Journal of Theoretical and Applied Information Technology
30th September 2025. Vol.103. No.18

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

NEURAL NETWORK-BASED ENSEMBLE APPROACH TO DETECT INFORMED ATTACKS FOR MULTIPLE TARGETS IN RECOMMENDER SYSTEMS

ASHISH KUMAR[1], YUDHVIR SINGH[2]
Department of Computer Science and Engineering
[1,2]UIET – Maharishi Dayanand University, Rohtak, India
[1]Priyadarshini Indira Gandhi Government College for Women, Jind, India
E-mail: [1]ashishpannu91@gmail.com, [2]dr.yudhvirs@gmail.com

ABSTRACT

Recommender systems (RSs) based on collaborative filtering (CF) are frequently used to deliver personalized services. However, profile injection attacks can easily affect them. As a result, attackers can easily manipulate the outcomes of these RSs. Informed attacks, which are also a type of profile injection attack, are very challenging to identify due to their high resemblance to genuine user profiles. Very limited research has been carried out to identify informed attack profiles, so there is a significant research gap to propose a technique to identify these profiles with good accuracy. In our experiment, we proposed a new data partition scheme for better training and testing of machine learning models. We injected informed attacks to promote and demote a specific item and a set of 10 items in the MovieLens dataset, and we proposed a novel neural network-based ensemble approach. Its performance is evaluated based on accuracy, precision, and recall by comparing it with that of a classical voting-based ensemble model (CVBEM), along with other supervised machine learning models in the detection of informed attacks. Robustness of performance is ensured by using k-fold cross-validation. We conducted our experiment in 24 attack scenarios with varying types of attacks, intentions, target item sizes, and attack sizes. Our study found that the proposed model's accuracy outperforms the other models' accuracy by a good margin of nearly 4% in most of the test scenarios. The CVBEM comes out as the second-best performer among all. The proposed model performs better not only in predicting biased users but is also more stable compared to traditional machine learning models.

Keywords: *Informed Attacks, Ensemble Model, Recommender System, Probe Attacks, Power User Attacks, Neural Network*

1. INTRODUCTION

Finding the most relevant information from the pool of data in a reasonable time is quite a challenging task. RSs solve this problem by generating high-quality recommendations in almost no time. RSs are now a crucial part of the majority of e-commerce platforms such as Flipkart, Amazon, EdX, YouTube, and Netflix [1, 2], etc. One of the most well-known and frequently used types of RSs is collaborative filtering (CF)-based RS. It is developed on the straightforward idea that if two people have similar tastes in the past, there's a good chance they will have similar tastes in the future as well. CF-based RSs are also of two types; one is an item-based RS and the other is a user-based RS [3]. Amazon uses item-based RS because it reduces the time complexity of computation compared to the user-based RS [4]. This is assumed because the count of users in a system is much higher than the count of items. Thus, finding the correlation among the items is faster. In a CF-based RS, the prediction of ratings is a two-step process. In the first step, the similarity is calculated between the target user for whom the prediction is to be made and all the remaining users of the system by using the Pearson correlation [5], as shown in Equation (1).

$$S_{x,y} = \frac{\sum_{i\in I}(r_{x,i}-\overline{r_x})(r_{y,i}-\overline{r_y})}{\sqrt{\sum_{i\in I}(r_{x,i}-\overline{r_x})^2}\sqrt{\sum_{i\in I}(r_{y,i}-\overline{r_y})^2}} \quad (1)$$

Here, $S_{x,y}$ represents how similar users x and y to each other. User x's rating of item i is represented by $r_{x,i}$. $\overline{r_x}$ is the average of all the ratings by user x. The subset of items I is i that are

rated by users x and y. The result of this equation falls between -1 to 1. To determine similarity in their research projects, BellCore [6] and LensGroup [7] both used Pearson Correlation. The second phase uses Equation (1) to filter out the top k nearest neighbors. The rating prediction is done by using Equation (2) [8]. $P_{x,i}$ is the user x's predicted rating for the item i.

$$P_{x,i} = \frac{\sum_{n \in Neighbor} (r_{n,i} - \overline{r_n}) s_{x,n}}{\sum_{n \in Neighbor} |s_{x,n}|} + \overline{r_n} \quad (2)$$

However, the cold-start problem is an issue with CF-based RSs [9, 10] and is highly susceptible to profile injection attacks [11, 12, 13]. In profile injection attacks, the attacker injects fake user profiles that look like genuine users to influence the rating prediction (push attack to promote and nuke attack to demote) for a particular item or set of items. The size of the attack and the size of the attack profile also play a significant role in affecting the predictions [14, 15]. Generally, attackers will automate this task, as it is hard to insert a large number of attack profiles manually. To avoid these profiles, the site owner tries to increase the cost of profile creation by making registration or a captcha mandatory before allowing users to enter the website [16]. Some types of attacks require more domain knowledge than others. The attacks that require more knowledge are hard to inject. For example, random attacks and bandwagon attacks are considered low-knowledge attacks, while segment attacks are considered high-knowledge attacks. Commercial platforms also widely use content-based and hybrid RSs [17]. A content-based RS works on the keywords and descriptions of items. It can be attacked by hacking, but it is not as vulnerable to profile injection attacks, as the item's description and keywords are filled in by the operator only on the platform. No permission is given to outsiders to modify the content of the product's description [18]. Identifying the attack profiles and neutralizing their effects from the recommendations generated is the need of the hour. The following are this study's main contributions:

- Focus on Informed Attacks: Unlike most prior works that mainly detect simple attacks (random, bandwagon, love-hate), this paper specifically targets informed attacks (probe and power-user attacks), which are much harder to detect because they closely resemble genuine users.
- New Data Partitioning Scheme: The authors propose a customized data partitioning approach to improve the training and testing process for machine learning models, which enhances the robustness of evaluation.
- Neural Network-Based Ensemble Model: Instead of a standard voting-based ensemble, the paper develops a novel neural network–based ensemble model that learns relationships between predictions from top-performing base models and actual labels, reducing overfitting/underfitting issues.
- Evaluation on Multiple Scenarios: The study conducts experiments across 24 attack scenarios with different attack sizes (1%, 10%, and 20%) and both single and multiple target items, providing a more comprehensive analysis compared to earlier works.
- Performance Improvement: The proposed approach consistently improves accuracy, precision, and recall — outperforming classical voting-based ensembles by around 4% accuracy on average.

This research builds directly on our earlier work [19], where we discussed the concept of informed attacks in recommender systems and provided preliminary insights into their characteristics. That study focused mainly on analyzing the impact of informed attacks on the recommender systems.

This study adopts a quantitative, experimental research approach using the MovieLens 100K dataset to detect informed attacks in recommender systems. The dataset was preprocessed to retain only user–item–rating data, after which probe and power-user attacks were artificially injected under 24 scenarios, varying by attack type, intention (push/nuke), target size (1 and 10 items), and attack size (1%, 10%, 20%). Key profile-based features such as degree similarity, length variance, RDMA, and WDMA were extracted to distinguish genuine and fake profiles. The data was randomly partitioned into 70% training and 30% testing sets, with further internal splitting for model selection. Five supervised machine learning models—Decision Tree, SVM, Random Forest, kNN, and Naïve Bayes—were trained and evaluated using 10-fold cross-validation, and the top three performers were selected. Their predictions were combined to train a neural network–based ensemble model, which was then tested on the unseen data. Performance was measured using accuracy, precision, and recall, and results showed that the proposed approach outperformed individual models and a classical

voting-based ensemble by approximately 4% in most scenarios.

This research work is organized into the following sections. The literature survey is presented in Section 2. Section 3 outlines the informed attack model, along with its types. Section 4 details the setup required for experiments, including the dataset's structure and various features that need to be extracted for identifying profiles, along with our proposed approach and the results of our experiment. The conclusion of our research, along with its future scope, is described in Section 5.

2. RELATED WORK

A few literary works in the domain of identifying biased user profiles from RSs are described in this section.

Aktukmak et al. [20] proposed a framework in which a sequential detection algorithm is taught after an expectation-maximization algorithm has trained a latent variable. This model produces a uniform distribution of users irrespective of age, gender, etc. Robustness is measured on the MovieLens, LastFm, and BookCrossing datasets. The mean detection delay (MDD) is used to evaluate the average delay in the sequential detection algorithm, while the area under the curve (AUC) metric is used to gauge the detection algorithm's accuracy. The mean AUC is reported to be between 0.1% and 1%. It has also been proven that the MDD of the proposed model performs slightly better than the generalized likelihood ratio algorithm. In the real world, attackers can adopt new strategies; in that case, this approach has limited robustness, and there is computational overhead in implementing the algorithm.

Zhang et al. [21] solved the problem of overtraining and reduced the cost of this process by proposing an unsupervised method based on divide and conquer to detect biased user profiles. It divides the attacks into standard and obfuscated behavior attacks. After that, it divides the profiles into clusters based on their extracted features. The proposed method does not require any prior knowledge. The authors used precision to measure their effectiveness on MovieLens-100K and the Netflix dataset for various types of attacks. The authors proved that their proposed techniques perform better in terms of accuracy and require less computational time in the identification of standard and obfuscated behavior attacks.

Fuzhi et al. [22] proposed an approach to detect group shilling attacks based on the graphs from the RSs based on collaborative filtering. The proposed approach is a three-step process involving first constructing a chart depicting user connections derived from rating behaviors and identifying a low-dimensional vector representation for every graph node. The candidate groups are obtained using the k-mean++ clustering algorithm. Finally, the suspected groups are identified based on the clustering algorithm, namely Ward's hierarchy. For evaluation purposes, F1-measure, recall, and precision metrics are used. The authors have proved that their proposed technique performs better on Amazon and Netflix datasets as compared to other baseline techniques, and computational cost is also improved, but it has less stability. However, this approach is ineffective in detecting shilling attacks if the graph is poorly constructed in the case of sparse data or noisy interactions.

Barbieri et al. [23] proposed a generative approach to introduce biased user profiles into the system. It generates new biased user profiles with minimal variation from genuine users in the system by utilizing the generative model, specifically a variational autoencoder, on the 100K MovieLens dataset, demonstrating that these biased user profiles are hard to detect. These profiles are converted into biased ones by rating the target item. The authors demonstrated that their approach outperforms other model-based systems by 3% to 5% at lower attack sizes. However, the larger the attack size, the poorer this approach performed. This model may not be flexible enough to detect attacks in real time if the system's underlying dynamics change quickly, leading to missed attack detection.

Rezaimehr et al. [24] designed a robust time and trust RS (T&TRS), which used a clustering algorithm to detect biased users. It determines the reliability value for all item ratings and classifies them as biased or unbiased. T&TRS considered rating time and implicit and explicit trust among users while creating the weighted user-user network, and it finds communities as users' nearest neighbors to anticipate unknown item ratings. It removes the doubtful users and items from the rating matrix by using the clustering algorithm and generates the top k items for the users based on their interests. The authors show that after the identification of biased users and items, the precision of recommendations increases in comparison to KMCF-U, KMCF-I, and TOTAR.

The existing work is mainly focused on identifying profile injection attacks, such as bandwagon, reverse bandwagon, random, average, and love-hate attacks from the RSs. Additionally,

these studies concentrate on promoting and demoting a specific item only. In this paper, we introduced a new type of attack, i.e., informed attacks, which is a highly unexplored type of attack. Additionally, we not only inject the attack to promote or demote a single target item, but also a group of target items as well. These attack profiles appear more similar to genuine users, while other attack types show sufficient deviation in the characteristics of genuine users. The existing machine learning models used for profile identifications have limitations, including poor accuracy against informed attack models, and their accuracy is highly unstable. To resolve these issues, a novel neural network-based ensemble approach is proposed, and its performance is evaluated against other models.

3. INFORMED ATTACK MODELS

The attacker modifies the system by injecting false user profiles and ratings that influence the system's suggestions to genuine users, ultimately leading them to make incorrect decisions. These fake profiles affect genuine users in a large number. An attack type that requires the least system knowledge is easy to implement, and vice versa; however, a low-knowledge attack generally causes less impact on the recommendations made by the system. This paper discusses informed attack models, which are highly knowledgeable attacks. Informed attack models are mainly of two types: one is a probe attack, and the other is a power user attack.

3.1 Probe Attack

In probe attacks, the attacker introduces some malicious user profiles. These profiles give ratings to seed items, which are nothing but randomly selected items. The system's average rating for a certain item is assigned to these seed items. These attack profiles also give ratings to the target item. A push attack provides the highest rating possible; conversely, in a nuke attack, the target item is assigned the lowest rating possible. The relationship between these biased user profiles and actual users is used to make recommendations for actual users about the target item. The attacker can gradually learn about the system's rating distribution through the probe attack [25]. The probe attack's algorithm is as follows:

Probe Attack Algorithm:

1. N number of fake user IDs are created (based on attack size).
2. Repeat steps 3 to 5 for each fake user.
3. A set of seed items is selected randomly.
4. A fake user ID assigns the seed item's average rating to that item.
5. Fake user ID assigns a maximum and minimum rating in case of a push and nuke attack, respectively to the target item.
6. k nearest genuine users are calculated for each biased user by Equation 1.
7. Biased recommendations are generated for k genuine users by using Equation 2.

3.2 Power User Attack

Users with the highest association with other users within the system are known as power users. Hence, they have a large number of neighbors [26]. Generally, many items in the system received ratings from these users, and when correlation is calculated, they have something in common with other users. Correlated users also influence the recommendations of other users in the system. The attacker chooses a group of system power users to serve as the attack profiles. The size of attacks determines the number of users in this set. The target item or set of target items received the maximum possible rating in the case of a push attack by the power users. Similarly, the minimum possible rating is assigned to the target item or set of target items in the case of a nuke attack [27, 28].

Power User Attack Algorithm:

1. Calculate the ratings (count) given by each user.
2. Filter out the top N users (power) who have given maximum ratings.
3. For each power user, assign the target item the maximum and the minimum rating in push and nuke attack.
4. Finally, biased recommendations are generated for genuine users based on power users.

One advantage of the probing attack over the power user attack is that it requires less domain expertise. RS selects only a small set of seed items randomly. After that, it selects additional items and generates ratings for them [29].

4. EXPERIMENTAL EVALUATION

The prediction shift in the ratings of the target item is calculated to measure the efficiency of an attack. These attack profiles are identified to

counteract the impact of the attack, and their predictions are not considered when making recommendations. In the last one and a half decades, several studies have been carried out to detect these attack profiles. These anomalies can be detected by using supervised and unsupervised learning models [30]. Supervised models require labeling of sample data. They are used when we know the type of attack. Unsupervised models are used for unknown types of attacks. In general, supervised models generate higher accuracy than unsupervised models. To increase the accuracy of identifying attack profiles, we use a specially designed ensemble method.

4.1 The Dataset

The MovieLens dataset is used in this study [31]. The dataset does not have demographic information on the users, and each user has given ratings for at least 20 movies. For our research, we have removed the timestamp, genre of the movie, etc., and kept only users, movies, and ratings in the dataset. The dataset has a total of 100836 ratings given to 9,724 movies by 610 users. All the user IDs are in the range of 1 to 610. There are 10 possible ratings, ranging from 0.5 to 5, with 0.5 being the lowest and 5 being the highest rating. A user has given a maximum of 2,698 ratings in the system and a minimum of 20. The most popular movie ID is 318 among all movies, as it has received the maximum number of 5 ratings. The average user-submitted rating is 165.305, while the average rating for each movie is 10.369. The difference between the two closest ratings is 0.5. The users have given a rating of 4 most of the time. Figure 1 describes the distribution of the ratings in the dataset.

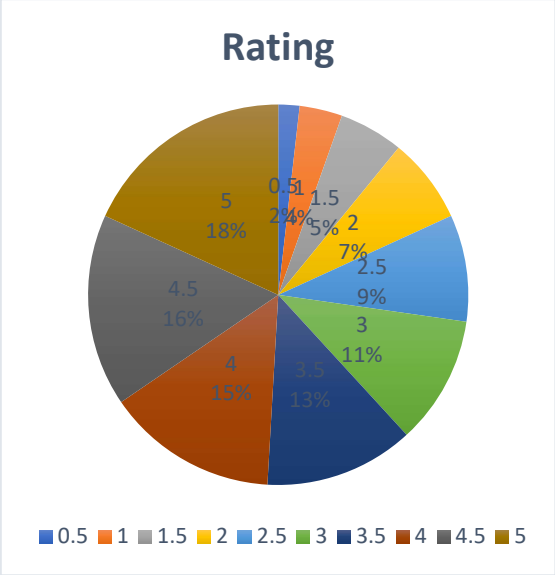

Figure 1: Distribution of Ratings In The MovieLens 100K Dataset.

4.2 Attributes to Detect Attacks

These are user profile properties that help us separate genuine users from fraudulent user profiles. There are several attributes based on which user profiles can be distinguished; some of these are [32]:

1) Degree similarity with top k neighbors ($DegSim_x$): it determines the similarity between users x and its k nearest neighbors as per Equation (3).

$$DegSim_x = \frac{\sum_{v=1}^{k} sim_{x,y}}{k} \quad (3)$$

2) Length variance ($LengthVar_x$): The basic idea of using this attribute is that a genuine user does not give ratings to thousands of items. If a user gives too many ratings in the system to maximize its impact, e.g., it gives a rating to the target item either maximum/minimum, and ratings to too many other items based on the property of the attack. It's possible that this user profile has not rated the items honestly. The system will mark it as a fake user profile and will not consider ratings from that user in the system. The variance in the length of the user x and the average length of other users in the system is measured by this attribute as given by Equation (4).

$$LengthVar_x = \frac{|l_x - \bar{l}|}{\sum_{k \in U}(l_k - \bar{l})^2} \quad (4)$$

Here, l_x is the length of the user profile x and $\bar{l}$ is the average length of all the user profiles in the system.

3) Rating deviation from the mean agreement (RDMA): the average variance of the ratings for

every item that user x has rated is measured by this attribute as per Equation (5).

$$RDMA_x = \frac{\sum_{i=0}^{N_x} \frac{|r_{x,i} - \overline{r_i}|}{t_i}}{N_x} \quad (5)$$

User x has given N_x ratings x. Item i has received t_i ratings. $r_{x,i}$ is the rating item i received by the user x.

4) Weighted deviation from mean agreement (WDMA): This attribute can be measured by Equation (6).

$$WDMA_x = \frac{\sum_{i=0}^{N_x} \frac{|r_{x,i} - \overline{r_i}|}{|t_i|^2}}{N_x} \quad (6)$$

4.3 Experimental Methodology

The paper follows a multi-stage conceptual model; initially dataset contains genuine users along with programmatically injected informed attack profiles. From this dataset, features are extracted such as DegSim, LengthVar, RDMA, and WDMA. Multiple supervised models are trained and tested on this specially partitioned dataset, and the top three performers are picked up, which are further combined with a neural network to generate our final ensemble model. The results are k-fold cross-validated and analyzed on the basis of precision, recall, and accuracy.

The attacks in this research are implemented in 24 distinct scenarios. Both attacks are injected to promote and demote the target items, and for both intentions, the number of target items is 1 and 10. For each attack scenario, the attacks are injected with various sizes of attack, i.e., 1%, 10%, and 20%, as summarized in Table 1.

Table 1: The Attack Scenarios of the Informed Attack Models.

Name of Attack	**Objective**	**Fixed Attribute** (Size of Target Item)	**Variable Attribute** (Size of Attack)
Probe Attack	Promote	1	1%, 10%, and 20%
		10	
	Demote	1	
		10	
Power User Attack	Promote	1	
		10	
	Demote	1	
		10	

The proposed approach is a three-stage process: (1) partition and randomization of data, (2) selection of models and their training and testing, (3) training and testing of the neural network. In the first stage, the shuffled dataset is divided into two parts; the training data (SET-I) is 70% of the dataset, and the testing data (SET-II) is the remaining 30% of the dataset.

In the second stage, five supervised machine learning models, namely Decision Tree, Support Vector Machine (SVM), Random Forest, k-nearest neighbors (kNN), and Naïve Bayes models, are selected. The SET-I is further divided into 70% training data (SET-III) and 30% testing data (SET-IV) without compromising the testing data, i.e., SET-II. The selected models are trained on SET-III and tested on SET-IV. For the final ensemble model, the top three performing models are chosen based on their accuracy on SET-IV.

In the third and final stage, the predictions of the selected models and their actual values are now used as training data (SET-V) for the neural network. It establishes a relationship between the three models predicted and the actual values. The ensemble model is tested on SET-II. Figure 2 describes the flow chart of the proposed approach.

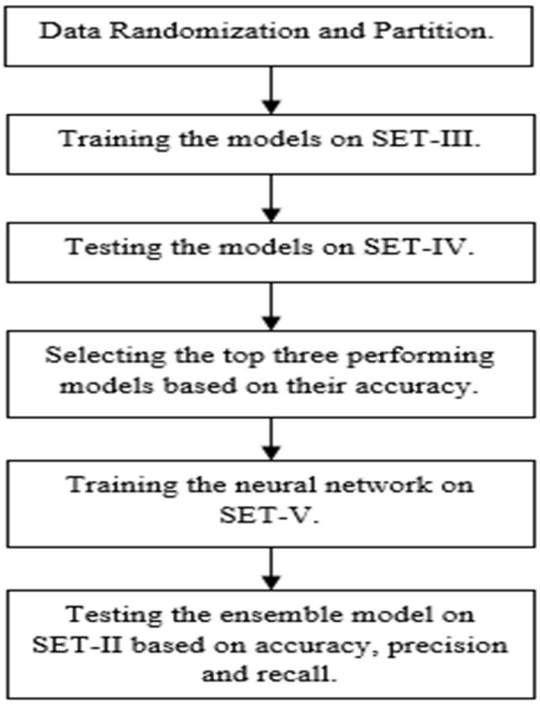

Figure 2: The Proposed Approach's Flow Chart.

Table 2 describes the basic details of supervised machine learning models used in our research. All the experimentation work is done in the R language.

Table 2: Basic Details of Classification Models Used.

Sr. No.	Model	Package	Method	Parameter	Description
1.	Decision Tree	rpart	rpart	MaxDepth=25, minSplit=15	A flow chart-like structure is used for both categorical and continuous variables.
2.	SVM	e1071	svm	Epsilon=0.2, degree=5, nu=1	Used to carry out classification, regression, and density estimation.
3.	Random Forest	randomForest	randomForest	Ntree=500, mtry=10	Used for classification and regression. Developed by aggregating the trees.
4.	kNN	class	knn	K=10	Used for both regression and classification. It is a nonparametric algorithm.
5.	Neural Network	neuralnet	neuralnet	Rep=3,algorithm="rprop+", stepmax=100000	Used to recognize patterns like humans.
6.	Naïve Bayes	e1071	naiveBayes	Na.action=na.pass, laplace=1	It is a non-linear classification algorithm

4.4 Performance Evaluation

The five supervised machine learning models tested on SET-IV, as mentioned above, are measured based on their accuracy. The Neural Network, a CVBEM, along with the top three performing supervised machine learning models, is tested on SET-II and evaluated based on accuracy, precision, and recall [33].

1) Accuracy (Acc): It is measured in percentage and defined as how accurately the machine learning model predicts the correct outcome as per Equation 7.

$$\text{Acc\%} = \frac{\text{Correct predictions}}{\text{All predictions}} * 100 \quad (7)$$

2) Precision (P): It is all about how good the model is at predicting a specific category as per Equation 8.

$$P = \frac{\text{Correct predictions of a specific category}}{\text{All predictions of that category}} \quad (8)$$

3) Recall (R): It measures how correctly the machine learning model identifies instances of a specific category from all instances of that category, as per Equation 9.

$$R = \frac{\text{Correct predictions of a specific category}}{\text{All instances of that category}} \quad (9)$$

4.5 Results

Accuracy is the main parameter based on which we select the top three performing models. The experiment is conducted in a total of 24 different attack scenarios. Table 3 shows the average accuracy of five supervised machine-learning models in these scenarios. To ensure the models do not suffer from the problem of overfitting and underfitting, *k*-fold cross-validation is used in this research [34]. The accuracy of the top three performing models is represented by bold values. At the end of 10 iterations, the average of all iterations is calculated, and based on the average, the accuracies of the models are compared. It is observed that Naïve Bayes gives the best accuracy; after that, Random Forest and SVM performed better than the other two remaining models. It is also observed that in three cases, the Decision Tree performed better than SVM, but the kNN performed the least among all the models. The Naïve Bayes, Random Forest, and SVM are the top three performers in terms of accuracy. Therefore, these models are used for the generation of the ensemble model. The predictions of these top-performing models and actual target values, along with the extracted features, are used to generate the SET-V, the neural network is trained on this set.

Table 3: The Average Accuracy of 10-Fold Cross-Validated Various Models.

Name of Attack	Objective	Fixed Attribute (Size of Target Item)	Variable Attribute (Size of Attack)	Decision Tree	SVM	Random Forest	kNN	Naïve Bayes
Powe-user attack	Promote	1	1	78.91	**81.11**	**81.49**	74.57	**83.52**
			10	79.46	**82.51**	**83.97**	75.72	**83.11**
			20	80.57	**84.39**	**85.03**	78.52	**85.00**
		10	1	**76.43**	76.08	**79.54**	71.64	**82.25**
			10	77.85	**78.95**	**82.69**	74.93	**84.46**
			20	78.27	**82.79**	**82.10**	73.00	**85.16**
	Demote	1	1	77.39	**81.74**	**81.74**	76.35	**84.27**
			10	79.24	**81.18**	**83.81**	76.19	**86.31**
			20	80.09	**82.19**	**84.72**	77.07	**86.79**
		10	1	76.34	**77.12**	**78.30**	71.63	**82.46**
			10	77.59	**78.73**	**82.84**	74.02	**84.69**
			20	**80.80**	80.44	**85.52**	73.40	**83.07**
Probe attack	Promote	1	1	79.62	**82.30**	**86.16**	73.63	**85.73**
			10	81.19	**84.71**	**86.16**	75.91	**86.06**
			20	81.96	**84.22**	**88.89**	75.99	**88.07**
		10	1	78.33	**79.11**	**80.76**	71.50	**82.61**
			10	80.42	**81.98**	**81.38**	73.76	**84.54**
			20	**81.70**	81.61	**81.32**	74.38	**85.11**
	Demote	1	1	78.33	**82.57**	**85.73**	74.08	**85.44**
			10	80.93	**86.89**	**86.88**	74.51	**86.04**
			20	82.98	**85.26**	**88.70**	75.07	**88.58**
		10	1	78.55	**79.35**	**80.01**	71.11	**83.67**
			10	79.65	**81.24**	**81.35**	73.57	**83.08**
			20	80.48	**83.66**	**82.79**	74.16	**85.70**

The structure of the SET-V is shown in Table 4 along with five sample values, which show the attributes of the original dataset and predictions generated by the Naïve Bayes, Random Forest, and SVM.

Table 4: Structure of SET-V with Sample Values.

userID	DegSim	LengthVar	RDMA	WDMA	Actual Target Value	Naïve Bayes Prediction	Random Forest Prediction	SVM Prediction
10	0.2535	4.2535	0.0235	0.0012	1	1	0	1
120	-0.5210	8.1425	0.5723	0.1241	1	1	1	1
52	0.8241	1.2145	0.1285	0.0175	0	0	0	1
675	0.1752	3.846	0.2751	0.0121	1	0	1	1
89	-0.2147	2.1241	0.1856	0.0195	0	0	0	0

The neural network is trained on the SET-V. It establishes a relationship between the predicted and actual target values of the top three performing models. The trained neural network is tested on SET-II. The predicted target value is compared with the actual target value to measure accuracy. The CVBEM, along with the top three performing models from the previous step, is compared with the proposed model. The precision and recall of our proposed model, the top three performing models, and the CVBEM are shown in Table 5. It is observed that both precision and recall are higher in our proposed model compared to others, which validates the better predictions of our proposed model. To ensure we get more accurate values, we use k-fold cross-validation.

Table 5: Precision and Recall of Our Proposed Model and Other Models in Different Scenarios.

Name of Attack	Objective	Fixed Attribute (Size of Target Items)	Variable Attribute (Size of Attack)	SVM		Random Forest		Naïve Bayes		CVBEM		Proposed model	
				P	R	P	R	P	R	P	R	P	R
Power-user attack	Promote	1	1	0.75	0.76	0.80	0.83	0.82	0.84	0.87	0.92	0.95	0.91
			10	0.81	0.82	0.80	0.80	0.86	0.87	0.91	0.88	0.91	0.93
			20	0.76	0.77	0.85	0.81	0.83	0.85	0.90	0.86	0.96	0.97
		10	1	0.79	0.82	0.80	0.85	0.82	0.83	0.89	0.91	0.94	0.92
			10	0.77	0.81	0.83	0.78	0.83	0.86	0.90	0.92	0.92	0.95
			20	0.79	0.76	0.78	0.81	0.84	0.85	0.89	0.91	0.94	0.95
	Demote	1	1	0.79	0.76	0.85	0.84	0.86	0.81	0.89	0.90	0.95	0.97
			10	0.76	0.75	0.79	0.84	0.83	0.86	0.91	0.86	0.90	0.91
			20	0.79	0.76	0.82	0.81	0.82	0.81	0.88	0.87	0.93	0.93
		10	1	0.81	0.79	0.83	0.85	0.86	0.85	0.90	0.86	0.97	0.94
			10	0.77	0.77	0.84	0.85	0.81	0.87	0.90	0.92	0.97	0.92
			20	0.83	0.78	0.83	0.83	0.86	0.87	0.92	0.88	0.96	0.96
Probe attack	Promote	1	1	0.75	0.82	0.83	0.80	0.83	0.84	0.92	0.88	0.93	0.94
			10	0.81	0.83	0.78	0.81	0.88	0.81	0.92	0.86	0.93	0.95
			20	0.79	0.81	0.82	0.85	0.88	0.82	0.87	0.90	0.95	0.95
		10	1	0.75	0.82	0.83	0.82	0.86	0.87	0.90	0.92	0.93	0.96
			10	0.77	0.78	0.85	0.82	0.87	0.81	0.86	0.87	0.93	0.91
			20	0.81	0.81	0.79	0.80	0.88	0.85	0.85	0.92	0.97	0.94
	Demote	1	1	0.82	0.78	0.86	0.80	0.82	0.87	0.87	0.85	0.93	0.96
			10	0.81	0.81	0.82	0.85	0.86	0.86	0.88	0.87	0.91	0.91
			20	0.82	0.79	0.84	0.85	0.85	0.82	0.88	0.87	0.94	0.93
		10	1	0.80	0.80	0.78	0.79	0.83	0.87	0.92	0.86	0.97	0.97

			10	0.82	0.75	0.85	0.79	0.87	0.82	0.90	0.90	0.97	0.92
			20	0.75	0.75	0.81	0.78	0.88	0.84	0.87	0.90	0.96	0.92

Figures 3 and 4 show the accuracy comparison of the top three performing models from the previous step, the proposed model, and the CVBEM in the case of a probe attack for both intentions, i.e., promoting and demoting a target item, and target items of size 10, respectively. The results clearly show that the proposed model's accuracy is much better than that of the CVBEM and the other three models, as it establishes a relationship between the actual target value and the value predicted by the top three performing models. It is observed that as the number of target items increases from 1 to 10, the accuracy of the models decreases, including our proposed model, because it increases the correlation of biased users with genuine users; as a result, it becomes difficult for the models to detect them. The accuracy of the models is almost stable, as expected, in push and nuke attacks at the same attack size.

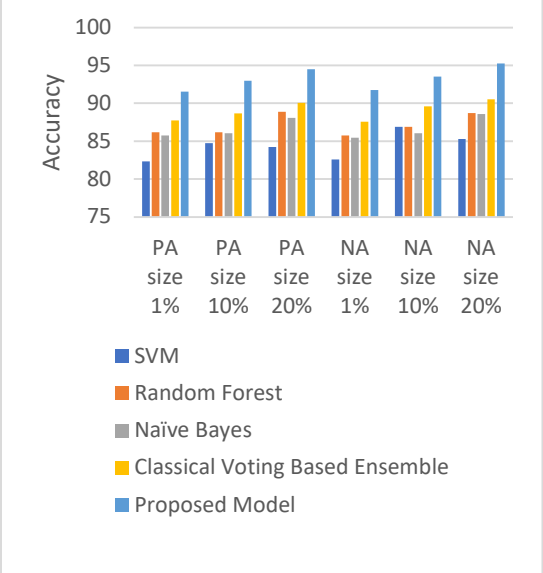

Figure 3: Accuracy Comparison of Models for Probe Attack in Push (PA) and Nuke (NA) Intention When Target Item Size is 1.

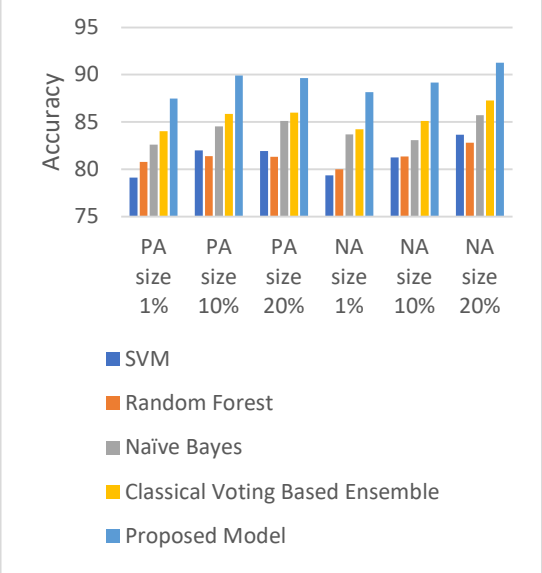

Figure 4: Accuracy Comparison of Models for Probe Attack in Push (PA) and Nuke (PA) Intention When Target Item Size is 10.

Figures 5 and 6 describe the accuracy comparison of SVM, Random Forest, Naïve Bayes, CVBEM, and our proposed model in the case of power user attacks for both intentions at target item sizes 1 and 10, respectively. It is observed that when the attack size is 1%, the accuracy of our proposed model is below 90% in all four cases, but as the attack size approaches 10%, the accuracy of our proposed model crosses 90%. It is also observed that the accuracy of our proposed model in the case of a power-user attack is lower than the probe attack because the power-user attack is more knowledgeable, and its generated user profiles are closer to those of genuine users.

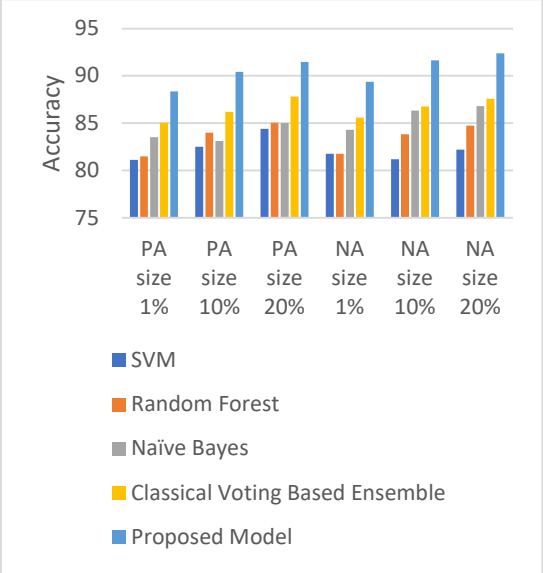

Figure 5: Accuracy Comparison of Models for Power User Attack in Push and Nuke Intention When Target Item Size is 1.

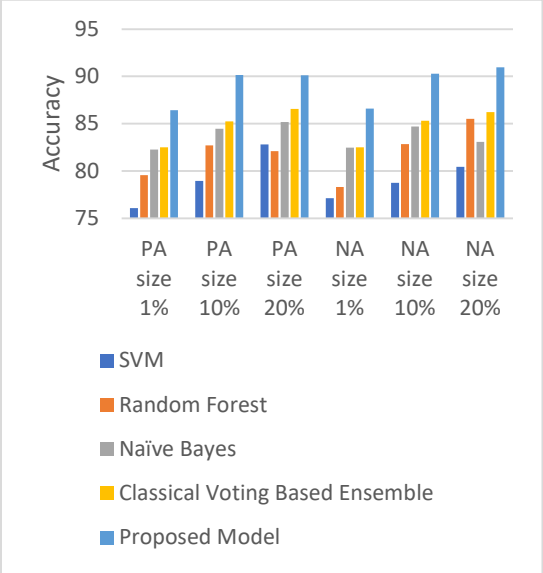

Figure 6: Accuracy Comparison of Models for Power User Attack in Push and Nuke Intention When Target Item Size is 10.

The robustness of the proposed model is ensured by using the *k*-fold cross-validation due to its simplicity and randomness compared to the other validation methods. The training and testing processes are cross-validated "*k*" times on different samples of data. Each time, data of the same size is selected randomly, and their results are compared. By using this validation, a better understanding of the performance of the model is developed, although minor deviations can happen in real scenarios.

For experimental purposes, the value of "*k*" is 10 in this research; that is, the training and testing of the model are repeated 10 times in all scenarios. The CVBEM also performs better than the other three models, except for the proposed model, in most iterations for measuring accuracy, precision, and recall. Through the analysis, it is clear that most of the attacks are identified correctly by our proposed model.

5. CONCLUSION AND FUTURE SCOPE

The primary objective of this research is to develop a more robust ensemble model for detecting informed attacks that aim to promote or demote a single item or a group of items. To achieve this, we designed a new data partition scheme for the training and testing of existing as well as our proposed model. Our proposed model is not just a regular ensemble model, but is developed by combining several other best-performing models on a more randomized and scalable dataset that yields more accurate and stable results than the CVBEM. To increase the effectiveness of the attack and make it hard to identify, we targeted a group of items so that the resemblance of both the biased and genuine users could be increased. This is the major research gap that we found in the previous studies in this domain. Our proposed model performs very well in this case as well. Although its performance decreases in this case, it is still better than its peers. We ensured the robustness of our proposed model by using *k*-fold cross-validation. In most of the attack scenarios, we achieved an accuracy of nearly 90% and surpassed the closest CVBEM by approximately 4%. The proposed model takes slightly more time in computation compared to the CVBEM, but that is negligible compared to the difference between the accuracies of both models.

The proposed approach is tested on only the MovieLens dataset, which may limit generalizability to other real-world datasets with different rating patterns. The study considers only probe and power user attacks; other informed or hybrid attack strategies remain unexplored. Further research can be carried out to identify the additional attributes so that the accuracy of the proposed model can be increased. Also, experiments can be done to tune the parameters of the models to improve the accuracy and computational complexity.

REFERENCES:

[1] Nguyen, Thi Thanh Sang, Hai Yan Lu, and Jie Lu. "Web-page recommendation based on web usage and domain knowledge." IEEE Transactions on Knowledge and Data Engineering 26.10 (2013): 2574-2587.

[2] Panagiotakis, Costas, Harris Papadakis, and Paraskevi Fragopoulou. "Detection of hurriedly created abnormal profiles in recommender systems." 2018 International Conference on Intelligent Systems (IS). IEEE, 2018.

[3] Bobadilla, Jesús, et al. "Recommender systems survey." Knowledge-based systems 46 (2013): 109-132.

[4] Lam, Shyong K., and John Riedl. "Shilling recommender systems for fun and profit." In Proceedings of the 13th international conference on World Wide Web, pp. 393-402. ACM, 2004.

[5] Zhang, Fuzhi, et al. "UD-HMM: An unsupervised method for shilling attack detection based on hidden Markov model and hierarchical clustering." Knowledge-Based Systems 148 (2018): 146-166.

[6] Hameed, Mohd Abdul, Omar Al Jadaan, and Sirandas Ramachandram. "Collaborative filtering based recommendation system: A survey." International Journal on Computer Science and Engineering 4.5 (2012): 859.

[7] Davoodi, Fatemeh Ghiyafeh, and Omid Fatemi. "Tag based recommender system for social bookmarking sites." 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. IEEE, 2012.

[8] Su, Xiaoyuan, and Taghi M. Khoshgoftaar. "A survey of collaborative filtering techniques." Advances in artificial intelligence 2009 (2009).

[9] Pitsilis, Georgios K., Heri Ramampiaro, and Helge Langseth. "Securing Tag-based recommender systems against profile injection attacks: A comparative study." arXiv preprint arXiv:1901.08422 (2019).

[10] Hill, Will, et al. "Recommending and evaluating choices in a virtual community of use." Proceedings of the SIGCHI conference on Human factors in computing systems. 1995.

[11] Konstan, Joseph A., et al. "Grouplens: Applying collaborative filtering to usenet news." Communications of the ACM 40.3 (1997): 77-87.

[12] Ricci, Francesco, Lior Rokach, and Bracha Shapira. "Recommender systems: introduction and challenges." Recommender systems handbook. Springer, Boston, MA, 2015. 1-34.

[13] Lops, Pasquale, Marco De Gemmis, and Giovanni Semeraro. "Content-based recommender systems: State of the art and trends." Recommender systems handbook (2011): 73-105.

[14] Isinkaye, Folasade Olubusola, Yetunde O. Folajimi, and Bolande Adefowoke Ojokoh. "Recommendation systems: Principles, methods and evaluation." Egyptian informatics journal 16.3 (2015): 261-273.

[15] Suganeshwari, G., and S. P. Syed Ibrahim. "A survey on collaborative filtering based recommendation system." Proceedings of the 3rd international symposium on big data and cloud computing challenges (ISBCC–16'). Springer, Cham, 2016.

[16] Linden, Greg, Brent Smith, and Jeremy York. "Amazon. com recommendations: Item-to-item collaborative filtering." IEEE Internet computing 7.1 (2003): 76-80.

[17] Burke, Robin, Bamshad Mobasher, Roman Zabicki, and Runa Bhaumik. "Identifying attack models for secure recommendation." In Beyond Personalization: A Workshop on the Next Generation of Recommender Systems. 2005.

[18] O'Mahony, Michael, Neil Hurley, Nicholas Kushmerick, and Guénolé Silvestre. "Collaborative recommendation: A robustness analysis." ACM Transactions on Internet Technology (TOIT) 4, no. 4 (2004): 344-377.

[19] Kumar, Ashish, et al. "IMPACT ANALYSIS OF PROFILE INJECTION ATTACKS IN RECOMMENDER SYSTEM." INFORMATION TECHNOLOGY IN INDUSTRY 9.1 (2021): 472-478.

[20] Aktukmak, Mehmet, Yasin Yilmaz, and Ismail Uysal. "Sequential attack detection in recommender systems." IEEE Transactions on Information Forensics and Security 16 (2021): 3285-3298.

[21] Zhang, Fei, et al. "Unsupervised contaminated user profile identification against shilling attack in recommender system." Intelligent Data Analysis 28.6 (2024): 1411-1426.

[22] Zhang, Fuzhi, et al. "Graph embedding-based approach for detecting group shilling attacks in collaborative recommender systems." Knowledge-Based Systems 199 (2020): 105984.

[23] Barbieri, Julio, et al. "Simulating real profiles for shilling attacks: A generative approach." Knowledge-Based Systems 230 (2021): 107390.

[24] Rezaimehr, Fatemeh, and Chitra Dadkhah. "T&TRS: robust collaborative filtering recommender systems against attacks." Multimedia Tools and Applications 83.11 (2024): 31701-31731.

[25] Panagiotakis, Costas, Harris Papadakis, and Paraskevi Fragopoulou. "Unsupervised and supervised methods for the detection of hurriedly created profiles in recommender systems." International Journal of Machine Learning and Cybernetics 11.9 (2020): 2165-2179.

[26] Chirita P, Nejdl W, Zamfir C (2005) Preventing shilling attacks in online recommender systems. In: WIDM '05: Proceedings of the 7th annual ACM international workshop on Web information and data management, New York, NY, USA, ACM Press pp 67–74.

[27] Wang, Qinyong, et al. "Fast-adapting and privacy-preserving federated recommender system." The VLDB Journal 31.5 (2022): 877-896.

[28] Anwar, Taushif, and V. Uma. "A study and analysis of issues and attacks related to recommender system." Convergence of ICT and Smart Devices for Emerging Applications. Springer, Cham, 2020. 137-157.

[29] Cohen, R., Sar Shalom, O., Jannach, D., & Amir, A. (2021, March). A Black-Box Attack Model for Visually-Aware Recommender Systems. In Proceedings of the 14th ACM International Conference on Web Search and Data Mining (pp. 94-102).

[30] Rezaimehr, Fatemeh, and Chitra Dadkhah. "A survey of attack detection approaches in collaborative filtering recommender systems." Artificial Intelligence Review 54 (2021): 2011-2066.

[31] MovieLens homepage: https://grouplens.org/datasets/movielens/.

[32] Garg, Satvik. "Drug recommendation system based on sentiment analysis of drug reviews using machine learning." 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence). IEEE, 2021.

[33] Powers, David MW. "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation." arXiv preprint arXiv:2010.16061 (2020).

[34] Zhang, Xinyu, and Chu-An Liu. "Model averaging prediction by K-fold cross-validation." Journal of Econometrics 235.1 (2023): 280-301.