Available online at www.ijpe-online.com

International Journal of Performability Engineering

vol. 19, no. 5, May 2023, pp. 291-302
DOI: 10.23940/ijpe.23.05.p1.291302

ShAD-SEF: An Efficient Model for Shilling Attack Detection using Stacking Ensemble Framework in Recommender Systems

Nittu Gotham[a,*], Karan Singh[a], Latha Banda[b], Purushottam Sharma[c], Chaman Verma[d], and S. B. Goyal[e]

[a]*School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, India*
[b]*Department of Computer Science and Engineering, ABES Engineering College, Ghaziabad, India*
[c]*Amity School of Engineering and Technology, Amity University, Noida, India*
[d]*Department of Media and Educational Informatics, Faculty of Informatics, Eötvös Loránd University, Budapest, Hungary*
[e]*Faculty of Information Technology, City University, Petaling Jaya, Malaysia*

Abstract

Recommender Systems helps users to find suitable products from massively available data on the internet. The most broadly applied recommendation method is collaborative filtering, which can also be subject to shilling attacks. Profile injection occurs when malicious users insert a few bogus profiles into the user-item rating database, which alters the result of the recommendation. In this paper, the shilling attack is simulated: a Random attack, segment attack, average attack, and bandwagon attack on the movie lens dataset, focusing on users with similar interests. To build trust in the system, fake profiles must be detected. Accuracy, attack size, and filler size computations were done for each model. Several machine learning algorithms are in use to classify these fake and original profiles. Here, four Machine Learning algorithms are compared and the most efficient models are KNN, random forest, and xgboost. To get more accuracy, the ensemble model used logistic regression as a meta classifier which is more accurate than individual machine learning algorithms. Our proposed model, which is stacking an ensemble model using logistic regression as a meta-classifier, will give the best accuracy in any case.

Keywords: attack models; collaborative filtering; ensemble model; recommender system (RS); shilling attack

1. Introduction

With the growing Recommender Systems (RS) information, finding the correct information on social networking sites (SNS) is challenging. The RS produces much information with the extent of customer satisfaction but needs some accuracy with personalized, relevant data. Specialized recommender systems that can filter information based on user preferences have been designed to address this problem. Users can explore new groups and organizations using the personalized recommendations features on Facebook and Instagram. Examples include recommendations for you, acquaintances you might have, and groups you ought to join. RS is used in many fields like music, movies, library recommendation, etc. It sorts the information according to the user's interest and suggests related items. Many e-commerce websites, such as Flip kart, Amazon, and Myntra, use RS technology [1] to filter items according to customer interest.

By contrasting the content of the product and the consumer profile, content-based systems make product recommendations to customers based on their prior purchasing behavior [2]. However, it suffers from scalability problems. The basic premise of collaborative filtering (CF) is that two similar users having similar interests in the future will be shared by those with similar likes in the past. Sometimes CF, also called social filtering based on the rating given by different similar users' recommendations, is generated. Amazon uses item-item CF [3]. It suffers from a cold star, shilling attack.

CF suffers from profile injection problems due to its openness. The motto of malicious users is to create a fake profile using tools or manually inserting it into the system's database. The negative user rates the products so that the promoted items are rated higher to promote the item (push attack) or the target item is rated lower to demote the item (nuke attack). This paper provides a stacking ensemble framework-based technique for shilling attack detection. Initially, we compared several

* Corresponding author.
E-mail address: puru.mit2002@gmail.com

supervised learning algorithms to detect fake profiles then we built our ensemble model. User-item rating matrix consists of genuine user profiles (Alex to Bob) and fake profiles (A1 to A3) injected by malicious users. There is a high chance of promoting his targeted item. Here, the attacker recommends movie 6 to Alex irrespective of his taste which leads to falling down the system accuracy as shown in Table 1.

Table 1. An example of a push attack and target product is Cinema 6

items / users	movie 1	movie 2	movie 3	movie 4	movie 5	movie 6	Correlation w.r.t Alex
Alex	5	2	3	3		?	
Bob	2		4		4	1	-1.00
Cavin	3	1	3		1	2	0.76
Dolly	4	3		3	3	2	0.94
Elias	3	3	2	1	3	1	0.21
Fenn		3		1	2		-1.00
Gia	4	2	3	1		1	0.72
harry		5		1	5	1	-1.00
A1	5		3		2	5	1.00
A2	5	1	4		2	5	0.89
A3	5	2	2	2		5	0.93
Correlation w.r.t Movie 6	0.85	-0.55	0.00	0.48	-0.59	?	?

There are many advantages and disadvantages of RS that robustness may achieve, which may be suitable for mobile applications. The most popular health applications use the RS technique, which has disadvantages like detecting malicious users, being time-consuming, and having some issues in preserving data. More details of the pros and cons of RS are shown in Table 2.

Table 2. Advantages and Disadvantages of Recommendation Systems and Applications

Advantages	Disadvantages	Applications
Good robustness in recommending accurate mobile applications with preserving the privacy	Detecting internal malicious users is issued	In Mobile Application.
Low latency and learner security are achieved via the fog computing approach, which is applied in the (Fog-Based Recommender System) FBRS.	Construction of the class and its associated subclass takes time, and the system setup is costly and time-consuming.	Fog based E-Learning
Preserving privacy in online medical recommendations for the e-Healthcare system by which users can find a suitable doctor.	No accuracy in preserving data	Online medical service
Estimates a user's rating based on their nearby neighbors and previous activity.	Data sparsity	Multimedia social network
We can find shilling attacks in the database, such as push and nuke attacks.	If the size of the data is tiny, we cannot find the shilling attack	Movielens rating
Unsupervised Machine learning like DSA-AURB and boosted decision trees take significantly less computation time to detect fake profiles	Only saw push attacks and DSA-AURB models are not good when the dataset is minimal.	YouTube video statistics

The rest of the paper is organized as follows: Related work is covered in part 2, and shilling attack and types are discussed in section 3. Section 4 discusses problem formulation. The proposed model is presented in Section 5 and the experimental findings are shown in Section 6. Finally, the conclusion is explored in Section 7.

2. Related work

Different researchers have developed various machine-learning techniques to improve the quality of recommendations on social networking sites. “Ridel and Lam” initially proposed two attack types: Random Bot and Average Bot attack [4]. They explained how the system recommends products after injecting the fake profile into the system. There are plenty of researchers who are working in this zone of research. The authors provide methods to detect a shilling attack from the system. All methods have different approaches such as supervised, semi-supervised, unsupervised, statistical methods, and some variable-based methods. [5-8].

``

Most researchers concluded that the average attack significantly impacts user-based collaborative techniques more than item-based collaborative techniques. This requires more system knowledge and rating pattern knowledge than segment attacks [9]. According to the experimental findings, segment attacks affect "User-based collaborative technique" more than "item-based collaborative technique." A novel attack detection method using Bisecting k-mean clustering was proposed in [10]. Sometimes ratings may be binary, such as the YouTube like/dislike feature, but these are also vulnerable to shilling attacks. Bilge et al. [11] used a rule-based approach, generic, and model-based attributes to identify fake profiles to remove harmful profiles injected into the system using a robust method based on the reliability concept proposed [12]. Matrix factorization was used to obtain each user's prediction of an item connected with a reliability value. To improve the low precision problem in some existing supervised learning models, [13] proposed a model to identify fake profiles using neural network backpropagation and ensemble framework. A comparative survey on profile injection attacks [14,15] was submitted using various profile injection attack detection algorithms. The movie lens 100K dataset is the benchmark for different shilling attack model detection. "Shilling attack detection" based on text Convolution Neural Network (CNN) was proposed in [16]. Deep learning techniques may detect shilling attacks without having feature extraction like any generic or model-specified attributes [17]. Unsupervised machine learning algorithms [18] such as CBS, DSA-AURB, PCA-VarSelect, and UD-HMM have different attack sizes like 3%, 5%, 8%, 10%, and fixed filler sizes of 3%. The experimental results say that the DSA-AURB system can identify shilling attacks more accurately than another unsupervised algorithm. In [19], they proposed a classification technique to detect profile injection attacks using detection attributes and type-specified attributes like Filler Mean-Variance (FMV), Filler Mean Different (FMD), FAC, and TFM. Performance evaluation is calculated in sensitivity and specificity, where sensitivity for SVM and C4.5 are good at identifying fake profiles. Still, kNN classification has a problem identifying when the filler size is small. Here is the comparison of different approaches given in Table 3.

Table 3 Comparison of different approaches

Method	**Domain type**	**Advantages**	**Disadvantages**	**Measuring-Parameters**
Bisecting k means clustering [20]	Collaborative Recommender Systems	Detects fake profiles using the child correlation value of the two sub-class is the same as that of the parent cluster.	The dataset shows all genuine profiles as fake if there is no bogus profile in the dataset.	Attack and filler sizes should be between 3% and 25%.
Rule-based approach and generic attributes [11]	Collaborative filtering techniques	By Calculating system-specified attributes to detect malicious profiles in binary rating such as like or dislike	Used only on binary rating data	attack size 1%
Matrix factorization [21]	Collaborative filtering	To remove harmful profiles that have been injected into the system using a robust method based on the reliability concept	instead of identifying bogus profiles, it finds doubtful prediction-making to the final recommendation.	Precision, recall, attack size, filler size
Using neural network backpropagation and ensemble framework [22]	Recommender systems	Detect bogus profiles in the first stage of EMD training. Validation data is generated in the following stage and base classification on BP is produced on training. In the final stage, the ensemble algorithm is applied	They identified limited attacks like bandwagon attacks, average attacks, and random attacks.	Precision, recall

In this, the description of the method and domain is explained, and the advantages and disadvantages of every technique are tabularized with the target of metrics. In some papers, the authors show the mark of precision and recall using machine learning techniques, whereas, in other articles, the target achievements were attack size and filler size. The research contribution of researchers in Recommendation Systems with parameters is shown in Table 4.

The parameters defined by researchers were attack size, prediction shift, average prediction shift, filler Size, etc., and are detailed here in this article, along with the most current contribution to recommendation systems. Among others, the detection attacks Push attack, Bandwagon random, Bandwagon average, and love/hate were employed on the platforms of WEKA, Python, and MATLAB. Most of the RS techniques were implemented with the dataset of Movielens.com.

Table 4. Recent research contributions Recommendation Systems

S.No	Techniques	Parameters	Detection Attack	Platform	Dataset
1	User-based, item-based collaborative algorithm[9]	Attack size (between 5% to 25%), filler size (const=50%), prediction shift, avg prediction shift	Push attack	WEKA, Python	Movie lens-100K
2	KNN and SVM[23]	Attack size (const=1%), filler Size (between 3% to 100%)	"Random, average, bandwagon random, bandwagon average, love/hate."	MATLAB, R2017b	Movielens public
3	Re-scale AdaBoost [24]	Attack size (between 1.1% to 27.6%), filler size (between 1.2% to 16.4%), classification error, detection rate, false alarm rate	Bandwagon, reverse bandwagon, segment	MATLAB, R2014a, WEKA	Movie Lens-100K
4	PCA [25]	Attack size (const =5%), filler Size (between 1% to 60%), precision, F-measure	Average, random bandwagon	MATLAB	Movielens
6	CNN [26]	Attack size, filler size, precision, recall	Random, average, bandwagon, segment	Python	Movielens
7	Ensemble technique (SVM+RF)[27]	Attack size (const=5% and10%), filler size (between1% to 50%), precision, recall, k-fold	Push attack	WEKA, Python	Moivielens-100K

3. Shilling Attack

An attacker creates an attack profile so that the target product should be recommended to the user. Figure 1 illustrates the construction of a generic shilling attack model. Each attack profile is made up of 4 sets of gathered items: the target items I_T, the collection of chosen items I_S, the set of not rated goods I_N, and the group of filler items selected at random I_F.

Set	I_S			I_F			I_N			I_T
Items	i_1^s	…	i_k^s	i_1^F	…	i_l^F	i_1^N	…	i_v^N	i_t
Rating	$\delta(i_1^s)$	…	$\delta(i_k^s)$	$\alpha(i_1^F)$	…	$\alpha(i_l^F)$	null	null	null	$\gamma(i_t)$

Rating given for (1 to k) selected items

Rating given for (1 to l) filler items

Unrated items for 1 to v in attack profile

Rating given for target item

Figure 1. A fundamental profile attack structure of ion Shilling attack

3.1. Features of Attack

The size of the attack, the level of expertise needed to mount the attack, the attacker's intention—whether to push or nuke—the product, as well as the number of customer ratings determine the type of attack. The following are the attack characteristics that an attacker must be aware of before developing an attack profile to skew the system's results.

1) **Attacker's Aim:** Initially, the attacker's intention should be clear whether to promote the product (push) by giving a maximum rating or to demote (nuke) the effect by providing a minimum rating.
2) **size of the attack:** The system is programmed to build various false profiles, and these profiles rate different products by assigning the highest possible rating to the intended target—the more the attack size, the more significant the chance to reach the target product to more users.
3) **Filler size:** The no. of items that must be rated by an attack profile surrounding the item means for it to appear to be a genuine profile.
4) **Knowledge level:** Knowledge about the system is an essential feature for an attacker. Sometimes attackers require common knowledge about the system for a low attack profile and to make a high attack profile, the attacker requires more knowledge.

``

3.2. Attack Models

a) Random attack
Items for filler are selected randomly, and the target item receives the max rating (push) or the min rating (nuke). It necessitates a minimal amount of system and product rating knowledge. Product ratings are given based on the mean of an inclusive method to promote or demote products. It could be more effective; it hinders system functioning rather than advancing the goods. The properties of push assault models are shown in Table 2.

b) Average attack
The items are chosen the same way as in the random attack model, but in the average attack, individual average ratings for each user product are used rather than the system mean. To assign ratings for filler items, one must understand the system database and how ratings are allocated for products. It is difficult to differentiate between a fraudulent user and a legitimate user profile. The attacker's ability to design a compelling attack profile depends on the model's effectiveness, as illustrated in Table 5.

c) Bandwagon attack
In this model, the selected item consists of popular products assigning high ratings to get many similarities between the malicious user and original user profiles. A low level of knowledge is required. If the filler set is selected the same as in a random attack, it is considered a bandwagon random attack. Bandwagon average attack is referred to if the filler set is chosen in the same manner as in an average attack.

Table 5. Characteristics of push attack models

Characteristics		Random attack	Average attack	Bandwagon attack	Segment attack
I_S	Item	Empty	Empty	Popular items	Favored by segment users
	Rating	Empty	Empty	r_{max}/r_{min}	r_{max}/r_{min}
I_F	Item	Selected Randomly	Selected Randomly	Selected Randomly	Selected Randomly
	Rating	Overall system means	Item avg mean	Overall system means	r_{max}/r_{min}
I_N	Item	Empty	Empty	Empty	Empty
	Rating				
I_T	Item	r_{max}/r_{min}	r_{max}/r_{min}	r_{max}/r_{min}	r_{max}/r_{min}
	Rating				
Attacker intent		Promote/demote an item	Promote/demote an item	Promote/demote an item	Promote/demote an item
Knowledge required		Low	high	low	Low-moderate
Effective		In user-based, item based	In user-based	In user-based	In item based
Cost		Low	high	low	low

d) Segment attack
It targets similar taste users likely to purchase the target product. It is used in item-based collaborative filtering techniques because it emphasizes characteristics of the product. The attack profile looks genuine by selecting a group of similar taste users in select items along with the target item. A maximum rating is given, and filler item products are rated with a minimum rating.

4. Problem Formulation

It is a challenging task to select a suitable product from millions of available products on the internet. CF technique helps the user to choose the product according to individual tastes. CF built the recommendation based on the commonality between users and products. Due to openness in nature, the system is susceptible to shilling attacks. Malicious users create fake profiles and insert them into the system, which may alter the product recommendation leading to a decrease in accuracy and loss of trust in the system. Fake profiles are created using manual or automatic tools, and these profiles may not get separated from the database because of their veracity. As a result, the system recommends products unrelated to the user's interests. We focus on “profile injection attacks” which is when malicious users are injected into the system without pre-knowledge about the system and rating pattern. They are two categories of attack: push attack, which aims to promote the target product to the user, and nuke attack, which aims to demote the target product. The stability and durability of the attack in the system are measured in filler and attack ratios. The attacker succeeds if the targeted product is promoted (push) to the user without prior knowledge and is cost-effective. The main objective of the study is to detect the shilling attack so that the accuracy of the good recommendation increases and builds trust in the system. The proposed work objectives are as follows:

1) To check the system’s robustness before and after inserting a fake profile.
2) To check how fake profiles behave on attack models like push and nuke attacks.

3) To develop an ensemble framework using different algorithms using voting methods to detect shilling attacks.
4) To check how the system behaves on different filler and attack ratios in detecting different attack models.
5) To test the classifier's performance used to detect the shilling attack in terms of "Recall, Precision, and F-measure".
6) To study and compare the existing techniques with the proposed framework.

5. Proposed Model

Figure 2 depicts the workflow for the project we've proposed. Using classification techniques, the following processes identify bogus profiles from user profiles stored in the system's database.

1) **Dataset selection:** The dataset collection is from Movielens.com, in which a dataset of 100k is used for testing the supervised learning algorithm.
2) **Data preprocessing:** We examined the dataset by taking the complete values and filling in the missing values; this dataset is preprocessed in accordance with our experimental needs.
3) **Generate detection attributes:** The training set comprises the detection attributes, including user id, detection attributes, and the label of genuine or fake profiles.
4) **Classification:** Actual and Fake profiles are generated using various supervised learning techniques such as KNN(K Nearest Neighbor), RF (Random Forest), NB (Naïve Bayesian), and XGBoost (eXtream Gradient Boosting)
5) **Identify the attack type:** The attack is based on the value assigned for targeted items.
6) **Evaluation of performance:** After classification, the classifiers' efficiency is computed in terms of Precision, Recall & F-measure, then validation is done using cross-validation.

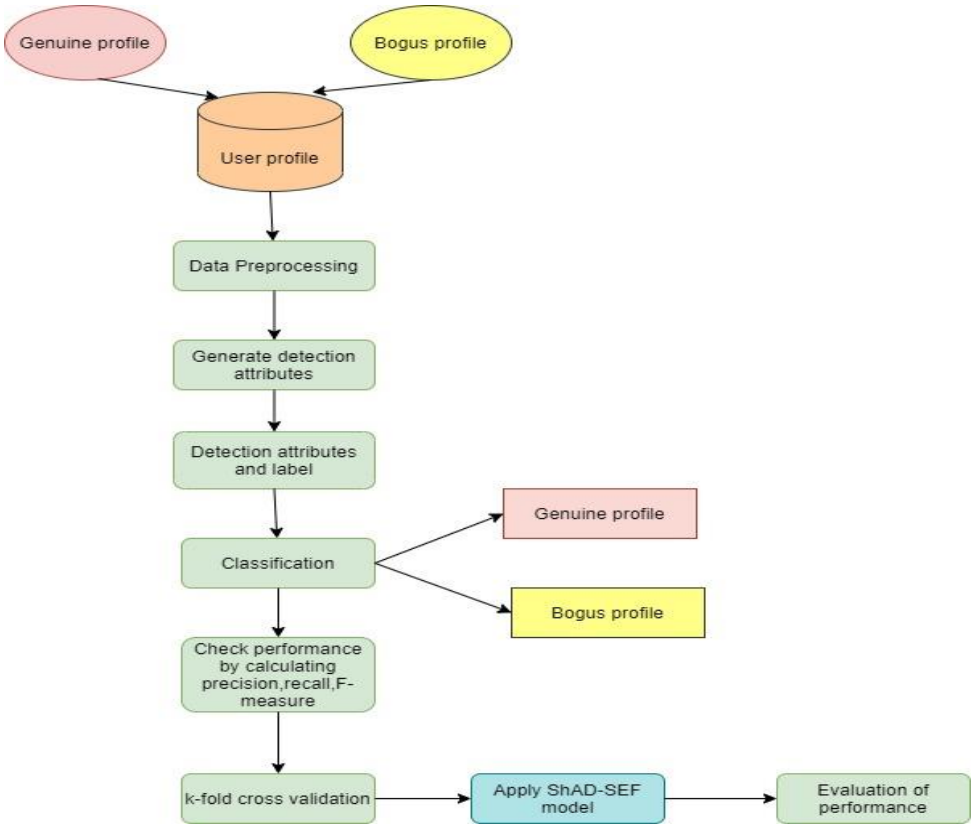

Figure 2. Steps to classify attacker's profile

5.1. Generic Attributes

i) Length Variance (Len_Var)

The length variance is the difference between the system profile's mean length and the customer profile's mean length. Here length refers to the number of movies the user profile has reviewed. Table 6 defines each symbol notation, and the equation of length variance is as Equation (1):

``

$$\text{Length\_Var} = \frac{|r_a - \bar{R}|}{\sum_{a \in U}(r_a - \bar{R})^2} \quad (1)$$

ii) Weighted degree of Agreement (WDA)
The difference between the item's user and mean ratings is divided by the number of times the item has been reviewed. The equation of WDA is as Equation (2):

$$\text{WDA} = \sum_{i=0}^{N_a} \frac{|r_{a,i} - \bar{r_i}|}{R_i} \quad (2)$$

iii) Rating deviation from mean agreement (RDMA)
RDMA identifies the fake profile from the sizeable original profile by measuring the average derivation per item divided by the total no. of ratings given to that item. The equation of RDMA is as Equation (3):

$$\text{RDMA} = \frac{\sum_{i=0}^{N_a} \frac{|r_{a,i} - \bar{r_i}|}{R_i}}{N_a} \quad (3)$$

iv) Weighted deviation from mean agreement (WDMA)
WDMA is the same as RDMA, but the difference is that in the denominator, it takes the square of the count of ratings given to the item. It mainly focuses on the deviation of weight on high-rating sparse items. The equation of WDMA is as Equation (4):

$$\text{WDMA} = \frac{\sum_{i=0}^{N_a} \frac{|r_{a,i} - \bar{r_i}|}{R_i^2}}{N_a} \quad (4)$$

Table 6. Symbol notation

r_a	count rating from user a
$r_{a,i}$	rating is given by the user to the item i
N_a	count number of items rated by the user a
$\bar{r_i}$	mean rating of the item i
R_i	count total number of ratings provided to the item i
$p_{a,f}$	the profile of user a from filler set
$p_{a,t}$	the profile of user a from target set
U	total users on the system

5.2. Model specified attributes

i) Filler Mean Variance (FMV)
FMV detects the average attack in the system. It has three types of items such as target items ($p_{a,t}$), other rated items ($p_{a,f}$), and all other unrated items. FMV computes the average between overall system mean and filler item. Items with low variance specify the average attack. The equation of FMV is as Equation (5):

$$\text{FMV} = \frac{\sum_{i \in p_{a,f}} (r_{a,i} - \overline{r_a})^2}{|p_{a,f}|} \quad (5)$$

Here $p_{a,f}$ = p_a - $p_{a,t}$
Where $p_{a,f}$ denotes the profile of user a from filler set, $p_{a,t}$ denotes the profile from the target set.

ii) Filler Mean Target Difference (FMTD)
It serves to identify attack like bandwagon. The difference in ratings between the target and filler items. The equation of FMTV is as Equation (6):

$$\text{FMTD} = \left| \frac{\sum_{i \in p_{a,t}} r_{a,i}}{|p_{a,t}|} - \frac{\sum_{i \in p_{a,f}} r_{a,k}}{|p_{a,f}|} \right| \quad (6)$$

6. Experimental Results

In order to detect fake profiles from the database, a movie lens 100k dataset was taken, and experiments were done on the Python platform. The training dataset is selected from a database that initially does not contain any fake profiles. The attacker's data was collected from ua_base data at several attack sizes like 1%, 5%,1 0%, and 15%, and filler sizes of 25% and 50% are inserted into the training dataset and are marked as counterfeit. The training dataset is used to generate the detection attributes for each profile. Different classification algorithms such as KNN, NB, RF, and XGBoost are trained, and then the results are analyzed based on evaluation metrics precision, recall, and f-measure. After analyzing the results of individual algorithms using majority voting, three algorithms were combined: RF, XGBoost, and Naïve Bayes, and we made the ShAD-SEF framework. Then, we calculated the precision, recall, and f-measure for the same training dataset using an ensemble framework. We got good results compared with another algorithm.

Table 7 contrasting several models for detecting against shilling attack at 25% filler size[1]

Attack Size	**1%**			**5 %**			**10 %**			**15 %**		
Models	**PR**	**RC**	**F-M**	**PR**	**RC**	**F-M**	**PR**	**RC**	**F-M**	**PR**	**RC**	**F-M**
KNN	0.9104	0.9103	0.9103	0.9108	0.9107	0.9107	0.9047	0.9046	0.9046	0.8854	0.8852	0.8852
RF	0.9183	0.9181	0.9180	0.9089	0.8977	0.8977	0.8959	0.8959	0.8959	0.9080	0.9080	0.9079
NB	0.8960	0.8961	0.8960	0.8858	0.8892	0.8963	0.8806	0.8648	0.8533	0.8907	0.8342	0.8104
XGBoost	0.9247	0.9441	0.9441	0.9214	0.9413	0.9413	0.9288	0.9278	0.9224	0.9243	0.9131	0.9131
ShAD-SEF (our model)	0.9597	0.9596	0.9596	0.9499	0.9599	0.9599	0.9324	0.9324	0.9325	0.9646	0.9624	0.9624

Multiple experiments were conducted, and the accuracy of the suggested model was higher than that of the traditional methods. Tables 7 and 8 illustrate the performance analysis for the "average, random, and bandwagon attack models" with fixed filler sizes of 25% and 50%, respectively, and variable attack sizes of 1%, 5%, 10%, and 15%. We found RF, XGBoost, and NB in this experiment and the ensemble framework that used these three algorithms with logistic regression as the metaclassifier achieves good accuracy.

From Figure 3, the precision for our proposed model ShAD-SEF is more compared to other models. When the attack size is 15%, the precision, recall, and f-measure are 96.4%, 96.2%, and 96.322% respectively, whereas when the attack size is 1%, the precision, recall, and f-measure are 95%, 95%, and 95% respectively.

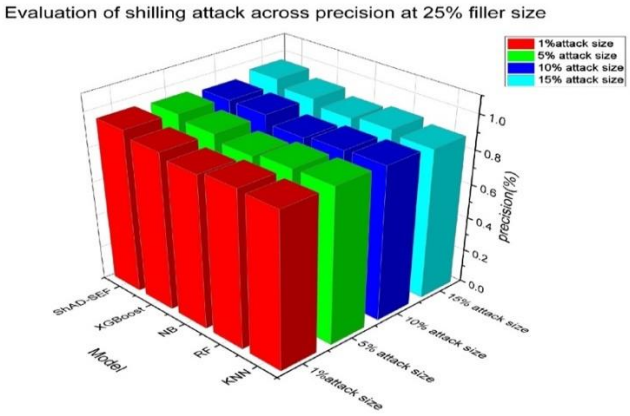

Figure 3. Comparison of precision at various attack sizes and fixed 25% filler size

Figure 4 shows that recall for NB is 89% and ShAD-SEF is 95% when the attack ratio is 1%, 5%, 10%, and 15%. recall for RF is 89%, our model is 93%, and the recall for KNN is 88%, and ShAD-SEF is 96%, respectively. According to the findings, ShAD-SEF performs admirably at varied attack sizes and 25% filler sizes.

``

Figure 5 shows the calculation of the f-measure at attack ratios of 1%, 5%, 10%, 15%, and 25%. For KNN, the f-measure is 91%, and our model is 95%; for NB, it is 89%; for KNN, it is 90%; for ShAD-SEF, it is 93%; and for xgboost, it is 91%. According to experimental findings, our ShAD-SEF model's f-measure exhibits good performance.

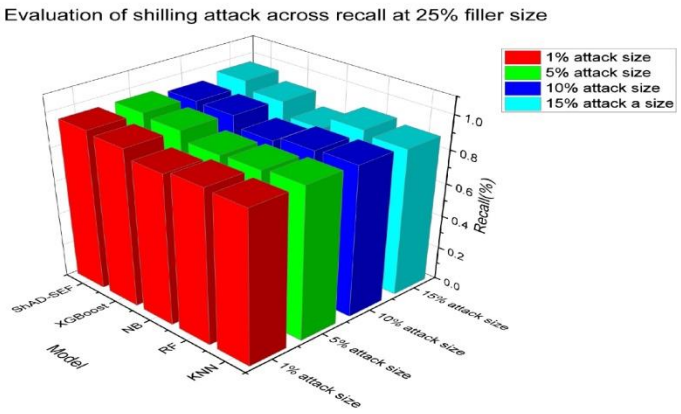

Figure 4. Comparison of recall at different attack sizes and fixed 25% filler size

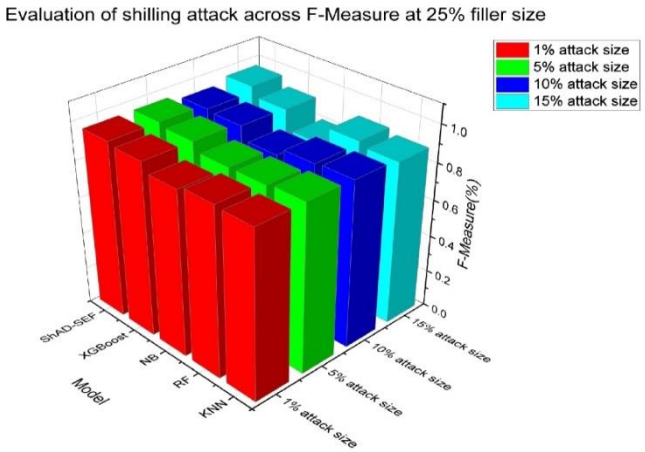

Figure 5. Comparison of F-Measure at different attack sizes and fixed 25% filler size

As seen from Table 8, we experimented with a fixed 50% filler size and various attack sizes like 1%, 5%, 10%, and 15%. With KNN at 1% attack size, we get precision, recall, and f-measure of 92%, 92%, and 92% respectively. For 5% attack size precision for random forest we get 92% precision, 94% recall, and 94% f-measure. For 10% attack size, we got precision for XGBoost as 91%, recall of 91% and f-measure of 91%. For 15% attack size for naïve bayes we got precision 77%, recall of 76%, and f-measure 76%. When the attack size is 1%, we got high accuracy for naïve Bayes. We got precision, recall, and f-measure of 96.7%, 96.6%, and 96.5% respectively. When the attack ratio is small, we get less accuracy. Our model performs well when the attack ratio is 15%. We obtained precision of 96%, recall of 96%, and f-measure of 96%. It is observed that our model performs well compared with other existing algorithms with 96% accuracy.

Table 8. Evaluation of different models for shilling attack detection at 50% filler size

Size of the Attack	**1%**			**5 %**			**10 %**			**15 %**		
Models	**PR**	**RC**	**F-M**	**PR**	**RC**	**F-M**	**PR**	**RC**	**F-M**	**PR**	**RC**	**F-M**
KNN	0.9253	0.9254	0.9259	0.9198	0.9198	0.9198	0.8684	0.8685	0.8684	0.8369	0.8368	0.8368
RF	0.8749	0.8722	0.8705	0.9220	0.9462	0.9459	0.8749	0.8722	0.8705	0.8001	0.7943	0.7901
NB	0.9670	0.9662	0.9659	0.8749	0.8722	0.8705	0.8001	0.7943	0.7901	0.7740	0.7672	0.7665
XGBoost	0.9564	0.9258	0.9256	0.8905	0.8998	0.8997	0.9162	0.9149	0.9148	0.9211	0.9297	0.9296
ShAD-SEF (Proposed Model)	0.9681	0.9680	0.9680	0.9249	0.9489	0.9489	0.9794	0.9694	0.9694	0.9696	0.9697	0.9697

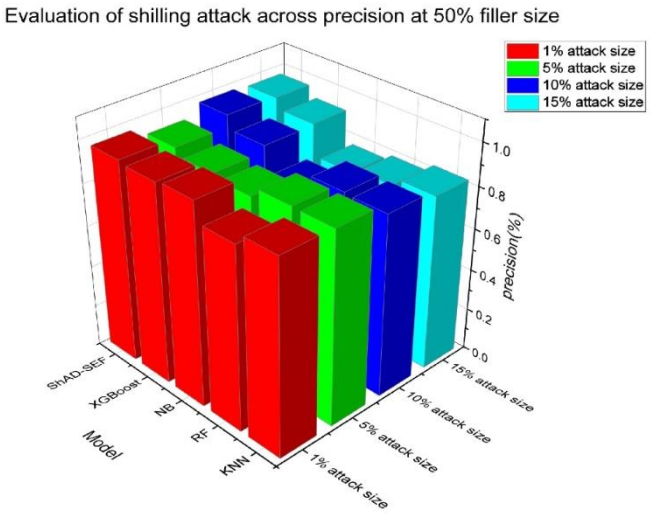

Figure 6. Comparison of precision at different attack sizes and fixed 50% filler size

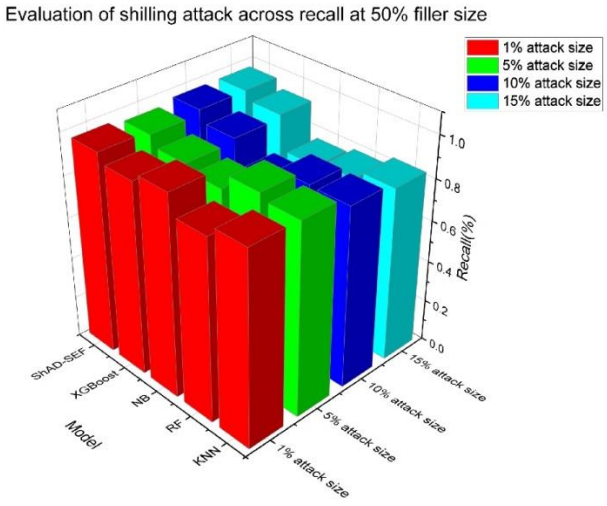

Figure 7. Comparison of recall at various attack sizes and fixed 50% size of the filler

Figure 6 shows the precision evaluation for different classification models, including KNN, RF, NB, and XGBoost, at various attack ratios of 1%, 5%, 10%, and 15% as well as at a fixed filler ratio of 50%. The precision for our ShAD-SEF model is 96% for NB, 83% for KNN, 87% for RF, and 97% for NB when evaluated at different attack ratios of 1%, 5%, 10%, and 15% respectively. Figure 7 shows the comparison of recall at various attack sizes and fixed 50% size of the filler.

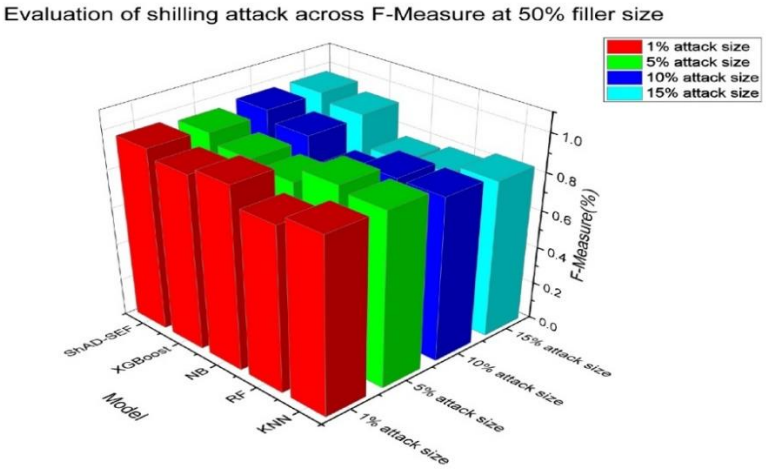

Figure 8. Comparison of F-Measure for various attack sizes and fixed 50% filler size

``

Whenever the attack ratio is 1% and the filler ratio is 50%, Figure 8 shows that the F-Measure for the ShAD-SEF compares to existing classification models quite favorably. When the attack ratio is 5% and the filler ratio is 50%, the f-measure for KNN is 92%, and ShAD-SEF is 96%. For xgboost, the f-measure is 89%, and our proposed model is 94%. For an attack ratio of 10% and filler ratio of 50%, the KNN's f-measure is 86%, and our proposed model is 96%.

7. Conclusion and Future Work

This paper focuses on many attacks against Recommender systems according to collaborative filtering. Attackers create a fake profile to promote the item by assigning a maximum rating and inject to the database with minimum knowledge about the system. Compared to other attack models, we discovered that while the typical attack significantly affects the system, it necessitates additional system understanding. It has been concluded that an item-based collaborative system has high security compared with a user-based coordinated system. Identifying the shilling profiles in the system will increase system trust. We created the training dataset by injecting some fake profiles by calculating generic attributes and model specified attributes. Different Machine Learning algorithms such as Naive bayes, KNN, Random Forest, and xgboost are implemented with different attack and filler ratios. Their performance is analyzed by calculating Precision, Recall and F-measure and it is determined that random forest, naïve bayes and xgboost give top accuracy compared with other algorithms. Our suggested ShAD-SEF model was put to the test using k-fold cross-validation to determine its robustness. In most situations, the accuracy of our proposed model ShAD-SEF is more than 94%. In some models if the attack size is low, the system does not perform well. By using the stacking ensemble approach and majority voting approach we observed that our ShAD-SEF model performs more accurate than existing models. There are many security issues in RS due to its openness. In the future, we can detect the different shilling models by having more model specified attributes to detect other shilling attack models. The more the parameters, the more accurate result we can obtain. In a collaborative recommender system, shilling attacks can be recognized using deep learning approaches.

References

1. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. and Riedl, J. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work* (pp. 175-186), 1994, October.
2. Davoodi, F.G. and Fatemi, O. Tag based recommender system for social bookmarking sites. In *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining* (pp. 934-940). IEEE, 2012, August.
3. Bobadilla, J., Ortega, F., Hernando, A. and Gutiérrez, A. Recommender systems survey. *Knowledge-based systems*, vol. 46, pp.109-132, 2013
4. Lam, S.K. and Riedl, J. Shilling recommender systems for fun and profit. In *Proceedings of the 13th international conference on World Wide Web* (pp. 393-402), 2004, May.
5. Bland, J.A., Petty, M.D., Whitaker, T.S., Maxwell, K.P. and Cantrell, W.A. Machine learning cyberattack and defense strategies. *Computers & security*, vol. 92, pp. 101738, 2020
6. Chen, K., Chan, P.P., Zhang, F. and Li, Q. Shilling attack based on item popularity and rated item correlation against collaborative filtering. *International Journal of Machine Learning and Cybernetics*, vol. 10, pp.1833-1845, 2019
7. Si, M. and Li, Q. Shilling attacks against collaborative recommender systems: a review. *Artificial Intelligence Review*, vol. 53, pp.291-319, 2020
8. Aashkaar, M. and Sharma, P. Enhanced energy efficient AODV routing protocol for MANET. In *2016 International Conference on Research Advances in Integrated Navigation Systems (RAINS)* (pp. 1-5). IEEE, 2016, May.
9. Kaur, P. and Goel, S. Shilling attack models in recommender system. In *2016 International conference on inventive computation technologies (ICICT)* (Vol. 2, pp. 1-5). IEEE, 2016, August.
10. Bilge, A., Ozdemir, Z. and Polat, H. A novel shilling attack detection method. *Procedia Computer Science*, 31, pp.165-174, 2014
11. Batmaz, Z., Yilmazel, B. and Kaleli, C. Shilling attack detection in binary data: a classification approach. *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, pp.2601-2611, 2020
12. Zhang, F. and Zhou, Q. Ensemble detection model for profile injection attacks in collaborative recommender systems based on BP neural network. *IET Information Security*, vol. 9, no. 1, pp.24-31, 2015
13. Wang, Y., Qian, L., Li, F. and Zhang, L. A comparative study on shilling detection methods for trustworthy recommendations. *Journal of Systems Science and Systems Engineering*, vol. 27, no. 4, pp.458-478, 2018
14. Sundar, A.P., Li, F., Zou, X., Gao, T. and Russomanno, E.D. Understanding Shilling Attacks and Their Detection Traits: A Comprehensive Survey. *IEEE Access*, vol. 8, pp.171703-171715, 2020
15. Hu, D., Xu, B., Wang, J., Han, L. and Liu, J. A Shilling Attack Model Based on TextCNN. In *2020 IEEE 3rd International Conference on Automation, Electronics and Electrical Engineering (AUTEEE)* (pp. 282-289). IEEE, 2020, November.
16. Ebrahimian, M. and Kashef, R. Efficient Detection of Shilling's Attacks in Collaborative Filtering Recommendation Systems Using Deep Learning Models. In *2020 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)* (pp. 460-464). IEEE, 2020, December.
17. Rani, S., Kaur, M., Kumar, M., Ravi, V., Ghosh, U. and Mohanty, J.R. Detection of shilling attack in recommender system for YouTube video statistics using machine learning techniques. *Soft Computing*, pp.1-13, 2021

18. Sharma, P., Saxena, K., and Sharma, R. Heart disease prediction system evaluation using C4.5 rules and partial tree doi:10.1007/978-81-322-2731-1_26, 2016
19. Williams, C.A., Mobasher, B. and Burke, R. Defending recommender systems: detection of profile injection attacks. *Service Oriented Computing and Applications*, vol. 1, no. 3, pp.157-170, 2007
20. Bilge, A., Ozdemir, Z. and Polat, H. A novel shilling attack detection method. *Procedia Computer Science*, vol. 31, pp.165-174, 2014
21. Alonso, S., Bobadilla, J., Ortega, F. and Moya, R. Robust model-based reliability approach to tackle shilling attacks in collaborative filtering recommender systems. *IEEE Access*, vol. 7, pp.41782-41798, 2019
22. Zhang, F. and Zhou, Q. Ensemble detection model for profile injection attacks in collaborative, 2014
23. Williams, C.A., Mobasher, B. and Burke, R. Defending recommender systems: detection of profile injection attacks. *Service Oriented Computing and Applications*, vol. 1, no. 3, pp.157-170, 2007
24. Yang, Z., Xu, L., Cai, Z. and Xu, Z. Re-scale AdaBoost for attack detection in collaborative filtering recommender systems. *Knowledge-Based Systems*, vol. 100, pp.74-88, 2016
25. Mehta, B., Hofmann, T. and Fankhauser, P. Lies and propaganda: detecting spam users in collaborative filtering. In *Proceedings of the 12th international conference on Intelligent user interfaces* (pp. 14-21), 2007, January.
26. Zhou, Q., Wu, J. and Duan, L. Recommendation attack detection based on deep learning. *Journal of Information Security and Applications*, vol. 52, pp. 102493, 2020
27. Zhou, W., Wen, J., Xiong, Q., Gao, M. and Zeng, J. SVM-TIA a shilling attack detection method based on SVM and target item analysis in recommender systems. *Neurocomputing*, vol. 210, pp.197-205, 2016

``