Wireless Personal Communications (2024) 137:259–286
https://doi.org/10.1007/s11277-024-11401-y

Detecting Unknown Shilling Attacks in Recommendation Systems

Pradeep Kumar Singh[1] · **Pijush Kanti Dutta Pramanik**[2] · **Nilanjan Sinhababu**[3] · **Prasenjit Choudhury**[4]

Accepted: 20 June 2024 / Published online: 4 July 2024

Abstract

Recommender systems are vulnerable to attacks because of their open nature. Counterfeit users give biased ratings to the items due to various objectives that may lead to the loss of user trust. The attackers use certain attack models with specific features. The existing attack detection techniques are typically attack-specific and work only when the attack features are known. They are unable to identify an unknown attack with unfamiliar features. To diminish this problem, in this paper, we propose a generalized solution that filters any attack irrespective of its design and features. We trained the classifiers with the ratings of the known authentic users using one-class SVM and PU learning models for detecting attacks, considering their ability to detect anomalies in the dataset caused by unknown attacks. The openly available MovieLens dataset has been used to assess our designed attack detection method. The experimental results show that all unknown attacks are successfully detected with 100% accuracy. The same detection accuracy is achieved for attacks with known features.

Keywords Recommender system · Top-N recommendation · Shilling attack · Attack features · Classifiers · Machine learning

✉ Pijush Kanti Dutta Pramanik
pijushjld@yahoo.co.in

Pradeep Kumar Singh
pradeepkumar.singh@galgotiasuniversity.edu.in

Nilanjan Sinhababu
nilanjansb95@gmail.com

Prasenjit Choudhury
prasenjit0007@yahoo.co.in

[1] School of Computer Science and Engineering, Galgotias University, Greater Noida, U.P., India

[2] School of Computer Applications and Technology, Galgotias University, Greater Noida, U.P., India

[3] Indian Institute of Technology, Kharagpur, W.B., India

[4] Department of Computer Science and Engineering, National Institute of Technology, Durgapur, W.B., India

1 Introduction

Recommendation systems have become an imperative element of e-commerce. Recommendation systems are designed to help online buyers make better selections from the large pool of internet-based products and services. The true beauty of recommendation systems is that they can predict and oblige users with the items they might prefer and like. Swayed by alluring proposals, even stray browsers ultimately become gratified buyers. A recommendation system's most basic and important element is the user's response. Users give feedback and ratings for the items they have used or browsed. The top-rated items are recommended to new users so that they can view the best product without going through the whole catalogue, thus enjoying a satisfying buying experience. However, can we truly trust these seemingly friendly recommendation systems? Raising this question is legitimate for a system that is built upon public input. Is it reasonable to presume that all the users are honest and altruistic? The suspect deepens, especially because a large amount of money is involved in the highly commercialized implementation of the recommendation system. The point is that the product that is suggested to us as the best probable item (based on ratings and reviews) is truly so? Unfortunately, the answer is 'no'. In fact, providing a biased recommendation depending on the user profile is not unfamiliar. For instance, a few years ago, the Wall Street Journal stated that Orbitz, a vacation website, was displaying an increased price to Mac Book owners for online flight and room booking [1]. Likewise, it is alleged that Google recommends lower-paying jobs to women candidates [2]. Just as biased news feeds are infamous for promoting and demoting politicians' credentials by selectively focusing only on the candidate's positive and negative points to manipulate the viewer's opinion.

1.1 The Shilling Attack Problem in Recommendation Systems

This malpractice of manipulating recommendation systems has taken an ugly outlook in terms of e-commerce. Fierce competition has caused companies to use unscrupulous means to bring their products to the attention of online buyers. They manipulate recommendation systems. We call this manipulation an attack. The attack may also come from the users. Due to their open nature, recommendation systems are prone to profile injection attacks wherein intruders insert false profiles with incorrect ratings into the system to bias recommendations. Generally, there might be three purposes of these attacks: (a) to bring one's own products to the top of the recommendation list, (b) to bring down a rival company's products from the top of the recommendation list, and (c) to play with the list to disrupt the overall recommendation system.

1.2 Problem Description

Attackers have adopted several attack schemes to manipulate recommendation systems. This has resulted in diminishing trust in recommendation systems, leading researchers to jump into protecting the purpose of a trustworthy recommendation system. Researchers have been able to identify attack models and train systems by means of machine learning techniques so that the systems can identify possible attacks and take necessary measures.

The feature-based attack detection model is the most successful approach for mitigating attacks if the profile of the attackers is known [3–5]. The performance of such methods degrades significantly in the context of unknown attack features. The major

drawback of feature-based detection is that every time a new attack is created, the attack design must be identified, and its features must be extracted. Detection techniques are constantly developing, but so are attack designs. Some attacks might not possess known features and hence escape detection. Smart attackers can always launch a new attack model with entirely new features. Chad et al. [6] generated an obfuscation attack and demonstrated that a small obfuscation can reduce the ability of these features to detect attacks.

The problem is formulated as follows:

Let U, I, R, F_{AK}, AU, and AT be the set of users and items, the user-item rating matrix, the set of known attack features, a set of authentic users, and a set of attackers, respectively. The feature-based shilling attack detection approaches estimate the function $f(g(U \times I \rightarrow R)) \in F_{AK} \rightarrow H_D$, which classifies the user's rating f(g(U, R), H_D), for a user with high detection accuracy (H_D). However, the detection accuracy of the detection based on features is reduced in the case of unknown attacks, and the approaches estimate the function f(g(U, R), L_D), i.e., $f(g(U \times I \rightarrow R)) \notin F_{AK} \rightarrow L_D$, which classifies the user's rating with low detection accuracy (L_D). $L_D \in AU \rightarrow AT$ and $L_D \in AT \rightarrow AU$, which means that low detection accuracy occurs due to classifying some attackers as authentic users and some authentic users as attackers.

1.3 Motivation

As the saying goes "devil's mind always stays a step ahead of those of saints," attackers have developed new attack models with different features whenever they have been encountered. This cat and mouse race is never going to end. This situation is shown in Fig. 1. To break this cycle, a detection technique is required that detects attackers irrespective of the way they have been designed or the features they possess.

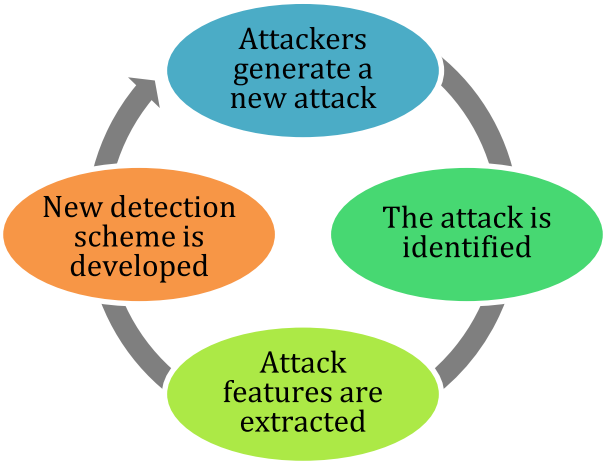


Fig. 1 The process of generating an attack, detecting it, and subsequently generating another new attack continues

1.4 Proposed Solution

The main issue of attack detection in recommendation systems is "if the statistical features of the attack model are not known beforehand, then the existing classifiers are unable to detect the attacks." This paper introduces a solution approach that can detect any attack on a recommendation system irrespective of prior knowledge of the attack features. The philosophy of our approach is that instead of acquainting with every bad feature, we shall edit the recommendation system about the properties of a good user. Anything not matching this knowledge is filtered out as a bad entity. Therefore, instead of training the recommendation system with the attack features, we trained the recommendation system with genuine ratings. If a recommendation system encounters any deviation from this genuine knowledge, it will be regarded as an attack.

In the proposed approach, whichever attack model the attackers try, our model will always detect that attack. The goal of our proposed solution approach is to achieve $H_D \in \mathrm{AT} \rightarrow \mathrm{AT}$ and $H_D \in \mathrm{AU} \rightarrow \mathrm{AU}$, i.e., high detection accuracy for identifying all the attackers and the authentic users correctly, irrespective of the attack type (known and unknown).

However, in this paper, we emphasize unknown attacks because the existing attack detection models perform satisfactorily in detecting known attacks but miserably fail to detect unknown attacks. Figure 2 shows the comparative performances between the existing attack detection schemes and our proposed approach. For an unknown attack, the existing attack detection schemes identify some attackers as authentic users and some authentic users and attackers as attackers. In comparison, our method identifies all the attackers correctly while identifying some authentic users as attackers. This will ensure that the system will be attack-free with some compromise.

In the proposed method, we used the concept of most trustworthy users, where instead of training the recommendation system for every possible attack, the classifier was labelled only by the properties of authentic users (most trustworthy users) from the training dataset, and from these properties, it could predict which users belong to the authentic group in the test dataset. Any user whose behaviour did not comply with an authentic user's was claimed to be a nonauthentic user or attacker. Let $U_{AU} \in \mathrm{U}$ and R_{AU} be the sets of authentic users and their ratings, respectively. Thus, the function of the proposed solution becomes f(R, R_{AU}), which classifies the user's rating with the high detection accuracy of attackers.

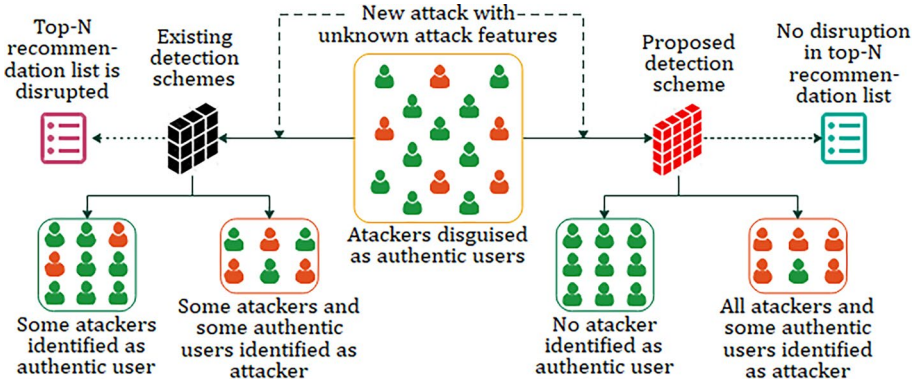


Fig. 2 Comparison of existing feature-based detection schemes and the proposed solution approach

When the model was tested with the test sets created by launching four known attacks and an unknown attack (obfuscated attack), it detected all of them by identifying the authentic users. We employed positive unlabelled (PU) learning and a one-class SVM (OSVM) as classifiers to test the proposed method.

1.5 Contributions

Listed below are the paper's main contributions, while Fig. 3 outlines the key aspects of the paper.

- The inability of feature-based detection methods to identify unknown attacks was demonstrated.
- A set of the most trustworthy users was generated by implementing a novel algorithm.
- A comparison of standard attack models and an unknown attack (obfuscated attack) is performed based on biasing, shuffling effects, and the hit ratio to determine the attack and filler size.
- For attack detection and verification, the following actions were performed:
- Binary classifiers such as support vector machines (SVM), J48, random forest, and naïve Bayes are trained using known shilling attacks and then evaluated using unknown attacks.
- PU learning using binary classifiers is learned through genuine users and evaluated by both known and unknown shilling attacks.
- The OSVM was trained with ratings of trustworthy users and tested with known and unknown attacks.

1.6 Paper Organization

The remainder of the paper is structured as follows. Section 2 provides the necessary theoretical background that is required to be familiar with this paper, which includes a brief description of the attack models, the impact of shilling attacks on recommendations, and

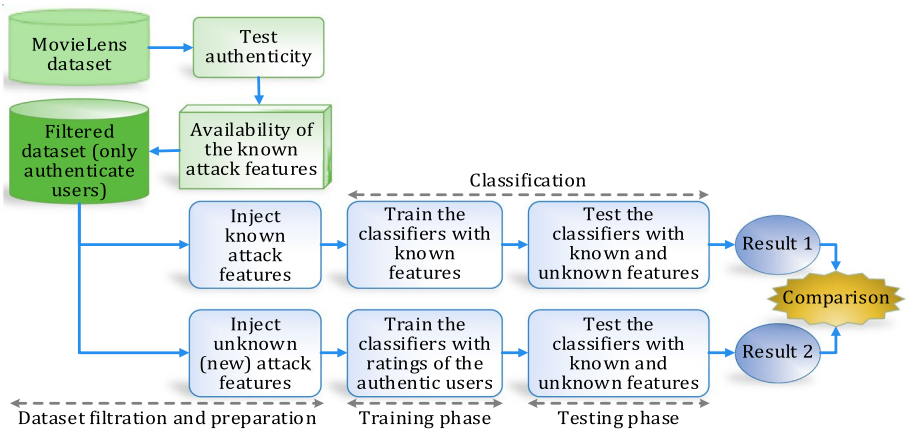


Fig. 3 A brief outline of this paper

the model for supervised shilling attack detection. Section 3 discusses related works on detecting shilling attacks. The problem description of this paper is discussed in Sect. 4. Section 5 presents a novel solution for detecting any attack on recommendation systems. Section 6 reports the analysis of the experimental outcomes. Finally, we conclude the paper in Sect. 7, stating the scope of further research by adding more value to this work.

2 Theoretical Background

Information retrieval techniques play an important role in recommender systems. Recommendation systems use various filtering approaches to retrieve information from users. Collaborative filtering is one of the most popular approaches used in recommendation systems. Figure 4 illustrates the conceptual framework of collaborative filtering, which consists of three different units: (i) data collection, (ii) missing rating prediction (using similarity metric and prediction approach), and (iii) top-N recommendation.

Data collection considers the user's rating information on a target item. A list of m users and n items is converted into a user-item rating dataset of size $m \times n$, which has a high possibility of vacant entry, i.e., a user does not rate the target item. Similarity measures and prediction approaches are applied to predict missing ratings. Then, the collaborative filtering-based recommendation system generates the top-N list of items to recommend the user based on the predicted rating. The complete process is pictorially shown in Fig. 4.

The generated Top-N lists will be insecure in the presence of shilling attackers in the dataset that is used in the recommendation. Basic terms related to shilling attacks, the impact of shilling attacks on top-N recommendations, and the existing detection schemes in the literature are discussed as follows.

2.1 Shilling Attacks

An attack begins with the creation of a large number of false profiles. Depending on the purpose, an attack may be a push or a nuk [1]. Each attacker's profile consists of four components, as listed in Table 1 [7]. Various standard attack models have been discussed in the literature [8–12]. The basic characteristics of these attack models are shown in Table 2. The effect of attacks on collaborative filtering-based recommendation systems varies on the attack size and filler size, which are described in Table 3.

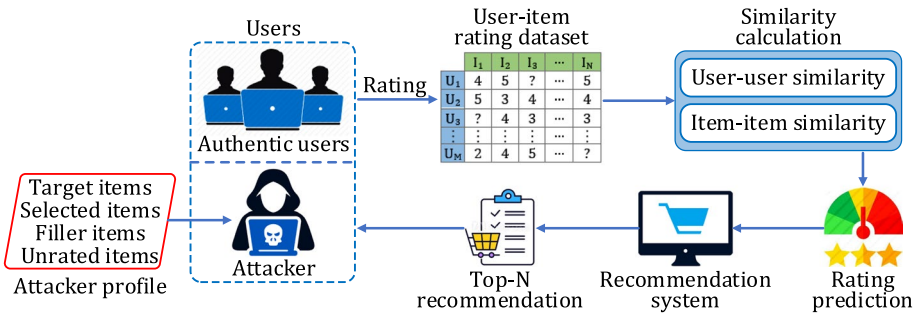


Fig. 4 Conceptual framework of collaborative filtering

Table 1 Components of an attacker's profile

Component	Description
Target items ($\boldsymbol{I_T}$)	Those items that are biased by attackers and have been given excessively high or low ratings
Selected items ($\boldsymbol{I_S}$)	Those items whose ratings are functionally determined by the different types of attacks
Filler items ($\boldsymbol{I_F}$)	These items are selected by attackers randomly and assigned random ratings
Unrated items ($\boldsymbol{I_N}$)	Those items whose ratings are not given by attackers

Table 2 Standard attack model

Attack	Property
Random attack	$I_S = \varnothing$ and $\rho(i) = N(\overline{r}, \sigma^2)$
Average attack	$I_S = \varnothing$ and $\rho(i) = N(\overline{r_i}, {\sigma_i}^2)$
Bandwagon attack	I_S includes some popular items with excellent ratings and $\rho(i) = N(\overline{r_i}, {\sigma_i}^2)$
Segment attack	I_S contains items that are identical to the target items and $\rho(i) = N(\overline{r_i}, {\sigma_i}^2)$

$\rho(i)$ is a function on item i that determines the rating patterns ofI_F. $N(\overline{r}, \sigma^2)$ represents the Gaussian distribution. σ^2 and $\overline{r}$ represent the variance and the mean rating of all items and users, respectively. $\overline{r_i}, {\sigma_i}^2$ show the mean and the variance of item i, respectively

Table 3 Filler size and attack size

Item	Description	Calculation (%)
Filler size	Total number of filler items in the dataset	$\frac{\textit{Total number of filler items}}{\textit{Total number of items}} * 100$
Attack size	Total number of attackers in the dataset	$\frac{\textit{Total number of attackers}}{\textit{Total number of users}} * 100$

2.2 Impact of Shilling Attacks

E-commerce companies use recommendation systems to offer the best service to their customers by suggesting products they may like. The top-N list ensures the most appropriate items from all items. The attacker's aim is to modify this top-N list. As shown in Table 4, three parameters are typically used to determine the impact of attacks on the target users' top-N recommendations.

Here, R_u shows the top-N list. $t = \{t_1, t_2, \ldots, t_k\}$ is the list of target items that have to be pushed. U is a list of users. N is the total number of items. T_a identifies the top-N list when an attack is produced.

The main objective of shilling attack detection is to model a classifier to distinguish between authentic users and shilling attackers. Therefore, several studies have used statistical pattern recognition to mitigate shilling attacks. There are two types of classifiers: (a) supervised learning models and (b) unsupervised learning models.

Learning from unlabelled data, i.e., unsupervised learning, is inherently difficult. As a result, it suffers from a high number of false positives. However, learning from labelled

Table 4 Deciding parameters of the impact of attacks [7]

Parameters	Description	Computational equation	Ideal value	
			Attacker	Detector
Hit ratio	Denotes modifications initiated by attackers to the top-N recommendation	$H_r = \sum_{u \in U} \frac{H_{u,t_i}}{\lvert U \rvert}$, where $H_{u,t_i} = 1$ if $t_i \in R_u$ and $H_{u,t_i} = 0$, otherwise	1	0
Biasing	Denotes changes of the target items in the top-N recommendation by attackers	$B = \frac{\sum_{i=1}^{N} B_i}{N}$, where $B_i = 1$ if $t_i \in T_a$ and $B_i = 0$, otherwise	1	0
Shuffling	Denotes the changes in the positions of items in the top-N list by attackers	$S = \frac{\sum_{i=1}^{N} S_i}{N}$, where $S_i = 0$ if, $T_i == T_{a_i}$ and $S_i = 1, T_i \neq T_{a_i}$.	1	0

Table 5 Existing solution approaches for detecting shilling attacks

Classification approach	Advantages	Disadvantages	References
Supervised	Uses less complex algorithms with high accuracy and is more reliable	Finds difficulties on dynamic data, i.e., online data	[3, 13–19]
Unsupervised	Uses more complex algorithms with moderate accuracy and is less reliable	Provides comparatively low accuracy on static data, i.e., offline data	[20–25]

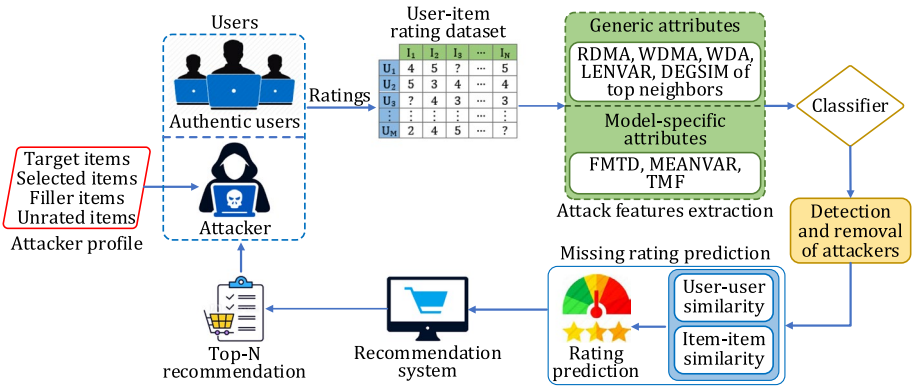


Fig. 5 Conceptual framework of a collaborative filtering-based recommendation system using a supervised shilling attack detection model

data (supervised learning) results in a very low number of false positives. Table 5 describes the classification approaches for detecting shilling attacks.

2.3 Supervised Shilling Attack Detection

Supervised classification is the most prevalent predictive model, so we consider it a baseline scheme for detecting shilling attacks. The conceptual framework of a collaborative filtering-based recommendation system using a classification-based shilling attack detection model is shown in Fig. 5.

2.3.1 Data Collection

The accuracy of the recommendation system is dependent upon the types of data used in the recommendation. Like two faces of a coin, there are two types of users in collaborative filtering, i.e., authentic users and shilling attackers. The two methods of rating data collection used in collaborative filtering are explicit and implicit. There is a list of m users, i.e., $\mathrm{U} = \{u_1, u_2, \ldots, u_m\}$, and a list of n items, i.e., $\mathrm{I} = \{i_1, i_2, \ldots, i_n\}$, in a traditional collaborative filtering-based recommendation system. Each user gives his opinion explicitly about the particular item in the form of a rating score. User activities involve implicit methods of data collection in collaborative filtering-based recommendation systems. These activities

Table 6 Notations used for attributes

Notation	Description	Notation	Description
N_u	Total items rated by user u	t_i	The total number of ratings that every user gave item i
k	The instance of nearest neighbours	$r_{u,i}$	The rating of i^{th} item given by user u
l_u	length of a user's profile u	$\overline{r_i}$	The average rating that item i obtained from all users
P_u	The profile of a user u	$\bar{l}$	The average length of a user's profile
$P_{u,T}$	The user's target item set	$\|P_u\|$	The number of ratings in the profile u
$P_{u,F}$	The set of filler items of user u	$Sim_{u,v}$	The similarity value between user u and v
$\|P_{u,T}\|$	The number of target items of user u	$\varnothing_{u,i}$	$\varnothing_{u,i}$= 1, if i ∈ $P_{u,T}$ and $\varnothing_{u,i}$ = 0, otherwise
$\|P_{u,F}\|$	The number of filler items of user u		

Table 7 Attributes and their computational equations

Generic attributes		Model-specific attributes	
RDMA	$RDMA_u = \frac{\sum_{i=0}^{N_u} \left\| \frac{r_{u,i} - \overline{r_i}}{t_i} \right\|}{N_u}$	FMTD	$FMTD_u = \left(\frac{\sum_{i \in P_{u,T}} r_{u,i}}{\|P_{u,T}\|} \right) - \left(\frac{\sum_{k \in P_{u,F}} r_{u,k}}{\|P_{u,F}\|} \right)$
WDMA	$WDMA_u = \frac{\sum_{i=0}^{N_u} \left\| \frac{r_{u,i} - \overline{r_i}}{t_i^2} \right\|}{N_u}$	MEANVAR	$MeanVar_{P_{t,u}} = \frac{\sum_{i \in (P_u - P_{u,T})} (r_{i,u} - \overline{r_i})}{\|P_u\|}$
WDA	$WDA_u = \sum_{i=0}^{N_u} \left\| \frac{r_{u,i} - \overline{r_i}}{t_i} \right\|$	TMF	$F_i = \frac{\sum_{u \in U} \varnothing_{u,i}}{\sum_{u \in U} \|P_{u,T}\|}$
DEGSIM	$DegSim_u = \frac{\sum_{v=1}^{k} Sim_{u,v}}{k}$		
LENVAR	$LenVar_u = \frac{\|l_u - \bar{l}\|}{\sum_{k \in U} (l_u - \bar{l})^2}$		

RDMA rating deviation from mean agreement, *WDMA* weighted deviation from mean agreement, *WDA* weighted deviation from agreement, *DEGSIM* degree of similarity, *LENVAR* length variance, *FMTD* filler mean target difference, *MEANVAR* mean–variance, *TMF* target model focus

of the user are (a) time spent searching for an item, (b) click behaviour, (c) movement of the mouse cursor, etc. The user-item dataset is generated using the abovementioned methods.

2.3.2 Feature Extraction

Usually, the attack profiles are based on standard attack models, and as a result, they appear to share some similarities. Consequently, attack profiles have statistically varied from those of genuine users. In the literature, a number of attributes that are common in user profiles have been extracted. In the case of real users and attackers, these attributes appear to display different trends. They have, therefore, taken an active role in identifying whether a profile is that of an attacker or a legitimate user.

Table 6 denotes the notations used in the attributes, and the computational equations of the attributes are given in Table 7.

2.3.3 Attacker Detection and Removal

The features explained in the previous subsection differ greatly in terms of their values for authentic users and attackers. Hence, these are used to train various classifiers to filter the attackers from the system, as shown in Fig. 6. However, the presence of redundant and irrelevant features reduces the accuracy of a classifier. The process of determining which characteristics are most suitable for classification is referred to as feature selection. It eliminates any features that aren't significant or redundant, which not only lowers the dimensionality of the data but also makes it possible for data mining algorithms to function more quickly and efficiently.

Feature selection is performed using attribute evaluators such as information gain, the Gini index, uncertainty, and correlation coefficients [3, 26]. MC-Relief has been used to extract five features out of the total for feature detection [27]. RDMA and its variants, such as WDMA and WDA, have the highest information gains. Different attack features are useful for different filler sizes; hence, no single feature can be considered the best. Length variance is an important feature for distinguishing large filler sizes, as genuine users generally do not rate so many items.

2.3.4 Prediction of Missing Ratings

Missing rating prediction is employed in a recommendation system based on collaborative filtering to adhere to the concept that if the predicted rating is high, there is a strong likelihood that a user will find the recommendation favourable. Therefore, the collaborative filtering-based recommendation system uses a number of similarity metrics and prediction techniques to forecast the target item's rating for the target user after shilling attackers have been eliminated [28]. Table 8 lists the most frequently employed computing equations for similarity measures and prediction approaches.

Here, Sim(u, v) represents the similarity of two users u and v, while $R_{i,u} and R_{i,v}$ show the ratings of two users u and v on item i, respectively. $\overline{R_i}$, $\overline{R_u}$, and $\overline{R_v}$ show the average ratings of item i, user u, and user v, respectively. $|R_u|$ and $|R_v|$ denote the total number of ratings given by users u and v, respectively; $|I_{uv}|$ denotes the total count of ratings given by both users' u and v; $k_{i,u}$ and $k_{i,v}$ show the ranks of the two users' ratings u and v, respectively, with respect to item i; and $\overline{k_u}$ and $\overline{k_v}$ denote the average ranks of users u and v, respectively,

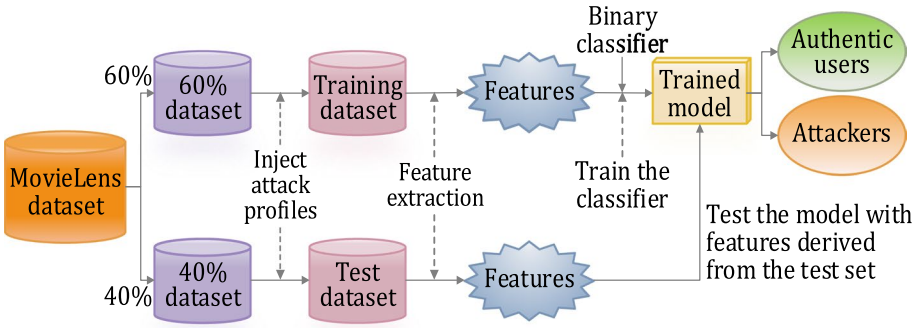


Fig. 6 Feature-based detection of shilling attacks

Table 8 Popularly used similarity measures and prediction approaches

Calculation	Purpose	Method used	Corresponding equation
Sim(u,v)	Calculating the similarity between users u and v	Cosine similarity	$\frac{R_{i,u}.R_{i,v}}{\|\|R_{i,u}\|\|^2 \|\|R_{i,v}\|\|^2}$
		Adjusted cosine similarity	$\frac{\sum_{i\in I}(R_{i,u}-\overline{R_i})(R_{i,v}-\overline{R_i})}{\sqrt[2]{(\sum_{i\in I}(R_{i,u}-\overline{R_i})^2}\sqrt[2]{(\sum_{i\in I}(R_{i,v}-\overline{R_i})^2}}$
		Euclidean distance	$\sqrt[2]{\frac{\sum_{i\in \boldsymbol{I_{uv}}}(R_{i,u}-R_{i,v})^2}{\|\boldsymbol{I_{uv}}\|}}$
		Jaccard similarity	$\frac{\|R_u\|\cap\|R_v\|}{\|R_u\|\cup\|R_v\|}$
		Pearson correlation	$\frac{\sum_{i\in I}(R_{i,u}-\overline{R_u})(R_{i,v}-\overline{R_v})}{\sqrt[2]{(\sum_{i\in I}(R_{i,u}-\overline{R_u})^2}\sqrt[2]{(\sum_{i\in I}(R_{i,v}-\overline{R_v})^2}}$
		Spearman correlation	$\frac{\sum_{i\in I}(k_{i,u}-\overline{k_u})(k_{i,v}-\overline{k_v})}{\sqrt[2]{(\sum_{i\in I}(k_{i,u}-\overline{k_u})^2}\sqrt[2]{(\sum_{i\in I}(k_{i,v}-\overline{k_v})^2}}$
$\widehat{r_{ui}}$	Predicting the ratings of the target item i	Mean centering	$\overline{r_u}+\frac{\sum_{v\in N_i(u)}sim(u,v)(R_{i,v}-\overline{R_v})}{\sum_{v\in N_i(u)}\|sim(u,v)\|}$

based on the ratings. Furthermore, $\widehat{r_{ui}}$ identifies the item i's predicted rating for target user u.

2.3.5 Top-N Recommendation

Based on the generated top-N list, a collaborative filtering-based recommendation system recommends items to the user [29–33]. Recommendation using the top-N list will be more authentic and more accurate if there are no shilling attackers in the dataset.

3 Related Work

The problem of shilling attacks was first discussed by O'Mahony et al. [34], who summarized various attack-building strategies and evaluated the robustness of memory-based collaborative filtering. Lam and Riedl [35] investigated the intent and effect of these attacks and introduced the random bot and average bot attack models. In addition, various other attack models have been discussed in the literature [36].

In their study, Si and Li [37] critically analyzed existing survey papers on shilling attacks, addressing their limitations. They provided a comprehensive overview of various types of shilling attacks and their deployment. Additionally, they delved into robust recommendation algorithms, profile injection attack tactics, shilling attack detection designs, and briefly explained evaluation measures for the suggested systems.

Several strategies have been developed to detect shilling attacks, most of which are centred on identifying and extracting the behaviour of attack profiles. Because attack profiles show a high degree of correlation, Zhang et al. [38] deployed spectral clustering to filter out attackers, while Zhang and Kulkarni [8] used a graph-based detection technique to filter out attackers. Bilge et al. [39] presented a novel unsupervised technique for shilling attack detection. Their approach utilizes a bisecting k-means clustering methodology,

which organizes attack profiles into leaf nodes of a binary decision tree. This method is particularly effective in identifying specific known attacks like bandwagon, segment, and average attack. In [40], the similarity in the structure of attack profiles was exploited to develop a PCA and a PLSA algorithm to segregate attackers from authentic attackers. These techniques fail to discern attackers with low correlation.

The behaviour of attackers is expressed in the form of attack features. Several detection techniques based on these attack features have been developed. Chirita et al. [41] identified a few features, naming them statistical metrics utilized to analyze user ratings, and introduced a new metric called the Rating Deviation From Mean Agreement. A naïve algorithm has been proposed to exploit these metrics. Zhuo et al. [4] used two metrics, RDMA and degree of similarity, to filter out most of the attackers and then applied their algorithm, target item analysis, to the filtered set of users. Williams et al. [42] further identified a few more features and studied their classification performance. Zhang et al. [43] introduced a new metric called *trust* and incorporated it with features to analyze its effectiveness. He et al. [44] used a rough set theory on these features for detection, but their method suffers from a high false-positive rate.

William et al. [45] used kNN, C4.5, and SVM trained on these features to enhance the robustness of the recommendation system. In an attempt to find the best possible combination of classifiers for feature-based classification, Bhebe et al. [26] proposed a methodology that utilizes k-nearest neighbor, support vector machine, and Bayesian network as the initial base classifiers. Kumar et al. [5] designed an ensemble model that compares six machine learning algorithms and used the best combiner strategy to develop an ensemble model for detecting attackers.

Chad et al. [6] deviated from standard attack models and proposed the idea of diverse and obfuscation attacks by applying noise injection, target shifting and user shifting. The application of these techniques reduces the performance of feature-based classifiers. Zhang et al. [46] studied these obfuscation attacks and derived a few features to correctly identify these attacks using an ensemble model, even though they are obfuscation attacks.

Lee and Zhu [47] proposed a methodology that employed "filler" ratings to identify a group of profiles. They thoroughly investigated various attributes of these profiles, offering empirical evidence of the key features of shilling attacks. They then presented a hybrid, two-phase approach for shilling attack detection. A multidimensional scaling technique was utilized to identify distinctive patterns supporting the detection and security of recommended activities. After that, clustering-based techniques were applied to differentiate among attack users.

4 Proposed Solution for Detecting Unknown Shilling Attacks

The feature extraction and detection of attacks eventually lead to an attack cycle, as discussed in Sect. 1. A solution that excludes the norm of identifying and extracting features is needed to break this cycle.

To implement our solution, the precondition is to identify a group of authentic users within the system who can never give biased ratings, termed the most trustworthy users. The question that follows is how do we find this set of users?

To find the set of most trustworthy users, we used the publicly available and widely used MovieLens dataset to perform our experiments. Since feature-based detection has

proven to be successful in detecting standard attacks, we follow the feature-based strategy used in [41] to test the MovieLens dataset for any known standard attacks. The modified algorithm used in the filtration of the most trustworthy users is defined in Algorithm 1. The proposed solution that identifies all known and unknown attacks is shown in Fig. 7.

Algorithm 1 Generation of the most trustworthy users

Input: user-item rating dataset
Output: a set of most trustworthy users
1. *A* = set of all users in the Movie Lens dataset
2. *Low_Features* (*LF*) = {FMD, MEANVAR}
3. *High_Features* (*HF*) = {RDMA, WDMA, WDA, DEGSIM, FMTD, LENVAR, FMTD, TMF}
4. *All_Features* = *LF* U *HF*
5. *Suspected_with_HF* = { }
6. *Suspected_with_LF* = { }
7. for Each *u* in *A* do
8. for each *f* in *All_Features* do
9. Compute f(u)
10. if *u* has high values in *HF* then
11. *Suspected_with_HF* = *Suspected_with_HF* U {*u*}
12. if *u* has low values in LF then
13. *Suspected_with_LF* = *Suspected_with_LF* U {*u*}
14. *Attackers* = *Suspected_with_HF* U *Suspected_with_LF*
15. *Most_trustworthy_users* = *A* - *Attackers*

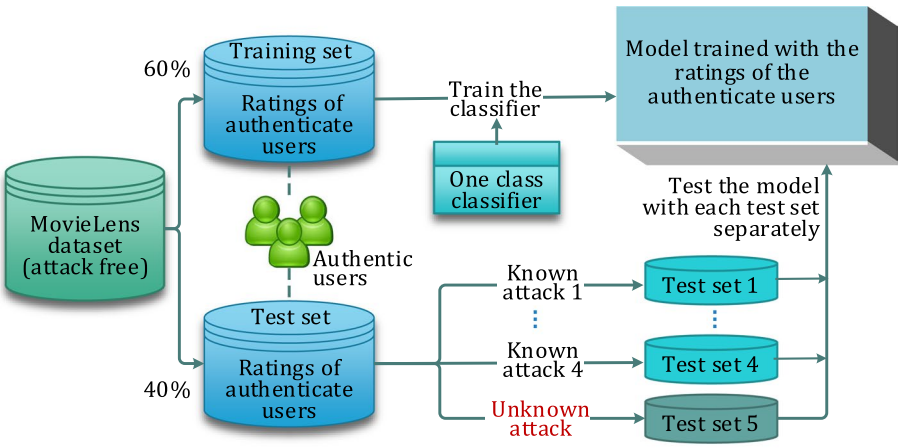


Fig. 7 The proposed solution that identifies all known and unknown attacks

5 Experimental Analysis

This section discusses the experimental details, including the dataset, methodology and performance analysis.

5.1 Dataset Description

The MovieLens dataset was collected to obtain a detailed solution that can detect all forms of attacks. Due to the lack of any normal attack functionality, the collected dataset is considered to be attack-free. As a result, it can be inferred that the MovieLens dataset only contains authentic user ratings. There are 943 users and1682 movies with 1,00,000 ratings.

5.2 Performance Metrics

The performance of different detection schemes is measured using standard metrics such as precision, recall, and F1-score, as shown in Table 9. Here, TP (true positive) shows to the count of attack profiles that are accurately identified as attackers. FP (false positive) represents the count of genuine profiles that are mistakenly identified as attackers. FN (false negative) indicates the count of attack profiles that are mistakenly identified as genuine users [7]. Furthermore, we calculated the hit ratio values subsequent to the completion of the detection process.

The overall performance of machine learning models is usually assessed using accuracy metrics. However, accuracy does not indicate various classification anomalies that might be present in the model. Assessing these anomalies is a basic requirement for models dealing with data that have an imbalanced cost associated with the various classes. In our problem, our objective is to defend a sensitive system that will not tolerate any attacker even if some of the authentic users are blocked. In this case, analyzing and improving the recall value will be the main objective of the analysis. Furthermore, to evaluate the overall performance, we consider the F1-score and accuracy to provide better insight into the performance of the proposed detection scheme.

5.3 Determining the Attack and Filler Size

To determine the appropriate attack size, we compared the biasing and shuffling effects and considered attack sizes where both values were 1. After that, on the considered attack size, a comparison of the hit ratio of several attack models is made to identify the filler size. We produced several dimensions of conventional push-type attacks. These varying attack sizes

Table 9 Performance metrics

Metric	Description	Ideal value	Calculation
Precision	Identifies how many attackers are predicted correctly from the total number of actual attackers	1	$\frac{TP}{TP+FP}$
Recall	Identifies how many attackers are predicted correctly from the total number of users	1	$\frac{TP}{TP+FN}$
F1-score	Is the harmonic mean of the precision and recall	1	$\frac{2*\text{Precision}*\text{Recall}}{\text{Precision}+\text{Recall}}$

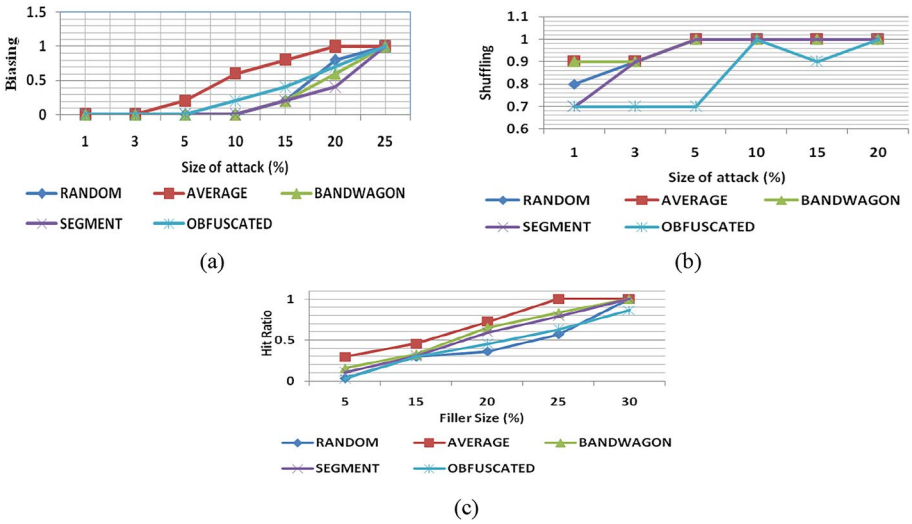


Fig. 8 **a** Biasing, **b** shuffling effects, and **c** the hit ratio of different attacks across various filler sizes on the overall top-N list

alter the top-N suggestion list due to their biasing and shuffling effects. The attack sizes are incrementally increased until the threshold t_s is reached, at which point the biasing, shuffle, and hit ratio values all become 1.

In Fig. 8, we can see how various attacks on the top-N list affect the bias, shuffling, and hit ratio. The biasing values of all attacks reach one at attack sizes of 20% and 25%. The shuffling value of all attacks reaches one at attack sizes of 5%, 10%, 15%, and 20%. Therefore, from the aforementioned observation, we select 25% for attack size because, at this attack size, all the attack models obtain biasing and shuffling values of one. Figure 8 shows the hit ratio of all attacks at various filler sizes for a 25% attack size. Figure 8 shows that at a 30% filler size, all attacks achieve the highest hit ratio. Therefore, for clearer observation in the experiments, we consider 25% and 28% attack sizes for known and unknown attacks, respectively. Furthermore, various filler sizes, i.e., 15%, 20%, 25%, and 40%, are considered for the selected attack sizes.

5.4 The Proposed Solution Approach

We used three different classifiers to detect known and unknown shilling attacks. For the experiment, we considered various attack sizes of different attack models, as discussed in Sect. 5.3. The experimental dataset comprised the ratings given by the set of most trustworthy users, divided into a 60% training set and a 40% test set. The attack profiles are generated on the test set. The classifiers were trained with the rating dataset given by the authentic users, labelled authentic. The testing technique involves the utilization of a test set, which consists of ratings provided by both genuine users and attackers.

Table 10 Performance of binary classifiers in detecting known and unknown (obfuscated) attacks at different attack sizes across various filler sizes based on P, R, F, and H

Attack type	Attack size (%)	Classifier	Filler size															
			15%				20%				25%				40%			
			P	R	F	H	P	R	F	H	P	R	F	H	P	R	F	H
Random attack	25	SVM	0.989	0.990	0.989	0	0.973	0.991	0.982	0	0.979	0.982	0.980	0	0.995	0.980	0.987	0
		J48	0.990	0.992	0.991	0	0.992	0.991	0.991	0	0.991	0.991	0.991	0	0.993	0.991	0.992	0
		Random Forest	0.978	0.981	0.979	0	0.985	0.996	0.990	0	0.997	0.976	0.986	0	0.982	0.975	0.978	0
		Naïve Bayes	0.980	0.989	0.984	0	0.997	0.996	0.996	0	0.977	0.989	0.983	0	0.991	0.992	0.991	0
Average attack	25	SVM	0.952	0.968	0.960	0	0.972	0.989	0.980	0	0.981	0.979	0.980	0	0.993	0.978	0.985	0
		J48	0.992	0.991	0.991	0	0.992	0.991	0.991	0	0.992	0.991	0.991	0	0.992	0.991	0.991	0
		Random Forest	0.980	0.979	0.979	0	0.983	1	0.991	0	0.995	0.971	0.983	0	0.980	0.972	0.976	0
		Naïve Bayes	1	1	1	0	0.996	0.994	0.995	0	0.974	0.987	0.980	0	0.993	0.990	0.991	0
Bandwagon attack	25	SVM	0.975	0.980	0.977	0	0.972	0.984	0.978	0	0.980	0.982	0.981	0	0.988	0.980	0.984	0
		J48	0.990	0.991	0.990	0	0.992	0.993	0.992	0	0.989	0.990	0.989	0	0.992	0.993	0.992	0
		Random Forest	0.978	0.980	0.979	0	0.983	0.986	0.984	0	0.991	0.977	0.984	0	0.983	0.986	0.984	0
		Naïve Bayes	0.997	0.989	0.993	0	0.996	0.993	0.994	0	0.980	0.992	0.986	0	0.995	0.990	0.992	0
Segment attack	25	SVM	0.982	0.970	0.976	0	0.972	0.989	0.980	0	0.982	0.978	0.980	0	0.995	0.980	0.987	0
		J48	0.991	0.990	0.990	0	0.992	0.991	0.991	0	0.993	0.991	0.992	0	0.990	0.993	0.991	0
		Random Forest	0.983	0.978	0.980	0	0.989	0.991	0.990	0	0.996	0.973	0.984	0	0.979	0.975	0.977	0
		Naïve Bayes	0.998	0.997	0.997	0	0.995	0.996	0.995	0	0.987	0.989	0.988	0	0.993	0.992	0.992	0
Obfuscated attack	28	SVM	0.685	0.440	0.536	0.205	0.695	0.554	0.617	0.219	0.742	0.578	0.650	0.321	0.797	0.612	0.692	0.482
		J48	0.652	0.471	0.547	0.150	0.679	0.523	0.591	0.223	0.712	0.567	0.631	0.352	0.732	0.689	0.710	0.451
		Random Forest	0.669	0.461	0.546	0.198	0.691	0.523	0.595	0.245	0.706	0.527	0.604	0.381	0.729	0.678	0.703	0.457
		Naïve Bayes	0.699	0.434	0.536	0.212	0.721	0.511	0.5981	0.261	0.745	0.586	0.656	0.319	0.789	0.611	0.689	0.487

5.4.1 Binary Classifiers: Trained with Known Attacks and Tested with Known and Unknown Attacks

The process of classifying the items of a collection into two different groups using a class label is known as binary classification. To classify authentic users and attackers using binary classifiers, along with authentic users, classifiers were also trained with known attackers, which were labelled as attackers. Table 10 presents the performances of different binary classifiers in detecting various attacks. We can observe that in the case of known attacks, the authentic users are well classified in this table due to their high precision, recall, and F1-score and low hit ratio, but in the case of unknown attacks, all of them attain low recall, F1-score, and high hit ratio. A low recall indicates that attackers from the test dataset were not detected very often, while a high hit ratio indicates that the top-N recommendation list was affected more significantly. Table 10 shows that the detection accuracy of the binary classifiers is significantly lower for unknown attacks than for known attacks.

5.4.2 Binary Classifiers with PU Learning: Trained with Ratings of Trustworthy Users and Tested with Known and Unknown Attacks

The detection of unknown attackers can be improved by using the PU learning technique with binary classifiers. The PU learning technique is used in different classifiers, such as SVM, J48, random forest, and naïve Bayes classifiers, to compare the performance [48]. The goal of PU learning is to train a binary classifier using a set of positive labelled and unlabelled samples. Combined with PU learning, the binary classifiers exhibit high recall values and low hit ratios. Table 11 shows that the same results for precision, recall, and accuracy across different filler sizes are obtained, indicating that they are independent of the filler size. Attack detection using PU learning has slightly decreased in the case of known attacks, while the detection accuracy of unknown attacks has significantly increased. However, the PU learning techniques SVM and J48 have become the most effective classifiers for known and unknown attacks, respectively, due to their high precision and recall compared to those of other classifiers.

5.4.3 One-class SVM: Trained with Ratings of Trustworthy Users and Tested with Known and Unknown Attacks

To further improve the recall, i.e., to detect all the attackers, we used the OSVM [49]. Table 12 shows an increased rate of attack detection for both known and unknown attacks, where known feature-based detection has failed. High recall values and a hit ratio of 0 denote the classification of the good number of attackers and the ineffective nature of attacks on the top-N list, respectively.

Table 11 Performance of PU learning with binary classifiers in detecting known and unknown (obfuscated) attacks at different attack sizes across various filler sizes based on P, R, F, and H

Attack type	Attack size (%)	Classifier	Filler size															
			15%				20%				25%				40%			
			P	R	F	H	P	R	F	H	P	R	F	H	P	R	F	H
Random attack	25	PU+SVM	0.743	0.853	0.794	0	0.743	0.853	0.794	0	0.743	0.853	0.794	0	0.743	0.853	0.794	0
		PU+J48	0.564	0.944	0.706	0	0.564	0.944	0.706	0	0.564	0.944	0.706	0	0.564	0.944	0.706	0
		PU+RF	0.581	0.848	0.690	0	0.581	0.848	0.690	0	0.581	0.848	0.690	0	0.581	0.848	0.690	0
		PU+NB	0.465	0.833	0.597	0	0.465	0.833	0.597	0	0.465	0.833	0.597	0	0.465	0.833	0.597	0
Average attack	25	PU+SVM	0.824	0.964	0.889	0	0.824	0.964	0.889	0	0.824	0.964	0.889	0	0.824	0.964	0.889	0
		PU+J48	0.795	0.854	0.823	0	0.795	0.854	0.823	0	0.795	0.854	0.823	0	0.795	0.854	0.823	0
		PU+RF	0.672	0.805	0.732	0	0.672	0.805	0.732	0	0.672	0.805	0.733	0	0.672	0.805	0.733	0
		PU+NB	0.420	0.733	0.534	0	0.420	0.733	0.534	0	0.420	0.733	0.534	0	0.420	0.733	0.534	0
Bandwagon attack	25	PU+SVM	0.553	0.910	0.688	0	0.553	0.910	0.688	0	0.553	0.910	0.688	0	0.553	0.910	0.688	0
		PU+J48	0.332	0.603	0.428	0	0.332	0.603	0.428	0	0.332	0.603	0.428	0	0.332	0.603	0.428	0
		PU+RF	0.474	0.861	0.611	0	0.474	0.861	0.611	0	0.474	0.861	0.611	0	0.474	0.861	0.611	0
		PU+NB	0.420	0.733	0.534	0	0.420	0.733	0.534	0	0.420	0.733	0.534	0	0.420	0.733	0.534	0
Segment attack	25	PU+SVM	0.780	0.898	0.834	0	0.780	0.898	0.834	0	0.780	0.898	0.834	0	0.780	0.898	0.835	0
		PU+J48	0.673	0.848	0.750	0	0.673	0.848	0.750	0	0.673	0.848	0.750	0	0.673	0.848	0.750	0
		PU+RF	0.664	0.744	0.702	0	0.664	0.744	0.702	0	0.664	0.744	0.702	0	0.664	0.744	0.702	0
		PU+NB	0.397	0.706	0.508	0	0.397	0.706	0.508	0	0.397	0.706	0.508	0	0.397	0.706	0.508	0
Obfuscated attack	28	PU+SVM	0.411	0.718	0.523	0	0.411	0.718	0.523	0	0.411	0.718	0.523	0	0.411	0.718	0.523	0
		PU+J48	0.661	0.918	0.769	0	0.661	0.918	0.769	0	0.661	0.918	0.769	0	0.661	0.918	0.769	0
		PU+RF	0.473	0.895	0.619	0	0.473	0.895	0.619	0	0.473	0.895	0.619	0	0.473	0.895	0.619	0
		PU+NB	0.398	0.643	0.492	0	0.398	0.643	0.492	0	0.398	0.643	0.492	0	0.398	0.643	0.492	0

Table 12 Performance of the OSVM classifier in detecting known and unknown (obfuscated) attacks at different attack sizes across various filler sizes based on P, R, F, and H

Attack type	Attack size (%)	Filler size															
		15%				20%				25%				40%			
		P	R	F	H	P	R	F	H	P	R	F	H	P	R	F	H
Random attack	25	0.530	0.992	0.691	0	0.534	0.989	0.694	0	0.532	0.994	0.693	0	0.529	0.991	0.690	0
Average attack	25	0.533	0.976	0.689	0	0.535	0.979	0.692	0	0.536	0.973	0.691	0	0.536	0.971	0.691	0
Bandwagon attack	25	0.539	0.993	0.699	0	0.533	0.997	0.695	0	0.537	0.994	0.697	0	0.538	0.996	0.699	0
Segment attack	25	0.531	0.994	0.692	0	0.527	0.988	0.687	0	0.528	0.989	0.688	0	0.530	0.990	0.690	0
Obfuscated attack	28	0.532	0.987	0.691	0	0.528	0.988	0.688	0	0.529	0.984	0.688	0	0.533	0.987	0.692	0

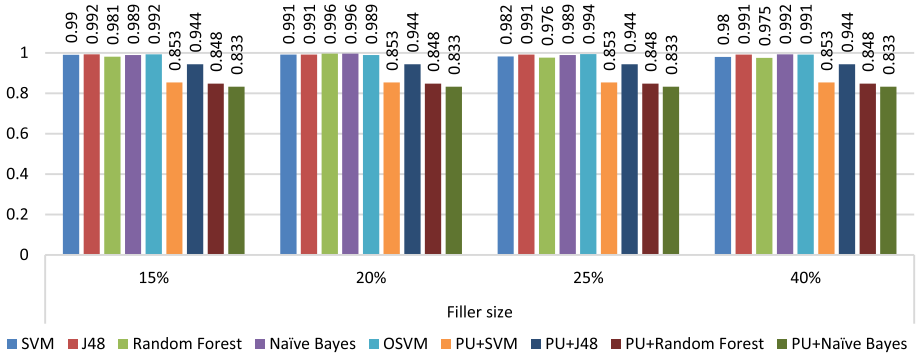


Fig. 9 Recall comparison of the proposed attack detection scheme with other classifiers in detecting random (known) attacks with a 25% attack across various filler sizes

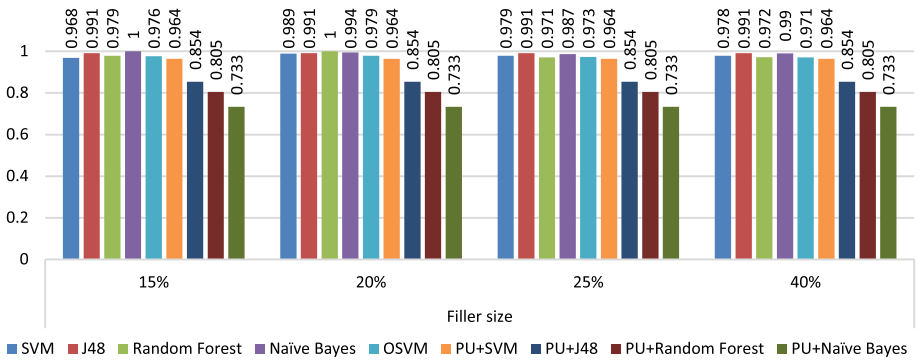


Fig. 10 Recall comparison of the proposed attack detection scheme with other classifiers in detecting average (known) attacks with a 25% attack across various filler sizes

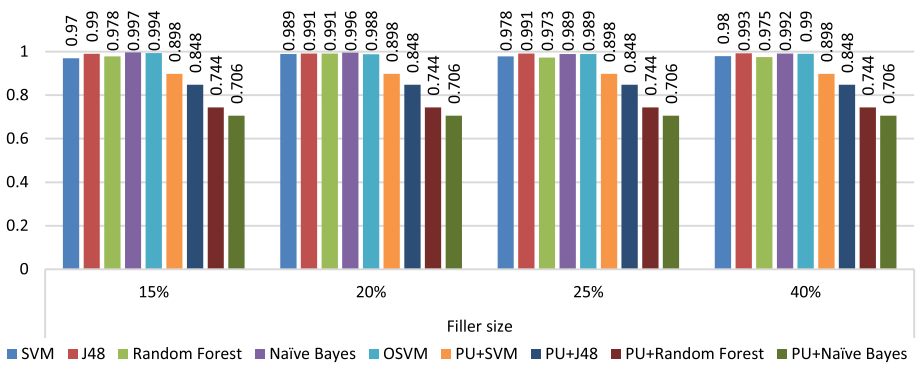


Fig. 11 Recall comparison of the proposed attack detection scheme with other classifiers in detecting a segment (known) attack with a 25% attack across various filler sizes

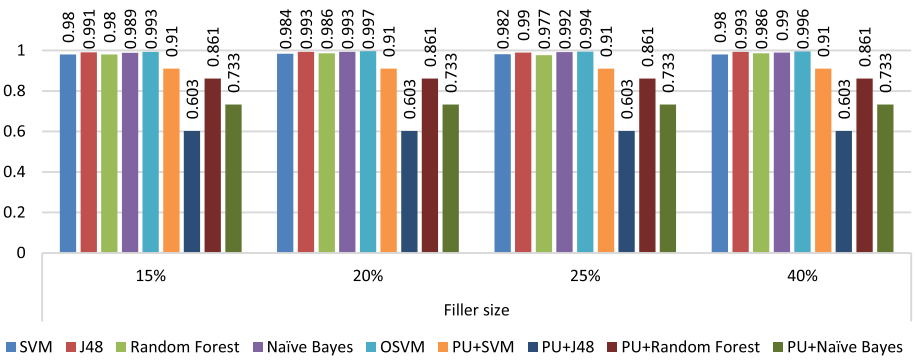


Fig. 12 Recall comparison of the proposed attack detection scheme with other classifiers in detecting bandwagon (known) attacks with a 25% attack across various filler sizes

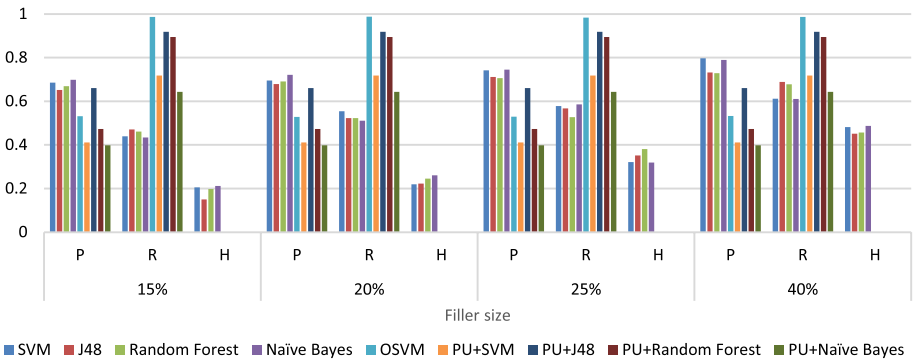


Fig. 13 Performance comparison of various classifiers in detecting obfuscated (unknown) attacks across various filler sizes

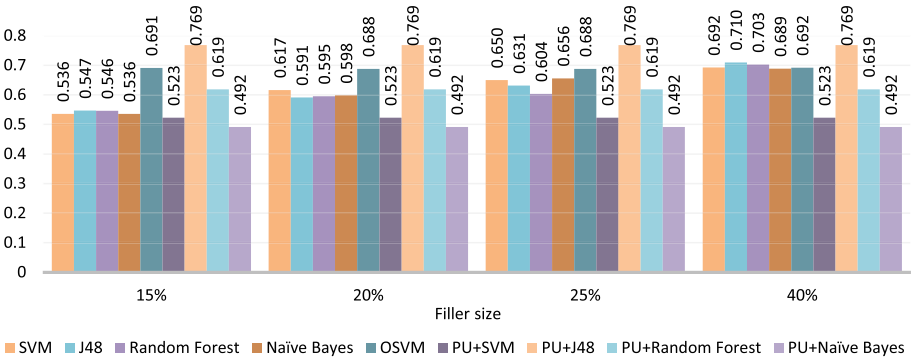


Fig. 14 F1-scores of different classifiers on obfuscation attacks for various filler sizes

6 Analysis and Discussion

Figures 9, 10, 11 and 12 compare the proposed shilling attack detection scheme with other classifiers commonly used for detecting known shilling attacks in terms of recall values. For this comparison, a 25% attack size was considered across various filler sizes (15%, 20%, 25%, and 40%). From these figures, we notice that all binary classifiers attain a slightly higher recall value than PU learning with binary classifiers and OSVM.

Figure 13 shows the comparison of all classifiers based on precision, recall, and hit ratio for the detection of an unknown attack. We notice that OSVM attained the highest recall value. For all types of attacks, the average recall obtained using the OSVM is 0.9876. This confirms that this approach can classify most attackers. Furthermore, for all the attacks, an absolute zero hit ratio was achieved, which indicates that there is no effect of the attacks on the top-N list, i.e., the top-N list remains unchanged. The combination of PU learning and binary classifiers also achieved a hit ratio of 0. However, OSVM provides a slightly lower precision than the binary classifiers but a higher precision than does the combination of binary classifiers and PU learning. A low precision value suggests that some of the authentic users might be wrongly classified as attackers.

Figure 14 shows a comparison of different classifiers for detecting an unknown attack based on the F1-score. At all filler sizes, PU learning with J48 achieves a high F1-score compared to other classifiers.

The experimental outcomes prove that the proposed approach is more successful in detecting unknown attacks. However, the classification rate for authentic users is not satisfactory. The aforementioned Tables 10, 11 12 and Figs. 9, 10, 11, 12, 13, and 14 illustrate that the proposed solution approach outperforms the feature-based detection scheme in terms of the detection accuracy of the unknown attack. We can conclude that the proposed solution approach with the OSVM is superior to any other classifier. However, we need to increase the detection accuracy of authentic users.

Although our goal was to achieve $H_D \in$AT$\rightarrow$ AT and $H_D \in$AU$\rightarrow$AU, our experimental results showed that we could achieve the first target completely and the second target to some extent. Our classifying approach could identify attackers as attackers with 100% accuracy, but the accuracy decreased when detecting authentic users correctly. In other words, we achieved $H_D \in$AT$\rightarrow$AT (high detection accuracy for identifying that an attacker occurs due to classifying all attackers as attackers) and $L_D \in$AU$\rightarrow$AT (low detection accuracy for identifying that an authentic user occurs due to classifying some authentic uses as an attacker).

Although detecting some authentic users as attackers decreases the overall accuracy of the system, it ensures that no attacker is passed through disguised as an authentic user. This makes the recommendation system completely attack-free.

7 Conclusion and Future Work

The problem of shilling attacks in recommendation systems may ruin their correctness and applicability. Feature-based detection has proven to be successful in eliminating standard known attacks. Unfortunately, this technique fails when a new attack with undefined features arrives. This paper addresses the limitations of recognized feature-based detection methods. We have demonstrated the ineffectiveness of established feature-based detection

methods in recognizing unfamiliar attacks. To overcome this problem, we propose a generalized solution that involves training a one-class classifier with the ratings of genuine users. This technique detects not only standard attacks but also obfuscation attacks. In fact, our proposed solution is able to detect any attack irrespective of its design or features. We used the MovieLens dataset, which has been proven to be authentic and attack-free, to establish our claim successfully. It should be noted that in evaluating the performance of our solution, we do not emphasize 100% detection of the attack profiles. Instead, detection is considered successful if it can reduce the number of attackers to a level where even if some attack remains undetected, the overall effect of an attack becomes negligible. Our proposed solution does exactly this.

The proposed method works with the precondition that the recommendation system should already have some genuine reviews and ratings from authentic users. However, one may raise the obvious and legitimate question of how to fulfil this criterion in every case. How do we populate the recommendation system, initially, with genuine ratings from authentic users? Of course, this is a nontrivial task. However, in today's digital world, abundant information is available everywhere. We have to draw off them judiciously. We can perform opinion mining in the web content, including social networks, blogs, and news portals, for the views of socially established people on different subjects, including products and services. Because of their social status, their opinions might be considered relatively bias-free and sincere. If these subjective opinions can be converted and mapped into a quantitative rating using statistical means, recommendation systems can be loaded with a decent amount of authentic and unadulterated ratings. For example, for a movie recommendation system, movie reviews from reputed dailies and magazines are twigged and converted into appropriately mapped numeric ratings. However, this is an extensive research challenge in itself, and therefore, researchers are encouraged to consider this particular concern.

Funding No funds, grants, or other support was received.

Data Availability Enquiries about data availability should be directed to the authors.

Declarations

Conflict of interest Regarding the topic discussed in this paper, the authors declared no relevant conflicts of interest.

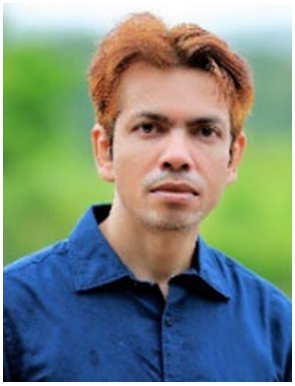
References

1. White, M.-C. (2017). Orbitz shows higher prices to mac users (**online**). Available: http://business.time.com/(2012)/06/26/orbitz-shows-higher-prices-to-mac-users/
2. Gibbs, S. (2017). Women less likely to be shown ads for high-paid jobs on Google, study shows (**online**). Available: https://www.theguardian.com/technology/(2015)/jul/08/women-less-likely-ads-high-paid-jobs-google-study. Accessed 11 May (2017)
3. Burke, R., Mobasher, B., Williams, C., & Bhaumik, R. (2006). Classification features for attack detection in collaborative recommender systems. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, *Philadelphia, PA, USA*.
4. Zhou, W., Wen, J., Koh, Y.-S., Alam, S., & Dobbie, G. (2014). Attack detection in recommender systems based on target item analysis. In *Proceedings of the International Joint Conference on Neural Networks*, *Beijing, China*.

5. Kumar, A., Garg, D., & Rana, P.-S. (2015) Ensemble approach to detect profile injection attack in recommender system. In *Proceedings of the international conference on advances in computing, communications and informatics*, *Kochi, India*
6. Williams, C., Mobasher, B., Burke, R., Sandvig, J., & Bhaumik, R. (2006). Detection of obfuscated attacks in collaborative recommender systems. In *Proceedings of the ECAI'06 workshop on recommender systems*, *Riva del Garda, Italy*.
7. O'Mahony, M., Hurley, N., & Silvestre, G. (2005). Recommender systems: Attack types and strategies. In *Proceedings of the twentieth national conference on artificial intelligence and the seventeenth innovative applications of artificial intelligence conference*, *Pittsburgh, Pennsylvania*.
8. Zhang, Z., & S. Kulkarni (2013). Graph-based detection of shilling attacks in recommender systems. In *IEEE international workshop on machine learning for signal processing*, *Southampton, UK*.
9. Mobasher, B., Burke, R. B. R., & Sandvig, J.-J. (2007). Attacks and remedies in collaborative recommendation. *IEEE Intelligent Systems, 22*(3), 56–63.
10. Mobasher, B., Burke, R., Bhaumik, R., & Williams, C. (2005). Effective attack models for shilling item-based collaborative filtering systems. In *Proceedings* of the *the WebKDD workshop*, Chicago, Illinois
11. Mobasher, B., Burke, R., Bhaumik, R., & Williams, C. (2007). Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness. *ACM Transactions on Internet Technology*. https://doi.org/10.1145/1278366.1278372
12. Kaur, P., & Goel, S. (2016). Shilling attack models in recommender system. In *International Conference on Inventive Computation Technologies*, *Coimbatore, India.*
13. Cao, J., Wu, Z., Mao, B., & Zhang, Y. (2013). Shilling attack detection utilizing semi-supervised learning method for collaborative recommender system. *World Wide Web, 16*(5–6), 729–748.
14. Wu, Z., Wu, J., Cao, J., Tao, D. (2012). HySAD: A semi-supervised hybrid shilling attack detector for trustworthy product recommendation. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, *Beijing, China*
15. Yang, Z., & Cai, Z. (2017). Detecting abnormal profiles in collaborative filtering recommender systems. *Journal of Intelligent Information Systems, 48*(3), 499–518.
16. Sheikhpour, R., Sarram, M., Gharaghani, S., & Chahooki, M. (2017). A survey on semi-supervised feature selection methods. *Pattern Recognition, 64*, 141–158.
17. Lv, S., Jiang, H., Zhao, L., Wang, D., Fan, M. (2013). Manifold based fisher method for semi-supervised feature selection. In *10th international conference on fuzzy systems and knowledge discovery, Shenyang, China.*
18. Wang, J., Yao, J., & Sun, Y. (2014). Semi-supervised local-learning-based feature selection. In *International joint conference on neural networks*, *Beijing, China*
19. Yang, Z., Cai, Z., & Yang, Y. (2017). Spotting anomalous ratings for rating systems by analyzing target users and items. *Neurocomputing, 240*, 25–46.
20. Lee, J., & Zhu, D. (2012). Shilling attack detection–a new approach for a trustworthy recommender system. *INFORMS Journal on Computing, 24*(1), 117–131.
21. Xia, H., Fang, B., Gao, M., Ma, H., Tang, Y., & Wen, J. (2015). A novel item anomaly detection approach against shilling attacks in collaborative recommendation systems using the dynamic time interval segmentation technique. *Information Sciences, 306*, 150–165.
22. Tabakhi, S., Moradi, P., & Akhlaghian, F. (2014). An unsupervised feature selection algorithm based on ant colony optimization. *Engineering Applications of Artificial Intelligence, 32*, 112–123.
23. Kumar, A., Garg, D., & Singh, P. (2017). Clustering approach to detect profile injection attacks in recommender system. *International Journal of Computer Applications, 166*(6), 7–11.
24. Hu, Y., Liu, K., & Zhang, F. (2017). Robust recommendation method based on shilling attack detection and matrix factorization model. In *2nd international conference on communications, information management and network security (CIMNS (2017))*, *Phuket, Thailand*
25. Zhu, P., Zuo, W., Zhang, L., Hu, Q., & Shiu, S. (2015). Unsupervised feature selection by regularized self-representation. *Pattern Recognition, 48*(2), 438–446.
26. Bhebe, W., Kogeda, O. (2015). Shilling attack detection in collaborative recommender systems using a meta learning strategy. In *International conference on emerging trends in networks and computer communications*, *Windhoek, Namibia*
27. Wang, Y., Zhang, L., Tao, H., Wu, Z., & Cao, J. (2015). A comparative study of shilling attack detectors for recommender systems. In *12th International Conference on Service Systems and Service Management*, *Guangzhou*, *China*
28. Singh, P., Pramanik, P., & Choudhury, P. (2019). A comparative study of different similarity metrics in highly sparse rating dataset. In V. Balas, N. Sharma, & A. Chakrabarti (Eds.), *Data*

Management, Analytics and Innovation. Advances in Intelligent Systems and Computing (Vol. 839, pp. 45–60). Singapore: Springer.
29. Singh, P.-K., Setta, S., Pramanik, P., & Choudhury, P. (2020). Improving the accuracy of collaborative filtering based recommendations by considering the temporal variance of top-N neighbors. In *International conference on innovative computing and communications*, *Ostrava, Czech Republic*
30. Singh, P., Pramanik, P., Debnath, N., & Choudhury, P. (2019). A novel neighborhood calculation method by assessing users' varying preferences in collaborative Filtering. In *Proceedings* of the *34th international conference on computers and their applications*, *Honolulu, Hawaii, USA*
31. Singh, P., Pramanik, P., & Choudhury, P. (2019). An improved similarity calculation method for collaborative filtering-based recommendation, considering the liking and dis-liking of categorical attributes of items. *Journal of Information and Optimization Sciences, 40*(2), 397–412.
32. Singh, P., Pramanik, P., & Choudhury, P. (2020). Collaborative filtering in recommender systems: technicalities, challenges, applications and research trends. In *New Age Analytics: Transforming the Internet through Machine Learning, IoT, and Trust Modeling* (pp. 183-215). Apple Academic Press
33. Jorge, A., Vinagre, J., Domingues, M., Gama, J., Soares, C., Matuszyk, P., & Spiliopoulou, M. (2016). Scalable online top-n recommender systems. In *International conference on electronic commerce and web technologies*, *Porto, Portugal*
34. O'Mahony, M., Hurley, N., Kushmerick, N., & Silvestre, G. (2004). Collaborative recommendation: A robustness analysis. *ACM Transactions on Internet Technology, 4*(4), 344–377.
35. Lam, S., & Riedl, J. (2004). Shilling recommender systems for fun and profit. In *Proceedings of the 13th international conference on World Wide Web, New York, USA*
36. Mobasher, B., Burke, R., Bhaumik, R., & Williams, C. (2007). Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness. *ACM Transactions on Internet Technology, 7*(4), 23.
37. Si, M., & Li, Q. (2020). Shilling attacks against collaborative recommender systems: A review. *Artificial Intelligence Review, 53*, 291–319.
38. Zhang, Z., & Kulkarni, S. (2014). Detection of shilling attacks in recommender systems via spectral clustering. In *17th international conference on information fusion*, *Salamanca, Spain*
39. Bilge, A., Ozdemir, Z., & Polat, H. (2014). A novel shilling attack detection method. *Procedia Computer Science, 31*, 165–174.
40. Mehta, B., & Nejdl, W. (2009). Unsupervised strategies for shilling detection and robust collaborative filtering. *User Modeling and User-Adapted Interaction, 19*(1–2), 65–97.
41. Chirita, P., Nejdl, W., & Zamfir, C. (2005). Preventing shilling attacks in online recommender systems. In Proceedings of the *the 7th annual ACM international workshop on Web information and data management*, Bremen, Germany.
42. Williams, C., Bhaumik, R., Burke, R., & Mobasher, B. (2006). The impact of attack profile classification on the robustness of collaborative recommendation. In *ACM SIGKDD conference on data mining and knowledge discovery, Philadelphia, PA, USA*
43. Zhang, Q., Luo, Y., Weng, C., & Li, M. (2009). A trust-based detecting mechanism against profile injection attacks in recommender systems. In *Third IEEE international conference on secure software integration and reliability improvement, NW Washington, DC, USA*
44. He, F., Wang, X., & Liu, B. (2010). Attack detection by rough set theory in recommendation system. In *IEEE international conference on granular computing*, *San Jose, CA, USA*.
45. Williams, C., Mobasher, B., & Burke, R. (2007). Defending recommender systems: Detection of profile injection attacks. *Service Oriented Computing and Applications, 1*(3), 157–170.
46. Zhang, F., & Chen, H. (2016). An ensemble method for detecting shilling attacks based on ordered item sequences. *Security and Communication Networks, 9*(7), 680–696.
47. Lee, J., & Zhu, S. D. (2012). Shilling attack detection—A new approach for a trustworthy recommender system. *INFORMS Journal on Computing, 24*(1), 117–131.
48. Narayan, R., Rout, J. K., & Jena, S. K. (2018). Review spam detection using semi-supervised technique. In P. Sa, M. Sahoo, M. Murugappan, Y. Wu, & B. Majhi (Eds.), *Progress in Intelligent Computing Techniques: Theory, Practice, and Applications Advances in Intelligent Systems and Computing* (Vol. 719, pp. 281–286). Berlin: Springer.
49. Deng, X., Li, W., Liu, X., Guo, Q., & Newsam, S. (2018). One-class remote sensing classification: One-class vs. binary classifiers. *International Journal of Remote Sensing, 39*(6), 1890–1910.

Dr. Pradeep Kumar Singh is currently working as Assistant Professor (Senior grade) in the Department of Computer Science and Engineering at Galgotias University, India. He obtained Ph.D. from the Department of Computer Science and Engineering, National Institute of Technology Durgapur (an Institute of National Importance under GoI, MHRD), India. He has worked in the organizing committee in the conference of ICBIM 2016, ICACSE 2019, and ICACSE 2021. In the past, he has completed his M.Tech degree from National Institute of Technology Durgapur in 2014. He has authored and co-authored about 18 international journal papers or conference proceedings. Currently his interested research areas are data mining, machine learning, and data analysis.



Dr. Pijush Kanti Dutta Pramanik is an associate professor in the School of Computer Applications and Technology at Galgotias University, Greater Noida, India. He has more than 13 years of teaching experience at the Graduate and Postgraduate levels. Pijush Kanti acquired a range of professional qualifications in the core and allied fields of Computer Science and Information Technology, viz., MIT, MCA, MBA (IT), M.Tech. (CSE), and M.Phil. (CS). He obtained Ph.D. in Computer Science and Engineering from the National Institute of Technology (an institute of national importance), Durgapur, India. He qualified UGC-NET in Computer Science and Applications as well as in Management. Pijush Kanti is actively engaged in research in the domains of healthcare informatics, crowd computing, IoT, mobile grid computing, edge computing, and recommendation systems and has published more than 70 research articles in WoS/Scopus indexed journals, edited books, and reputed international conferences. He also edited a few special issues for WoS/Scopus-indexed journals. He also serves as a reviewer for reputed journals like Frontiers in Energy Research, Frontiers in Public Health, IEEE Access, International Journal of Communication Systems, Education and Information Technologies, Mobile Information Systems, International Journal of Healthcare Information Systems and Informatics, etc. and several edited books.

Nilanjan Sinhababu is a Ph.D. research scholar at Indian Institute of Technology (IIT), Kharagpur, India. His broad field of research is IR-LLM. His other research interests include Recommendation Systems, Data Analytics and Crowd Computing. He received his MS (Master of Science by Research) degree from IIT Kharagpur in 2021, where he also served as a Junior Project Officer (JPO) in Sponsored Research and Industrial Technology (SRIC) project named 'Data Analytics and Reliability Assessment of Proof Information System' (DARA), sponsored by DRDO, PXE Chandipur, India. In addition, he is currently serving as a Teaching Assistant in 'Programming in Java' course offered by NPTEL, IIT Kharagpur. He received his B.Tech degree in Computer Science and Engineering from SETGOI, Durgapur, India

and completed his internship from National Institute of Technology, Durgapur. He served as a Technical Trainer at Ardent Collaboration, Durgapur. He secured top position in National Network Security Championship 2016 (CISCO) held at IIT Kanpur, India. He was selected as a participant of British Council Young Global Citizen Summit 2012 held in Odissa, India.

Dr. Prasenjit Choudhury is an Associate Professor in the Department of Computer Science and Engineering at National Institute of Technology, Durgapur, India. He has completed his Ph.D. in Computer Science and Engineering from the same institute. He has published more than 75 research papers in international journals and conferences. His research interest includes wireless network, information retrieval, data analytics, crowd computing, and recommendation systems.