (Push)	0.394	0.999	2.823	3.782	2.329
AA (Push)	3.661	11.924	20.288	26.850	32.002
BA	1.561	5.334	10.335	15.697	17.275
SA	-0.040	3.236	7.224	9.811	9.525
RA (Nuke)	0.448	0.394	1.343	1.137	1.309
AA (Nuke)	9.001	23.092	32.38	38.286	40.274
RB	1.281	3.016	5.232	7.743	9.404
L/H	4.276	9.684	15.396	19.101	21.627

As it is seen from the tables, average attack model is the most successful model. But, the best results are gained when it is undisguised. For the remaining attack models, even if the best results are gained when the attack models are as in their undisguised forms for most of them, the other generation schemes may increase their performance depending on given votes to the filler items. The tables show that increasing attack size values enhance performance of the attack models for three generation schemes.

Tables 5.24, 5.25, and 5.26 represent effects of varying filler size values on performances of the attack models generated by the schemes described in Sections 4.1, 4.2, and 4.3, respectively.

Table 5.24. Effects of varying filler sizes on performance (no masking)

Filler Size (%)	1	3	6	10	15
RA (Push)	0.075	-1.166	0.299	1.690	1.737
AA (Push)	17.175	33.635	43.548	50.505	54.721
BA	32.694	38.225	41.321	43.046	41.065
SA	7.945	11.251	13.173	13.756	11.945
RA (Nuke)	13.328	13.415	10.172	6.757	3.037
AA (Nuke)	26.329	34.942	39.385	42.053	43.650
RB	9.436	17.018	18.437	17.544	15.724
L/H	15.875	17.561	17.977	18.074	18.187

Table 5.25. Effects of varying filler sizes on performance (multi-group w/o full privacy)

Filler Size (%)	1	3	6	10	15
RA (Push)	-0.174	-0.694	0.233	1.970	1.620
AA (Push)	6.657	15.024	22.011	26.076	29.843
BA	1.885	5.372	9.724	14.988	16.197
SA	3.697	8.443	11.073	11.533	9.241
RA (Nuke)	13.48	13.383	10.318	6.967	3.820
AA (Nuke)	22.757	31.242	36.148	38.903	41.024
RB	2.490	9.118	11.784	11.514	11.230
L/H	16.131	19.194	20.664	22.651	23.224

Table 5.26. Effects of varying filler sizes on performance (multi-group with full privacy)

Filler Size (%)	1	3	6	10	15
RA (Push)	-0.008	0.422	1.856	3.387	2.329
AA (Push)	6.590	17.372	25.179	29.631	32.002
BA	9.459	11.955	15.050	17.050	17.275
SA	7.546	10.653	12.265	12.571	9.525
RA (Nuke)	10.229	8.471	6.265	3.220	1.309
AA (Nuke)	18.749	28.611	34.579	38.057	40.274
RB	11.750	11.915	11.283	10.691	9.404
L/H	12.201	14.706	16.995	20.091	21.627

As seen from Table 24, Table 25, and Table 26, the ratio shift values of average attack model for three schemes increase with incremental filler size values. For random, bandwagon, and reverse bandwagon attack models, increasing filler size values may enhance performances of the attack models generated by the algorithm in Section 4.1 due to the given ratings to the filler items. When the attack profiles are disguised as in Section 4.2, then the reversed votes given to the filler items may increase ratio shift values. For the mentioned attack models, the algorithm in Section 4.3 provides to increase filler item sets of their profiles. Increasing filler items may enhance performances of the profiles depending on the given votes to them. For three schemes, ratio shift values of the average and love/hate attack models increase with incremental filler size values.

Tables 5.27, 5.28, and 5.29 represent effects of varying θ values on performances of the attack models generated by the schemes in Sections 4.1, 4.2, and 4.3, respectively. Ratio shifts of the attack models generated by the scheme in Section 4.1 become more similar to their disguised versions when θ approaches to 1. When θ is 1, they give the same results. Also, the attack models generated as in Section 4.3, perform nearly successful as their other forms depending on the filled items for full privacy with $\theta = 1$ value. Many attacks perform successfully when θ approaches to 1 for the three schemes. For some attack models like love/hate attack model, their disguised and full privacy provided forms may perform more effectively when θ closes to 0.5 depending on their filler item rating strategy.

Table 5.27. Effects of varying θ values on performance (no masking)

θ	**0.51**	**0.6**	**0.7**	**0.85**	**1**
RA (Push)	2.844	3.692	1.737	5.102	5.071
AA (Push)	2.392	49.654	54.721	59.357	63.243
BA	22.887	38.819	41.065	44.339	47.385
SA	13.014	13.340	11.945	16.419	19.145
RA (Nuke)	2.683	3.234	3.037	4.341	3.067
AA (Nuke)	-5.684	37.249	43.650	45.266	45.673
RB	-20.498	9.440	15.724	19.256	18.674
L/H	16.210	16.187	18.187	18.151	18.288

Table 5.28. Effects of varying θ values on performance (multi-group w/o full privacy)

θ	**0.51**	**0.6**	**0.7**	**0.85**	**1**
RA (Push)	3.669	3.099	1.620	4.649	5.071
AA (Push)	4.547	16.899	29.843	50.346	63.243
BA	10.575	12.617	16.197	35.355	47.385
SA	9.616	10.000	9.241	14.698	19.145
RA (Nuke)	2.261	2.783	3.820	4.068	3.067
AA (Nuke)	-0.967	33.468	41.024	43.996	45.673
RB	-5.391	6.651	11.230	17.175	18.674
L/H	23.128	22.679	23.224	21.124	18.288

Table 5.29. Effects of varying θ values on performance (multi-group with full privacy)

θ	**0.51**	**0.6**	**0.7**	**0.85**	**1**
RA (Push)	4.534	4.621	2.329	5.684	5.983
AA (Push)	5.232	19.122	32.002	52.426	63.773
BA	10.940	12.507	17.275	35.737	47.088
SA	10.914	10.621	9.525	15.222	19.179
RA (Nuke)	0.187	0.613	1.309	2.059	1.016
AA (Nuke)	-2.492	31.722	40.274	44.110	45.555
RB	-6.850	5.135	9.404	15.786	17.243
L/H	20.151	20.984	21.627	20.098	15.866

Tables 5.30, 5.31, and 5.32 show effects of varying M values on performances of the attack models generated by the schemes in Sections 4.1, 4.2, and 4.3, respectively. When M is 1, the attack models generated utilizing the method in Section 4.2, perform as successful as their undisguised versions and the ratio shift values of their full privacy provided forms are so similar to their other versions due to the filled items. As it is obvious in the tables, incremental M values decrease ratio shift values of most of the attack profiles for three schemes since it increases privacy.

Table 5.30. Effects of varying *M* values on performance (no masking)

M	**1**	**2**	**3**	**5**
RA (Push)	4.942	4.418	1.737	3.313
AA (Push)	57.902	55.688	54.721	52.303
BA	43.902	45.313	41.065	40.986
SA	17.442	16.379	11.945	12.258
RA (Nuke)	5.525	2.505	3.037	2.293
AA (Nuke)	47.179	40.405	43.650	39.854
RB	21.196	14.246	15.724	15.432
L/H	21.253	15.790	18.187	15.251

Table 5.31. Effects of varying *M* values on performance (multi-group w/o full privacy)

M	**1**	**2**	**3**	**5**
RA (Push)	4.942	5.020	1.620	2.914
AA (Push)	57.902	37.236	29.843	24.134
BA	43.902	30.295	16.197	9.266
SA	17.442	14.115	9.241	9.200
RA (Nuke)	5.525	2.204	3.820	2.505
AA (Nuke)	47.179	38.384	41.024	36.195
RB	21.196	11.370	11.230	7.402
L/H	21.253	19.667	23.224	21.527

Table 5.32. Effects of varying *M* values on performance (multi-group with full privacy)

M	**1**	**2**	**3**	**5**
RA (Push)	5.871	5.572	2.329	4.386
AA (Push)	58.199	40.021	32.002	25.527
BA	43.548	31.811	17.275	11.082
SA	17.389	14.066	9.525	9.491
RA (Nuke)	3.283	0.708	1.309	0.045
AA (Nuke)	46.848	37.917	40.274	35.052
RB	19.915	10.303	9.404	7.3871
L/H	19.461	18.916	21.627	18.725

Effects of varying *f* values on performances of the attack models generated by the schemes in Sections 4.1, 4.2, and 4.3, are represented in Tables 5.33, 5.34, and 5.35, respectively. When *f* is varied, the ratio shift values of attack models change depending on θ, *M*, and the votes given to the randomly filled items. For average, love/hate, and segment attack models generated as in Sections 4.1 and 4.2, the ratio shift values are supposed to decrease with incresing *f* values when θ is large enough and *M* is small enough. For three shilling attack generation schemes, filling unrated items of random, bandwagon, and reverse bandwagon attack models equals to increasing filler size values of them. Hence, as it is mentioned before, the ratio shift values of mentioned attack models may increase or decrease with incremental filler size values depending on the votes given to the filler items. Since MLP is sparse, the number of filled items is low even if *f* is 100. If the data set is dense or filler size is much more than 15, the value of *f* will be more effective on the successes of the average, segment, and love/hate attack models. Varying *f* values do not have a distinct impact on performances of the attack models for three schemes since the other factors such as *M*, θ, votes of the filler items, and sparsity of the data set play a role on influence of *f* value.

Table 5.33. Effects of varying *f* values on performance (no masking)

***f* (%)**	**100**	**50**	**25**	**10**
RA (Push)	3.118	1.737	1.646	-0.897
AA (Push)	56.337	54.721	55.296	51.393
BA	44.008	41.065	41.12	37.932
SA	13.285	11.945	11.913	8.484
RA (Nuke)	4.780	5.351	6.106	6.717
AA (Nuke)	45.279	46.651	47.72	48.566
RB	16.278	18.484	19.502	20.259
L/H	19.014	20.121	21.544	22.422

Table 5.34. Effects of varying *f* values on performance (multi-group w/o full privacy)

***f* (%)**	**100**	**50**	**25**	**10**
RA (Push)	2.768	1.620	1.700	-1.071
AA (Push)	27.841	29.843	30.197	26.534
BA	16.689	16.197	16.178	13.552
SA	10.551	9.241	9.171	5.858
RA (Nuke)	5.128	5.230	6.250	6.638
AA (Nuke)	42.517	44.104	45.145	45.90
RB	11.830	12.624	14.284	15.550
L/H	24.507	25.843	26.753	27.849

Table 5.35. Effects of varying *f* values on performance (multi-group with full privacy)

***f* (%)**	**100**	**50**	**25**	**10**
RA (Push)	4.462	2.329	1.926	-0.821
AA (Push)	35.321	32.002	31.372	26.643
BA	21.130	17.275	16.594	13.739
SA	11.186	9.525	9.319	5.661
RA (Nuke)	1.735	3.790	4.657	6.028
AA (Nuke)	41.760	43.572	45.090	46.025
RB	8.791	11.34	13.351	14.700
L/H	21.527	25.124	26.494	27.743

Briefly, even if NBC-based CF masked with multi-group RRT scheme with providing full privacy, manipulating produced predictions effectively is still possible. When attack models are generated as in Section 4.1 and inserted to the system's database, they have acceptable effects on produced predictions with θ = 0.7, M = 3, and f = 50 values. Even though the average attack models require high knowledge about the system, it is still the most successful attack model among the other attack models generated as in Section 4.1. Even if the attack models are generated as in Section 4.2, they still successfully manipulate the produced predictions with obstructed detectabilities. Also, differences between undisguised and disguised forms of some attack profiles like average and bandwagon attack profiles seen in the tables are explained in Section 5.2.3. When the required

conditions discussed in Section 5.2.3 are provided, the average attack model becomes the most effective model for the second scheme. Moreover, to aggravate detectability of the proposed attack profiles, they can be disguised as genuine users. This time, the produced predictions are still manipulated in favor of the attackers while detactabilities of the attack profiles get difficult. For three schemes, average, bandwagon, reverse bandwagon, and love/hate attacks are successful. When approppriate conditions are provided, the full privacy provided forms of them can be preferred for manipulating predictions in order to obstruct detectability.

6. DETECTING SHILLING ATTACK PROFILES ON BINARY MASKED DATA

Effectiveness of a recommender system depends mostly on quality of data, so dealing with shilling attacks is extremely important. As it is shown in Section 5, even if data is binary and masked, manipulating produced predictions is still possible. Detecting bogus profiles is one of the influential ways of coping with profile injection attacks. In the literature, there are many detection schemes based on statistical methods, classification, clustering, variable selection, and other techniques (Gunes et al., 2014). Since the ratings can be only 1 or 0, classification is preferred rather than other methods in terms of performance and accuracy for this work. In Section 5, the certain generation strategy of shilling attack models are considered while regenerating them for binary masked data. For detecting modified attack profiles on binary masked data, some generic attributes derived from profiles by utilizing the work proposed by Chirita et al. (2005). Some derived attributes are adapted for binary masked data. The derived attributes used in the proposed shilling attack detection scheme for binary masked data are explained as follows:

1. **Dissimilarity in user's profile (dup):** This metric is generated utilizing the metric called as *standard deviation in user's profile* (Chirita et al., 2005). This metric represents how a profile differs its mode. For love/hate and segment attack models, this metric is supposed to be so small because of their filler item filling strategy. Especially for random, bandwagon, and reverse bandwagon attack models, the metric closes to 0.5. The metric is calculated as follows for binary data:

$$\text{dup} = \frac{\#\ of\ rated\ items\ of\ a\ profile\ differs\ from\ mode}{\#\ of\ rated\ items\ of\ a\ profile} \qquad (6.1)$$

2. **Agreement with other users (aou):** The metric is called as *degree of agreement with other users* (Chirita et al., 2005). It is adapted for binary masked data. It represents how much a profile agrees with the other profiles.

Whatever value of attack size is, this metric reaches its maximum value 1 for average attack model. The metric has also high values for segment and love/hate attack models except small values of attack size. For random, bandwagon, and reverse bandwagon attack models, this metric will be around 0.5. The metric is calculated as follows for binary data:

$$\text{aou}_a = \frac{\sum_{i=1}^{i=k} count}{k}\ , \qquad count = \begin{cases} 1, & if\ a_i = m_i \\ 0, & otherwise \end{cases} \tag{6.2}$$

In Equation 6.2, *k* represents the number of items rated by user *a* and m_i is the mode of item *i*.

3. **Similarity with top-*N* Neighbors (avgSim):** This metric is adapted for binary data. Since attack profiles are generated with a certain strategy, they look like each other so much. Thus, when attack size is large enough, the attack profiles have high similarities with their top *N* neighbors. *N* is chosen as 25 for this thesis. In order to calculate similarities between user profiles, Pearson's correlation is utilized. The metric is calculated as follows (Chirita et al., 2005):

$$avgSim_a = \frac{\sum_{i=1}^{i=N} W_{ai}}{N} \tag{6.3}$$

4. **Disagreement with possible target items (dti):** Shilling attackers try to increase popularity of an item with lower popularity or vice versa. This metric is derived for average, love/hate, and segment attack models. Possible target items are chosen as the ones with lower popularities for average and segment attack models. The items with higher popularities are chosen as target items for love/hate attack model. In terms of average attack model, since all filler items are filled with their own mode values, it has low **dti** value. For segment attack model, all filler items are filled with 0, so it also has lower **dti** value. For love/hate attack model, since 1 is assigned to all filler items, it has lower **dti** value. For other attack models, this metric is not a distinct property in that filler items are filled randomly. The metric is

calculated as follows, where *k* represents the number of items rated by user *a* among possible target items and $\boldsymbol{m_i}$ is the mode of item *i*:

$$\mathrm{dti}_a = \frac{\sum_{i=1}^{i=k} count}{k}, \qquad count = \begin{cases} 0, & if\ a_i = m_i \\ i, & otherwise \end{cases} \tag{6.4}$$

In order to detect shilling attack profiles, above mentioned metrics are derived from binary masked data. Binary data is disguised utilizing two algorithms as one- and multi-group full privacy provided RRT schemes. θ and f are set to 0.7 and 50, respectively for both schemes. M is set to 3 for multi-group RRT scheme. Expected values of derived attributes for attack models on binary data are shown in Table 6.1. Calculated values of the attributes differ from their original values depending on M, θ, f, and chosen shilling attack generation scheme. How to calculate the metrics on binary data masked with one- and multi-group RRT schemes are discussed in Sections 6.1 and 6.2, respectively.

As it is obvious in Section 5, the best results are gained when attack profiles are undisguised forms as in Section 4.1 for any of the mentioned masking algorithms. To obstruct detectability of them, they can be also generated as in Section 4.2 and 4.3 with admissible effectiveness. In order to classify profiles as bogus, the values of metrics for attack profiles will be different for each shilling attack generation scheme. The values of metrics for each shilling attack model deviate depending on θ, M, and f values.

Table 6.1. Expected values of derived attributes for each attack model on binary data

	dup	**aou**	**avgSim**	**dti**
RA (Push)	high	average values	average values	-
AA (Push)	high	high	high	low
BA	high	average values	average values	-
SA	low	high	high	low
RB	high	average values	average values	-
L/H	low	high	high	low

6.1. Calculation of Metrics with One-group RRT Scheme

In one-group RRT scheme with full privacy, all ratings of items are reversed together or all of them remain the same depending on chosen random number and θ. The details are discussed in Section 3.2. Some steps are explained as follows aiming to calculate values of the metrics from masked data.

- **Dissimilarity in user's profile:** Due to the one-group RRT scheme, even if a profile is disguised, its **dup** value does not change. Hence, Eq. (6.1) can be used in order to calculate **dup** value of a profile on binary masked data.
- **Agreement with other users:** Modes of items are needed for calculating aou for a profile. Modes of items are calculated described in Section 4.1 on binary masked data. Then, **aou** value is calculated by using the gained **aou** value on the user's rating vector with probability θ and the gained **aou** value on the user's reversed rating profile with probability 1-θ.

$$aou_a = aou_{R_a} * \theta + aou_{R'_a} * (1 - \theta) \tag{6.5}$$

- **Similarity with top-*N* Neighbors:** In order to calculate similarity between two masked user profiles R_a and R_b, partial similarity is calculated between R_a and R_b with probability θ^2 primarily. Then, partial similarities are calculated between R_a and R'_b , R'_a and R_b, and R'_a and R'_b with probabilities θ*(1-θ), θ*(1-θ), and (1-θ)^2, respectively. All partial similarities are then added for gaining similarity between two masked user profiles. Used similarity function also plays a role on the difference between gained similarity and original similarity between two users. Thus, when Pearson's correlation is used and θ is set to 0.7, the gained similarity is 0.16 times of its original value.
- **Disagreement with possible target items:** The procedure as in calculating **aou** for a profile is used for calculating **dti** value of a user profile.

The calculated values of the metrics may differ from their expected values shown in Table 6.1 depending on θ and f values. Thus, the intervals which derived attributes' values may occupy should be determined so tenderly.

6.2. Calculation of Metrics with Multi-group RRT Scheme

In multi-group RRT scheme, items are partitioned into M groups for a user profile. For each group, the user selects a random number and sends true or reversed ratings depending on the selected random number and pre-determined value of θ. For multi-group scheme, the mentioned metrics of a profile are calculated similarly as in Section 6.1. This time, 2^M combinations of representations of the profile and probabilities are utilized instead of two representations of the user profile in order to calculate values of metrics.

The calculated values of the metrics will differ from their expected values much more than the values calculated with one-group RRT scheme. Thus, while specifying the intervals in which derived attribute values may occupy, for each attack model, considering M and θ values sensitively is extremely important.

6.3. Experimental Evaluation

Several experiments are performed in order to show how the proposed detection scheme performs on binary databases masked with one- and multi-group full privacy provided RRT schemes in terms of effectiveness. Used data set and evaluation criteria are described in Section 6.3.1 and experimental results are shown in 6.3.2.

6.3.1. Data set and evaluation criteria

A set of different trials are performed using MLP data set aiming to measure effectiveness of the proposed detection scheme. MLP is a publicly available data set including numeric and discrete ratings. In order to perform experiments, the

ratings are translated into binary form. If the ratings is bigger than or equal to 3, then it is transformed into 1. Otherwise, it is transformed into 0.

Precision, recall, and F1 measure are chosen as evaluation metrics for measuring how accurately the proposed detection scheme performs. Assuming that the number of attack profiles classified correctly as fake is **A**, the number of genuine profiles classified as fake is **B**, and the number of attack profiles are not detected is **C**, then precision, recall, and F1 measure are calculated as follows:

$$Precision = \frac{A}{(A+B)} \tag{6.6}$$

$$Recall = \frac{A}{(A+C)} \tag{6.7}$$

$$F1\ Measure = \frac{2PrecisionRecall}{Precion+Recall} \tag{6.8}$$

6.3.2. Experimental results

In order to show effectiveness of the proposed detection scheme on binary masked data, the data are disguised with both one- and multi-group full privacy provided RRT schemes. The masking parameters are chosen as their optimum values, which are experimentally shown in the work (Kaleli and Polat, 2007). Thus, θ is set to 0.7 for both RRT schemes and M is set to 3 for multi-group RRT scheme. Also the f parameter is chosen as 50 for both schemes. Then, the attack profiles are disguised as genuine users in order to obstruct detactability of them while manipulating produced predictions as successfully as their undisguised forms. Later, the generated attack profiles are injected to the database with varying filler size and attack size values for measuring effectiveness of the proposed detection scheme. All experiments are performed for 50 nuke and 50 push items. Lastly, in order to show effectiveness of the proposed scheme on attack profiles generated as in Section 4.1 and 4.2, some experiments are performed.

6.3.2.1. One-group RRT scheme with full privacy

Some experiments are performed in order to specify how varying filler size and attack size values affect the performance of the proposed scheme on disguised attack profiles. Attack size is set to 15 while filler size values are varied from 3 to 15. Also, filler size is set to 15, while varying attack size from 3 to 15. During the experiments the attack profiles are generated as in Section 4.3 using one-group RRT scheme as genuine users.

Fig. 6.1 shows that the proposed detection scheme detects attack profiles more successfully with increasing attack size values with recpect to precision. The best results are gained with average and love/hate attack models due to their generation strategies. Increasing filler size values provide all of the attack models to show more specific properties like high similarity. Thus, they become more detactable with incremental filler size values.

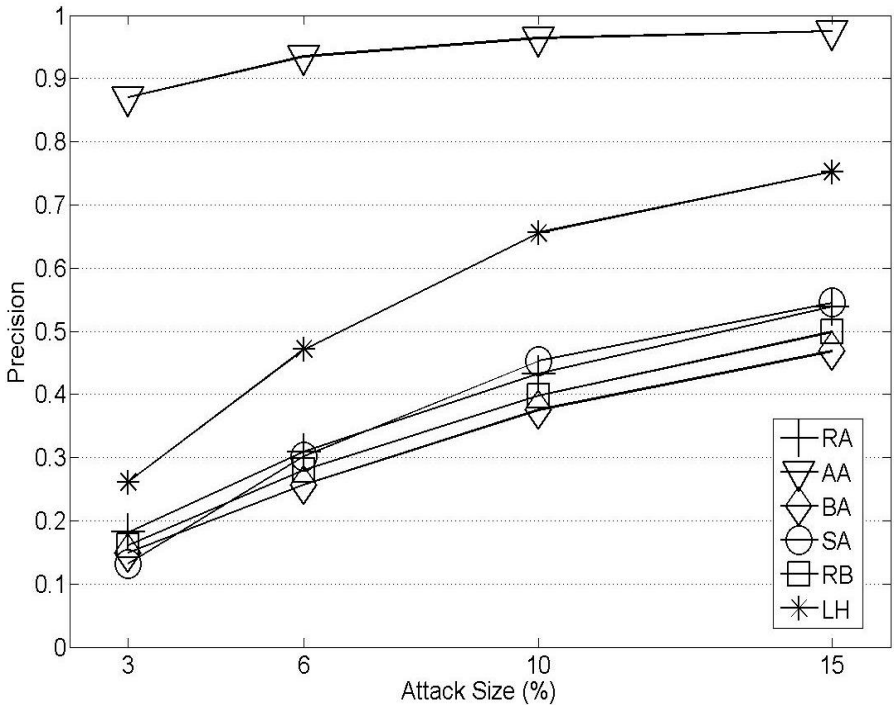

Figure 6.1. Effects of varying attack sizes on performance (precision)

Fig. 6.2 shows that varying attack size values improve performance of the scheme in terms of recall. Due to the generation strategies of random, bandwagon, and average attack models, most of their profiles are detected as independently from attack size values. The impact of varying attack size values is more significant on the performance of the scheme for remaining attack models. Since with increasing attack size values make the profiles of segment or love/hate profiles look like each other so much, the recall values are less for them when attack size is 3.

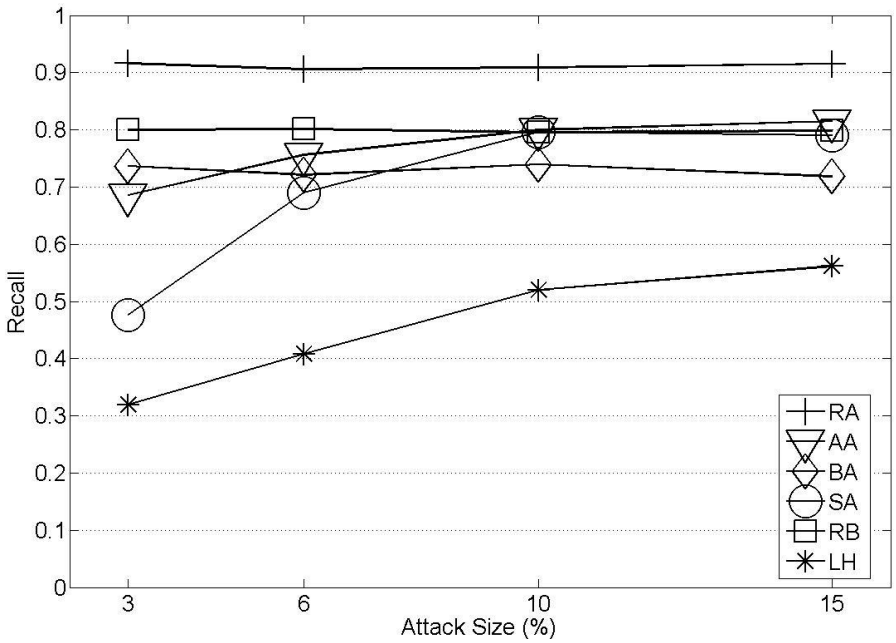

Figure 6.2. Effects of varying attack sizes on performance (recall)

Fig. 6.3 represents F1 measure values with varying attack size values to present an overall picture. As it is seen from Fig. 6.3, when random, bandwagon, and reverse bandwagon attack profiles are disguised as in Section 4.3, they do not differ so much from their undisguised forms as in Section 4.1 in terms of derived attributes. Also, f makes the filler size values of mentioned attack models to increase. Thus, they are more detectable with increasing attack size values. However, since derived attribute values are not so specific, they have smaller precision values even though most of their profiles are detected. For segment and love/hate attack models, f is more significant on the derived attribute values. All

the calculations for derived attributes change differently with *f* value. Depending on the amounts of differences between attributes' gained values and their expected values, the proposed scheme can detect their profiles successfully with higher attack size values. Due to the generation strategy of average attack model, the proposed scheme is more effective on average attack profiles with M = 1 and optimum values of θ and *f* parameters.

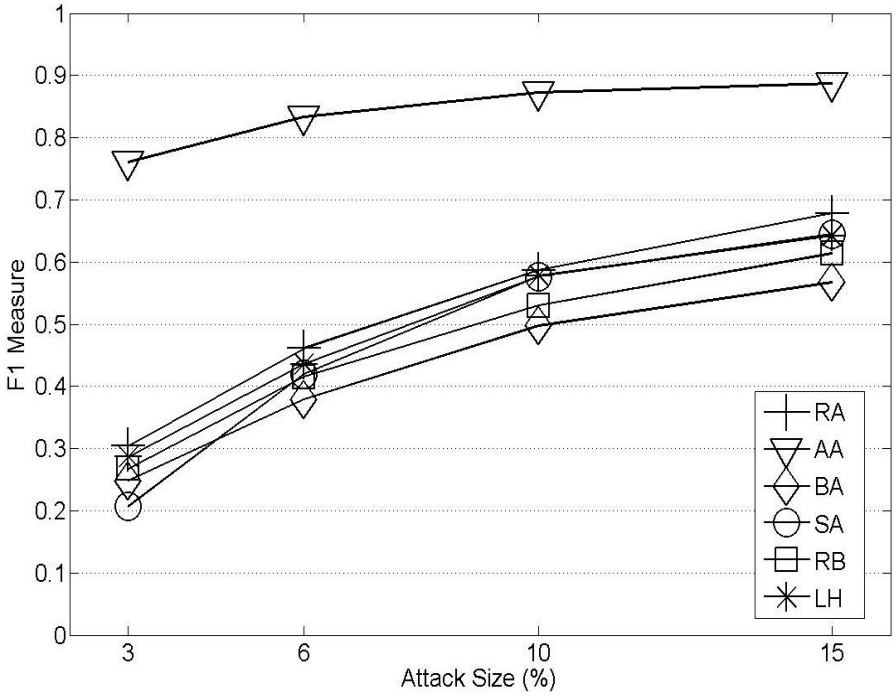

Figure 6.3. Effects of varying attack sizes on performance (F1 measure)

Fig. 6.4 shows how varying filler size affects the performance of the proposed scheme in terms of precision. As it is seen from Fig. 6.4, the best results are gained with average attack model with respect to precision depending on its generation strategy. Since attack size is large enough during this experiment, the precision values for love/hate and segment attack profiles are nearly the same with smaller varying filler size values. If the filler size is so larger, the effect of *f* parameter increases; thus, precision values will decrease. Since *f* parameter make filler size values of random, bandwagon, and reverse bandwagon attack models to increase, the proposed algorithm detects them more successfully with increasing filler size values in terms of precision.

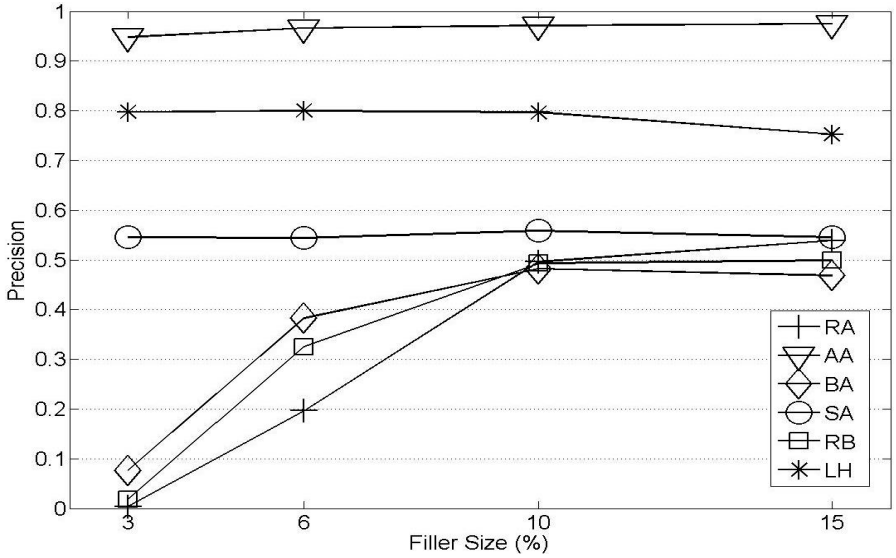

Figure 6.4. Effects of varying filler sizes on performance (precision)

Fig. 6.5 represents impacts of varying filler sizes on performance of the proposed scheme in terms of recall. Increasing filler sizes enhance performance of the scheme for random, bandwagon, and reverse bandwagon attack models. Since filler size is an influential factor of average attack model, the proposed scheme detects average attack profiles more successfully with incremental filler size values. Since *f* parameter is so effective on segment and love/hate attack models, the performance of the proposed scheme decreases when filler size is so higher.

For overall picture, F1 measure values are calculated and represented in the Fig. 6.6 for showing effects of varying filler size values on performance of the scheme. Since disguised forms of random, bandwagon, and reverse bandwagon attack models may not differ so much from their undisguised forms in terms of structure, the algorithm detects their profiles more successfully with increasing filler size values. Even if the average attack profiles are disguised, they become more detactable by the proposed scheme with increasing filler size values. Since segment and love/hate attack models' values of derived attributes differ from their expected values depending on masking parameter *f*, the algorithm becomes less effective on their profiles when filler size is so large.

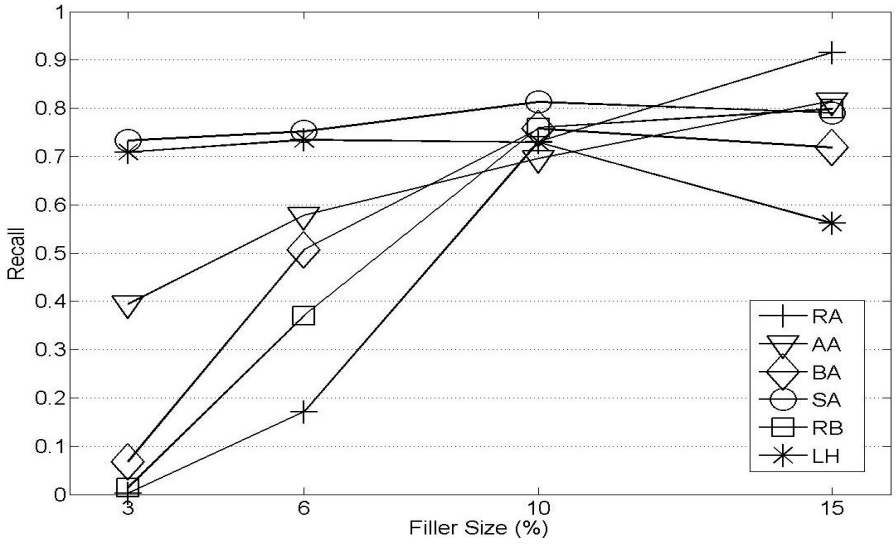

Figure 6.5. Effects of varying filler sizes on performance (recall)

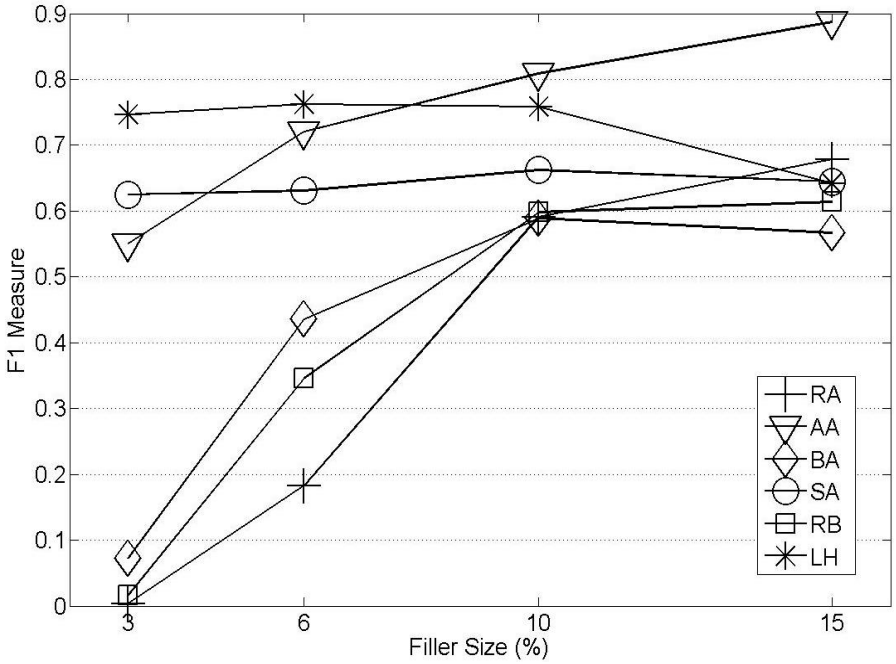

Figure 6.6. Effects of varying filler sizes on performance (F1 measure)

In order to show how the proposed scheme performs if the attack profiles are generated as in Section 4.1 and Section 4.2, different trials are performed with setting filler size and attack size values at 15. According to the results shown in Table 6.2, if the attack profiles are generated without disguising as in Section 4.1,

their derived attributes' values are so similar to the values supposed to be. However, since the algorithm is designed for detecting disguised attack profiles, it considers masking parameters. Thus, the algorithm detects all of the attack profiles more successfully in terms of recall while having less precision values compared to recall values. As it is seen, the algoritm detects all of the attack profiles successfully.

If the attack profiles are generated as in Section 4.2, the values of their derived attributes are affected with masking parameters except *f*. With compared to their full privacy provided versions, segment, love/hate, and average attack models are detected more successfully. For random, bandwagon, and reversebandwagon attack models have similar F1 measure values for three generation schemes.

Table 6.2. Effects of attack generation methods on the detection algorithm's performance

Generation Scheme	Attack Model	Precision	Recall	F1 Measure
Attack profiles are generated by the scheme discussed in Section 4.1	RA	0.534	0.874	0.663
	AA	0.975	0.829	0.896
	BA	0.493	0.795	0.608
	SA	0.614	1.000	0.761
	RB	0.509	0.802	0.617
	L/H	0.848	1.000	0.918
Attack profiles are generated by the scheme discussed in Section 4.2	RA	0.534	0.875	0.663
	AA	0.975	0.825	0.893
	BA	0.487	0.774	0.598
	SA	0.552	0.799	0.652
	RB	0.503	0.800	0.617
	L/H	0.801	0.712	0.754
Attack profiles are generated by the scheme discussed in Section 4.3	RA	0.537	0.913	0.676
	AA	0.975	0.821	0.89
	BA	0.467	0.719	0.566
	SA	0.556	0.821	0.663
	RB	0.450	0.804	0.616
	L/H	0.750	0.558	0.639

6.3.2.2. Multi-group RRT scheme with full privacy

To show how the proposed detection scheme works with varying filler and attack sizes on binary masked data, some trials are performed, where data is disguised with multi-group scheme with full privacy. The attack profiles are disguised as in Section 4.3. Attack size is set to 15 while filler size values are varied from 3 to 15. Also, filler size is set to 15, while varying attack size from 3 to 15. During the experiments θ, M, and f are set to 0.7, 3, and 50, respectively.

Fig. 6.7 represents impacts of varying attack sizes on performance of the detection scheme in terms of recall. As it is seen from Fig. 6.7, the best results are gained for random, bandwagon, and reverse bandwagon attack models since their derived attribute values are similar to their values supposed to be. Multi-group RRT scheme significantly affects derived attribute values of average, love/hate, and segment attack profiles. Thus, recall values of such models are lower. The algorithm works worse with smaller attack sizes for all attack models. Increasing attack size values mostly improve performance o the scheme for all attack models. Since attack profiles are disguised with multi group RRT scheme, if attack size is so large, the derived attribute values of attack profiles may differ from their expected values especially for love/hate, average, and segment attack models.

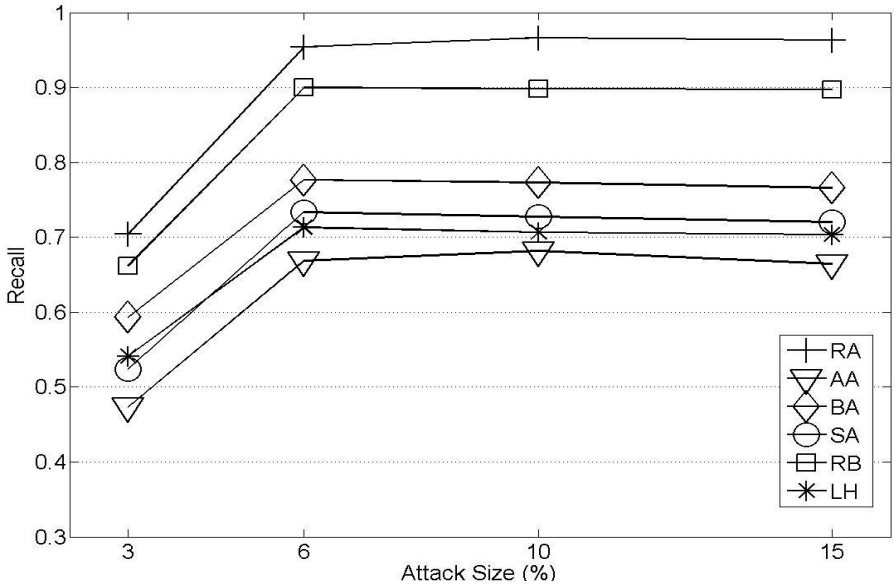

Figure 6.7. Effects of varying attack sizes on performance (recall)

Effects of varying attack size values on performance of the detection scheme in terms of precision is shown in Fig. 6.8. Due to the masking scheme and the generation strategies of random, bandwagon, and reverse bandwagon attack models, their precision values are higher than the other attack models. Because they have similar derived attribute values with the ones supposed to be. But, the genuine users' derived attribute values may change much more depending on *M*. As it is seen in Fig. 6.8, increasing attack size values enhance performance of the scheme in terms of precision for all attack models.

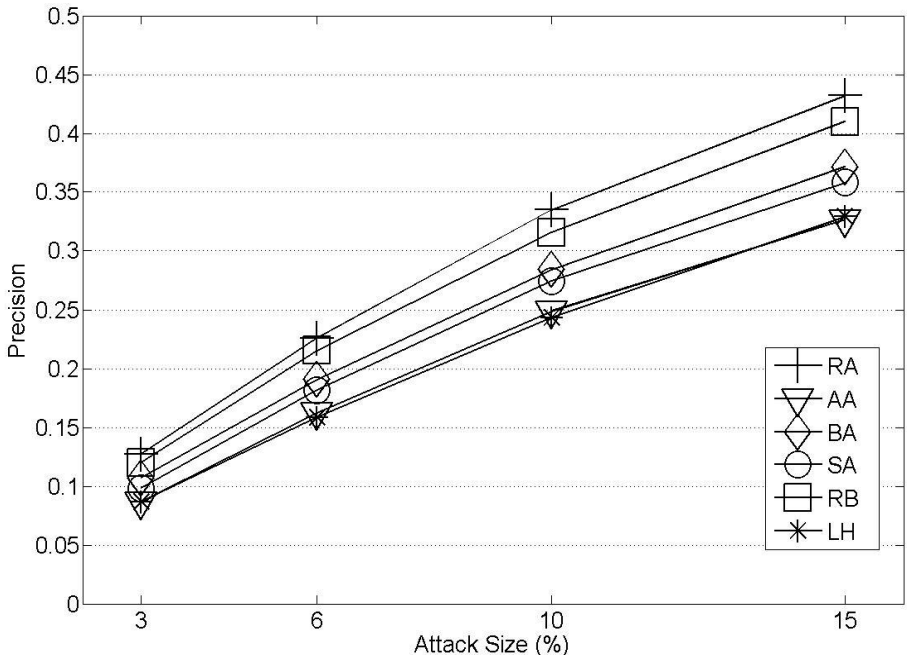

Figure 6.8. Effects of varying attack sizes on performance (precision)

For overall picture, F1 values are shown in Fig. 6.9 to specify how the detection algorithm performs with varying attack size values. As it is seen in Fig. 6.9, increasing attack size vaues enhance performance of the algorithm for all attack models. Since data and attack profiles are disguised with multi-group RRT scheme, the least affected attack models are random, bandwagon, and reverse bandwagon attack models depending on their generation strategies. Also, filling unrated items makes filler size values of the mentioned attack profiles to increment, they display their derived attribute values more specifically. Due to the

number of groups and *f* value, derived attribute values of love/hate, segment, and average attack profiles display much more differences from their expected values. Hence, in order to provide a balance between precision and recall values, the values which provide to classify profiles as fake using derived attribute values should be chosen carefully by considering *M*, θ, and *f* values.

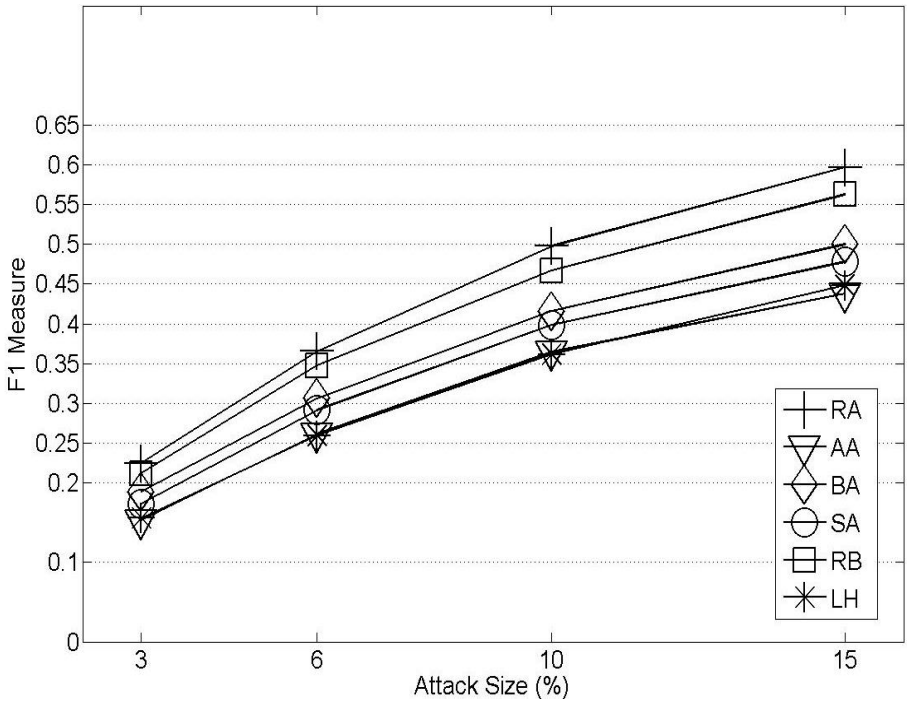

Figure 6.9. Effects of varying attack sizes on performance (F1 measure)

In Fig. 6.10, impressions of varying filler size values on performance of the proposed detection algorithm in terms of recall are represented. As it is shown in the figure, increasing filler size values enhance performance of the proposed scheme with respect to recall. The proposed detection scheme detects random, bandwagon, and reverse bandwagon attack profiles more successfully depending on their generation strategy and masking parameters. As it is seen in the figure, recall values for bandwagon and reverse bandwagon attack models decrease after filler size is 10. The reason of this situation is the chosen values which provide to classify profiles of mentioned attack models correctly utilizing derived attribute values. Since undisguised forms of average, segment, and love/hate attack profiles have so significant structures, their disguised forms differ so much from their

original structures. Thus, chosing the intervals in which derived attributes values may occupy play an important role on detection ability of the proposed detection scheme. While specifying those values, the number of filled items, which increase with incremental filler size values, is considered more carefully. Thus, specific increases do not happen for segment and average attack models in terms of recall.

Fig. 6.11 shows how varying filler size values affect performance of the proposed detection scheme in terms of precision. As it is seen in the figure, increasing filler size values mostly enhance performance of the scheme for all of the attack models. The best results are gained with random and bandwagon attack models with higher filler size values depending on their generation strategies.

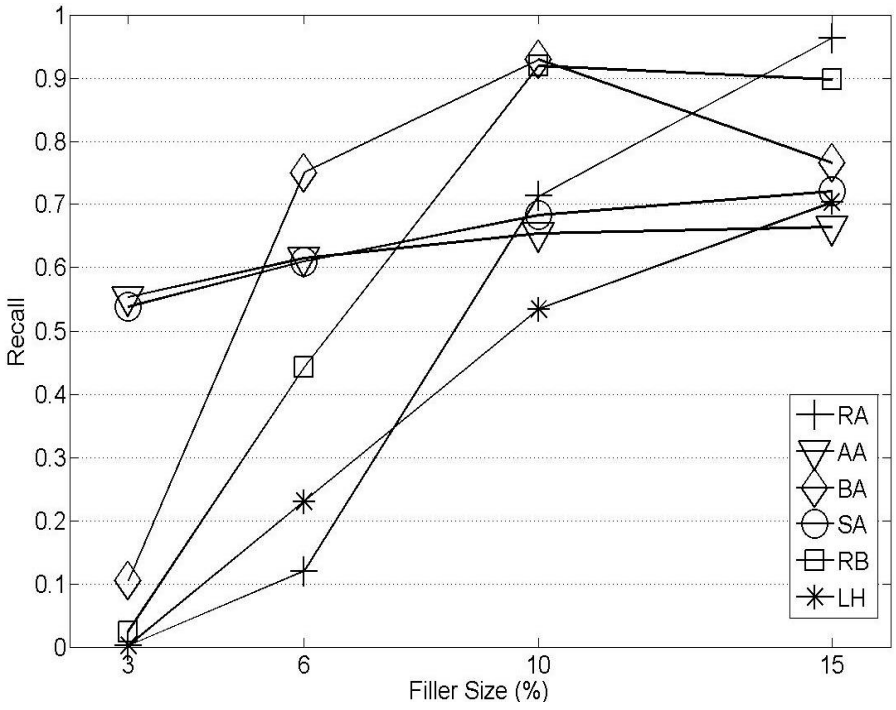

Figure 6.10. Effects of varying filler sizes on performance (recall)

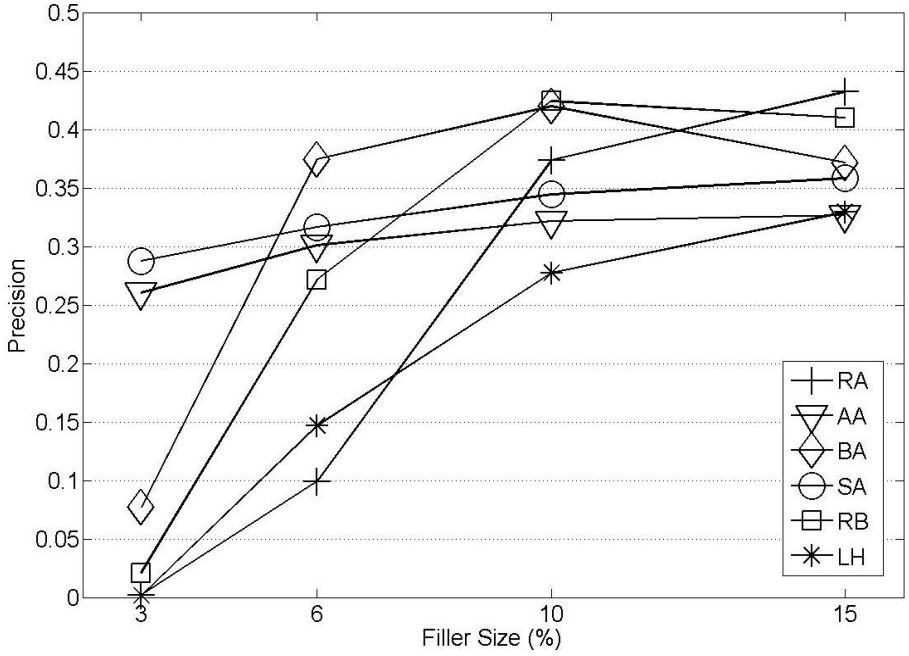

Figure 6.11. Effects of varying filler sizes on performance (precision)

In order to give an overall picture, F1 measure values are calculated and represented in Fig. 6.12. As it is obvious in the figure, incremental filler size values mostly improve performance of the schemes for all attack models. With higher filler size values, the best results are gained with random, bandwagon, and reverse bandwagon attack models because of their generation strategies. Even though they are disguised as in Section 4.3, their derived attributes' values are similar to their expected values. Also, for mentioned attack models, increasing filler size values mean incrementing the number of filled items for a profile. The values of the profiles' derived attribute values are supposed to be more similar to their expected values. However, this mentioned situation may be hindered depending on M value. Thus, the profile may not be detected by the proposed scheme. The undisguised forms of average, segment, and love/hate attack models have specific generation structures. However, the disguising parameters especially M and θ values obstruct so much to detect their profiles when they are disguised. Because their specific generation structures change. Since the ranges of derived attributes' values is determined considering f, θ, and M values more appropriately, their profiles are detected by the scheme more successfully than the others when filler size is 3.

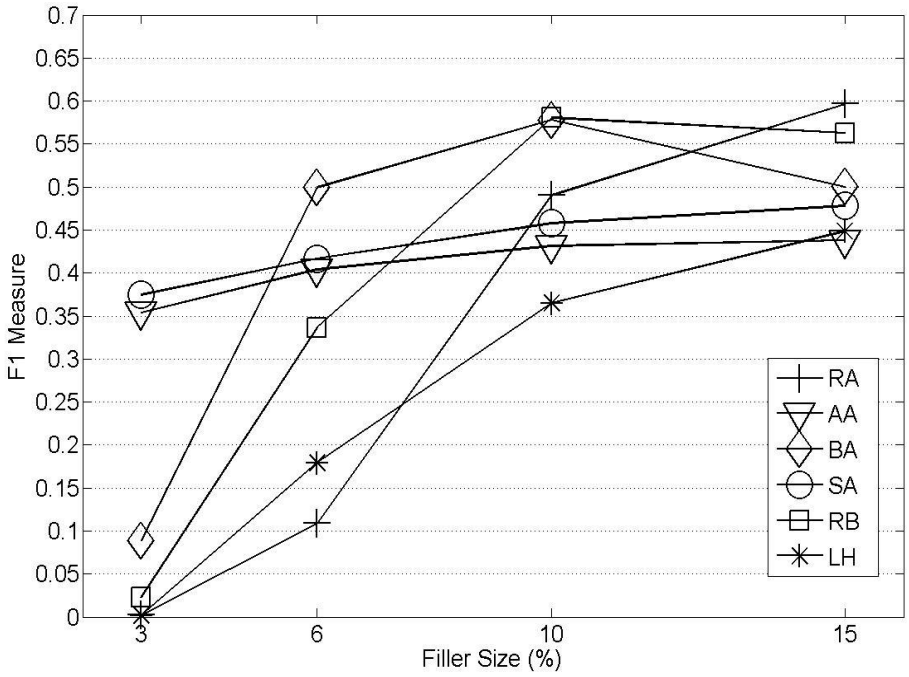

Figure 6.12. Effects of varying filler sizes on the performance (F1 measure)

In order to show how the proposed scheme performs when the attack profiles are generated as in Section 4.1 and Section 4.2, some experiments are performed and represented in Table 6.2. During the experiments filler size, attack size, θ, M, and f values are set to 15, 15, 0.7, 3 and 50, respectively. As it is obvious in Table 6.3, recall values are high for all of the attack models when they are generated as in Section 4.1. Because their derived attribute values' are so similar to the attributes' expected values. But, precision values are lower compared to recall values since the detection algorithm is designed by considering masking parameters. As it is seen, the algoritm detects all of the attack profiles successfully. If attack profiles are generated as in Section 4.2, recall values of segment, love/hate, and average attack models become lower depending on masking parameters. Because the mentioned attack profiles' derived attribute values differ from their expected values much more. Precision values of six attack models are similar to the precision values of the attack models generated with the algorithm discussed in Section 4.1 or 4.3. The intervals which the derived attribute values of the attack profiles may occupy are determined by considering θ, M, and f values for the proposed detection scheme. Thus, the specified interval

values are an important factor on precision and recall values for all of the attack models generated by any of the discussed generation schemes. As it is seen from the table, the proposed detection scheme detects successfully all of the attack models.

Table 6.3. Effects of proposed shilling attack generation schemes on performance of the proposed detection algorithm

Generation Sceheme	Attack Model	Precision	Recall	F1 Measure
Attack profiles are generated by the scheme discussed in Section 4.1	RA	0.426	0.913	0.581
	AA	0.418	1.000	0.590
	BA	0.419	0.934	0.578
	SA	0.362	0.715	0.480
	RB	0.427	0.948	0.589
	L/H	0.399	1.000	0.571
Attack profiles are generated by the scheme discussed in Section 4.2	RA	0.427	0.919	0.583
	AA	0.319	0.645	0.426
	BA	0.418	0.929	0.577
	SA	0.397	0.844	0.540
	RB	0.43	0.954	0.593
	L/H	0.205	0.367	0.263
Attack profiles are generated by the scheme discussed in Section 4.3	RA	0.432	0.954	0.597
	AA	0.327	0.669	0.438
	BA	0.371	0.776	0.500
	SA	0.358	0.734	0.478
	RB	0.410	0.900	0.563
	L/H	0.329	0.713	0.449

7. CONCLUSIONS AND FUTURE WORKS

Privacy-preserving collaborative filtering schemes produce accurate recommendations while protecting privacy. Accuracy of the produced predictions depends on quality of the data. In this dissertation, six well-known shilling attack models are modified in order to manipulate predictions on binary data masked with randomized response techniques. Also, a detection scheme is proposed in order to detect attack profiles on binary disguised data.

Three generation schemes are proposed aiming to manipulate predictions on binary data disguised with randomized response techniques. The best results are gained when attack profiles are generated without disguising. Also, either generating attack profiles by disguising or generating them by providing full privacy is also effective on produced predictions. For one-group randomized response technique, performances for three generation schemes are very similar. When data is disguised with multi-group randomized response technique, for random, segment, and love/hate attack models, performances of generation schemes are similar to each other. However, for attack models requiring calculations of items' modes, the attack profiles manipulate predictions worse when they are generated by disguising or providing full privacy. However, the mentioned attack models can perform as successfully as their undisguised versions by chosing target items carefully. Items' properties such as number of given votes and masking parameters are extremely important factors on the difference between ratio shift values of unmasked and masked forms of attack profiles. When required conditions are provided, masked forms of attack profiles manipulate predictions as successfully as their unmasked versions with obstructed detectabilities.

Three generation schemes are applied to naïve Bayesian classifier-based collaborative filtering algorithm masked with randomized response techniques in order to measure robustness of the scheme under shilling attacks. According to the results, naïve Bayesian classifier-based privacy-preserving collaborative filtering scheme is vulnerable to shilling attacks. For three generation schemes, the best results are gained with average attack model. The proposed generation schemes

are also effective on produced predictions for bandwagon, segment, reverse bandwagon, and love/hate attacks. Morever, the performances of the generation schemes enhance usually with θ values, which are close to 1 or 0, increasing filler size and attack size values, and decreasing M and f values. In order to provide effectiveness on produced predictions and an obstructed detectability, disguising attack profiles with full privacy gives influential results. However, if only important skill is manipulability for attackers without considering detactability, generating attack profiles without disguising should be a reasonable choice.

Classification is a tool which can be used for detecting attack profiles. Since attack profiles are generated by a certain strategy, classification is used for detecting bogus profiles on binary masked data by considering masking parameters. For this purpose, four attributes are adapted and derived as dissimilarity in user's profile, aggreement with other users, similarity with top-N neighbors, and disaggrement with possible target items. The mentioned attributes play an important role on classifying masked bogus profiles correctly. The gained values of the attributes from binary masked data will differ from their expected values depending on θ, M, and f values. Thus, the intervals in which derived attribute values may occupy, should specify so sensitively for balancing precision and recall values. According to the results, when M is 1, the algorithm detects attack profiles more successfully comparing with M = 3 value. When binary data is disguised with one-group scheme, the best results are gained with average, segment, and love/hate attack models. For multi-group scheme, the best results are gained with random, bandwagon, and reverse bandwagon attack models since average, segment, and love/hate attack models are affected much more by masking parameters because of their generation strategies. Also, the performance of the proposed detection scheme mostly enhances with increasing filler and attack sizes for disguising schemes. Morever, if the intervals in which derived attribute values may occupy, are specified differently, distinct results may be gained.

Finally, enhancing performance of the proposed detection scheme and applying proposed shilling attack generation schemes to other privacy-preserving collaborative filtering algorithms can be performed as future works.

REFERENCES

Anderson, J. R., Michalski, R. S., Carbonell, J. G., and Mitchell, T. M. (1986), "*Machine learning: An artificial intelligence approach*," Morgan Kaufmann.

Berkovsky, S., Kuflik, T., and Ricci, F. (2012), "The impact of data obfuscation on the accuracy of collaborative filtering," *Expert Systems with Applications*, **39** (5), 5033-5042.

Bhaumik, R., Williams, C. A., Mobasher, B., and Burke, R. (2006), "Securing collaborative filtering against malicious attacks through anomaly detection," *Proceedings of the 4th Workshop on Intelligent Techniques for Web Personalization*, Boston, MA, USA.

Bilge, A., Gunes, I., and Polat, H. (2014a), "Robustness analysis of privacy-preserving model-based recommendation schemes," *Expert Systems with Applications*, **41** (8), 3671-3681.

Bilge, A., Ozdemir, Z., and Polat, H. (2014b), "A novel shilling attack detection method," *Procedia Computer Science*, **31**, 165-174.

Burke, R., Mobasher, B., Bhaumik, R., and Williams, C. A. (2005), "Collaborative recommendation vulnerability to focused bias injection attacks," *Proceedings of the Workshop on Privacy and Security Aspects of Data Mining*, Houston, TX, USA, 35-43.

Burke, R., Mobasher, B., Williams, C. A., and Bhaumik, R. (2006), "Detecting profile injection attacks in collaborative recommender systems," *Proceedings of the 8th International Conference on E-Commerce Technology*, San Francisco, CA, USA, 23-30.

Canny, J. (2002), "Collaborative filtering with privacy via factor analysis," *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Tampere, Finland, 238-245.

Cao, J., Wu, Z., Mao, B., and Zhang, Y. (2013), "Shilling attack detection utilizing semi-supervised learning method for collaborative recommender system," *World Wide Web*, **16** (5-6), 729-748.

Cheng, Z. and Hurley, N. J. (2010), "Robustness analysis of model-based collaborative filtering systems," *Lecture Notes in Computer Science*, **6206**, 3-15.

Chirita, P. A., Nejdl, W., and Zamfir, C. (2005), "Preventing shilling attacks in online recommender systems," *Proceedings of the 7th Annual ACM International Workshop on Web Information and Data Management*, Bremen, Germany, 67-74.

Cranor, L. F. (2004), "'I didn't buy it for myself': Privacy and E-commerce personalization," *Designing Personalized User Experiences in eCommerce*, 57-73.

Du, W. and Zhan, Z. (2003), "Using randomized response techniques for privacy-preserving data mining," *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Washington, DC, USA, 505-510.

Gao, M., Yuan, Q., Ling, B., and Xiong, Q. (2014), "Detection of abnormal item based on time intervals for recommender systems," *Scientific World Journal*, **2014**, Article ID 845897, 8 pages.

Gunes, I., Bilge, A., Kaleli, C., and Polat, H. (2013a), "Shilling attacks against privacy-preserving collaborative filtering," *Journal of Advanced Management Science*, **1** (1), 54-60.

Gunes, I., Bilge, A., and Polat, H. (2013b), "Shilling attacks against memory-based privacy-preserving recommendation algorithms," *KSII Transactions on Internet and Information Systems*, **7** (5), 1272-1290.

Gunes, I., Kaleli, C., Bilge, A., and Polat, H. (2014), "Shilling attacks against recommender systems: A comprehensive survey," *Artificial Intelligence Review*, **42** (4), 767-799.

Gunes, I. and Polat, H. (2015), "Hierarchical clustering-based shilling attack detection in private environments," *Proceedings of the 3rd International Symposium on Digital Forensics and Security*, Ankara, Turkey, 1-7.

Han, J., Kamber, M., and Pei, J. (2011), "*Data mining: Concepts and techniques*," Morgan Kaufmann.

Hurley, N. J., Cheng, Z., and Zhang, M. (2009), "Statistical attack detection," *Proceedings of the 3rd ACM Conference on Recommender Systems*, New York, NY, USA, 149-156.

Kaleli, C. and Polat, H. (2007), "Providing private recommendations using naïve Bayesian classifier," *Advances in Intelligent Web Mastering*, **43**, 168-173.

Kaleli, C. and Polat, H. (2013), "Robustness analysis of naïve Bayesian classifier-based collaborative filtering," *Lecture Notes in Business Information*, **152**, 202-209.

Long, Q. and Hu, Q. (2010), "Robust evaluation of binary collaborative recommendation under profile injection attack," *Proceedings of the IEEE International Conference on Progress in Informatics and Computing*, Shanghai, China, 1246-1250.

Mehta, B., Hofmann, T., and Fankhauser, P. (2007), "Lies and propaganda: Detecting spam users in collaborative filtering," *Proceedings of the 2007 International Conference on Intelligent User Interfaces*, Honolulu, HI, USA, 14-21.

Mehta, B. and Nejdl, W. (2009), "Unsupervised strategies for shilling detection and robust collaborative filtering," *User Modeling and User-Adapted Interaction*, **19** (1-2), 65-97.

Mild, A. and Reutterer, T. (2001), "Collaborative filtering methods for binary market basket data analysis," *Lecture Notes in Computer Science*, **2252**, 302-313.

Miyahara, K. and Pazzani, M. J. (2000), "Collaborative filtering with the simple Bayesian classifier," *Proceedings of the 6th Pacific Rim International Conference on Artificial Intelligence*, Melbourne, Australia, 679-689.

Mobasher, B., Burke, R., Bhaumik, R., and Williams, C. A. (2007), "Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness," *ACM Transactions on Internet Technology*, **7** (4), 23-60.

Mobasher, B., Burke, R., and Sandvig, J. J. (2006), "Model-based collaborative filtering as a defense against profile injection attacks," *Proceedings of the*

21st National Conference on Artificial Intelligence, Boston, MA, USA, 1388-1393.

O'Mahony, M. P., Hurley, N. J., and Silvestre, G. C. M. (2005), "Recommender systems: Attack types and strategies," *Proceedings of the 20th National Conference on Artificial Intelligence*, Pittsburgh, PA, USA, 334-339.

O'Mahony, M. P., Hurley, N. J., and Silvestre, G. C. M. (2002), "Towards robust collaborative filtering," *Lecture Notes in Computer Science*, **2464**, 87-94.

O'Mahony, M. P., Hurley, N. J., and Silvestre, G. C. M. (2004), "An evaluation of neighbourhood formation on the performance of collaborative filtering," *Artificial Intelligence Review*, **21** (3-4), 215-228.

O'Mahony, M. P., Hurley, N. J., and Silvestre, G. C. M. (2003), "Collaborative filtering–safe and sound?," *Lecture Notes in Computer Science*, **2871**, 506-510.

Polat, H. and Du, W. (2005a), "Privacy-preserving collaborative filtering," *International Journal of Electronic Commerce*, **9** (4), 9-36.

Polat, H. and Du, W. (2005b), "Privacy-preserving top-*N* recommendation on horizontally partitioned data," *Preceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, Paris, France, 19-22.

Polat, H. and Du, W. (2006), "Achieving private recommendations using randomized response techniques," *Lecture Notes in Computer Science*, **3918**, 637-646.

Rashid, A. M., Lam, S. K., Karypis, G., and Riedl, J. T. (2006), "ClustKNN: A highly scalable hybrid model- & memory-based CF algorithm," *Proceeding of 12th ACM International Conference on WebKDD*, Philadelphia, PA, USA.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. T. (2001), "Item-based collaborative filtering recommendation algorithms," *Proceedings of the 10th International Conference on World Wide Web*, Hong Kong, 285-295.

Senyurek, E. and Polat, H. (2013), "Effects of binary similarity measures on top-*N* recommendations," *Anadolu University Journal of Science and Technology*, **14** (1), 55-65.

Su, X. F., Zeng, H. J., and Chen, Z. (2005), "Finding group shilling in recommendation system," *Proceedings of the 14th International Conference on World Wide Web*, Chiba, Japan, 960-961.

Vozalis, M. G. and Margaritis, K. G. (2007), "Using SVD and demographic data for the enhancement of generalized collaborative filtering," *Information Sciences*, **177** (15), 3017-3037.

Warner, S. L. (1965), "Randomized response: A survey technique for eliminating evasive answer bias," *Journal of the American Statistical Association*, **60** (309), 63-69.

Williams, C. A., Mobasher, B., Burke, R., Sandvig, J.J., and Bhaumik, R. (2006), "Detection of obfuscated attacks in collaborative recommender systems," *Proceedings of the Workshop on Recommender Systems, in Conjunction with 17th European Conference on Artificial Intelligence*, Riva del Garda, Trentino, Italy, 19-23.

Williams, C. A., Mobasher, B., Burke, R., and Bhaumik, R. (2007), "Detecting profile injection attacks in collaborative filtering: A classification-based approach," *Lecture Notes in Computer Science*, **4811**, 167-186.

Wu, Z., Zhuang, Y., Wang, Y. Q., and Cao, J. (2012a), "Shilling attack detection based on feature selection for recommendation systems," *Dianzi Xuebao(Acta Electronica Sinica)*, **40** (8), 1687-1693.

Wu, Z., Wu, J., Cao, J., and Tao, D. (2012b), "HySAD: A semi-supervised hybrid shilling attack detector for trustworthy product recommendation," *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Beijing, China, 985-993.

Xia, H., Fang, B., Gao, M., Ma, H., Tang, Y., and Wen, J. (2015), "A novel item anomaly detection approach against shilling attacks in collaborative recommendation systems using the dynamic time interval segmentation technique," *Information Sciences*, **306**, 150-165.

Yan, X. and Van Roy, B. (2009), "Manipulation robustness of collaborative filtering systems," *Management Science*, **56** (11), 1911-1929.

Zhang, S., Chakrabarti, A., Ford, J., and Makedon, F. (2006), "Attack detection in time series for recommender systems," *Proceedings of the 12th ACM*

SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, 809-814.

Zhang, Z. and Kulkarni, S. R. (2014), "Detection of shilling attacks in recommender systems via spectral clustering," *Proceedings of 17th International Conference on Information Fusion*, Salamanca, Spain, 1-8.