International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 05 | May - 2022 Impact Factor: 7.185 ISSN: 2582-3930

Detection of Shilling Attack in a Recommender System

Aditya Chilla[1], Surya Kambhampati[2], Nikhil Popuri[3] , Venkatesh Narra[4] , Srikanth Thota[5]

[1-4] GITAM Institute of Technology & GITAM University, Visakhapatnam

[5] Associate Professor, GITAM Institute of Technology & GITAM University, Visakhapatnam

---***---

Abstract - In recent times commendation programs have become an integral part of our daily life, from grocery compliments to recommendation movies. People would not like to devote their time to find the best thing on the list according to their needs. The level of recommendations is fundamental for users to recommend anything. The Joint Recommendation Program recommends items for customer engagement and is a widely used and proven way to provide recommendations. Based on user ratings, it recommends that particular item. Since this recommendation program is based on ratings, it is straightforward for attackers to create false profiles and inject biased profiles in very large numbers. These types of attacks are called shin attacks divided into push (hacker attack) and nuke attack (opposite object). This attack is detected using aggregation, separation, element extraction, and possible methods.

***Key Words*:** Shilling Attacks, Recommender Systems, Ratings, Nuke, Push

2.INTRODUCTION

Recommender Systems is used to assist users with relevant suggestions based on previous searches or based on preferences. These recommendation systems are divided into interactive, content-based, information-based, demographic, Utility-Based, and Hybrid recommender systems. The shared filtering model recommends anything based on correlations amongst members are evaluated based on their ratings, and updated proposals are generated based on user comparisons. A content-based model is a recommendation system that uses keywords to describe events. Which means that if you search for anything like mobile on one of the e-commerce websites, it will show all the mobile stuff, and we can filter the search by cost or rating, etc. This is an example of a CB recommender program. The knowledge-based model is based on an explicit user requirement. Combining different models integrated hybrid recommender system id makes the recommendation more accurate. The DB program aims to classify individuals based on traits and generate cohort predictions. The UB recommender application offers suggestions for each user-item based on computational consumption.

CFRS is a method for gathering preferences or taste data from several people to develop automated predictions about their interests. Sensory and monitoring data, such as mineral exploration, natural sensors in broad regions, or many sensors; financial data have all been employed using these approaches.

2 Users with Similar Measuring Patterns in CFRS: Look for users with similar measurement patterns to an active user (the user to whom the prediction belongs). To determine the active user's guess, consider the ratings from those like-minded people discovered in step 1.

Create an object matrix that identifies the relationship between pairs of items. Check the matrix for the current user's preferences and compare that user's data. Due to the fact that this recommendation tool is dependent on ratings, attackers may easily construct bogus profiles and insert biassed profiles in huge numbers. Strict assaults are the name for these sorts of attacks.

Shilling, also known as profile injection, is a CFRS attack in which the user creates a phoney profile in order to provide suggestions for their gain. Different amounts of knowledge regarding the distribution of estimates are required for different types of attacks. Cash assaults impact all system users and can severely harm recommendation systems. For the targeted / active users, injecting false profiles will change the suggestions and the evaluation of items in unattended objects. In the assault process, there are three categories of participants:

- Users (both actual and hostile)
- Items (movies, videos, books, and so on)
- Profiles

This shell strike consists of two stages: push and nuke. A push assault pushes something in the list of suggestions. When anything is taken from the list of recommendations, the opposite is true of nuclear assaults. This may be accomplished by injecting biased/correct proportions of the targeted assault, and as a result of the detrimental effect on the suggestion list, users' confidence will be weakened. To keep the suggested systems safe from assaults, the negative impacts should be eliminated by recognising the profile attacks.

The attacking purpose, which characterises the attacker's arm, and whether the attacker is inserting biassed or real profiles are two attack aspects that influence the classification of assaults. The second parameter is profile size, which indicates how many ratings an attacker has allocated to an attack profile. The third factor is the assault size, defined as the number of profiles added to the system by an attacker.

3. LITERATURE SURVEY

Gaurav Arora , Ashish Kumar and Gitanjali Sanjay Devre created a mechanism for recommendation Recommendations are based on a variety of factors, including user interest, user history, user location, and many more. One thing is common in all of the above aspects, and that is individuality. The engine recommends users based on their preferences, however there are items in the market that are worth considering that a user is unaware of. The engine must also recommend these items to the consumers. Nonetheless, due to the "individuality" limitation, these engines do not recommend items that are not in the box.

The hybrid movie recommendation engine has overcome this individuality constraint. The engine will recommend movies to users based on their preferences and movies rated by other users who are similar to the user.

These recommendation systems employ various methods, including content-based approaches, collaborative approaches, knowledge-based approaches, utility-based approaches, hybrid approaches, and so on.

Rui-sheng Zhang, Qi-dong Liu, Chun-Gui, Jia-Xuan Wei, Huiyi-Ma Markovian factorization of matrix process (MFM) was presented as a new model family. On the one hand, MFM models, such as time SVD++, can capture temporal dynamics in the dataset. On the other hand, they have clean probabilistic formulations, allowing them to adapt to a broad range of collabs. In this family, two simple example models are introduced for the prediction of movie ratings employing time-stamped rating data. The experimental analysis utilising the MovieLens dataset shows that the two models, although being simple and primitive, have comparable or even superior performance than time SVD++ and a standard tensor factorization model.

Aghili G, Shajari M, Khadivi S, Morid MA To identify assaults in the movie recommender system, an ANN technique based on movie characteristics was developed. They assume that consumers evaluate movies in their chosen genre, whereas attackers score movies in other genres randomly. This means that every real user has a specific genre in mind. They utilised a backpropagation ANN to extract profile features with one hidden layer. They calculated user similarity before clustering them using the K-NN technique to provide suggestions.

Fuzhi Zhang, Zoning Zhang, Peng Zhang, and Shilei Wang Based on the hidden Markov model and hierarchical classification, they suggested an unsupervised technique for detecting shilling assaults. First, we utilise a hidden Markov model to model user history evaluation behaviours and determine each user's level of suspicion by studying the sequence of user preferences and the difference in ranking behaviours between legitimate and malevolent users. Then we apply hierarchical clustering, which groups people based on their level of suspicion, and selects a group of attack users. In tests, the suggested technique outperforms the Movie Lens 1 M and Netflix datasets.

In terms of detecting performance, the reference technique is used. We present an unsupervised strategy for identifying hidden Markov-based shilling attacks to address the previous shortcomings hierarchical clustering and modelling is referred to as UDHMM. The proposed approach for evaluating behaviours looks at the difference between legitimate and attacking users. For the first time, we used historical user rating data to create a user rating item sequence, then used the hidden Markov model to create the user chain of alternatives and provide metrics to differentiate between legitimate and violating users in rating behaviours. UDHMM outperforms PCA-VarSelect and UD-Kmeans in terms of detection precision, but it falls short of CBS. PCA-VarSelect and UD-Kmeans, which are both nearly as excellent as CBS, have lower recall than UD-HMM.

Sheng Zhang, Amit Chakrabarti, James Ford, and Fillia Makedon

The study aimed to discover strategies for detecting a wide range of suggested assaults. The following observation serves as the starting point for our investigation. Suppose we assume that the attack configurations are introduced into the system over a short period of time1. In that case, most sorts of shilling attacks (section 2) have one element in common: they cause changes in the target products' rating distribution over the attack's length.

A push assault, for example, will force the target element's note distribution to focus on high notes (or the highest possible score) throughout the length of the attack, regardless of the attack type. In a nuclear strike, the target element's rating distribution will focus on the bottom (or lowest) rating.

A time series can give substantial diagnostic capabilities in identifying many attacks since the research evaluates the distribution of points for each of the above factors. Unlike earlier techniques, the concept of considering shilling assaults as events that disrupt the delivery of notes is novel. Who determines if a user's dating profile is prejudiced (attack) or not based on how they interact with people in general? There are two key advantages of using time series to identify attacks.

For starters, it detects previously impossible threats to identify using other approaches. The records of each attack are examined independently. Attack the resulting profile with various assaults (such as sample attacks) that are nearly discernible when merely looking at the individual scoring model. Systematic mining to score distribution change is the best way to find them. Second, there are the time-varying irregular distributions. The series may uncover assaults that were previously unknown or unknown. Compared to rule-based or supervised empirical classifiers, this is a huge improvement. We should point out that the time-series technique can also uncover useful data that isn't a harmful aberration. A simple example would be a book that got famous immediately due to a specific occurrence.

4.METHODOLOGIES

COLLABORATIVE FILTERING

Collaborative filtering (CF) may be divided into two types: user-based collaborative filtering and item-based collaborative filtering. Both user-based and item-based CF recommender systems employ a consistent procedure to create suggestions. To commence, assess whether individuals or things are similar or dissimilar. Then weight the similarities to emphasise the users (or items) who have the most impact over whether similarity or dissimilarity is determined. Finally, predictions are made based on people's (or items') evaluations and their similarities.

1) **User-based collaborative filtering**: The most often used collaborative filtering method is the kNN-based algorithm. If there is a rating on item I, the data is represented as a user-item Matrix R, with Ru,i reflecting the rating given by user u for item I or null if there is no rating on item i. The Pearson correlation is then used to calculate the degree of similarity between users.

$$W_{uv} = \frac{\sum_{i \in I}(R_{ui} - \overline{R_u})(R_{vi} - \overline{R_v})}{\sqrt{\sum_{i \in I}(R_{ui} - \overline{R_u})^2 (R_{vi} - \overline{R_v})^2}}$$

where I is the set of items that users u and v both rated, Rui is the rating user u gave to item i, and R¯ u is the average rating of user u.

2) **Item-based collaborative filtering**: The item-item similarity is another prominent CF method. Cosine-based similarity can be used to calculate the similarity between items u and v:

$$Sim(u, v) = \frac{\vec{u} \cdot \vec{v}}{\|\vec{u}\| \cdot \|\vec{v}\|}$$

4.1 ATTACK METHODS

- Average attack
- Bandwagon attack
- Random attack

Average approach: The average approach model is more advanced than the random attack model in that it requires information of each item's average rating in the recommender system. Attackers use a normal distribution to rate things in the filler set at random, with the deviation set to the average rating of the filler items being reviewed and the mean score of the filler objects being appraised. By implementing the average attack model, attackers are disguised and are more difficult to distinguish from legitimate users, having a greater impact on suggestions. The selected item's ratings are set to either the highest or lowest possible score. highest or least permissible rating dependent on the goal of the assault, much like in the random attack paradigm.

Bandwagon approach: In order to improve the likelihood that these phoney profiles have many neighbours, bandwagon assault profiles provide a random rating to a selection of things and a maximum rating to top-rated items. Attack of the Bandwagon Add profiles with high ratings for "blockbusters" (in the specified items); filler items with random values. Bandwagon Because popular things will have numerous ratings, and rating values are comparable to many other user profiles, this will logically lead to more neighbours.

Random approach: Filler objects should be given random values, while target items should be given high or low ratings. The random attack model is a basic attack in which the injected profile uses a normal distribution and the standard deviation around the system's average rating to score a collection of randomly-picked fillers. They then assign a maximum or minimum permissible rating to the collection of target objects based on the attack's goal. An attacker would rate the target object as 5 for a push attack and 1 for a nuke attack if the rating scores for a recommender system are between 1 and 5, where 1 indicates an unfavourable rating and 5 represents a good rating.

TABLE I
FEATURES OF THE ATTACK MODELS

Attack model	I_S(Selected Items)	I_F(Filler Items)	I_T(Target Items)	I_E(Unrated Items)
Random Attack	$\emptyset$	$r(I_F)$ = random ratings	$r(I_T) = r_{max}/r_{min}$	$r(I_E) = \emptyset$
Average Attack	$\emptyset$	Mean of each item	$r(I_T) = r_{max}/r_{min}$	$r(I_E) = \emptyset$
Bandwagon Attack	Popular items,$r(I_S) = r_{max}$	$r(I_F)$ = random ratings	$r(I_T) = r_{max}/r_{min}$	$r(I_E) = \emptyset$

4.2 Metrics for attack detection

- Rate deviation from the mean agreement
- Deg sim

In statistical terms, attack profiles differ from real profiles. The two key distinctions are assigned to the target item (items) and the rating distribution among the filler items.

As a result, many metrics for computing the correlation of differences have been proposed. We will look at two measures in this section: RDMA, and DegSim.

AN EXAMPLE OF RATING MATRIX AND ATTACK PROFILES.

	Item_1	Item_2	Item_3	Item_4	Item_5	....	Item_n
User_1	5	2	3	0	0	....	5
User_2	2	0	4	1	2	....	3
User_3	4	2	3	0	5	....	0
User_4	0	3	0	3	4	....	3
....	....	....	....	....	....	....	....
User_m	2	0	4	1	2	....	3
Attacker_1	2	1	0	0	5	....	4
Attacker_2	2	2	0	0	5	....	3
Attacker_3	1	2	0	0	5	....	2
....	....	....	....	....	....	....	....
Attacker_p	2	0	0	0	5	....	4
Count(5)	2	2	2	2	9	....	3

Rate Derivation From Mean Agreement: This is applied in attack models to compare rating patterns between malicious and authentic profiles. Measures the inverse rating frequency for a collection of target items and the divergence of agreement from other users on those things. The following formula may be used to determine RDMA:

$$RDMA_u = \frac{\sum_{i=0}^{N_u} \frac{|r_{u,i} - \bar{r}_i|}{NR_i}}{N_u}$$

Where Nu is the number of items user you rated, ru, i is the rating given by user you to an item I, NRI is the overall number of ratings in the system given to item i.

Degree of Similarity with Top Neighbours: Many analysts claim that maybe an attack profile exhibits a significant similarity metric with the user's nearest 25 neighbours. This hypothesis was confirmed in a previous investigation, which discovered that the most efficient attacks include introducing a huge number of profiles with extremely identical traits. The Degree of Similarity with Top Neighbors (DegSim) feature captures this idea. The DegSim attribute is derived based on the average Pearson correlation of the profile's k closest neighbours:

$$DegSim = \frac{\sum_{u=1}^{k} W_{uv}}{k}$$

Where Wuv is the Pearson correlation between the user, you, and user v, and k is the number of neighbors.

4.3 Generating Attack Profiles

There are a few parameters while injecting the fake profiles, which are

Attack size: The ratio of the injected spammers to genuine users.

Filler Size: The ratio of the filler items to all items.

Selected size: The ratio of the selected items to all items.

Link size: The ratio of the users maliciously linked by a spammer to all users. targetCount: The count of the targeted items. Target score: The score given to the target items.

Threshold: An item with an average score lower than the threshold may be chosen as one of the target items.

Min count: the item with a rating count larger than MinCount may be chosen as one of the target items.

Max Count: An item with a rating count smaller than maxCount may be chosen as one of the target items.

An attacker can inject the fake profile for the threshold values using these parameters. The dataset consists of the userId: the user's identity, itemId: is the item’s id. For example, let's consider movie recommendations. The different movies are the different item Ids.

Rating: It represents a particular user rating of a particular item,

Timestamp: It represents the timestamp of the particular user rating the particular item. All the ratings are more minuscule than the threshold values, i.e., the ratings more diminutive than the target score get the biased rating.

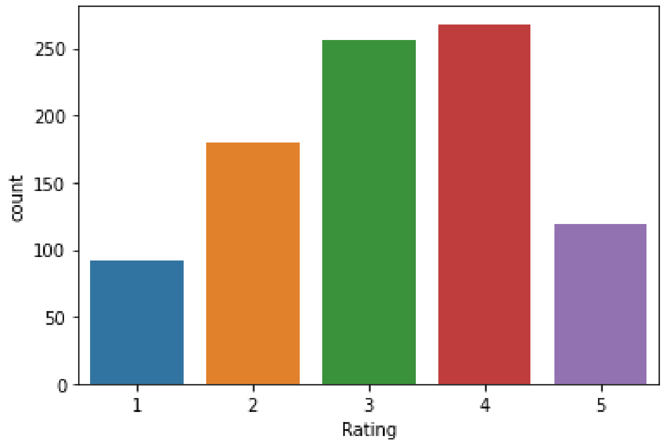


Fig1, Rating chart before attack

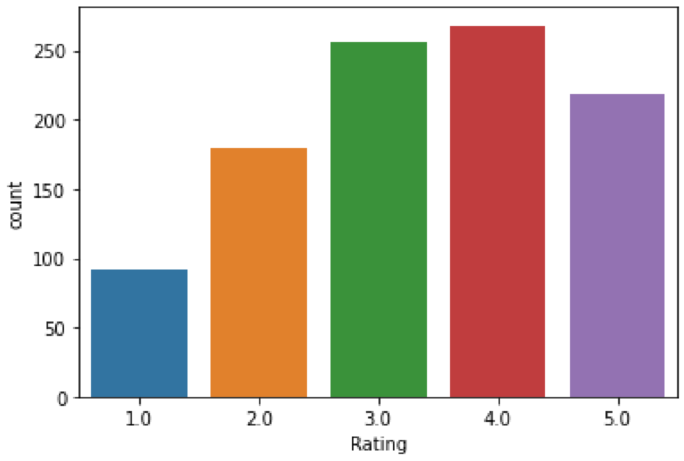


Fig2. Rating Chart after an attack

4.4 ATTACK DETECTION MODELS

Supervised Methods

- **K- Nearest Neighbor Algorithm Model**
- **Support Vector Machine Algorithm model**
- **Random Forest Classifier Algorithm model**

Supervised Models:

A machine learning algorithm learning a function that converts input to an output based on examples of input-output pairings is known as supervised learning (SL). It uses tagged training data and a set of examples to infer a function. Each example in supervised learning includes an input object (usually a vector) and a desired output value (also called the supervisory signal). A supervised learning algorithm examines the training data and generates an inferred function that may be applied to new situations. In the best-case scenario, the system will be able to accurately estimate class labels for cases that have not yet been observed. This necessitates the learning algorithm's ability to "reasonably" generalise from training data to unknown situations.

K Nearest Neighbor

K-Nearest Neighbour is a Supervised Learning-based Machine Learning algorithm that is one of the most basic.

The Proposed technique assumes that the new case/data and previous cases are similar and assigns the new case in the most similar category to the previous ones. It stores all available data and classifies a new data point based on its similarities to the known information. The K NN software can

effectively sort new information into a good suite category when it occurs. The technique can be used for combining regression and classification, it is more commonly employed for classification. It's a non-parametric algorithm, which means it makes no guesses it's functioning with. It's also regarded as a lazy learner algorithm because that doesn't learn from the training set real quick; instead, it maintains the dataset and works on it eventually.
The KNN algorithm simply preserves the information during the training phase, when new data is obtained, it is classified in a class that is quite equivalent to the new data.

KNN Usage:

Imagine there are two categories, Category A and Category B, and we have a new data point x1. We want to determine which of these categories this data point corresponds to. A K-NN algorithm is needed to address this type of problem. We can quickly determine the group or class of a dataset using K-NN.

- Working of KNN: Select the number K of the neighbors, calculate the Euclidean distance of K number of neighbors, take the K nearest neighbors as per the calculated Euclidean distance. Keep track of observations in each category. among these k neighbours, then assign new data points to the category with the largest data points.

Selecting the K value in KNN

- Because there is no one-size-fits-all method for determining the greatest value for "K," we must experiment with several options to find the best one. The most popular K value is 5.
- A very low K number, such as K=1 or K=2, might be noisy and cause outlier effects in the model.
- Large K levels are excellent, although they may cause problems.

Calculating using the euclidean distance formula
Initially, we should estimate the number of neighbours and then choose the k value through the Elbow method.
The Euclidean distance between the data points will then be calculated. The Euclidean distance is the distance between two points that we learned about in geometry class. It is calculated by taking the square root of the sum of squares of (X2-X1) and (y2-y1).

Support Vector Machine:
One of the most prominent Supervised Learning techniques for classification and regression issues is the Support Vector Machine, or SVM. However, it is most commonly employed in Machine Learning for Classification challenges.
The SVM method aims to find the optimal line or decision boundary for categorising n-dimensional space into classes so that new data points may be placed in the proper category rapidly. A hyperplane is the word for the optimum choice limit.
The maximum points/vectors that aid generate the hyperplane are chosen via SVM. Support vectors are the maximum instances, and the method is called a Support Vector Machine.

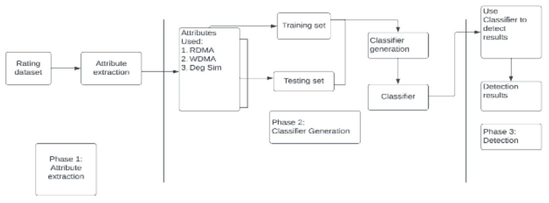
Support Vectors: The data points or vectors closest to the hyperplane that affect the hyperplane's position are called Support Vector. Since these vectors support the hyperplane, hence called a Support vectors.

Random Forest:
Random Forest is a well-known the supervised attempting to learn branch of artificial intelligence. It is based on the idea of ensemble learning, which entails combining multiple classifiers to solve a complex problem and improve the model's performance.
According to the name, "Random Forest is a classifier that includes a classification tree on various subsets of a given dataset and takes the average to improve the dataset's projected accuracy."
Instead of relying on a single decision tree, the random forest gathers forecasts from each tree and predicts the ultimate output based on the majority of votes.
The larger the number of trees in a forest, the more diverse it becomes.
Rather than relying on a single decision tree, the random forest gathers forecasts from each tree and predicts the ultimate output based on most votes.
The greater the number of trees in a forest, the better the forest's accuracy and prevents the problem of overfitting.

Uses of Random Forest :

- It takes less time to train compared to other algorithms.
- Even a big dataset predicts high accuracy output and runs quickly.

- When a considerable amount of the data is missing, it can still maintain accuracy.



5. Experimentation

5.1 DATASET:

The studies were conducted using the widely used MovieLens 100k and 1M datasets from the University of Minnesota. There are 100,000 ratings from 943 people on 1682 movies in the 100K dataset, and 1,000 ratings from 6,000 users on 4,000 movies in the 1M dataset. Each user has rated at least 20 movies. A movie may be rated from 1 to 5 by each user. The lowest number is one, while the greatest number is five.

5.2 METRICS

Here we will be talking about the parameters that were used for detecting attack profiles in a given dataset. These values chosen act as a threshold value for determining whether the given account is a malicious user or not. We set up a threshold limit for RDMA and DegSim. If the calculated value of RDMA and DegSim is greater than the threshold value, then the profile is considered a malicious user.

When setting RDMA and DegSim threshold values, we want to achieve values with a lot of separabilities because it's simpler to tell the difference between legitimate and attack profiles when there are a lot of separabilities. We tweaked the settings to create a minimal number of false negatives and positives. We noted that the intervals between DegSim values for the profiles were less than the intervals between RDMA values for the profiles in these studies.

We set the values threshold values of RDMA and DegSim by using the formulas:

$$\epsilon_{DegSim} = \lambda \sum_{u=1}^{n} \frac{DegSim_u}{n}$$

and

$$\epsilon_{RDMA} = \gamma \sum_{u=1}^{n} \frac{RDMA_u}{n}$$

We choose a different weight for λ and γ. We were more comfortable with the results of our experiments when λ = 1 and γ = 0.6. Setting these weights, we notice that the false negatives rate is lower and there are hardly any false positives. As we pointed out previously, the threshold value of RDMA and DegSim are generous thus allowing false-negative profiles into the set of SuspectedAttackers.

We used the attack size at 50, filler size at 1%, and target rate at 5%.

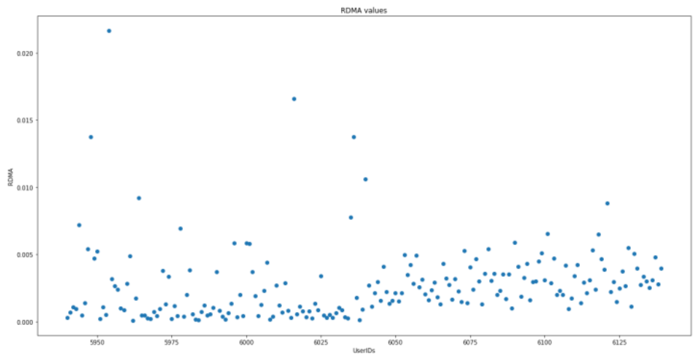


Fig3. Rate of RDMA values for each userId.

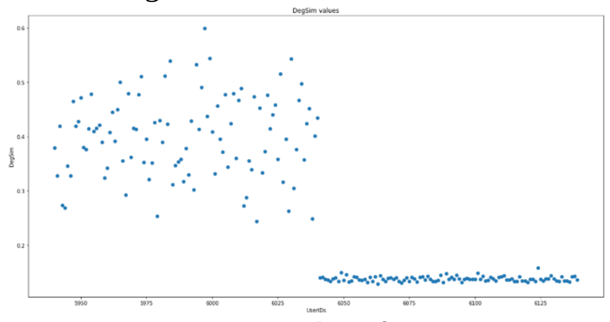


Fig 4 DegSim values for userIDs.

5.3 Evaluation Metrics:

The evaluation metrics that were used for determining the results are

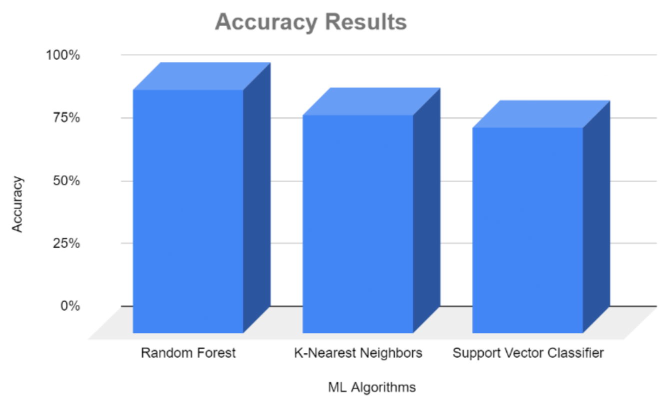
Precision : True Positive / (True Positive + False Positive)

Recall : True Positive / (True Positive + False Negetive)

F1- Score : 2 * ((Precision * Recall) / (Precision + Recall))

6. RESULTS and DISCUSSION

We discovered that random assaults had a greater detection rate than typical attacks. Nuke assaults have a lower false-positive rate than push attacks. As a result, nuclear assaults have a lower false-positive rate than push attacks. When nuke attacks are carried out, the target item is rated low (usually 1). Still, because the distribution of rating 1's in the dataset is lower, it's easier to distinguish a nuke attack from a push attack. The target item is rated high (usually 5). The distributions of rating 5 are higher.
We examined the data matrix without inserting attack profiles using RDMA and DegSim, and discovered three profiles that were wrongly labelled as attack profiles. The noteworthy aspect of introducing attack profiles into the rating matrix is that the misclassified profiles are almost always the same set. It's conceivable that this is because they have qualities in common with attack profiles. These three qualities are referred to as natural noise in the recommender system. Because simple recommender systems have millions of profiles, removing some of them will not influence the eventual conclusion of the recommender system.
The algorithms we employed for this work were K-Nearest Neighbors, Random Forest Classifiers, and the Support Vector Classifier technique, which had the best accuracy.



7. INFERENCE

Because of their open nature, recommender systems are vulnerable to malevolent users. Based on their rating habits, we employed two statistical criteria to identify attackers. K-Nearest Neighbors, Random Forest, and Support Vector Machines are the algorithms we utilised in this research to identify Random assaults in the Collaborative filtering recommender system. The algorithms are supervised algorithms that require labelled data to assist distinguish between legitimate and criminal users. It was calculated using the RDMA and DegSim methodologies to get the threshold values. When compared to the other algorithms, the Random Forest approach proved to be more accurate.

8. REFERENCES

[1] Bhaumik et al. - 2012 - A Clustering Approach to Unsupervised Attack Detection in Collaborative Recommender Systems
[2] Zhang et al. - 2018 - UD-HMM: An unsupervised method for shilling attack detection based on hidden Markov model and hierarchical clustering.
[3] Panagiotakis et al. - 2020 Unsupervised and Supervised Methods for the Detection of Hurriedly Created Profiles in Recommender Systems.
[4] Shilling Attack Detection in Recommender System Using PCA and SVM: Proceedings of IEMIS 2018, Volume 2.
[5] Zunping Cheng, Neil Hurley Effective diverse and obfuscated attacks on model-based recommender systems.
[6] Zhou Wei, Wen Junhao Attack detection in recommender systems based on target item analysis.
[7] Fatemeh Rezaimehr, Chitra Dadkhah " A survey of attack detection approaches in collaborative fltering recommender systems".
[8] Zhongmin Cai, Yang Zhihai "Detecting abnormal profiles in collaborative filtering recommender systems"
[9] S. Zhang, A. Chakrabarti, J. Ford, and F. Makedon, "Attack detection in time series for recommender systems,"
[10] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: an open architecture for collaborative filtering of netnews".