Robustness Analysis of a Novel Model-Based Recommendation Algorithms in Privacy Environment

İhsan Güneş[1*]
[1] Department of Computer Technologies, Eskisehir Technical University
Eskisehir, Turkey
[e-mail: igunes@eskisehir.edu.tr]
[*]Corresponding author: İhsan Güneş

Received November 11, 2023; revised March 22, 2024; revised April 27, 2024; accepted May 16, 2024; published May 31, 2024

Abstract

The concept of privacy-preserving collaborative filtering (PPCF) has been gaining significant attention. Due to the fact that model-based recommendation methods with privacy are more efficient online, privacy-preserving memory-based scheme should be avoided in favor of model-based recommendation methods with privacy. Several studies in the current literature have examined ant colony clustering algorithms that are based on non-privacy collaborative filtering schemes. Nevertheless, the literature does not contain any studies that consider privacy in the context of ant colony clustering-based CF schema. This study employed the ant colony clustering model-based PPCF scheme. Attacks like shilling or profile injection could potentially be successful against privacy-preserving model-based collaborative filtering techniques. Afterwards, the scheme's robustness was assessed by conducting a shilling attack using six different attack models. We utilize masked data-based profile injection attacks against a privacy-preserving ant colony clustering-based prediction algorithm. Subsequently, we conduct extensive experiments utilizing authentic data to assess its robustness against profile injection attacks. In addition, we evaluate the resilience of the ant colony clustering model-based PPCF against shilling attacks by comparing it to established PPCF memory and model-based prediction techniques. The empirical findings indicate that push attack models exerted a substantial influence on the predictions, whereas nuke attack models demonstrated limited efficacy.

Keywords: Ant colony clustering, Collaborative filtering, Privacy, Recommendation system, Shilling attacks

1. Introduction

The rapid advancement of Internet technologies has led to an increased focus on e-commerce. In contemporary times, there is a notable inclination among many individuals towards engaging in online purchasing. E-commerce enterprises facilitate the online purchase of a diverse range of products, including but not limited to books, music CDs, and food items, among others, by customers. E-commerce platforms offer a vast array of products, presenting customers with a multitude of choices from which they must select. The proliferation of available options has correspondingly increased the volume of information that consumers must evaluate in order to make informed decisions regarding the products that best meet their needs. Hence, e-commerce platforms employ collaborative filtering (CF) technologies to aid customers in making informed product choices. According to authors, these websites provide support to online merchants in order to optimize their revenues and/or profitability by providing guidance to their customers [1, 2].

Collaborative filtering (CF) systems primarily involve the tasks of generating consumer suggestions and delivering individualized information about items. The purpose of these solutions is to facilitate efficient object retrieval for users and protect the system from unnecessary information. Data mining techniques are employed to effectively handle the similarity among vast quantities of data, ranging from thousands to even millions of data points. These systems primarily encompass three fundamental processes: the collection and representation of data, the evaluation of similarities, and the execution of computations to generate recommendations. The objective of CF is to establish a correlation between individuals and the available dataset, with the aim of conducting a more comprehensive assessment of similarities and providing recommendations. The issue surrounding the quantification of similarity holds significant importance. There are numerous approaches within the field of CF that can be employed to determine similarity. Based on a certain assumption, it is believed that consumers who share similarities will exhibit similar preferences for CF. It can be inferred that users with similar characteristics are likely to have similar collaborative filtering preferences [3]. In general, predictions can be categorized into two main types: individual predictions pertaining to specific entities, and top-N lists comprising a selection of items that are likely to be of interest to active users.

CF systems yield remarkably favorable outcomes. Nevertheless, the extensive utilization of these technologies has also unveiled notable challenges [4, 5]. The primary challenges encountered in this context pertain to the preservation of privacy and the vulnerability to shilling attacks. Additional challenges encompass issues pertaining to precision, scalability, sparsity, synonymy, and similar concerns.

Insufficient privacy protection in the CF system may cause users to withhold their data or provide inaccurate information. Customers require assurance that their personal data is safeguarded. Therefore, gathering high-quality user data for collaborative filtering (CF) purposes is a challenging task. Inadequate user data quality leads to subpar recommendations and imprecise predictions for users. Implementing privacy measures in the system can improve the collection of reliable and accurate data. Conventional recommendation systems prioritize accuracy by collecting and analyzing large amounts of data. However, privacy-focused recommendation systems prioritize user privacy and data control while still striving to offer relevant recommendations. The selection between the two alternatives relies on the particular needs and principles of users and organizations concerning privacy and personalization.

Several studies have been suggested in the academic literature [6-8] to address the aforementioned issues. Extensive research has been conducted on privacy and shilling attacks

due to their significant impact on the overall efficacy of CF schemes. Several strategies have been implemented in order to uphold confidentiality and generate dependable recommendations [9-11]. In a similar vein, a multitude of scholars have proposed algorithms aimed at mitigating shilling attacks and enhancing the security of CF schemes [12-15]. Canny conducted the initial investigation into privacy concerns within CF services, proposing two potential solutions known as Privacy-Preserving Collaborative Filtering (PPCF) solutions [16]. Initially, the author presents a novel approach to CF that ensures the preservation of personal data confidentiality. The individual employs a probabilistic factor analysis model as the foundation for his methodology. A peer-to-peer protocol provides a level of privacy protection. In the second schema, he presents an alternative paradigm wherein users possess full autonomy over their log data.

The utilization of privacy-preserving collaborative filtering (PPCF) methodologies has been identified as a viable means to safeguard personal data, as suggested by Polat and Du [17]. The act of incorporating arbitrary numerical values into authentic ratings is a form of data manipulation. The numbers are selected in a random manner from a pre-established distribution. When a value, denoted as x, is concealed, the addition of a random number, represented as r, will result in the perturbation of x by a factor known as randomized perturbation techniques (RPT), as well as the resulting value of x + r.

Malicious actors or entities may introduce fraudulent profiles into the database of the CF system with the intention of manipulating the projected forecasts to their advantage. This phenomenon is alternatively known as shilling. According to previous studies [18, 19], it is possible that CF systems may not possess the capability to effectively counteract shilling attacks. The utilization of counterfeit user profiles has the potential to manipulate system recommendations through the actions of malevolent customers, suppliers, or competitors. The primary objective of these assaults is to alter the outputs of the system. For example, certain acts of deception aim to manipulate individuals into making purchases of specific products, whereas others strive to diminish the popularity of certain products [19, 20]. Before carrying out a shilling attack, it is crucial that the attackers have a good understanding of the recommender system they plan to target. This information may include various data points, such as the mean rating and the measure of variability for each item and/or user in the user-item matrix, the distribution of ratings, and similar statistics. In order to mitigate shilling attacks on recommendation systems, it is necessary to employ specialized detection techniques [21-23] and robust recommendation algorithms [24-26].

The efficacy of an attack is assessed by researchers through the utilization of the stability of prediction metric [27, 43]. This metric quantifies the ratio of accurately predicted target items that remain unchanged. Stability is assessed by employing a pre-established threshold. The power of attack metric evaluates the efficacy of an assault, while also considering the stability of prediction. This statistic refers to the mean shift in prediction towards a specific goal value, encompassing all target users and products. The metric known as Prediction Shift measures the alteration in the anticipated rating of an item following an attack. The hit ratio quantifies the relative effectiveness of an attack on a propelled object when compared to alternative targets. This statistical measure can be employed to evaluate the actual impact of a push attack on recommendations.

Nevertheless, despite the concealment of data in PPCF schemes, there remains a vulnerability to potential shilling or profile injection attacks. There is some research examining the efficacy of PPCF's defense mechanisms against shilling attacks. In a study conducted by [28], an investigation was carried out to assess the resilience of two memory-based algorithms against various attacks. The memory-based schemes that have been investigated include the

k-nearest neighbors (k-NN) algorithm and correlation threshold-based approaches. The study conducted by [29] investigated the resilience of model-based algorithms against profile injection attacks. The model-based schemes employed in this study include the k-means algorithm, singular value decomposition (SVD), item, and discrete wavelet transform (DWT) based PPCF schemes.

Several studies in the current literature have examined ant colony clustering as a basis for non-privacy collaborative filtering schemes. This paper introduces a novel model-based PPCF scheme that utilizes the benefits of the ant colony clustering algorithm for cluster formation, while also ensuring privacy protection. Additionally, the scheme is tested for its robustness against shilling attacks, which are a common challenge faced by collaborative filtering schemes. At the same time, this study represents the first investigation of the robustness of ant colony-based PPCF schemes against shilling attacks. This paper conducts experiments to evaluate the robustness of this scheme against shilling attacks. Furthermore, the acquired outcomes are contrasted with various model-based PPCF schemes implemented in various studies in the literature.

An overview of the article's contributions:

1. The initial step involves the application of an ant colony clustering approach utilizing a PPCF schema.
2. A series of comprehensive experiments utilizing real data is performed to assess the resilience of the ant colony clustering model-based PPCF scheme in the face of six attack models.
3. The current study compares a colony clustering-based PPCF schema with a previously implemented other model-based PPCF schema in terms of its robustness against six-shilling attack models.

The subsequent sections of the paper are organized as follows. Section 2 provides a review of related studies and briefly highlights the distinctions between this work and previous research. Section 3 provides a description of the preliminary works. In Section 4, this paper provides information about the developed scheme and describes the algorithm of the scheme. Section 5 provides a detailed account of experiments conducted using real data, along with the corresponding outcomes. Section 6 concludes the study and discusses future work.

2. Related Work

The preservation of personal data has assumed heightened significance within contemporary society. Canny presents two strategies for ensuring the security of personal data on CF systems [16, 30]. These methodologies enable individuals to encrypt and decrypt their personal data while ensuring the preservation of their privacy. In order to address the concern of privacy on CF, Polat and Du [17] employed randomized perturbation techniques (RPT). The efficacy of online filtering systems has been enhanced through the utilization of model-based collaborative filtering algorithms. Numerous studies have been conducted in the existing body of literature pertaining to the implementation of model-based PPCF schemes. In their study, Polat and Du [31] propose a PPCF scheme that employs singular value decomposition (SVD) to safeguard user privacy in SVD-based collaborative filtering (CF) while simultaneously improving its scalability. Bilge and Polat [32] illustrate the methodology for conducting k-means clustering on collaborative filtering (CF) schemes while upholding user privacy. Luo et al. [33] developed a highly effective clustering-based recommender system that protects user privacy. Homomorphic encryption protects user data while Collaborative Filtering generates recommendations. The system uses secure clustering to divide data into groups

before making recommendations to reduce excessive information. Experiments show that the proposed system is effective, scalable, and makes accurate recommendations. Hedge et al. [34] conducted a comprehensive analysis of privacy-preserving clustering techniques. They implemented and evaluated four efficient clustering protocols that ensure complete privacy. The assessment evaluated the practicality of these protocols in real-world scenarios and highlighted unresolved challenges. The protocols were evaluated in terms of communication, computation, and clustering quality. The authors discussed the importance of assessing the quality of secure clustering and implementing privacy-preserving soft clustering. They highlighted the difficulties in maintaining privacy during the clustering process. Catak et al. [35] introduced innovative privacy-preserving clustering techniques utilizing homomorphic encryption schemes that are compatible with high-performance computation platforms, such as cloud systems. The text examines the calculation of distance matrices that preserve privacy for clustering algorithms and assesses the performance of the proposed model using different metrics. The authors also examined previous research on privacy-preserving machine learning models and demonstrated a practical implementation of privacy preservation on the clustering training model using a partially homomorphic Paillier cryptographic system.

The utilization of CF and PPCF strategies, extensively implemented by e-commerce platforms with the aim of enhancing sales, may exhibit susceptibility to shilling or profile injection attacks. The notion of a shilling assault was initially proposed by [19, 24], while Dellarocas [36] examined unethical behaviors related to reputation reporting systems. The main aim of the study was to improve the dependability of online reputation systems through the identification of fraudulent activities. As stated by O'Mahony et al. [19, 24] , recommender systems are susceptible to attacks aimed at influencing specific recommendations. Multiple studies have been carried out to analyze and identify prospective attacks, detect them, strengthen the ability of recommender systems to withstand such attacks, create strong algorithms to defend against known attacks, and conduct cost-benefit evaluations. Furthermore, there exist other scholarly investigations that provide a synthesis of recent progressions within this particular domain. Several scholars focused on the analysis of shilling attacks and their influence on recommendation systems.

Mobasher et al. [37, 38] conducted a classification of attack types depending on their dimensions, taking into account the information required to identify the attack, the purpose behind the attack, and the severity of the attack. The paper's conclusion highlighted specific instances of attacks, providing illustrative examples. The authors conducted a thorough analysis of not only the different types of attacks, but also the evaluation metrics, detection techniques, and the definition of shilling attacks. Burke et al. [39] presented several crucial areas for future investigation in resilient CF systems, encompassing attack models, methods, profiling techniques, detection, and evaluation.

The investigation conducted by Sandvig et al. [12] was restricted to exclusively examining robust model-based algorithms. Zhang [40] conducted an examination of a limited range of attack types, attack detection techniques, and assessing metrics. The studies conducted by Burke et al. [39], Burke et al. [26] and Mobasher et al. [37] examine different facets of shilling attacks. In a comprehensive literature review conducted by Gunes et al. [41], an extensive examination of the existing research pertaining to shilling attacks was presented. In addition, the authors conducted an analysis of attack descriptions that encompassed specific details, detection methodologies, the design of robust algorithms, cost-benefit evaluations, and metrics Si and Li [42] conducted a survey on shilling attacks in collaborative filtering recommender systems (CFRSs). The paper explores different attack strategies, detection schemes, and robust recommendation algorithms. In addition, the authors provide an explanation of evaluation

metrics and propose future research directions to enhance the accuracy and resilience of detecting shilling attacks in CFRSs. The review offers a thorough analysis of the subject matter and highlights potential avenues for future investigation. Kaya and Kaleli [43] analyzed the vulnerability of multi-criteria top-n recommendation methods to manipulations and shilling attacks. They introduced a new shilling attack strategy and evaluated the robustness of these systems against the attacks. The study used real-world datasets and proposed new approaches for selecting powerful items and identifying target products required for the attack model.

Model-based studies are frequently employed to enhance the performance of CF and PPCF schemes. Several studies in the literature have explored the use of ant colony model-based CF schemes. In their study, Wu et al. [44] suggested the utilization of the ant algorithm for user clustering. This approach seeks to reduce the expenses incurred in searching, alleviate the influence of initial clustering centers and clustering numbers related to the K-Means clustering technique, and improve the speed at which nearest neighbors are queried in CF recommendation systems. As stated in the report, the experiment demonstrated the efficiency of user clustering through the utilization of the ant colony method. Furthermore, it effectively addressed the concern pertaining to new users who are not recommended, thereby enhancing the precision of the cooperation filtering suggestion algorithm.

The fuzzy ant-based recommender system (FARS) was proposed by Nadi et al. in [45]. The FARS methodology is employed to extract user preferences in online platforms by analyzing web server log files. Ant-based clustering methods are employed to appropriately categorize users into specific groups. Ant-based algorithms play a crucial role in providing optimal solutions. Upon the completion of the recommendation process, the pheromone allocated to each cluster is subsequently updated in preparation for future utilization. The accuracy and recall metrics are utilized to quantify the precision and comprehensiveness of the generated recommendations. Based on their research findings, the implementation of the recommended approach for user grouping is expected to yield improved accuracy in generating recommendations. In their study, Liao et al. [46] proposed an improvement to the ant colony-based CF algorithm for the purpose of enhancing its performance. This enhancement involves incorporating a preliminary phase of user clustering, which is ascertained through user preferences as indicated by the presence of pheromones. It is pertinent to mention that the quantity of users is considerably more than the representation of pheromones in the algorithm.

Sobecki and Tomczak [47] provided recommendations for student courses based on Ant Colony Optimization (ACO). The efficacy of ACO has been demonstrated in effectively addressing a range of optimization problems. The authors illustrated the efficacy of ACO in effectively addressing the task of predicting students' final grades upon completion of university courses. The Trust-based Ant Recommender System (TARS), as proposed by Bedi and Sharma [48], is designed to produce effective recommendations by integrating the concept of dynamical trust among users and employing the principles of ant colonies to determine the most optimal and compact neighborhood. The assertion made by the authors is that providing supplementary information to explain recommendations pertaining to the power and degree of connection in the trust graph, the items being recommended, and the number of neighbors present in predicting ratings can enhance the decision-making abilities of active users.

Numerous ant colony-based CF algorithms have been implemented and documented in the academic literature. Nevertheless, the existing literature lacks any research on privacy-preserving CF schemes. Simultaneously, the extent to which the ant colony-based CF and PPCF algorithms can withstand shilling attacks has not been observed in any research study. This article examines the robustness of the ant colony clustering-based PPCF scheme against shilling attacks. In this particular context, it is of utmost significance to undertake a study of

this nature, as it pertains to the model in question, for the initial instance.

3. Preliminaries

3.1 Privacy Protection by Randomization

Online sellers should prioritize achieving accuracy and privacy to effectively attract customers. However, preserving privacy requires a certain level of degradation in user data, which subsequently reduces accuracy, creating an inherent conflict between these two objectives. Therefore, a certain degree of accuracy must be sacrificed, while privacy measures need to be carefully adjusted to achieve a balanced trade-off. Privacy-preserving techniques in recommendation systems aim to provide personalized recommendations to users while protecting their sensitive data [49-51]. The following are some commonly used techniques:

Randomized perturbation techniques (RPT): These methods introduce random noise or perturbations to the data to prevent user re-identification while generating accurate recommendations. Randomized perturbation techniques protect user data in recommendation systems by adding randomness to data. The methods included in this approach ensure ease of implementation, preserve statistical value, allow for inherent clustering, and enable lossless transformation. These methods produce accurate recommendations while protecting user data.

Homomorphic encryption: This method enables computations on encrypted data without the need for decryption, thereby safeguarding sensitive user data while producing accurate recommendations. It is particularly useful for protecting user data during recommendation generation in recommendation systems. The technology's computational costs range from moderate to high, making it suitable for cloud computing and Internet of Things (IoT) applications.

Differential privacy: This technique aims to protect the privacy of user data by adding random elements to the data before sharing it. This ensures that the shared data does not reveal any confidential information about individual users. The approach incurs minimal computational costs and does not involve any additional communication costs. However, it may slightly affect the model's performance.

Secure multi-party computation: This technique enables multiple entities to collaborate in performing a computation on their respective inputs while preserving the privacy of those inputs. It guarantees that no party gains access to more information than what can be inferred from the output. It has low computational costs but increases communication costs.

Federated learning: This methodology allows for the training of recommendation models using intermediate parameters instead of actual user data. This facilitates collaboration between data platforms while adhering to privacy regulations.

Cryptographic-based: These methods use cryptographic techniques to protect sensitive user data while providing accurate recommendations. Encryption and other cryptographic methods ensure the confidentiality and security of user information, thus preserving privacy in recommendation systems. It incorporates robust privacy and reliability while minimizing the trade-off between security and accuracy. However, these techniques are computationally expensive and may not provide guaranteed reliability.

Each technique has its own advantages and limitations briefly described above. The study used the randomized perturbation technique (RPT), which is known for its ease of use and high accuracy. The utilization of RPT facilitates the achievement of successful privacy applications. According to Agrawal and Srikant [52], it is recommended to utilize these methodologies. RPT incorporates the addition of a random value, denoted as r, to a private

data item referred to as x. This is done in order to obfuscate the value that is being transmitted. The main objective of a random number is to produce an established allocation of data values, which are then saved in a database using the format of $x + r$. This distribution can then be retrieved and used as needed. When evaluating recommendation systems, data that is aggregated, as opposed to individual data, are typically used. As a result, they are able to effectively generate recommendations by utilizing aggregated perturbed data. When it comes to PPCF schemes, one of the primary goals of the privacy preservation process is to stop the server from discovering the genuine ratings and objects that have been given those ratings. By utilizing a process of generating random values and subsequently incorporating them into the existing rates, it is possible to acquire perturbed data.

Moreover, users have the capability to generate arbitrary values in order to incorporate a selection of unrated elements that have been randomly chosen. When generating random numbers, individuals utilize either a Gaussian or uniform distribution characterized by a mean (μ) of zero and a standard deviation (σ) [17]. In the context of the PPCF scheme, users initially employ the z-score method to standardize their evaluations. The server determines the values of σ_{max} and β_{max}. The term " β_{max}" refers to the upper limit of the fill rate for unrated elements that are intended to be populated with random values. Subsequently, it enables users to acquaint themselves with the aforementioned information. Every user, denoted as u, selects a value for u within the interval [0, σ_{max}] and a value for βu within the interval [u, β_{max}]. Polat & Du [31] summarized the sequential steps of the data disguising process below:

1. For each user's ratings, z-score values are calculated.
2. The server chooses σ_{max} and β_{max} values and notifies every user of them.
3. Every user u selects β_u and β_u percentage of their items without ratings to be populated with random numbers.
4. Prior to completing random number distribution, every user u chooses the standard deviation σ_u of the random numbers. Coin tosses are then used to decide whether random numbers will have a uniform or Gaussian distribution.
5. Users generate random numbers (r_{uj} values) for both genuine and unrated items that are chosen to be filled during the post distribution selection phase. Then, each user hides their z-score values by randomly adding values ($z'_{uj} = z_{uj} + r_{uj}$). Finally, every user assigns the matching random numbers to the selected unrated items.
6. In the last stage of the process, users will send their masked vectors to the server that has been designated.

3.2 Privacy Based Ant Colony Clustering Algorithm

In their study, Shelokar et al. [53] developed an ant colony optimization algorithm with the aim of resolving clustering challenges. The software ants make use of a pheromone matrix, which functions as an adaptive memory mechanism, to direct and coordinate the movement of other ants in the direction of the most effective clustering solution. The outcome of the objective function and the rate of evaporation both have an effect on the amount of pheromone that is deposited at a particular location (i, j), which corresponds to the assignment of sample i to cluster j. These two factors are responsible for the majority of the variance in the results. The evaporation rate serves as a forgetting factor, enabling the exploration of alternative clustering locations for item i. The ACO algorithm for data clustering is suitable in scenarios where the number of clusters is predetermined and they exhibit clear differentiation. The authors conduct a comparative analysis between the ACO algorithm and other stochastic algorithms, such as the genetic algorithm, in order to assess its effectiveness. The technique was implemented and subsequently evaluated on various simulated and real datasets. The

researchers reported highly favorable results in terms of solution quality, average function evaluations, and processing time. The clustering algorithm depicted in **Fig. 1** was utilized in this study. Several modifications were implemented on this algorithm to align it with the data utilized in the study. Shelokar et al. [53] offered an all-encompassing explanation of the algorithm depicted in **Fig. 1** within their scholarly publication.

The authors developed an ant colony algorithm with the goal of achieving an optimal cluster distribution by minimizing the sum of squared Euclidean distances between each object and its corresponding cluster center. This was done in order to achieve an optimal cluster distribution. In order to come up with solutions, this methodology takes into account a group of R agents. An initial string representation of the solution, denoted by the letter S and having a length of N, is created by the agent. Each individual element of the string represents one of the test samples. At the outset, there is no information stored in the solution string. The number of the cluster that the test sample is part of is represented by the value that is given to an element in the solution string that is denoted by the letter S. Following the creation of a population that is comprised of R trial solutions, an additional local search is carried out in order to enhance the fitness of these solutions. The pheromone matrix is subject to updates, the nature of which are determined by the quality of the solutions generated by the agents. The agents generate better solutions by making use of the altered pheromone matrix, and the stages described above are carried out in an iterative manner until the desired number of iterations has been reached [53]. Ant colony clustering algorithms offer several advantages. Firstly, they are capable of finding optimal or near-optimal solutions for clustering problems. Secondly, they utilize computational resources efficiently, as shown by the low average number of function evaluations and processing time. Finally, in terms of solution quality, genetic algorithms outperform other commonly used stochastic/heuristic methods such as simulated annealing and tabu search. These algorithms can effectively handle complex combinatorial and function optimization problems on a large scale. This study applies the ant colony clustering algorithm to PPCF systems to take advantage of its benefits in cluster formation.

As previously stated, the literature contains various methods that seek to protect user privacy in collaborative filtering schemes. One such method, proposed by Polat and Du [54], employs random perturbation techniques to achieve this goal while still providing accurate recommendations. The proposed scheme allows servers to collect private data without greatly compromising user privacy, achieving a balance between privacy and accuracy. The authors conducted experiments using the Jester and MovieLens datasets to verify the accuracy of the results, showing that predictions on randomized data are very close to the original ratings. Polat and Wu [54] demonstrated the ability to provide recommendations with an accuracy loss of 0.0835 in an experiment using the MovieLens Million Data (MLM) dataset described in section 5.1. The MAE of 0.0835 indicates that the results are very close to those generated from the original data, given that the rating range for the MLM dataset is from 1 to 5. In this study, RPT technique, which is easy to implement and has high accuracy, was used to ensure privacy.

Numerous methods have been proposed in the literature to address the scalability issue, which is another challenge faced by collaborative filtering schemes. Model-based CF and PPCF algorithms create a model based on system data, particularly user ratings. There are several types of model-based algorithms, including cluster models, probabilistic models, Bayesian networks, rule-based methods, and dimensionality reduction methods. The aim of the clustering technique is to divide the data set into distinct groups of users. The bisecting k-means algorithm [55] a modified version of the k-means clustering algorithm, is used for this purpose. Bilge and Polat [32] describe the procedure for performing k-means and bisection k-

means clustering on collaborative filtering (CF) schemes while maintaining user privacy. Bilge and Polat [32] used k-means and bisection-k-means algorithms to improve scalability while maintaining user privacy. Similar to other studies in the literature, there are many approaches that aim to preserve privacy while increasing scalability.

The primary objective of this study is to assess the robustness of the developed scheme. Additionally, a test was conducted to examine the impact of privacy on accuracy. In this experiment, prediction calculations were performed for 50 items across all users. MAE calculations were conducted by considering the users who had actually rated these items. The same experiment was conducted for both Ant colony clustering-CF and Ant colony clustering-PPCF. The privacy parameters described in the section 3.1, β_{max} = 25% and σ_{max} = 2, were consistently upheld. MAE values of 0.819 and 0.909 were obtained for Ant colony clustering-CF and Ant colony clustering-PPCF, respectively. According to this test result, there was an accuracy loss of 0.09, similar to the results of Polat and Wu [54]. This is an acceptable loss of accuracy for the 1-5 rating range. The MAE value is susceptible to alterations due to the interplay of various factors, including privacy parameters (β_{max} and σ_{max}) and parameters such as target items and the number of neighbors employed during prediction calculations. Depending on the variation of these parameters, an accuracy loss of between 0.08 and 0.15 is anticipated. The influence of these parameters necessitates further investigation in a separate study. In future studies, we will examine ant colony clustering model-based PPCF schemes in detail, with a particular focus on their accuracy and scalability.

Memory-based collaborative filtering (CF) algorithms use either the entire or a subset of the user-item database to generate predictions. They compute the similarity between the active user and all other users, and then identify the nearest neighbors. In model-based CF/PPCF, a model is first constructed, and then only the most similar users are selected from that model. When clustering algorithms are used as the model, users are initially grouped into clusters based on their similarities. When an active user requests a prediction for a target item, their similarity to each cluster center is calculated. The user is then associated with the most similar cluster, and similarity is calculated only with the users in that cluster. The n most similar users are identified as neighbors, and the prediction calculation is performed using those neighbors. To create a more scalable CF/PPCF scheme, similarity is calculated only with the users in the associated cluster, rather than with all users. Therefore, the model-based ant colony clustering PPCF is more advantageous than user-based CF schemes in terms of both scalability and privacy.

Another advantage of selecting a model-based collaborative filtering scheme in our study is its greater robustness compared to the memory-based approach. Research in the field of literature has conducted comparisons between the resilience of memory-based and model-based systems. Mobasher et al [56] conducted a comparison between k-means clustering and k-nn-based collaborative filtering (CF) algorithms. Their findings indicate that k-means algorithm demonstrates greater robustness. Similarly, Bilge et al. [29] conducted a comparison between four model-based PPCF models and the user-based PPCF model to assess their robustness. The findings indicated that the model-based PPCF schemes exhibited greater robustness. Hence, it can be inferred that model-based schemes exhibit greater resilience in both CF and PPCF schemes.

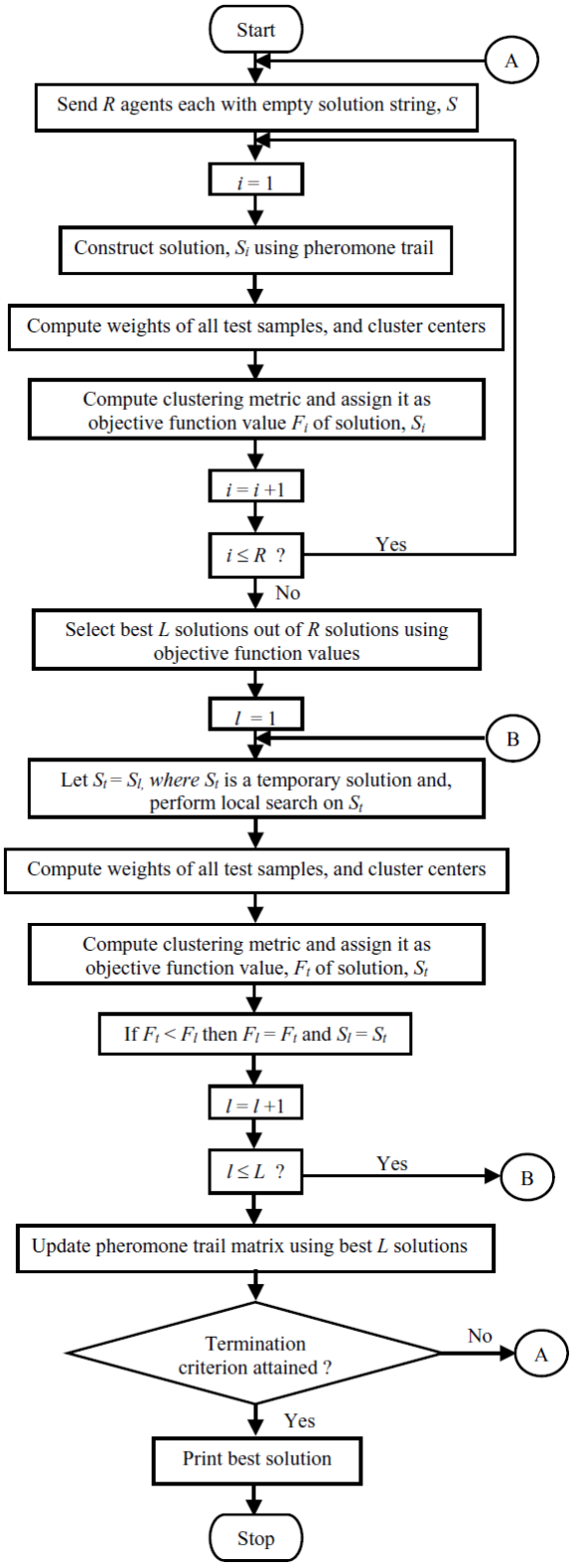

Fig. 1. Ant colony clustering algorithm [53]

3.3 Shilling Attack Models

A shilling attack is a type of manipulation where fake profiles are created to influence the recommendations made by a recommender system. Shilling attackers create fake accounts and allocate points strategically to manipulate the rankings of specific items, either by boosting them higher (push attack) or lowering them (nuke attack) in recommendation systems. These actions may be taken with the intention of deceiving users or gaining an unfair advantage, such as promoting a product that the attacker personally sells. Typically, shilling attacks are detected by attackers through the insertion of an attack profile, as illustrated in **Fig. 2**. This approach is initially addressed by Bhaumik et al. [57], Mobasher et al. [38], and Mobasher et al. [37] with the intention of misleading the collaborative filtering (CF) system. These profiles can be categorized into four distinct groups. The initial step undertaken by the assailant involves the identification of a collection of items, denoted as I_S, in conjunction with the selection of a particular rating function. These actions serve to define the characteristics of the attack. Furthermore, a rating mechanism is employed to selectively determine an alternative set of entities, with the intention of hindering the identification of a potential assault. In essence, a rating function, denoted as, is employed to produce a bias towards a specific object. The objects that have not been assigned a rating are represented as "I" in **Fig. 2** for the duration of the inventory. The act of impersonating legitimate individuals and fabricating false profiles is carried out by a malevolent user. It is crucial to acknowledge that the profile of a user is defined by their choices regarding various items, that are depicted as vectors. The user subsequently submits these attack profiles to the recommender system for targeting purposes, effectively injecting them into the database of the targeted system.

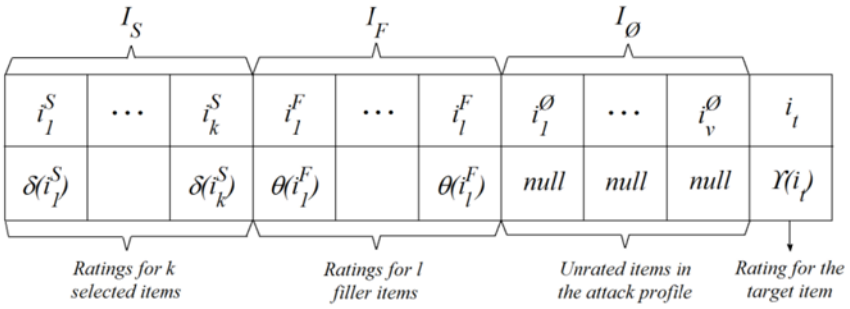

Fig. 2. General form of an attack profile

In general, the identification of an attack involves the insertion of multiple attack profiles into a recommender system's database, with the intention of inducing a bias towards specific target items. According to the research conducted by Lam and Riedl [58], attacks can serve various purposes and can be classified according to both the intent of the attack and the specific information required. As push attacks, the most common types of profile injection or shilling attacks are random, average, bandwagon, and segment; as nuke attacks, reverse bandwagon and love/hate are utilized [18, 38]. In their study, Mobasher et al. [38] provide an overview of various attack types, such as random, average, bandwagon, segment, love/hate, and reverse bandwagon, as documented in **Table 1**.

Table 1. Attack types according to intent and required knowledge

Attack Type	*Intent*		*Required Knowledge*		
	Push	*Nuke*	*Low*	*High*	*Informed*
Random	✓	✓	✓		
Average	✓	✓		✓	
Bandwagon	✓		✓		
Segment	✓		✓		
Reverse Bandwagon		✓	✓		
Love/Hate		✓	✓		

Based on the information presented in **Table 1**, shilling attacks can be categorized into two distinct types, namely push and nuke, depending on their intended objective. Likewise, these attacks can be categorized into three distinct groups: low, high, or informed attacks, depending on the level of expertise required. While certain strikes possess the sole capability of propelling or annihilating an object, there exist others that can serve both functions simultaneously. As indicated in **Table 1**, attacks generally necessitate minimal information. On the other hand, average attacks necessitate a substantial amount of expertise. In order to carry out an attack on a recommender system, the attacker must possess a certain level of understanding about the system they are targeting. This includes knowledge about the algorithm used, the users, the items being recommended, and the ratings associated with them [58]. For instance, the bandwagon attack model requires popular items information in the system. The bandwagon attack attempts to mimic the behavior of users within the system by assigning a high rating to popular items. The more closely it resembles actual users, the greater its ability to manipulate the system. The subsequent elucidation provides a concise overview of the prevailing attack categories.

Random attack: In this instance of a random attack, there is no set of items that have been chosen. The value of each filler item set is determined using a normal distribution, with the mean and standard deviation of the system's overall rating serving as the parameters of the distribution. In an effort to boost the item's popularity, it has been given the highest possible rating value.
Average attack: It's almost like an attack at random. The values of filler item sets are determined based on a normal distribution that is centered around the item's mean.
Bandwagon attack: Products that receive the highest possible rating value are selected from among those that are popular and receive typically high ratings. Items used as filler are chosen at random from a normal distribution that is centered on the mean value of the system. The item that was evaluated received the maximum rating value that could possibly be given.
Segment attack: A comparison can be made between this attack and the bandwagon attack. The one and only difference is that certain items are chosen from a specific group that may appeal to a particular group of users, such as purchasers of excellent movies, horror films, etc. The sole distinction lies in the selection of specific items from a specific group that may appeal to a particular group of users.
Reverse bandwagon attack: In contrast to the bandwagon attack, the selection process favors unpopular entities. The minimum rating value is established for the specific item that is being targeted.
Love/hate attack: The set that has been chosen does not contain any elements. The highest rating value is allocated to filler items selected at random, whereas the minimum rating value

is assigned to the target item with the intention of significantly reducing its popularity.

4. Shilling Attacks Against Ant Colony Clustering Model-Based Prediction Schemes with Privacy

Memory-based or model-based approaches are the two most common classifications for PPCF methods. Memory-based strategies that also emphasize confidentiality are the most fundamental examples of heuristic methods. Using methods like these, one can generate predictions with relative ease. Because memory-based algorithms run online, adding a new user or product to the collection is a straightforward process that can be done whenever needed. It is not necessary to conduct a content analysis on the items that have been recommended. When it comes to products with comparable ratings, the mechanism works very well. On the other hand, the sheer amount of data that these systems need to process can be detrimental to their ability to scale. Due to the limited amount of data available, the system might not be able to make a recommendation for a new user when that user first logs into the system. In contrast, model-based CF algorithms that protect users' privacy generate a model based not only on forecasts but also on ratings provided by users. The implementation of these algorithms is more challenging, despite the fact that they outperform memory-based algorithms in terms of scalability and sparsity. For the purpose of addressing sparsity and scalability issues, model-based ant colony CF schemes have been developed and published in the research literature [45, 46].

However, there are many studies in the literature that examine different challenges such as privacy and shilling attacks in model-based recommender systems using different clustering algorithms. Bilge et al. [29] addressed privacy and shilling attack challenges in CF algorithms by using the k-means clustering method. The researchers assessed the resilience of the k-means PPCF algorithm against six different assault models. Wei et al. [59] introduced a technique known as (p, l, α)-diversity to enhance the current k-anonymity method in PPCF used in recommender systems. The objective of the strategy is to enhance privacy protection during the recommendation process by intensifying privacy preservation and minimizing information loss. Deng et al. [60] presented a novel K-medoids clustering recommendation algorithm for collaborative filtering that is based on probability distribution. The algorithm calculates item similarity by employing a modified version of the Kullback-Leibler (KL) divergence, and it searches for cluster centers by increasing the contribution sum of distance to its maximum value.

This study presents the implementation of a PPCF scheme based on ant colony optimization. Notably, the investigation of privacy concerns in ant colony-based collaborative filtering schemes has not been previously explored. Furthermore, following the resolution of the privacy issue, the robustness of this scheme against shilling attacks is examined via experimental analysis. Extensive experiments with real data are carried out as part of this body of work in order assess the resilience of ant colony clustering model-based PPCF algorithms when put up against six different attack models.

The algorithm depicted in **Fig. 1** went through a few iterations of refinement in order to accommodate the application of the ant colony algorithm to recommender systems. In their work, Shelokar et al. [53] consider the optimal solution to be the minimum sum of the squared Euclidean distances between the cluster centers of each element. In our study, the optimal cluster distribution is the maximum sum of the similarities of each element to the cluster center. In our modified algorithm, the Pearson correlation similarity measure is used instead of the

Euclidean distance. When assigning the elements to the cluster, the similarity between each element and the cluster center is calculated, and the assignment process is performed by considering the largest similarity value. The Pearson correlation similarity metric formula is shown in **Eq. 1** below.

In this study, a model is first created using a modified ant colony algorithm. When clustering algorithms are used as a model, users are initially grouped into clusters according to their similarities. The prediction algorithm is then applied to this model and run. When an active user requests a prediction for a target item, their similarity to each cluster center is calculated. The user is then associated with the most similar cluster and similarity is calculated only with users in this cluster. The n most similar users are defined as neighbors, and the prediction calculation is performed using these neighbors. The similarity is computed only with users in the associated cluster, rather than all users, thus increasing the scalability of the PPCF scheme. In this study, all users in the dataset are considered as test users in sequence. For push and nuke attacks, 50 items were selected. Predictions are calculated separately for these items before and after the attack profiles are added. Then the effect of attacks on prediction is calculated with prediction shift measure.

When a user who is currently engaged in the system requests a prediction for a specific item, denoted as q, the server first establishes a's degree of similarity to each cluster center by employing the Pearson Correlation similarity measure in the manner described below:

$$w_{ac} = \frac{\sum_{j=1}^{m}\left(v_{aj} - \overline{v_a}\right)\left(v_{cj} - \overline{v_c}\right)}{\sigma_a \sigma_c} \tag{1}$$

Where c stands for the center of the cluster, v_{aj} is the rating that user a gave item j, $\overline{v_a}$ and $\overline{v_c}$ are the vector mean values of user a and the corresponding cluster center respectively, and σ_a and σ_C are the standard deviations of user a and the corresponding cluster center respectively.

The prediction algorithm suggested by Herlocker et al. [2], which is also utilized in PPCF frameworks, incorporates z-score normalization to apply variance weighting to neighboring data points. Users report their evaluations as normalized by their z-score in accordance with this system rather than providing their actual ratings. The predicted score for user a on item q is calculated by taking the weighted average of the z-scores of the users that are immediately adjacent to user a [2].

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

N is the number of neighbors selected for a specific cluster. $\overline{v_a}$ denotes the average value of the user *a*'s vector. w_{au} represents the similarity between the active user *a* and the user's neighbors. z_{uq} represents the z-scores of the neighbors on item *q*.

The implementation steps are generally summarized as follows:

- First, disguise the experimental data, which will be described in Section 5. The method for disguising the data is explained in detail in Section 1.
- Modify the ant colony clustering algorithm (using the Pearson correlation similarity metric instead of the Euclidean metric).
- Determine optimal agent, cluster and iteration size. Run the ant colony clustering

algorithm on hidden data to create clusters.

- After distinguishing each user as either a test or active user, the remaining users are allocated to the training set. The system experienced attacks on all target items for all test users, and predictions were generated both before and after injecting the attack profiles. When calculating the predictions for the active user, the PPCF scheme based on the ant colony clustering model-based PPCF schema selects neighbors from the cluster to which the user belongs rather than from all users. This increases the scalability of the scheme.
- Shilling profiles are generated and added to the nearest cluster.
- After adding shilling profiles, predictions are recalculated for all users and targets.
- Various metrics can be used to evaluate the efficiency of profile injections. The primary metric commonly employed to assess the effectiveness of shilling attacks is the prediction shift. This metric quantifies the average alteration in the anticipated rating of a targeted item following the attack.
- Finally, using the prediction shift metric defined in Section 5, we compute how much shilling attacks manipulate the prediction results. For the purpose of analyzing the effects of shilling attacks, two control factors known as filler size and attack size were utilized.

Briefly, the steps of the algorithm are shown below:
Algorithm 1 (ant colony clustering-based PPCF)

1. *load Data (MLP)*
2. *dData ← disquse(Data)*
3. *Set ant colony algorithm parameters: cluster size k, agent size S and iteration size t*
4. *Run ant colony-based PPCF algorithm and calculate prediction Result for all users and target items*
5. *Add shilling profiles for all attack models*
6. *Run ant colony -based PPCF algorithm and calculate prediction Result2 for all users and target items on data adding shilling profiles (do for all attacks model)*
7. *Calculate differences with Result – Result2 and prediction shifts for all attack models*

4.1 Costs Analysis

It is crucial to assess the proposed scheme in terms of both offline and online costs. While offline costs do not directly impact performance, they must be considered to provide a comprehensive analysis of the offline workload overload. In the scheme proposed in this study, the model building phase with ant colony clustering is performed off-line and the prediction calculation is performed on-line.

4.1.1 Clustering Phase

The complexity of this step depends on the number of ants and the size of the problem space. If each ant constructs its solution independently, the complexity could be *O(snm)*, where *s* is the number of ants, *m* is the number of items and *n* is the number of users. After each iteration, the pheromone trails are updated based on the quality of the solutions found. The complexity of this step is typically *O(s)*, where *s* is the number of ants.
The algorithm's scalability depends on the size of the problem space and the number of ants used. As the problem space or the number of ants increases, the computational requirements

of the algorithm may also increase significantly. Large-scale clustering problems may require a large number of ants to explore the solution space effectively, leading to higher computational complexity.

Model-based CF/PPCF approaches generate a model off-line and work on reduced data, which helps to overcome scalability and sparsity issues. This is followed by the grouping of users into c clusters using different clustering approaches offline. Although offline costs do not have a significant impact on performance compared to online overheads, they still need to be analyzed to provide a report on the size of the offline workload. The ant colony clustering algorithm employed in this study has a negligible impact on performance, as it operates in an off-line mode, even when the complexity increases in proportion to the number of agents and the size of the dataset.

4.1.2 Prediction Phase

The calculation of an item prediction for a user accessing the system is performed online. During an online interaction, when an active user wants a prediction for a target item, she sends her known ratings and a query to the server. The server calculates the similarity between user a and each cluster center. The calculation of these similarities by dot product is performed in $O(mc)$ time, where m is the number of items and c is the number of clusters. Once the cluster to which the user belongs is determined, the exact similarities between the active user and the rating profiles are determined. Thus, the estimation of online predictions can be approximated with a complexity of $O(Nm)$, where N is typically much smaller than n in systems facing scalability issues. n denotes all users, while N denotes the set within the identified cluster.

5. Experimental Evaluation

To assess the efficacy of our shilling attack models on ant colony clustering model-based PPCF algorithms, actual data-based tests are carried out. The current assessments utilized two control variables, specifically the size of the filler and the size of the attack. The use of effective shilling hits is a topic that has been investigated in the published research, as shown by the works of Bhaumik et al. [57] and Mobasher et al.[38]. The filler size corresponds to the proportion of unoccupied cells that must be filled in created profiles, using the rating function designated as θ, to prevent the identification of malicious activity, as described in Section 3 of the research done by Bhaumik et al. [57].

The term "attack size" pertains to the number of attack profiles that need to be installed, and this number is directly correlated with the number of customers in the system [38]. The PPCF parameters, β_{max} = 25% and σ_{max} = 2, are consistently upheld. According to Bilge and Polat [61], these values offer a satisfactory degree of personal privacy.

In the ant colony algorithm, the value of the parameter denoting the number of agents (S) is assigned as 10, the parameter representing the cluster size (k) is set to 6, and the parameter indicating the number of iterations (t) is defined as 20. As previously discussed, our selection criterion at the end of these iterations is to identify the cluster distribution with the highest fitness value. In Ant colony clustering algorithm, initially the elements are randomly distributed into clusters.

5.1 Date Set and Evaluation Criterion

The trials were carried out utilizing the readily accessible MovieLens dataset. The data was gathered by the GroupLens research team, as recorded on their website (http://www.grouplens.org). The dataset consists of 100,000 ratings given to 1,682 movies by a total of 943 individuals. It is acknowledged that discrete ratings ranging from 1 to 5 are present within the given set. A variety of metrics can be employed to assess the efficacy of profile injections. In their study, Burke et al. [26] assessed the effect of deployed shilling attack models by evaluating the metric of prediction shift, which is commonly used for this purpose. Prediction shift refers to the average change in the prediction made for a specific object, when comparing the prediction before and after the assault is carried out.

5.2 Experimental Methodology

The experiments employed a methodology that encompassed *all-but-one* experimental approach. In each iteration, a single user is assigned as the active user, while the remaining users constitute the training set. Additionally, two distinct collections comprising 50 films each are generated specifically for the intention of push and nuke attacks. To achieve a representative sample from the original dataset, a random assortment of movies was selected, with the aim of ensuring their distribution across various rating ranges. Shilling attacks target specific sets of items, as it is not meaningful to attempt to push predictions for items that already have high scores or to nuke poorly rated items. Hence, the set pertaining to push attacks comprises things with rating averages spanning from 1 to 3, whereas the set linked to nuke assaults comprises goods with rating averages spanning from 3 to 5. The experiments targeted all items for all users in the system and estimated attack profiles before and after injection. Prediction shift values were then calculated to demonstrate relative changes in estimated recommendations for each attack model.

5.3 Empirical Results

This section provides empirical findings obtained by manipulating different controlling parameters and discusses their significance.

5.3.1 Effect of Filler Size Parameter

The paper conducted experiments to evaluate the impact of disguised push and nuke assault models, with varying filler size values, on ant colony-based PPCF. The success of an attack is directly influenced by the size of the filler, as fillers serve as the foundation for infiltrating the community of real users during the recommendation procedure. Given that the initial value of β_{max} was set at 25%, the filler size was manipulated within the range of 3% to 25% throughout the experimental trials. Meanwhile, the attack size remained constant at its maximum value of 15%.

The **Fig. 3**, presented below illustrates the shift values of prediction for four push attack models when implemented in the ant-based PPCF scheme. Overall, the findings indicate a favorable shift in predictions ranging from 0.6 to 1.3. On a scale from one to five stars, the resulting prediction shifts are deemed significant. Based on the empirical findings, it is evident that the ant-based PPCF scheme is susceptible to shilling attacks, as shown in **Fig. 3**. As expected, the bandwagon and segment attack models, which exhibit superior efficacy, have demonstrated better effectiveness in comparison to other models. Since the attack models use popular products, their impact is likely to be greater because they are more likely to interact with more users.

Moreover, the impact decreases as the size of the filler rises, since a greater number of fillers allows the system to more effectively discern attack patterns and categorize them together. Nevertheless, the cluster discrimination process has a tendency to merge attack profiles that have comparable properties. Consequently, as shown in **Fig. 3**, when the size of the filler increases, the effectiveness of attacks decreases.

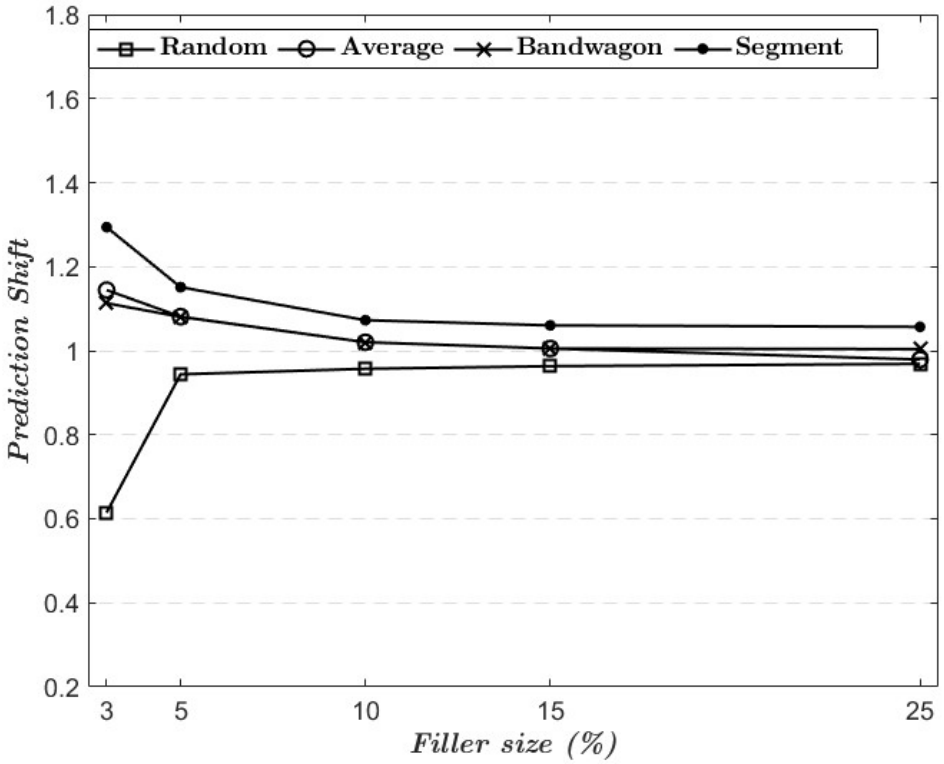

Fig. 3. Prediction shift for different sizes of filler (in push attack models)

The **Fig. 4** presented below illustrates the calculated values for prediction shift that have been derived from the models used to simulate a nuke attack. The **Fig. 4** illustrates the negative prediction shift values obtained in the Reverse Bandwagon and love/hate attack models, ranging from -0.62 to -0.32. The utilization of nuke attacks has resulted in a modestly adverse impact on the shift value of predictions. Furthermore, as elucidated in the preceding paragraph, the impact of the attacks diminished with an increase in the filler size value, similar to the push attack models.

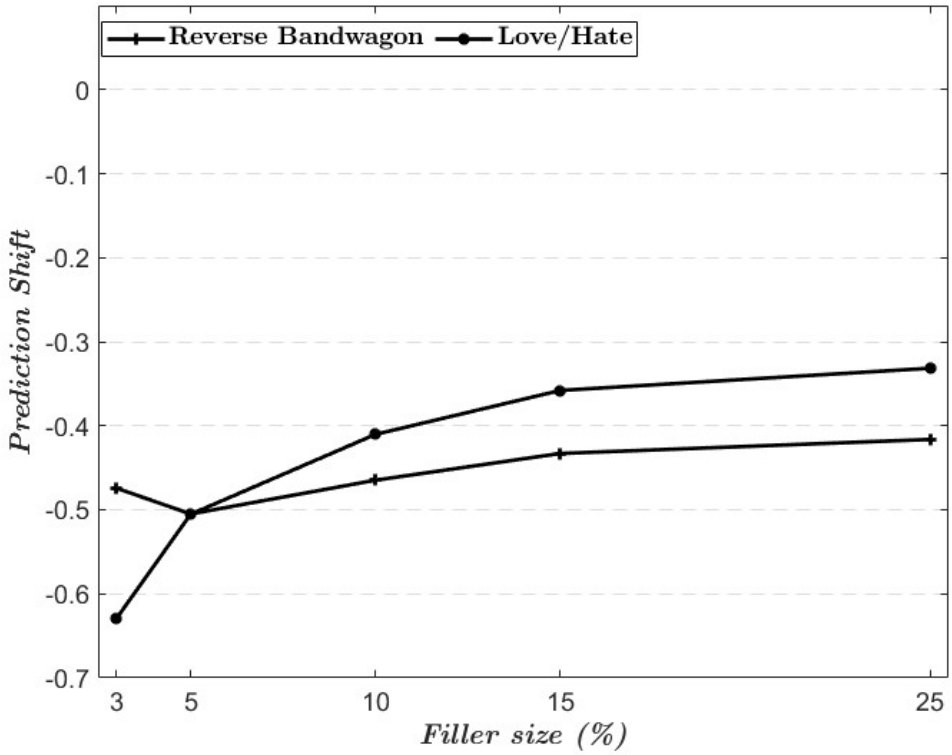

Fig. 4. Prediction shift for different sizes of filler (in nuke attack models)

5.3.2 Effect of Attack Size Parameter

Subsequently, a subsequent set of experiments was undertaken, wherein the attack sizes were varied from 1% to 15%, with the intention of investigating the impact of the quantity of injected profiles on the phenomenon of prediction shift. Throughout this series of investigations, the fill size was consistently maintained at a level of 15%, a value that was hypothesized to have optimized the observed effect.

As depicted in the **Fig. 5** presented below, it is evident that Segment, Average, and Bandwagon attacks exhibit a slightly higher degree of success when compared to the random attack model. Typically, an observed prediction shift ranging from 0.3 to 0.85 is commonly observed, as shown in **Fig. 5**. Nevertheless, it is important to highlight that beyond the random attack model, there is a decrease in the prediction shift value as the attack size increases. The observed phenomenon can be ascribed to the proliferation of attack profiles exhibiting similarities, thereby forming a distinct cluster that may be distinct from the target profile for the attack. In the context of the random attack model, it is observed that due to the complete randomness in the generation of attack profiles, there is a possibility for the distribution of distinct attack profiles to occur across various clusters. In contrast, segment attacks possess a higher degree of influence due to their targeted nature, focusing on specific cohorts of users.

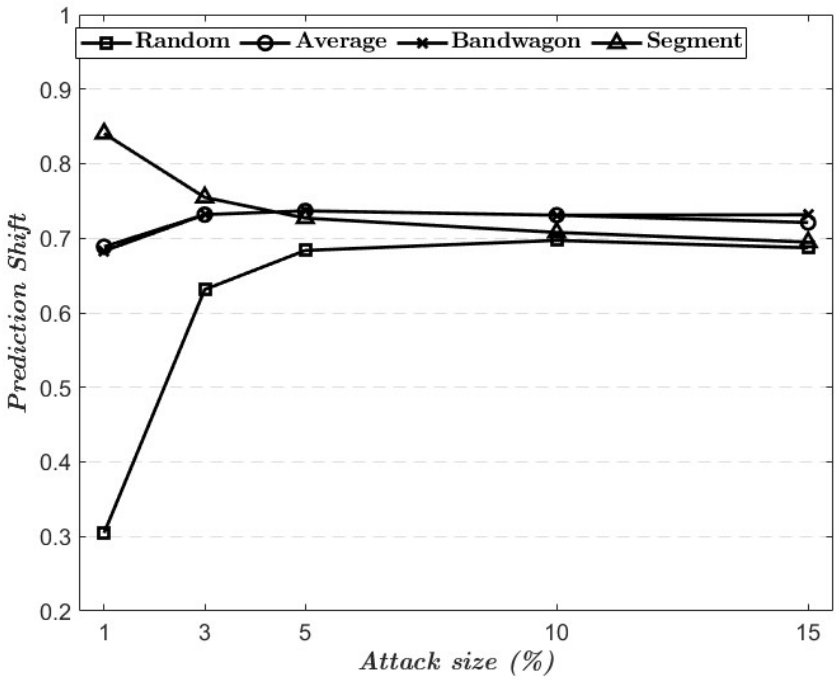

Fig. 5. Prediction shift for different sizes of attack (in push attack models)

The presented **Fig. 6** illustrates the obtained negative prediction shift values within the Reverse Bandwagon and love/hate attack models, spanning a range from -0.26 to -0.12. Based on the findings, it can be observed that the prediction shift values of nuke attack models are comparatively lower than those of push attack models.

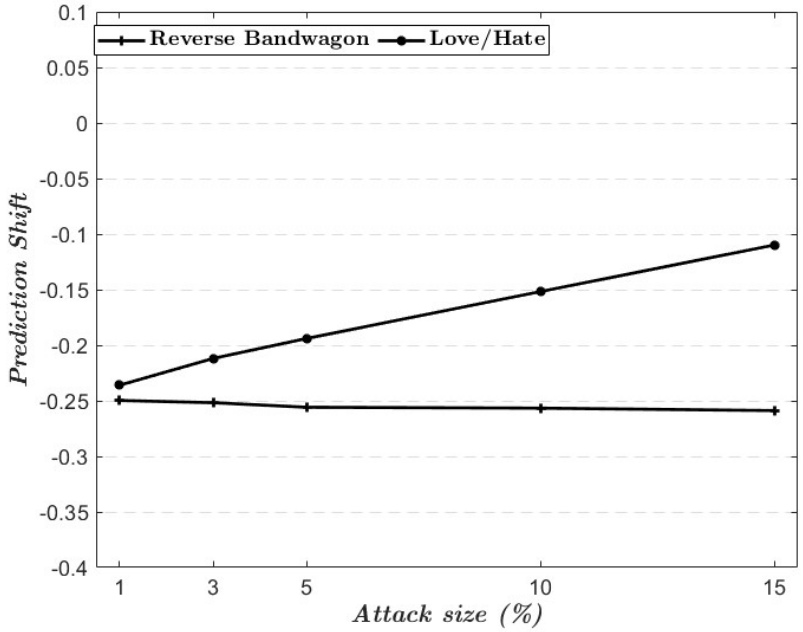

Fig. 6. Prediction shift for different sizes of attack (in nuke attack models)

5.4 Overall Comparison

In previous studies, Gunes et al. [28] have investigated the resilience of memory-based k-NN PPCF. Bilge et al. [29] evaluated the robustness of four model-based PPCF (DWT, k-means, SVD and item-based) schemes against shilling attack models. Bilge et al. [29] previously compared memory and model-based algorithms. In this study, the ant colony-based PPCF is added to the comparison table and shown in **Table 2**.

The findings from **Table 2** and other studies in the existing literature indicate that model-based schemes tend to exhibit greater robustness compared to memory-based schemes [12, 56]. The attack models of Average, Bandwagon, and Segment, which possess higher complexity and necessitate a deeper understanding of the system, exhibited more successful results compared to the remaining models. To execute the aforementioned attack models, it is imperative to possess certain system-related data, including average rate information and popular items. Based on the provided information, attack models are formulated.

According to the findings presented in **Table 2**, the SVD and item-based algorithms demonstrate the highest resilience against shilling attacks. The ant colony algorithm and the k-means algorithm, when employed in conjunction with the cluster method, exhibit comparable prediction shift values. When conducting a comparative analysis between the ant colony algorithm and k-means algorithm with respect to attack models, it can be observed that the former exhibits more robustness in the context of Random, Bandwagon, and segment attack models. In other attack models, the k-means scheme exhibits a relatively higher level of resilience. The ant colony clustering PPCF attack model that has been developed exhibits greater robustness compared to the conventional k-NN based memory-based algorithm.

Table 2. Prediction shift for memory, model and ant colony PPCF schemes

Algorithm type	***Shilling attacks***					
	Random	Average	Bandwagon	Segment	Reverse BW	Love/Hate
Memory-based PPCF						
k-NN	1.343	0.545	1.377	1.523	-1.753	-0.168
Model-based PPCF						
DWT	0.600	1.032	0.877	0.601	-0.562	-0.021
k-means	1.230	0.572	1.093	1.467	-0.298	-2.083
SVD	0.000	0.000	0.000	0.000	-0.001	-0.000
Item-based	0.018	0.021	0.018	0.080	-0.017	-0.018
Ant colony	**0.963**	**1.005**	**1.0052**	**1.060**	**-0.433**	**-0.358**

To explain the meaning of prediction shift, consider the following example: if a push attack has a prediction shift value of 1.5 and a user's prediction rate value for product q is 3, the resulting value will be 4.5 after the attack. The system will then recommend the product to the user, assuming that they will like it. For the 1-5 rating scale dataset used in this study, attacks with a prediction shift value above 1 can be considered successful. Typically, products with a prediction value above 4 are recommended to users. Due to the system manipulation by the attacker, the product that should not have been recommended was suggested to the user.

In addition to the **Table 2** above, various studies in the literature have examined robustness. However, due to differences in attack models, datasets, robustness metrics, and privacy methods used in these studies, they could not be included in the table. Nonetheless, some of the findings from these studies are mentioned in the following paragraph. Yılmazel and Kaleli [62] discussed the challenges of producing accurate recommendations for customers with

arbitrarily distributed rating preferences and proposes methods for enabling data holders' collaboration while protecting privacy. The authors analyzed the robustness of proposed arbitrarily distributed data-based recommendation methods against well-known shilling attack types. In their study, the authors used the same data set and attack models as in this study. However, the data was stored arbitrarily distributed instead of centralized. The authors reported results ranging from 0.2 to 1.5 for push attacks and from -0.2 to -1.7 for nuke attacks for the six attack models. The authors concluded that recommendation methods are vulnerable to shilling attacks, even with privacy protection. Turk and Bilge [63] examined the robustness of multi-criteria collaborative filtering (MCCF) algorithms against shilling attacks. The authors discussed the vulnerabilities of these algorithms and proposes alternative attacking schemes. The study indicated that multi-criteria collaborative filtering (MCCF) algorithms are highly vulnerable to manipulations, as demonstrated by empirical results on real-world data. The authors utilized a multi-criteria preference dataset with a rating scale of 1-13, crawled on the Yahoo!Movies platform. In the dataset, users have an overall rating as well as multi-criteria ratings on four sub-aspects of the film domain: acting, directing, visuals and story. The authors reported results ranging from 0.0 to 3.0 for push attacks and from 0.0 to -4.5 for nuke attacks for the six attack models. Alonso et al. [64] presented a robust model-based reliability approach to address shilling attacks in collaborative filtering recommender systems. It introduced a method based on matrix factorization to obtain reliability values for user-item predictions, aiming to neutralize shilling attacks. The method was tested through experiments, demonstrating its effectiveness in neutralizing shilling attacks, particularly on sparse datasets.

6. Conclusion and Future Work

CF algorithms are widely utilized in diverse domains, with a specific focus on e-commerce platforms. Recommendation programs offer advantages to both consumers and e-commerce vendors. In addition to their inherent advantages, collaborative filtering methodologies pose several challenges. The main challenges associated with these schemes pertain to the privacy protection, susceptibility to shilling attacks and scalability problem.

Model-based recommender systems are generally regarded as more advantageous compared to memory-based approaches because of their improved efficiency in online settings. There exist prediction systems that currently prioritize the preservation of privacy and utilize model-based methodologies to effectively generate recommendations while maintaining the privacy of consumers. This study utilizes four push and two nuke shilling attack strategies to assess the efficacy of the ant-colony privacy-preserving collaborative filtering model. These methodologies utilize concealed data to execute altered iterations of random, average, segment, bandwagon, reverse bandwagon, and love/hate shilling attacks. In addition, a series of empirical experiments were conducted to assess the resilience of these prediction techniques against the six attack models.

Based on the empirical findings, it can be observed that the segment, average, and bandwagon models exhibit higher levels of effectiveness compared to other models, as they specifically target particular groups. The models used to simulate nuke attacks did not demonstrate significant impact. Consequently, the push attack models yielded prediction shift values ranging from 0.3 to 1.3. Based on the observed values, it can be concluded that the ant-colony based PPCF scheme exhibits susceptibility to shilling attacks.

This study utilizes a database that stores data in an obscured manner, rendering the creation of direct attack models challenging even if the data is disguised. PPCF algorithms offer greater benefits compared to CF algorithms when it comes to defending against attacks involving data

hiding. The limitation of the developed system is the need for predetermination of the number of clusters. In future studies, the algorithm could automatically determine the number of clusters based on the data size. Currently, there are many open-source recommendation systems available. Our method can be applied to these systems by making necessary code changes. Additionally, this system could be made publicly available as a service by developing an interface. Users can upload their datasets and calculate predictions for desired users and items.

Given that attack profiles are generated through a specific algorithmic process, it is probable that they will exhibit certain similarities. In the subsequent phase of the study, the objective is to develop a plugin that leverages this inherent similarity and groups the attack profiles into a single cluster, thereby isolating them from the system. Consequently, these fictitious profiles will be automatically identified and removed from the system independently of the attack models, preventing them from manipulating the system. It is of the utmost importance to ensure that genuine profiles are not removed from the system, as this would result in a reduction in the scheme's overall accuracy.

This study investigated the robustness of the developed scheme, however, in future work, we will investigate and compare the accuracy and scalability of ant colony clustering model based PPCF schemes, focusing on their accuracy and scalability. Additionally, there are plans to develop new model-based PPCF schemes that are more resilient to these attack models.

References

[1] J. Ben Schafer, J. A. Konstan, and J. Riedl, "E-commerce recommendation applications," *Data Mining and Knowledge Discovery*, vol. 5, no. 1–2, pp. 115-153, 2001. Article (CrossRef Link)

[2] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 5–53, 2004. Article (CrossRef Link)

[3] M. Grcar, "User profiling: Collaborative filtering," in *Proc. of SIKDD 2004 at Multiconference IS*, pp. 75–78, 2004.

[4] J. Ben Schafer, D. Frankowski, J. Herlocker, and S. Sen, "Collaborative filtering recommender systems," in *The Adaptive Web: Methods and Strategies of Web Personalization*, pp. 291–324, 2007. Article (CrossRef Link)

[5] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowledge-Based System*, vol. 46, pp. 109–132, 2013. Article (CrossRef Link)

[6] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in Artificial Intelligence*, vol. 2009, 2009. Article (CrossRef Link)

[7] B. M. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering," in *Proc. of The Fifth International Conference on Computer and Information Technology*, pp. 291–324, 2002.

[8] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Analysis of recommendation algorithms for e-commerce," in *Proc. of The 2nd ACM Conference on Electronic Commerce*, pp. 158–167, 2000.

[9] G. Nayak and S. Devi, "A survey on privacy preserving data mining: approaches and techniques," *International Journal of Engineering Science and Technology*, vol. 3, no. 3, pp. 2127–2133, 2011.

[10] R. Agrawal and R. Srikant, "Privacy-preserving data mining," in *Proc. of the 2000 ACM SIGMOD International Conference on Management of Data*, pp. 439–450, 2000. Article (CrossRef Link)

[11] H. Polat and W. Du, "Achieving private recommendations using randomized response techniques," in *Proc. of International Conference on Knowledge Discovery and Data Mining*, pp. 637-646, 2006. Article (CrossRef Link)

[12] J. J. Sandvig, B. Mobasher, and R. D. Burke, "A survey of collaborative recommendation and the robustness of model-based algorithms.," *IEEE Data Eng. Bull.*, vol. 31, no. 2, pp. 3–13, 2008.
[13] C. A. Williams, B. Mobasher, and R. Burke, "Defending recommender systems: detection of profile injection attacks," *Service Oriented Computing and Applications*, vol. 1, pp. 157–170, 2007. Article (CrossRef Link)
[14] P.-A. Chirita, W. Nejdl, and C. Zamfir, "Preventing shilling attacks in online recommender systems," in *Proc. of The 7th Annual ACM International Workshop on Web Information and Data Management*, pp. 67–74, 2005. Article (CrossRef Link)
[15] R. Burke, B. Mobasher, C. Williams, and R. Bhaumik, "Classification features for attack detection in collaborative recommender systems," in *Proc. of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 542–547, 2006. Article (CrossRef Link)
[16] J. Canny, "Collaborative filtering with privacy," in *Proc. of 2002 IEEE Symposium on Security and Privacy*, IEEE, pp. 45–57, 2002. Article (CrossRef Link)
[17] H. Polat and W. Du, "Privacy-preserving collaborative filtering using randomized perturbation techniques," in *Proc. of Third IEEE International Conference on Data Mining*, IEEE, pp. 625–628, 2003. Article (CrossRef Link)
[18] R. Burke, B. Mobasher, R. Bhaumik, and C. Williams, "Collaborative recommendation vulnerability to focused bias injection attacks," in *Proc. of International Conference on Data Mining: Workshop on Privacy and Security Aspects of Data Mining*, 2005.
[19] M. P. O'Mahony, N. J. Hurley, and G. C. M. Silvestre, "Promoting recommendations: An attack on collaborative filtering," in *Proc. of the 13th International Conference on Database and Expert Systems Applications*, pp. 494-503, 2002. Article (CrossRef Link)
[20] M. P. O'Mahony, N. J. Hurley, and G. C. M. Silvestre, "Recommender systems: Attack types and strategies," in *Proc. of the 20th National Conference on Artificial Intelligence*, pp. 334–339, 2005. Article (CrossRef Link)
[21] C. A. Williams, B. Mobasher, and R. Burke, "Defending recommender systems: detection of profile injection attacks," *Service Oriented Computing and Applications*, vol. 1, pp. 157–170, 2007. Article (CrossRef Link)
[22] R. Burke, B. Mobasher, C. Williams, and R. Bhaumik, "Detecting profile injection attacks in collaborative recommender systems," in *Proc. of the 8th IEEE International Conference on E-Commerce Technology and the 3rd IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services*, pp. 23-23, 2006. Article (CrossRef Link)
[23] I. Gunes and H. Polat, "Detecting shilling attacks in private environments," *Information Retrieval Journal*, vol. 19, pp. 547–572, 2016. Article (CrossRef Link)
[24] M. P. O'Mahony, N. J. Hurley, and G. C. M. Silvestre, "Towards robust collaborative filtering," in *Proc. of AICS: Irish Conference on Artificial Intelligence and Cognitive Science*, pp. 87-94, 2002. Article (CrossRef Link)
[25] B. Mehta, T. Hofmann, and W. Nejdl, "Robust collaborative filtering," in *Proc. of the 2007 ACM Conference on Recommender systems*, pp. 49–56, 2007. Article (CrossRef Link)
[26] R. Burke, M. P. O'Mahony, and N. J. Hurley, "Robust collaborative recommendation," in *Recommender Systems Handbook*, 2015, pp. 961–995. Article (CrossRef Link)
[27] M. O'Mahony, N. Hurley, N. Kushmerick, and G. Silvestre, "Collaborative recommendation: A robustness analysis," *ACM Transactions on Internet Technology*, vol. 4, no. 4, pp. 344–377, 2004. Article (CrossRef Link)
[28] I. Gunes, A. Bilge, and H. Polat, "Shilling Attacks Against Memory-Based Privacy-Preserving Recommendation Algorithms," *KSII Transactions on Internet & Information Systems*, vol. 7, no. 5, pp. 1272-1290, 2013. Article (CrossRef Link)
[29] A. Bilge, I. Gunes, and H. Polat, "Robustness analysis of privacy-preserving model-based recommendation schemes," *Expert Systems with Applications*, vol. 41, no. 8, pp. 3671–3681, 2014. Article (CrossRef Link)

[30] J. Canny, "Collaborative filtering with privacy via factor analysis," in *Proc. of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 238–245, 2002. Article (CrossRef Link)

[31] H. Polat and W. Du, "SVD-based collaborative filtering with privacy," in *Proc. of the 2005 ACM Symposium on Applied Computing*, pp. 791–795, 2005. Article (CrossRef Link)

[32] A. Bilge and H. Polat, "A comparison of clustering-based privacy-preserving collaborative filtering schemes," *Applied Soft Computing*, vol. 13, no. 5, pp. 2478–2489, 2013. Article (CrossRef Link)

[33] J. Luo, X. Yi, F. Han, X. Yang, and X. Yang, "An Efficient Clustering-Based Privacy-Preserving Recommender System," in *Proc. of International Conference on Network and System Security*, pp. 387–405, 2022. Article (CrossRef Link)

[34] A. Hegde, H. Möllering, T. Schneider, and H. Yalame, "Sok: Efficient privacy-preserving clustering," in *Proc. of on Privacy Enhancing Technologies*, pp. 225-248, 2021. Article (CrossRef Link)

[35] F. O. Catak, I. Aydin, O. Elezaj, and S. Yildirim-Yayilgan, "Practical implementation of privacy preserving clustering methods using a partially homomorphic encryption algorithm," *Electronics*, vol. 9, no. 2, pp. 229-249, 2020. Article (CrossRef Link)

[36] C. Dellarocas, "Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior," in *Proc of the 2nd ACM Conference on Electronic Commerce*, pp. 150–157, 2000. Article (CrossRef Link)

[37] B. Mobasher, R. Burke, R. Bhaumik, and J. J. Sandvig, "Attacks and remedies in collaborative recommendation," *IEEE Intelligent Systems*, vol. 22, no. 3, pp. 56–63, 2007. Article (CrossRef Link)

[38] B. Mobasher, R. Burke, R. Bhaumik, and C. Williams, "Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness," *ACM Transactions on Internet Technology*, vol. 7, no. 4, pp. 23-es, 2007. Article (CrossRef Link)

[39] R. Burke, B. Mobasher, R. Zabicki, and R. Bhaumik, "Identifying attack models for secure recommendation," in *Proc. of the Beyond Personalization 2005 Workshop, Int. Conference on Intelligent User Interfaces*, pp. 19-25, 2005.

[40] F. Zhang, "A survey of shilling attacks in collaborative filtering recommender systems," in *Proc. of 2009 International Conference on Computational Intelligence and Software Engineering*, pp. 1–4, 2009. Article (CrossRef Link)

[41] I. Gunes, C. Kaleli, A. Bilge, and H. Polat, "Shilling attacks against recommender systems: a comprehensive survey," *Artificial Intelligence Review*, vol. 42, pp. 767–799, 2014. Article (CrossRef Link)

[42] M. Si and Q. Li, "Shilling attacks against collaborative recommender systems: a review," *Artificial Intelligence Review*, vol. 53, pp. 291–319, 2020. Article (CrossRef Link)

[43] T. T. Kaya and C. Kaleli, "Robustness Analysis of Multi-Criteria Top-n Collaborative Recommender System," *Arabian Jorunal of Science and Engineering*, vol. 48, no. 8, pp. 10189–10212, 2023. Article (CrossRef Link)

[44] Y. Wu, Y. Du, and L. Li, "A research of collaborative filtering recommendation based on ant colony algorithm," in *Proc. of 2011 International Conference on Uncertainty Reasoning and Knowledge Engineering*, pp. 58–61, 2011. Article (CrossRef Link)

[45] S. Nadi, M. H. Saraee, A. Bagheri, and M. Davarpanh Jazi, "FARS: Fuzzy ant based recommender system for web users," *International Journal of Computer Science Issues*, vol. 8, no. 1, pp. 203-209, 2011.

[46] X. Liao, H. Wu, and Y. Wang, "Ant collaborative filtering addressing sparsity and temporal effects," *IEEE Access*, vol. 8, pp. 32783–32791, 2020. Article (CrossRef Link)

[47] J. Sobecki and J. M. Tomczak, "Student courses recommendation using ant colony optimization," in *Proc. of Asian Conference on Intelligent Information and Database Systems*, pp. 124–133, 2010. Article (CrossRef Link)

[48] P. Bedi and R. Sharma, "Trust based recommender system using ant colony for trust computation," *Expert Systems with Applications*, vol. 39, no. 1, pp. 1183–1190, 2012. Article (CrossRef Link)
[49] Z. Batmaz and C. Kaleli, "Methods of privacy preserving in collaborative filtering," in *Proc. of 2017 International Conference on Computer Science and Engineering*, pp. 261–266, 2017. Article (CrossRef Link)
[50] M. Asad, S. Shaukat, E. Javanmardi, J. Nakazato, and M. Tsukada, "A comprehensive survey on privacy-preserving techniques in federated recommendation systems," *Applied Sciences*, vol. 13, no. 10, p. 6201, 2023. Article (CrossRef Link)
[51] D. Pramod, "Privacy-preserving techniques in recommender systems: state-of-the-art review and future research agenda," *Data Technologies and Applications*, vol. 57, no. 1, pp. 32–55, 2023. Article (CrossRef Link)
[52] R. Agrawal and R. Srikant, "Privacy-preserving data mining," in *Proc. of the 2000 ACM SIGMOD International Conference on Management of Data*, pp. 439–450, 2000. Article (CrossRef Link)
[53] P. S. Shelokar, V. K. Jayaraman, and B. D. Kulkarni, "An ant colony approach for clustering," *Analytica Chimica Acta*, vol. 509, no. 2, pp. 187–195, 2004. Article (CrossRef Link)
[54] H. Polat and W. Du, "Privacy-preserving collaborative filtering," *International journal of electronic commerce*, vol. 9, no. 4, pp. 9–35, 2005. Article (CrossRef Link)
[55] M. Steinbach, G. Karypis, and V. Kumar, "A comparison of document clustering techniques," in *Proc. of KDD Workshop on Text Mining*, pp. 525-526, 2000. Article (CrossRef Link)
[56] B. Mobasher, R. Burke, and J. J. Sandvig, "Model-based collaborative filtering as a defense against profile injection attacks," in *Proc of the 21st National Conference on Artificial Intelligence*, pp. 1388-1393, 2006. Article (CrossRef Link)
[57] R. Bhaumik, C. Williams, B. Mobasher, and R. Burke, "Securing collaborative filtering against malicious attacks through anomaly detection," in *Proc. of the 4th Workshop on Intelligent Techniques for Web Personalization*, 2006.
[58] S. K. Lam and J. Riedl, "Shilling recommender systems for fun and profit," in *Proc. of the 13th International Conference on World Wide Web*, pp. 393–402, 2004. Article (CrossRef Link)
[59] R. Wei, H. Tian, and H. Shen, "Improving k-anonymity based privacy preservation for collaborative filtering," *Computers & Electrical Engineering*, vol. 67, pp. 509–519, 2018. Article (CrossRef Link)
[60] J. Deng, J. Guo, and Y. Wang, "A Novel K-medoids clustering recommendation algorithm based on probability distribution for collaborative filtering," *Knowledge-Based Systems*, vol. 175, pp. 96–106, 2019. Article (CrossRef Link)
[61] A. Bilge and H. Polat, "An improved privacy-preserving DWT-based collaborative filtering scheme," *Expert Systems with Applications*, vol. 39, no. 3, pp. 3841–3854, 2012. Article (CrossRef Link)
[62] B. Y. Yilmazel and C. Kaleli, "Robustness analysis of arbitrarily distributed data-based recommendation methods," *Expert Systems with Applications*, vol. 44, pp. 217–229, 2016. Article (CrossRef Link)
[63] A. M. Turk and A. Bilge, "Robustness analysis of multi-criteria collaborative filtering algorithms against shilling attacks," *Expert Systems with Applications*, vol. 115, pp. 386–402, 2019. Article (CrossRef Link)
[64] S. Alonso, J. Bobadilla, F. Ortega, and R. Moya, "Robust model-based reliability approach to tackle shilling attacks in collaborative filtering recommender systems," *IEEE access*, vol. 7, pp. 41782–41798, 2019. Article (CrossRef Link)

İhsan GÜNEŞ completed his undergraduate education in Kocaeli University Computer Engineering Department in 2001. Güneş then completed his master's and doctoral studies at Anadolu University Computer Engineering Department in 2005 and 2015, respectively. He is an Assistant Professor at the Department of Computer Technologies, Eskisehir Technical University, Eskisehir, Turkey. His current research interests include recommendation systems, data mining, learning analytics.