Shilling Attack Detection in Recommender Systems

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering

in

Computer Science and Engineering

Submitted By

Parneet Kaur

(Roll No. - 801432018)

Under the supervision of

Dr. Shivani Goel

Assistant Professor (CSED)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

PATIALA – 147004

June 2016

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, "***Shilling Attack Detection in Recommender Systems***", in partial fulfillment of the requirements for the award of degree of Master of Engineering in ***Computer Science and Engineering*** submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of ***Dr. Shivani Goel*** and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

Parneet Kaur

(Parneet Kaur)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

Shivani Goel

(Dr. Shivani Goel)
Assistant Professor (CSED)

Countersigned by

(Dr. Maninder Singh)
Head
Computer Science and Engineering Department
Thapar University
Patiala

(Dr. S.S Bhatia)
Dean (Academic Affairs)
Thapar University
Patiala

ACKNOWLEDGEMENT

I am very much thankful to my guide ***Dr. Shivani Goel***, Assistant professor, Computer Science and Engineering Department, Thapar University, Patiala for her guidance, patience, moral support, motivation, advice, positive attitude and sincere efforts. She has been very cooperative and solved my queries whole heartedly. It would not have been possible for me to explore and work on such a vast topic without her support. She has given me new ideas for this research work and shown me a right direction to achieve my goals.

I am also heartily thankful to **Dr. S.S. Bhatia**, Dean of Academic Affairs, **Dr. Maninder Singh**, Head of CSED and **Dr. Ashutosh Mishra**, PG Coordinator for providing the entire infrastructure for learning.

I am also thankful to my friends for their cooperation, help and showing me the right path.I would like to thank the almighty, my parents and my family members for giving me courage and strength to carry out the research work. They are continuous source of support, concern, encouragement, love and strength for me.

Parneet Kaur

801432018

ME (CSE)

ABSTRACT

Recommender systems help customers to select relevant products from millions of choices available on the internet. These systems can be based on various approaches: content based filtering, collaborative filtering, knowledge based approach and hybrid filtering. Recommender systems which are based on collaborative filtering are vulnerable to "shilling attacks" due to their open nature. Malicious users inject a few unscrupulous shilling profiles into the database of ratings for altering the system's recommendation, due to which some inappropriate items are recommended by the system. As a result, the performance of the system may degrade. In this thesis, we simulated shilling attacks namely random, average, bandwagon and segment on Movie-Lens dataset, which focused on a set of users having similar interests. Biased ratings of the items are also introduced in the system. The results show that although segment attack has impact on item based collaborative filtering, still it has higher robustness than user based collaborative filtering approach.

To preserve the trust of the recommender system, it is required to identify and remove the fictitious profiles from the system. Therefore, machine learning classifiers and detection attributes are used to distinguish the attacker's profiles. Five classification algorithms are compared and a new model is proposed by integrating two models with high performances using majority voting method. The proposed model outperforms in most of the cases. In the experiments, it is proved that the combination of random forest and adaptive boosting algorithm is more accurate than simple random forest model.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 Recommender System

With the rapid growth of internet, the data on the web is increasing at a very high rate. It becomes a difficult task to retrieve useful information. In recent years, recommender systems (RSs) have become popular in e-commerce which helps the customers to select items of their interest by predicting the ratings that would be given by them to an item [1]. These systems are used in various application areas i.e. online bus bookings, online shopping, online hotel bookings, movie recommendations, online cab bookings, finding research articles and so on. Now, it has become a monotonous task to provide recommendations to the users by filtering the items of their choice. Recommendations depend upon various factors like user's rating given to collection of items, their age, occupation, likes and dislikes, gender *etc*. There are many recommender systems for financial services, collaborators, experts, restaurants, Twitter followers etc. For example, **Amazon** uses item-to-item collaborative filtering algorithm. When any customer selects an item to purchase, then the system recommends him/her other items based on the similarity of those items with the actual item he/she searches. **Lenskit** is an open source toolkit for researching and building recommender systems. It implements well-regarded collaborative filtering algorithms. **Netflix** provides streaming television series and movies globally. **LinkedIn** is a social networking site which recommends the job a user might like, companies in which he might has interest or groups he might want to follow. Google, Myntra, Twitter, Pandora Radio, Del.icio.us, Last.fm etc. are the sites that incorporate recommendation engines.

1.2 Business Characteristics of a Recommender System

It plays an important role for both users and for service providers. The purpose of a RS for service provider is [1]:

i. Growth in sale: The main aim of any online seller is to sell a large number of items. Recommender engine plays an important role. It helps the buyer in finding the items that might be of his/her interest.

ii. Increase user satisfaction level: Recommender systems should be well designed in order to raise the user's satisfaction.

The purpose of a RS for users is:

i. Finds items of choice: It helps in finding items based on their ranks, and then recommends items to a particular user's interest. It would be difficult to search for an item from millions of available items. It also lists out the items in the sequence, in which the user may purchase those items.

ii. Limits the set of choices: When customer is not able to choose which item he/ she should buy, recommender system narrow down the choices and recommends only those items that are most similar to his/her choice.

iii. Items exploration: Sometimes user only wants to know the new items in the market, and then recommender system shows only those items which are relevant for him, so that he finds that item useful and buys it.

1.3 Recommendation Techniques

Recommendation systems have changed the way static websites used to interact with their customers. Instead of providing an inanimate experience in which customer searches and purchase items, these systems have increased the interaction to provide a better experience. RSs provide recommendations to individual customers based on their past searches and purchases, and on the behavior of other similar users. These systems provide recommendations to the users by using one of the following approaches [2]:

- Content based filtering
- Collaborative filtering
- Knowledge based filtering
- Hybrid filtering

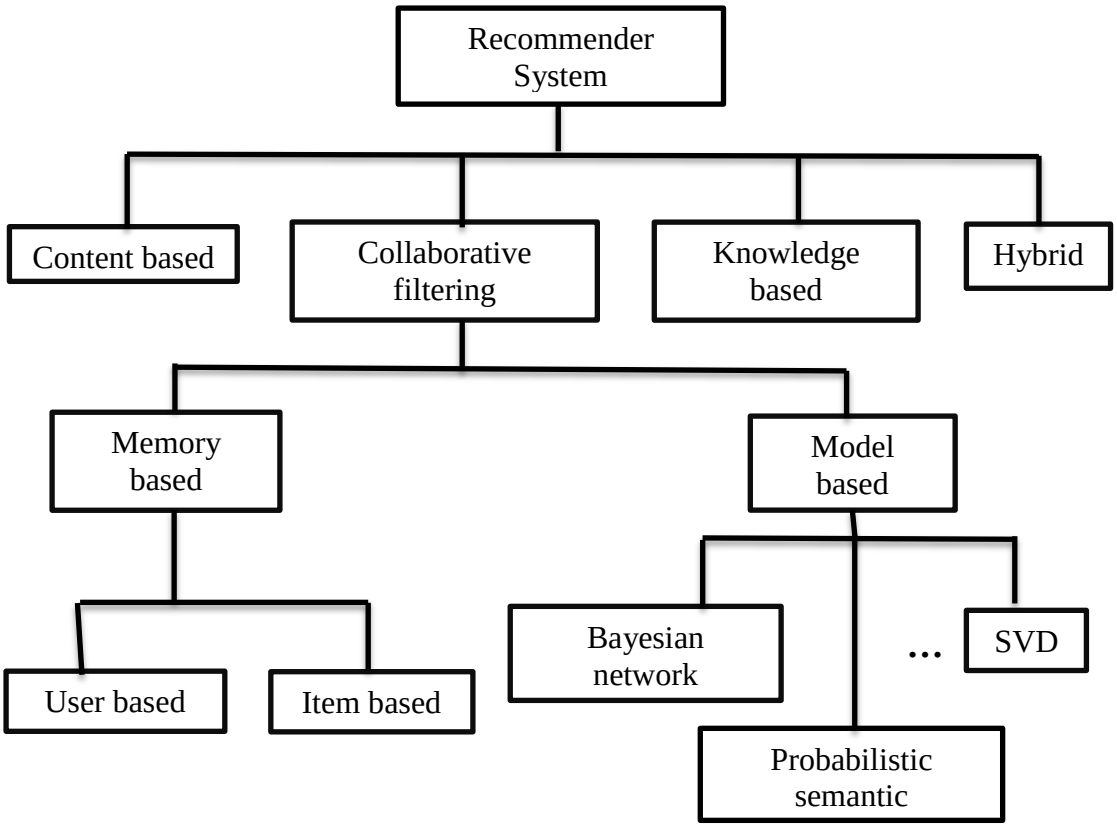

Figure 1.1 Classifications of Recommendation Techniques

1.3.1 Content based Filtering

Content-based filtering recommends items to a user by comparing the content of the items and the profile of the user. It is also known as cognitive filtering. Each item's content is represented as a set of terms that are present in a document. The profile of the user is represented with the same terms and is constructed by examining the item's content that has seen by the user. It concentrates more on the characteristics of items. Content based filtering techniques suggest those items which are similar to the previously liked items of the user [2]. The user's profile is generated based on the history of the interaction of user with the system, on the user's preference model and on the weighted mechanism of item's characteristics. The weight represents the importance of item's characteristics to the user. For example, if a customer has purchased a novel on flipkart then the recommender system gives more preferences to the user for purchasing the novels with similar information from their online book store.

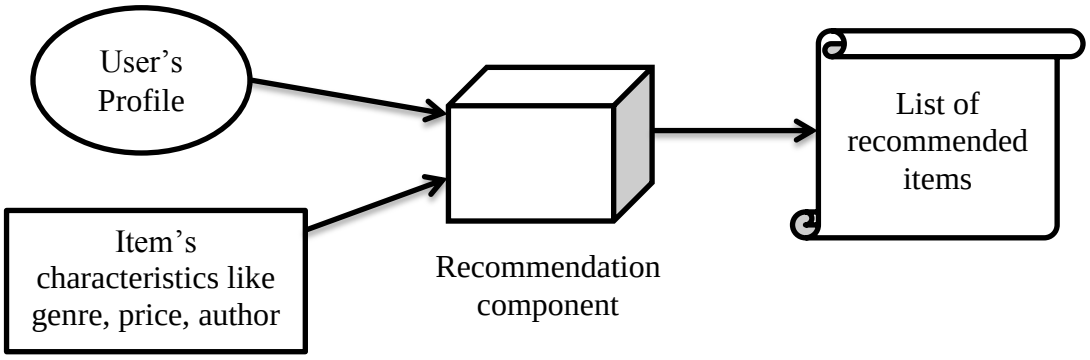

Figure 1.2 Content based Recommender System

Content-based filtering system may face many problems. First, terms can either be created manually or automatically. When the terms are created automatically a method needs to be selected which can obtain these terms from items. Second, it is difficult to find whether the system is able to learn the preferences of the user from his previous actions and use them on other type of contents [2].

Pandora Radio, Internet Movie Database, Rotten Tomatoes etc. are the recommender systems that make use of content-based filtering approach.

1.3.2 Collaborative Filtering

Collaborative filtering (CF) techniques are used widely to design recommender systems. It is also known as social filtering. Recommendations are generated by the system using information about ratings for different users. The system locates similar users with a rating history peer to the present user, predicts the ratings for the new user and generates recommendations on the basis of the choice of his neighbors [3]. In other words, it builds a database of user's preferences for items. When a new user enters the system, similarities between the other users are calculated. Items which are liked by the neighbors are recommended to him because those items might be of his choice. For example, a person wants to watch a movie, he might ask for suggestions from his friends. He would have more trust on the recommendations provided by his like-minded friends rather than those provided by the system. This would help in deciding which movie he should watch.

The benefit of CF approach is that it does not depend on the content analyzed by the machine. Therefore it is able to recommend complex items more efficiently.

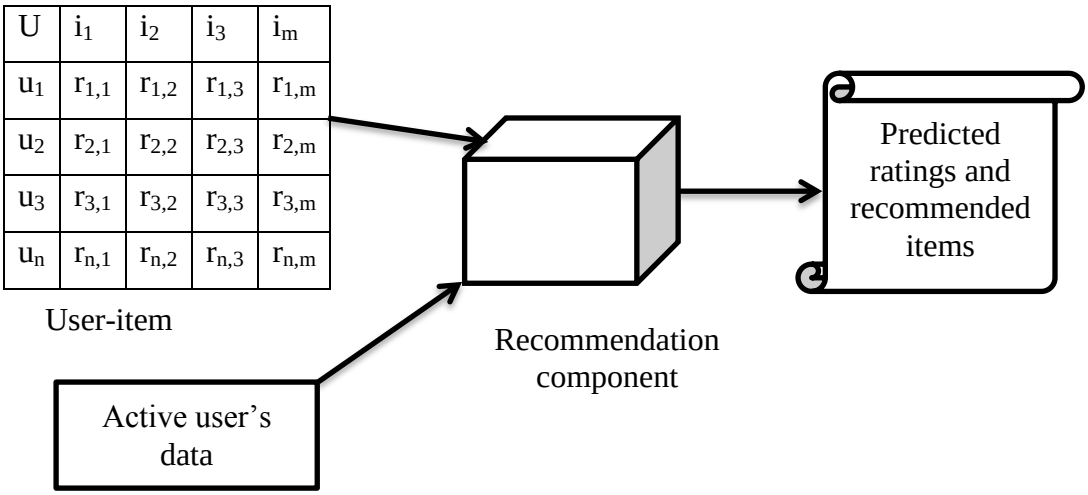

Figure 1.3 Collaborative Filtering Recommender System

There are two challenges for collaborative filtering recommender systems (CFRSs).

1) To improve the scalability of CF algorithms. Today, there is a demand of searching hundreds of the millions most similar neighbors. But, the existing CF algorithms have performance issue; they take more time to search relevant information from such a large amount of available information.
2) To improve the recommendation quality. The accuracy of the system is important. It should recommend only those items that are relevant to the user in order to maintain the trust of the user on it.

1.3.2.1 Memory based collaborative filtering

Memory based algorithms uses the input user-item matrix to predict the ratings for an item [4]. These systems utilize various statistical methods to find the similar set of users. Once the neighbors are found, different algorithms are used to predict the ratings for the user. There are two types of approaches in memory based CF: user based CF and item based CF.

A) User based Collaborative Filtering

A standard CF algorithm which finds the users who are k most similar to the targeted users and uses these user's preferences to predict the ratings is *k nearest neighbor*

(kNN) algorithm [3]. In order to provide recommendations to user, two steps are followed:

i) Calculate similarity: The first step is to find the similarities between the users using any statistical techniques. Following algorithms can be used to find it:

a) Pearson's correlation score: It can be computed as [3]:

$$S_{a,b} = \frac{\sum_{q \in I} (r_{a,q} - \bar{r}_a) * (r_{b,q} - \bar{r}_b)}{\sqrt{\sum_{q \in I} (r_{a,q} - \bar{r}_a)^2} * \sqrt{\sum_{q \in I}^{k} (r_{b,q} - \bar{r}_b)^2}} \quad \text{.....(1)}$$

where, $r_{a,q}$ and $r_{b,q}$ are the ratings given by user 'a' and its neighbor 'b' respectively, to an item q. *I* is an item set that contains all the items. The most similar users are chosen after finding the similarities between the users.

b) Eucledian distance: This metric can be used to compute the similarity between the users. It can be calculated by taking the root of the sum of squared differences between each of their rating.

$$S_{a,b} = \sqrt{\sum_{q \in I} (r_{a,q} - r_{b,q})^2} \quad \text{..... (2)}$$

ii) Prediction: Once the neighbors are found, rating is predicted for item q and target user 'a' by using (3):

$$P_{a,q} = \bar{r}_a + \frac{\sum_{a \in M} S_{a,b} (r_{b,q} - \bar{r}_b)}{\sum_{b \in M} |S_{a,b}|} \quad \text{..... (3)}$$

where, $r_{b,q}$ represents user's rating who have given rating to item q, $\bar{r}_b$ denotes overall mean rating, M contains k most similar users.

B) Item Based Collaborative Filtering

This CF algorithm is based on similarities between the items [3]. kNN algorithm uses to find the k peer items. Similarities between the items are calculated using following methods in addition to those described in previous part.

a) Cosine-Based Similarity: In this, two items are taken as vectors in n dimensional user-item space. The similarity between the two items is calculated by computing the cosine of the angle between these vectors.

$$S_{a,b} = \cos(\vec{p}, \vec{q}) = \frac{\vec{p}.\vec{q}}{|\vec{p}|^2 * |\vec{q}|^2} \quad \text{..... (4)}$$

Once the similarities are calculated, a set of k items are selected that are most alike to the targeted item and predictions are generated using the following formula:

$$P_{a,w} = \frac{\sum_{x \in N} r_{a,x} * S_{w,x}}{\sum_{x \in N} S_{w,x}} \quad \text{..... (5)}$$

where, N is a set of k similar items, $S_{w,x}$ denotes the similarity between items w and x and $r_{a,x}$ represents the predicted rating of item x for user 'a'.

The *advantages* of memory based CF are:

- It is a simple algorithm to implement.
- Prediction quality is better.
- It uses the whole database to make predictions; therefore, it is easy to update the database.

The *disadvantages* of memory based CF are:

- Memory based RS is very slow because it uses the whole database to make predictions.
- These systems are less scalable.
- Sometimes, it does not make predictions for the user if there is no common item between him and all the other users who have given rating to target item.

1.3.2.2 Model based collaborative filtering

In this approach, the system uses training data and learns the complex patterns and then makes predictions for the test users depending upon these learned models [5]. The dataset is partitioned into training and testing dataset, the models are trained using training dataset and performance of the models is analyzed on testing dataset.

Sometimes the whole dataset is used to make predictions. Bayesian network, clustering and rule based approach can be used to build the models. Bayesian network develops the probabilistic models for CF problems. If the ratings of user are *categorical*, then classification techniques can be used as CF models and if they are *numerical*, then SVD and regression models can be used. To find the association between the common purchased items, rule based approach is used. The *advantages* of model based RSs are:

- These systems are scalable which means models developed from the model based approach are small even for large datasets.
- They are faster than memory based systems.

The main *disadvantage* of the model based system is its inflexibility. It is difficult to add data to these systems as constructing a model is resource consuming and time consuming.

1.3.3 Knowledge based recommender systems

These systems utilize knowledge structure to infer the user's preferences and requirements. These systems have knowledge about the type of item liked by a user, so that a relationship can be formed between user's requirements and appropriate recommendation for him.

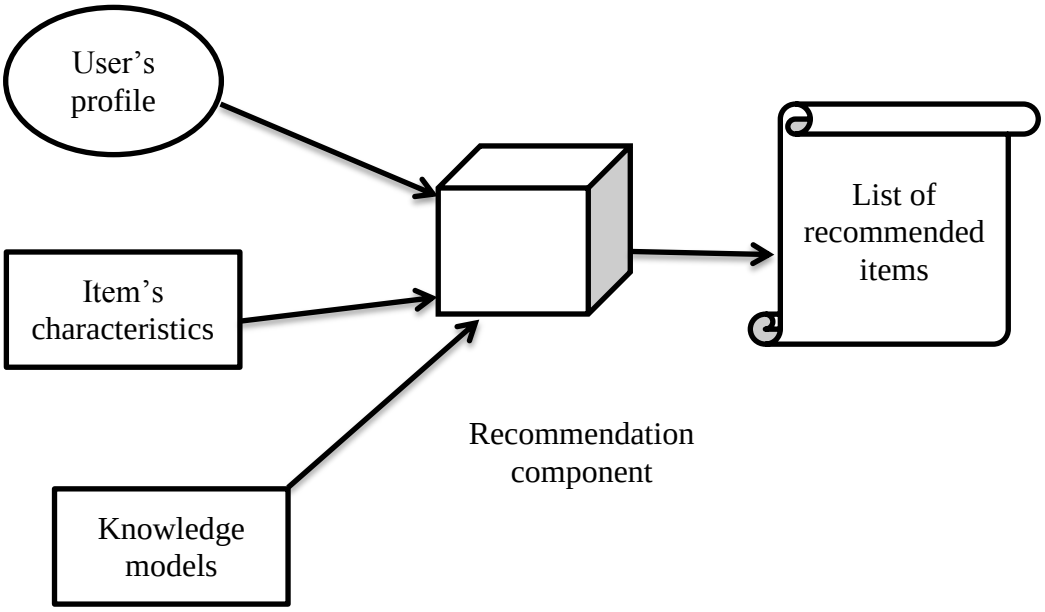

Figure 1.4 Knowledge based Recommender System

1.3.4 Hybrid recommender systems

Single recommender system approach is not much effective. Hybrid recommender systems could generate recommendations more accurately. They can be constructed by integrating content based and collaborative filtering approach. These systems overcome the problems of the traditional recommender systems. Previous studies demonstrate that performance of hybrid recommender system is much better than CFRSs or content based systems [6]. Netflix is the system that make use of hybrid RS. They provide suggestions by using the search and watch history of the peer users (i.e. CF) as well as by the movies which have similar features as those rated by the user (i.e. content based filtering).

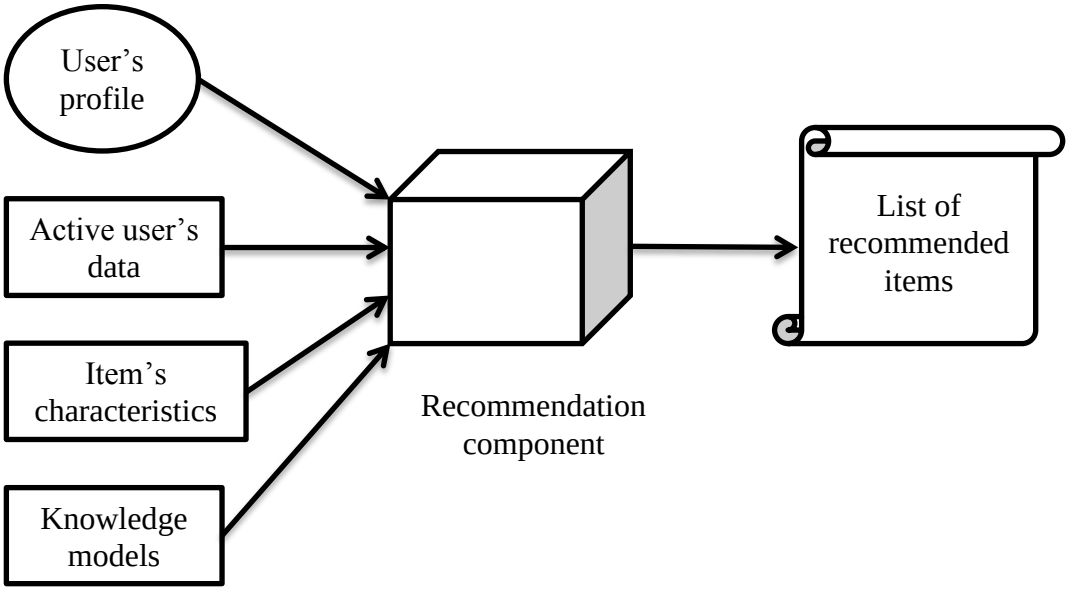

Figure 1.5 Hybrid Recommender System

1.4 Challenges in Recommender System

Recommender systems experience many challenges which can be described as follows [2]:

1.4.1 Data Sparsity

Sparsity of data is one of the major challenge faces by the systems. It affects the quality of recommendations. It arises due to the items that are not rated by most of the

users. CF approach suffers from this problem because it depends on the rating matrix in most cases.

1.4.2 Shilling attack

Recommender systems based on collaborative filtering are vulnerable to "shilling attacks" due to their open nature [7]. It is also known as profile injection attacks. Malicious users inject a few unscrupulous shilling profiles into the database of ratings for altering the system's recommendation, due to which some inappropriate items are recommended by the system. Shilling attacks can be categorized as *nuke attack* and *push attack*. In nuke attack, attacker gives lowest score to target items in order to demote them whereas, in push attack, attacker gives highest score to target items in order to promote them. The motive of an attacker is to construct attack profiles using attack models which have high influence and require minimum knowledge.

1.4.3 Cold Start problem

Cold start problem arises with the entry of a new user or a new item in the system. It becomes difficult to provide recommendations as very less information is available about the new user and the new item, no rating is available corresponding to the new item [2]. Therefore, in such cases, collaborative filtering doesn't provide appropriate recommendations. But in case of the new item, content based approach can provide relevant recommendation as it doesn't depend on any information about the previous rating of other users.

1.4.4 Black Sheep and Gray Sheep problem

These two are very important issues of recommender systems. Black sheep problem arises when user's preference matches with a very few or no users. Gray sheep problem refers to a situation when user's preference doesn't correlate with any other user or group of users. Due to these problems, recommender systems fail to perform efficiently and accurately.

1.4.5 Scalability

It is the property of the system which specifies its capability of handling the growing data growing in a graceful way. But CF method fails to handle large amount of data,

as a result it gives inaccurate output. Techniques which are proposed to handle this scalability issue are based on the approximation mechanisms.

1.5 Evaluation metrics of Recommender System

There are several metrics that can be used to measure the performance and accuracy of the RS. These metrics can be categorized as:

- Predictive accuracy metrics, for example, mean absolute error (MAE). It calculates up to what extent a system can predict the ratings of users.
- Classification accuracy metrics, such as precision, recall, F-measure. These metrics are used to measure how well a system is able to distinguish the items accurately.
- Coverage metrics, such as prediction coverage, catalogue coverage. They measure the percentage of items for which the system can provide recommendations and the percentage of the present items ever recommended to the user.
- Rank accuracy metrics such as Pearson's product-moment correlation.

1.5.1 Mean Absolute Error (MAE)

It calculates the mean of the absolute difference between the actual ratings and the predicted ratings [7].

$$MAE = \frac{\sum_{\{a,q\}} |p_{a,q} - r_{a,q}|}{\#r} \quad \text{..... (6)}$$

where, $p_{a,q}$ denotes the predicted rating of item q for user a, $r_{a,q}$ is the actual rating and $\#r$ represents the total count of the ratings. Lower the value of MAE better is the prediction.

1.5.2 Root Mean Squared Error (RMSE)

It is one of the important metrics that can be used to measure the performance of the system. It can be computed as:

$$RMSE = \sqrt{\frac{\sum_{\{a,q\}} (p_{a,q} - r_{a,q})^2}{\#r}} \quad \text{..... (7)}$$

1.5.3 Precision, Recall, F-measure

These metrics are used to measure the performance of the classifiers. Precision measures the fraction of relevant items retrieved out of all retrieved items. For example, the percentage of recommended movies those are actually good.

$$precision = \frac{TP}{TP+FP} = \frac{|good\ movies\ recommended|}{|all\ recommendations|} \quad \ldots\ldots\ldots\ldots\ldots\ldots (8)$$

where, false positives (FP) means the count of genuine profiles that are incorrectly classified, true positive (TP) means the fake profiles identified accurately.

Recall determines the fraction of relevant items retrieved out of all relevant items. For example, the percentage of all good movies recommended.

$$recall = \frac{TP}{TP+FN} = \frac{|good\ movies\ recommended|}{|all\ good\ movies|} \quad \ldots\ldots\ldots\ldots\ldots\ldots (9)$$

where, false negatives (FN) is the fake profiles that are not correctly identified. The F-measure combines precision and recall and gives equal weightage to each of them.

$$F - measure = \frac{2*precision*recall}{precision+recall} \quad \ldots\ldots\ldots\ldots\ldots\ldots (10)$$

1.6 Structure of Thesis

Chapter 2: This chapter explains the attack models and classification techniques. The work done by the researchers in this field has also been described in this chapter.

Chapter 3: In this chapter, the problem statement and its objectives have been discussed.

Chapter 4: The proposed framework and the tools used for its implementation are explained in this chapter.

Chapter 5: The experimental results and its analysis have been done in this chapter.

Chapter 6: The results have been concluded and future work has been explained.

CHAPTER 2

LITERATURE REVIEW

2.1 Evolution of Recommender Systems

Recommender systems play an important role in our day to day life. Today, people depend on recommender systems because these systems help them in selecting items of their choice. The idea of recommender system is not new; earlier the recommendations were made offline i.e. the items were suggested by the known people. But now, the large numbers of items and services are available online so the online recommender systems have gained popularity. When similar types of items are being sold by so many online retailers then these system play an important role. The recommender systems make use of several techniques to produce the best and accurate results. The recommender system can be based on content based filtering, collaborative filtering, knowledge based filtering and hybrid filtering which uses the combination of these approaches.

2.2 Overview of Collaborative Filtering

Recommender systems based on collaborative filtering provide recommendations on the basis of liking or disliking of similar users [2]. Collaborative filtering based recommender systems which are available freely are described below:

Lenskit: It is a toolkit for building recommender systems and uses collaborative filtering algorithms.

Crab: It is a python based component used to build recommender systems.

Apache Mahout: It is a java based machine learning library, which creates recommender system by using collaborative filtering approach.

Vogoo PHP Lib: It is a powerful collaborative filtering based system using which personalized features can be added to the websites.

Collaborative filtering technique is partitioned into two types: memory based CF and model based CF [3]. Memory based CF uses sub sample or whole dataset to generate recommendations which lead to the scalability problem. User based CF is the most commonly algorithm but item based CF is more robust than user based algorithm.

Model based CF algorithm uses user database to train a model and then recommendations are generated.

Hybrid recommender systems are gaining popularity day by day because they overcome the problems of content based and collaborative filtering based systems.

2.3 Profile Injection Attack

Profile injection attacks refer to the promotion of the attacker's item or demotion of his opponent's item. In such attacks, the fictitious user creates a large number of attack profiles using any automated tool and injects them into the database of the system benefits. As a result, the system would generate recommendations that are irrelevant to the genuine user due to which he might lose his trust on that particular recommender system. Insertion of the fake profiles can be done either using some tool or manually. If an attacker wants to insert large number of fake profiles in the system, then it is not an optimal way to insert them manually. In that case, he should use some automated tool to generate and insert them into the system.

Profile injection attacks can be categorized into two types i.e. push attack or nuke attack [7]. The type of attack depends on the aim of the attacker. If he wants to promote his item then it is push attack and if he wants to demote the item of his opponent then the nuke attack is mounted. In push attack, maximum rating is assigned to the target items where as in nuke attack, minimum rating is given to the target item in order to demote it.

2.3.1 Elements of Attack

An attack has attack profiles which bias the recommender system's results to their benefits, biased data and a set of target items. These attacks can be categorized on the basis of following elements:

i. Knowledge required for mounting an attack
ii. Attack size
iii. Profile size
iv. Intent of the attack

i) Knowledge required for mounting an attack: Some efforts are required to mount an attack against the system. Gathering the knowledge about the system is one such

effort. The attacks are categorized into two types based on the amount of knowledge needed by an attacker about the system: high knowledge attack and low knowledge attack. In high knowledge attacks, an attacker must know the distribution of ratings in a system. A low knowledge attack is one that doesn't require details of the system. From attacker's perspective, the attack that has more impact and requires less knowledge is the best attack to mount.

ii) Attack Size: Size of attack is the number of unscrupulous profiles inserted by the attacker. Generating and injecting the fake profiles automatically require less cost and effort.

iii) Profile Size: It is defined as a number of items rated in an attack profile. Providing ratings to an item requires less cost as compared to that required for creating the attack profile.

iv) Intent of an attack: The attacker assigned rating to the target items based on the purpose of attack. A malicious user may inject biased profiles to make less popular item to be more likely "push" and most popular item to be less likely "nuke".

The example of push attack is shown in Table 2.1:

Table 2.1 An example of push attack that favors target item, book 3

	book1	**book2**	**book3**	**book4**	**book5**	**book6**	**Correlation with Allie**
Allie	**2**	**5**	**?**		**3**	**3**	
User1		2	1	4		4	-1.00
User2	1	3	2	1		3	0.76
User3	5		1	5	1		-1.00
User4	3	4	2	3	3		0.94
User5	3	3	1	3	1	2	0.21
User6	2	4	1		1	3	0.72
User7	3			2	1		-1.00
Attack1		**5**	**5**	**2**		**3**	**1.00**
Attack2	**1**	**5**	**5**	**2**		**4**	**0.89**
Attack3	**2**	**5**	**5**		**2**	**2**	**0.93**
Correlation with book3	-0.55	0.85		-0.59	0.48	0.00	

Consider a book recommender system which contains six books and seven genuine users. Allie created her profile from previous visits and come back for new recommendations. Table 2.1 shows the profiles of the genuine users and the profiles of attackers. These attack profiles assigned maximum rating to target item i.e. book3. Let the system is using traditional user based CF approach that predicts the rating of book3 for Allie by finding the similar users. Before attack, user similar to Allie is user 4. The predicted rating corresponding to book3 is 2 which mean Allie also dislikes it. But after the insertion of three attack profiles i.e. attack1, attack2 and attack3, attack1 is successfully mounted. Attack1 profile becomes peer to Allie. The predicted rating for book3 is now 5 which mean Allie would like this item.

Now, suppose the system is using item based CF approach. The predicted rating for book3 is calculated by finding the similarity between the items. This method prevents the attack against the system because malicious user does not have any control on the other users for their ratings. If some knowledge about the ratings distribution is obtained, then attack can be successful. From the Table 2.1, it can be observed that item book2 is famous among a group of users to which Allie also belongs.

The attack profile is created such that maximum rating is assigned to both book2 and book3. This may increase the similarity between these two items and there is a high probability that book3 will recommended to Allie.

2.3.2 Shilling Attack Models

An attack has attack profiles which bias the recommender system's results to their benefits, biased data and a set of target items. The motive of an attacker is to construct attack profiles using attack models which have high influence and require minimum knowledge. There are four attack models that can be used for mounting the push attack*: random attack, average attack, bandwagon attack, segment attack* [11]. The structure of an attacker's profile is shown in Table 2.2.

Table 2.2 Structure of attacker's profile

I_S			I_F			I_N			I_T		
I_1^S	…	I_k^S	I_1^F	…	I_m^F	I_1^N	…	I_q^N	I_1^T	…	I_n^T
$\delta(I_1^S)$	…	$\delta(I_k^S)$	$\beta(I_1^F)$	…	$\beta(I_m^F)$	Null	…	null	$\gamma(I_1^T)$	…	$\gamma(I_n^T)$

A profile for mounting an attack consists of k-dimensional ratings vector; k is the number of items in the RS. The k-dimensional vector is divided into four sets: I_S, I_F, I_N, I_T. The details of these four sets of items are discussed below:

- *I_S:* A set of randomly selected items that have some relationship with the target items. Their ratings are generated by the function $\delta(I_k^S)$.
- *I_F:* A set of filler items, selected randomly whose ratings are generated by the function β (I_m^F).
- *I_N:* A set of items those are not rated.
- *I_T:* A set of target items. All target items are assigned rating to maximum i.e. $\gamma(I_n^T)=r_{max}$ (push attack) or minimum i.e. $\gamma(I_n^T)=r_{min}$ (nuke attack).

2.3.2.1 Push Attack Models

a) Random Attack

It is a low knowledge attack, in which filler items (I_F) are selected in a random manner and rate them by using normal distribution with standard deviation and mean rating of the system [11]. In this model, the selected item set is empty i.e. I_S=null. The set of targeted items are rated with minimum or maximum depending on the type of attack i.e. nuke or push. For example, rating is in between 1 and 5, where 5 means liked item and 1 means disliked item, therefore, in push attack $r_{target}=5$ and in nuke attack, $r_{target}=1$. In the experiment, it has been shown that this attack is not much effective in user based as well as in item based algorithm.

b) Average Attack

It is more sophisticated than other models. But it is impractical to implement because it requires knowledge about the system, as it uses individual average ratings for each item instead of global mean of the system [11]. Attackers select filler items randomly and rate the items in the database using normal distribution with mean and standard deviation of individual item. Attackers are difficult to distinguish when compared to

actual users, therefore they have large impact on the system's result. Rating pattern of targeted items is same, as in random attack. It has been shown in our experiments that, this model is not much effective. This attack model is considered as high knowledge attack as it requires the average rating of individual item. The experiments have shown that average attack is highly effective on user based algorithm when the average ratings are assigned to a small subset of items in the database, thus reducing the knowledge requirement. However, this attack is not much effective in case of item based algorithm.

c) Bandwagon Attack

In this model, attacker takes advantage of Zipf's law distribution of popularity and generates the biased profiles that contain most popular items [11]. Popular items are those items that are rated by lots of users. Therefore, there is a high possibility that attackers become similar to the actual users. I_S, a set of frequently rated items, therefore, these items together with the items in a target set, I_T are assigned maximum ratings. The items in filler set are chosen in a same way as in random attack. This attack model is considered as low knowledge attack because in order to determine the popular products in any product space, knowledge required about the system is less.

d) Segment Attack

It requires less knowledge about the system. The basic concept behind this attack is to popularize the target items among a group of targeted users [15]. For example, an author of a romantic novel want his novel to be recommended to the readers who are the lovers of "The Notebook" (another romantic novel), not to the ones who like comics.

The fictitious user determines a set of segmented items which have high chances of being preferred by targeted users, who belong to his/her particular segment. Maximum rating is assigned to segmented items i.e. I_S. To maximize the attack's impact, items in the filler set, I_F are assigned ratings to the minimum, i.e. r_{min}=1, thus maximize the variations between the item similarities.

These push attacks are summarized in Table 2.3.

Table 2.3 Features of push attack models

Attack Models	I_F		I_S		I_T (push/nuke)	I_N
	Ratings	Items	Ratings	Items		
Random	overall mean	randomly selected	Empty		r_{max}/r_{min}	Empty
Average	item mean	randomly selected	Empty		r_{max}/r_{min}	Empty
Bandwagon	overall mean	randomly selected	r_{max}/r_{min}	popular items	r_{max}/r_{min}	Empty
Segment	r_{min}/ r_{max}	randomly selected	r_{max}/r_{min}	Segment items	r_{max}/r_{min}	Empty

2.3.2.2 Nuke Attack Models

The push attack models can also be used for mounting nuke attack. This can be achieved by assigning minimum rating i.e. r_{min} to the target item instead of r_{max}. But previous research shows that the attack models that are effective for pushing items are not necessarily effective for nuke attacks. There are two attack models specifically for nuking items and both of these attack models can be considered low knowledge attacks as they do not need any system specific knowledge.

a) Reverse Bandwagon Attack

It is a variation of the bandwagon attack. In this attack, minimum ratings are assigned to the items in a target set and to the selected items. It increases the possibility that system would generate low predicted ratings for those items. The selected items are those which have ratings below the average. This attack has less impact on user based systems. But, it is very effective attack against item based RS.

b) Love-Hate Attack

This attack is very simple to mount because it requires no knowledge. The items in the filler set are assigned maximum rating i.e. r_{max}. Minimum ratings are provided to the target items. This attack can be used as push attack by switching r_{max} and r_{min}. Very less knowledge is required to mount an attack. Prior studies demonstrate that

this attack is not very effective when used as a push attack but it is one of the most effective nuke attacks in the user based CF systems.

These nuke attacks are summarized in Table 2.4.

Table 2.4 Features of nuke attack models

Attack Models	I_S		I_N	I_F		I_T (push/nuke)
	ratings	Items		ratings	Items	
Reverse Bandwagon	r_{min}/r_{max}	unpopular items	null	system mean	randomly selected	r_{max}/r_{min}
Love-Hate	Empty		null	r_{min}/r_{max}	randomly selected	r_{max}/r_{min}

The term "shilling" was first given by Riedl and Lam who introduced two attack models: Random Bot and Average Bot to inject attackers' profiles into the system [7]. They explored the factors that may affect the effectiveness of the attacks. These factors include which algorithm has been used, whether the system is generating predictions or recommendations, whether the attacks are detectable or not and the characteristics of the item being attacked. In this, automated CF was used for generating the recommendations. Prediction shift and mean absolute error (MAE) are the evaluation metrics that have been used to measure the effectiveness of the attack. These evaluation metrics are most suitable for the systems that produce predictions for the items. The performance of algorithm is measured before and after the attack. Using prediction shift they concluded that average attack has highest impact in user-user as well as in item-item algorithm whereas random attack has lowest impact on the system.

The effectiveness of attack is also influenced by the rating distribution of target item. They hypothesized that the characteristics of rating distribution of item have influence on the impact of attack on that item. Features of an item include popularity, entropy and likability. The ratings of unpopular items and items with high spread of ratings can be manipulated easily. The items which are liked by most of the users are easy to be pushed. It has been proved that item based algorithm offers advantage over user based algorithm because in item based algorithm, attacks are not much successful in altering the system's result.

Herlocker presented an algorithmic structure to perform collaborative filtering and novel algorithm which would improve the accuracy of prediction algorithms [8]. Automated collaborative filtering approach has been used to predict that how much a user would like an item not rated by him. They concluded by analyzing prior data gathered from movie prediction site. New invention is there in each of the three steps of prediction algorithm based on neighborhood. According to them, Pearson's correlations as well as Spearman correlation perform equally well. If the rating scale is discrete then Spearman correlation was used to measure the similarity. If the rating is continuous then Pearson's correlation may be used for finding the similar items. If rating is binary then different approach can be used.

An improved non-personalized average algorithm has been discovered by Herlocker to generate predictions when there is less knowledge of the user [8]. The improved algorithm to compute deviation from mean average has been found.

$$p_{a,i} = \bar{r}_a + \frac{\sum_{u=1}^{n}(r_{u,i} - \bar{r}_u)}{n} \qquad \ldots\ldots\ldots\ldots\ldots\ldots (11)$$

Sarwar examined the item based filtering technique which uses user-item matrix to find the similarity between the items and use these similarities to generate recommendations [9]. Different item based prediction algorithms have been analyzed. Cosine similarities between the items and item-item similarities are used to compute correlation between the items. Predictions are generated using regression and weighted sum techniques and these algorithms are compared with standard kNN algorithm. They concluded that item-item approach generates better quality of predictions than kNN approach. This improvement is stable for different neighborhood size. Item neighborhood is static which is pre-computed and has high online performance. It has been proved experimentally that item based algorithm performs better and provides better quality of prediction than user based algorithm.

At Amazon.com, recommendation algorithm has been used to customize the online store for every customer. The store changes radically on the basis of customer's interests. Conversion rates and click through are the important measure of web based and email advertising. To solve the problem of recommendations following approaches can be used: traditional collaborative filtering, search based techniques

and cluster models [10]. These methods are compared with item- to- item collaborative filtering. The online computation of item-item algorithm scales independently of the count of users and items in the database and produces high quality of predictions.

Two algorithms which focus on finding similarities between the users are cluster models and collaborative filtering. Search based methods and item to item algorithm focus on the correlation between items instead of users. The main motive of cluster method is to assign target user to the cluster having most similar users. Performance and scalability of cluster model is better than traditional CF algorithms because user is compared to the number of segments instead of whole customer base. The search based method is also known as content based method. The items rated and purchased by the user are given and based on that algorithm a query is built to search other famous items by same author, director or with closer subjects or keywords. This algorithm performs well only when the user has few purchases and ratings. The quality of recommendations is poor in all the cases. The following item to item algorithm is better than other algorithms, which calculates the similarity between an item and all related items [10]:

For each item in catalog, I_1
For each customer C who purchased I_1
For each item I_2 purchased by customer C
Record that a customer purchased I_1 and I_2
For each item I_2
Compute the similarity between I_1 and I_2

This algorithm scales independently of the number of the users and the catalog size. It only depends on the number of titles purchased by a user. It performs fast even for large datasets.

Item based CF has more security than user based CF algorithm. Mobasher experimentally proved that although item based approach has more security than user based approach still it suffers from profile injection attacks [11]. They proposed a novel attack model which concentrates on a subset of customers having similar choices and proved that this attack can be successful against item based approach. The attack profiles were created using random, average, bandwagon, segment and favorite

item attack model. Attacks consist of attack profiles that contain biased data associated with malicious users. User based and item based algorithm was used to produce predictions. Prediction shift and hit ratio are the evaluation metrics that have been used to measure the effectiveness of the attacks and concluded that low cost segmented attack deploys successfully against item based CF system.

Various attack strategies: popular attack strategy, probe attack strategy and rating strategies have been introduced [12]. While creating the attack profiles, there are two important issues that should be taken care of. The first one is the selection of products that should be rated by the attacker and the second issue is assigning rating to those selected items. Popular attack strategy limits the number of items to be considered for specific subdomain. The selection of popular items minimizes the cost of attack. The advantages of using this strategy for creating the attack profiles are:

- The popular items have chances of getting high ratings from the users.
- The similarity between the items rated by attacker and genuine users is increased.
- Each malicious profile has a higher chance of being the neighbor of genuine user because of which the cost required to create the number of attack profiles gets reduced.

But, this approach has a limitation that the attack profiles created using popular attack strategy are easily detectable, particularly when the large count of attack profiles are generated. Probe attack strategy overcomes the drawback of popular attack strategy. In this, a small number of items are selected and rated initially; the malicious user progressively generates the fake profiles which could match the items rated by genuine users. The high similarity between the genuine user and original user is assured. This approach has another advantage that it is a low knowledge attack because it requires less knowledge about the domain. Rating strategy has been adopted specifically for user based approach. For calculating the predictions, deviation from mean is considered. Automated collaborative technique has been used to predict the ratings for the target users. They used good predictions (GP) metrics for evaluating the push attacks and bad predictions (BP) for evaluating the nuke attacks. GP for an item j is defined as the number of times the equation (12) holds true:

$$p_{u,j} : p_{u,j} \geq \delta_g, \forall\, u \in U_j \quad \text{..... (12)}$$

δ_g denotes the threshold rating, U_j represents the all original users who have given rating item j. Similarly for nuke attacks, BP is the number of times the following equation (13) is true.

$$p_{u,j} : p_{u,j} \leq \delta_b, \forall\, u \in U_j \quad \text{..... (13)}$$

In the experiments, researchers set the value of threshold rating = r_{max}-1 in case of push attacks and ½ (r_{max}-r_{min}) in case of nuke attacks. Therefore, popular attack strategy should be used only when attacker should have full knowledge of the domain. Probe strategy should be used when one is having less domain knowledge and attack profiles created using this approach are difficult to detect.

The various types of attack models have been discussed in [13]. Random, average, probe, bandwagon, segment are the push attacks that have been used and love/hate attack and reverse bandwagon attacks are used for nuke attacks. Traditional item based CF algorithm and user based CF algorithm has been used to predict the ratings. The stability of the attacks is measured by using hit ratio and prediction shift. They also incorporated model based technique. K-means and Probabilistic latent semantic analysis (PLSA) models are used for generating the predictions and recommendations. It has been proved experimentally that PLSA has more security than kNN or k-means algorithms. Other unsupervised niche clustering may be more robust than these algorithms. The dataset is divided into five categories:

- Low average rating, low density
- High average rating, low density
- Low average rating, medium density
- High average rating, medium density
- High average rating, high density

Their results suggested that it is difficult to detect items those are rated highly but the approach detects these attacks effectively against low density items, which are more vulnerable. The results showed that it is not possible for a single method to remove the attack threat. So these techniques should be integrated to detect the attacks.

O'Mahony analyzed the robustness of the collaborative filtering technique [14]. Robustness means the capability of making recommendations inspite of noisy ratings of items. Stability and accuracy are the two facets to robustness. They proposed a framework to measure the stability of the recommendation algorithms.

Segment based attacks against CFRSs has been introduced which ensures that the item pushed by the attacker would be recommended to the target users [15]. Segment attack requires very little details of the system for mounting the attack. In this, only a small number of users in a segment are targeted to be attacked. The pushed item is likely to be recommended to the users in this segment. The attack has been mount on item based as well as on user based CF algorithm. Hit ratio and prediction shift are used to measure the stability and effectiveness of the attacks. The results show that segment attack is more effective against item based algorithm than user based.

2.4 Shilling Attack Detection using Classification

The most popular data mining techniques, used in identifying the attack profiles in the recommender systems based on collaborative filtering are described in this chapter. Data mining consists of the following steps: data preprocessing, data analysis and data interpretation. Figure 2.1 shows the flow of data mining process and various algorithms used.

2.4.1 Data Preprocessing

There is a step where processing of data i.e. data cleaning, filtering and transformation before analyzing the data. Some of the preprocessing methods are discussed below [16].

2.4.1.1 Sampling

We often have datasets of large size and processing of such a large data is very costly. The dataset is partitioned into samples. The samples of data are selected with the help of sampling. Sampling can be done using various methods; the most popular sampling method is random sampling. In this, there is an equal probability of selecting every item. Stratified sampling is another method used for sampling in which data is partitioned into samples on the basis of some characteristics. These samples are used as training and testing datasets. The training dataset is used to train the predictive

models and testing dataset is used for the evaluation of the models to ensure that they perform well.

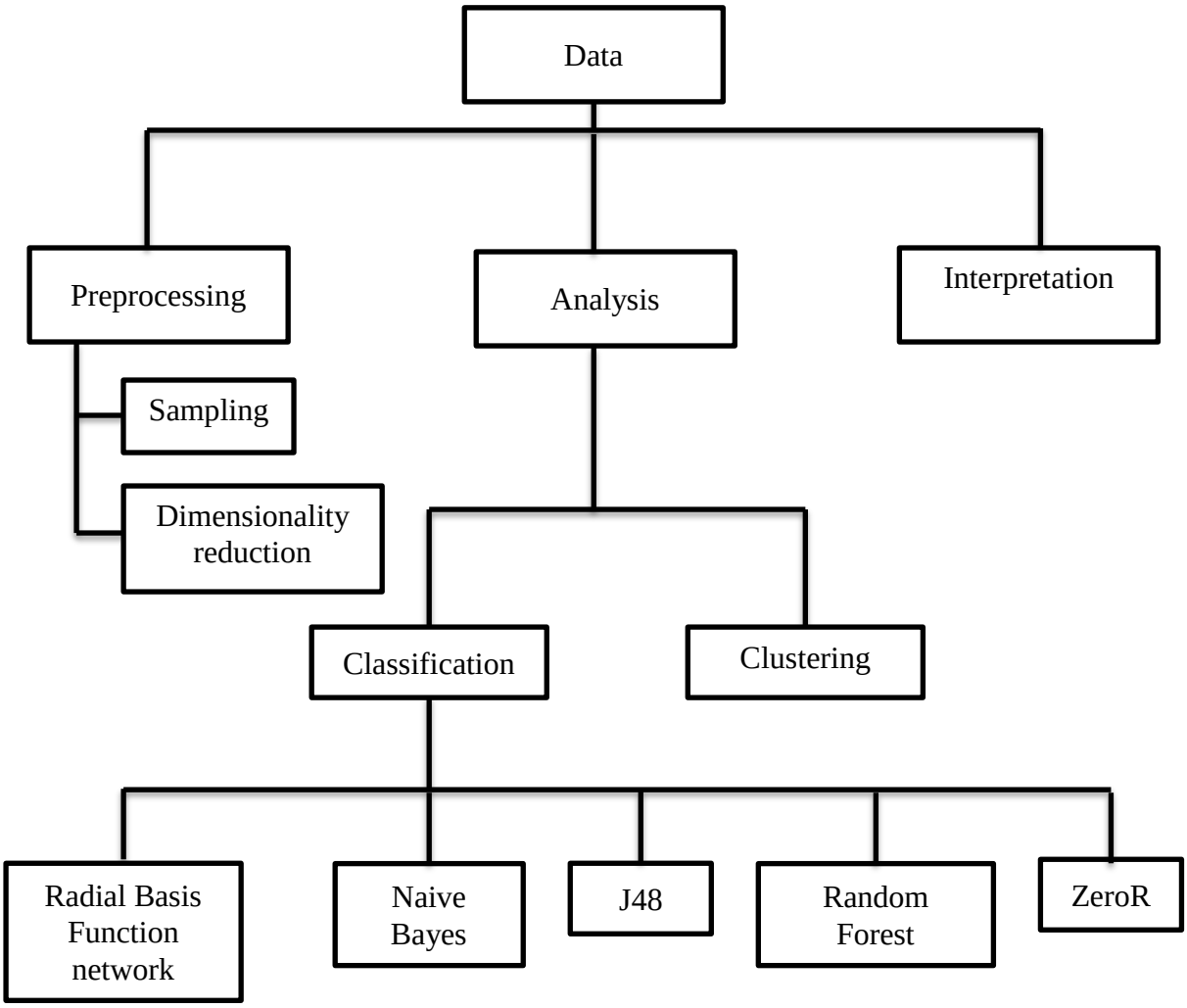

Figure 2.1 Flow of Data Mining process

2.4.1.2 Dimensionality Reduction

It is a technique which is used to downsize the data. It converts a set of high dimensional vector to low dimensional space and preserves the metrics among them. Dimensionality Reduction methodology is a data pre-processing step. It finds an appropriate low dimensional representation of the original data. If the dimensionality is diminished, the accuracy and efficiency of data analysis will be improved. It is possible that datasets have characteristics that are in large dimensional space and also have sparse information in that space. The notion of density and distance between the points, that are critical for outlier detection and clustering, becomes less meaningful in

high dimensional space. This is known as curse of dimensionality. Dimensionality reduction approach overcomes this problem by transforming original higher dimensionality space into a low dimension space. Its results are so directly applicable to computation of the predicted value that now this is considered to be an approach to design, rather than a pre-processing technique. The machine learning and data mining techniques might not be effective for huge dimensional data because of curse of dimensionality.

The major techniques of dimensionality reduction are:

- Feature selection.
- Feature Extraction (reduction).

a) Feature selection

Feature selection approach tries to find out a subset of original variables (features). Three strategies are there: filtering, wrapping and embedded approach. There are cases in which data analysis such as classification and regression can be done in the reduced space more accurately than in the original space. It is a process which chooses an optimal subset of features according to objective function. Its objectives are to reduce dimensionality and remove noise, speed of learning, to improve mining performance, predict accuracy and simplicity of mined results.

b) Feature extraction

It is also known as feature reduction. The mapping of original higher dimensional data onto a lower dimensional space is called feature reduction. The data transformation may be linear, as in principal component analysis, but many nonlinear dimension reduction techniques are also there. Then the lower dimensional representation for a given set of data points of q variables is computed. The main difference between feature selection and reduction are: in feature selection, only subset of original features is selected and they are continuous versus discrete and in feature reduction all original features are used and the transformed features are linear combinations of the original features. The goals of techniques of dimensionality reduction are: high efficiency**,** high effectiveness, able to handle both irrelevant and redundant features, less costly than existing subset evaluation methods, not pure individual feature

evaluation, and not traditional heuristic search methods. Limitations of existing methods in feature selection are:

- Individual feature evaluation: Focus on identifying relevant attributes without handling feature redundancy. Its time complexity is O (N).
- Feature subset evaluation: Rely on minimum feature subset heuristics to handle redundancy implicitly while pursuing relevant features and its time complexity is at least O (N^2).

2.4.2 Data Analysis

After the preprocessing step, data need to be analysed to obtain relevant information. In this section, a brief description of classification models is provided.

2.4.2.1 Classification

Classification is a technique which maps a label space and attributes space [16]. Label space represents the class to which the instances belong and attributes space represents the features of the instances present in the dataset. For example, a classifier implements a movie recommender system and classifies movies into one of the three categories: hit, average or flop. In machine learning, classification is known as a supervised technique. Various classification models are discussed below.

a) Random Forest

Random forest is an ensemble learning technique for classification and regression. It merges the concept of feature selection and bagging. Its training algorithm uses the general method of bootstrap aggregating. In this algorithm, multiple trees are built at training time and the class is produced as output [16]. It eliminates the problem of overfitting. When users don't have knowledge of which model they should use, then random forest is a good model to use. Users can use this model when they need to build a decent model in a short period. Lesser is the correlation between the trees of the forest, lesser will be the forest error rate. Strong classifier is the one with low error rate. Large datasets can be used to train its classifier. It automatically identifies the important variables as well as calculates the missing values. This model gives

accurate results even when the data is sparse. The forests generated on one dataset can be used on different datasets in future.

b) J48 Decision Tree

J48 is a Java implementation of C4.5 decision tree in WEKA tool. C4.5 is an algorithm which is used to create decision tree [17]. It is an extended version of ID3 algorithm. C4.5 is also known as a statistical classifier because the decision tree created by this algorithm is used for classification. C4.5 uses information entropy to construct the decision trees from a training dataset in a similar manner as created by ID3 algorithm. It handles both discrete and continuous attributes. It prunes the trees after its creation. It replaces useless branches by leaf nodes.

c) Radial Basis Function Network

Radial basis function network (RBF n/w) is an artificial neural network which uses radial methods as activation methods. This model outputs a linear combination of neuron specification and radial methods of the input. It is two-layer feed forward network, which means three layers are there in RBF n/w: an input layer, a hidden layer and an output layer. The nodes in the output layer execute linear summation functions whereas nodes in hidden layer execute a set of non-linear RBF such as, Gaussian functions. The learning process is very fast in RBF networks. The training network is partitioned into two phases: first phase is to determine the weights from input layer to hidden layer and in second phase, weights from hidden layer to output layer are determined as shown in figure 2.2 [18].

d) Naive Bayes

Naive Bayes model is a probabilistic classifier which is based on Bayes's theorem. These classifiers are scalable and require a number of parameters for learning process. They assume that the value of a particular parameter is not dependent on the value of any other parameter. For example, a vegetable may be considered as cabbage having a diameter of 15 cm, green in color. The Naive Bayes classifier considers that these values are independent of each other. There is a probability that the vegetable is a cabbage regardless of any association between diameter and color parameter.

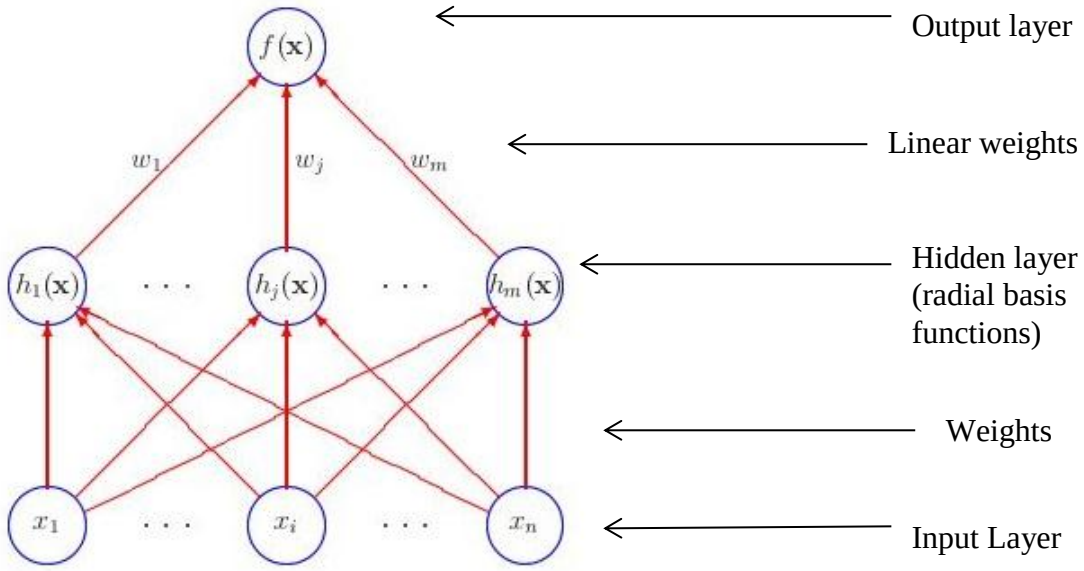

Figure 2.2 Architecture of RBF network [18]

e) ZeroR

ZeroR is a very simple technique used for classification. It ignores all the attributes present in the dataset and only depends upon the class to which an instance belongs. It performs prediction about the majority class. It only determines and sets the baseline performance as a standard for other classification techniques. The predictability power of this algorithm is zero.

The attack was detected using classification techniques. Six supervised models were compared and the researchers observed that neural network, random forest and SVM have higher performance [16]. They ensemble these models and built a new model which outperforms in most of the cases.

Three classification techniques: kNN, SVM, C4.5 have been used in [17]. Both generic and model specific attributes are identified to detect the fake profiles and improve the robustness of the recommendation engine. To measure the performance of classifiers, sensitivity and specificity metrics were used. MAE was used to evaluate the accuracy of the prediction algorithms. The sensitivity results demonstrated that C4.5 and SVM were able to detect the fictitious profiles accurately. But they perform oppositely in context of specificity. It is difficult for kNN algorithm to detect the attack profiles at low filler size. The overall performance of SVM is best.

Chirita introduced an algorithm of evaluation metrics to detect and remove the attack profiles from the system [19]. A novel algorithm to protect the recommender systems was proposed and evaluated. It maintains the good quality of the predictions. The metrics that are used to identify the attack profiles are: Standard deviation in ratings, degree of agreement, degree of similarity, number of differences in prediction, RDMA. An improved algorithm is a two-step process:

1) Average similarities using top neighbors for all the users are computed.
2) Select only those users who have mean similarity < ½ (max mean similarity). Then a function which computes the chance that a user is an attacker is associated with each value of RDMA. The function which is used in detection is expressed in equation (14).

$$f(x) = \frac{1}{e^{\alpha} - 1}\left(e^{\alpha \frac{x - Avg\_RDMA}{1 - Avg\_RDMA}} - 1\right) \quad \text{..... (14)}$$

Furthermore, integration of the mean similarity metric and RDMA results in strong metric that was used to detect the attackers. The method of finding the similarity between the users has been modified as follows:

$$W'_{i,j} = W_{i,j} * (1 - PS_j) \quad \text{..... (15)}$$

where, $W_{i,j}$ denotes the similarity between user i and user j, PS_j represents the probability of being attacker. Probability PS for genuine user is 0 whereas for malicious user is 1.

Mobasher and Burke studied various detection attributes and used kNN algorithm detect the fake profiles and to improve the strength of the system [20]. They studied various generic and type specific attributes to identify the malicious profiles. It has been shown that the classification technique which uses type specific attributes as well as generic attributes can successfully detect the fake profiles in the system. Recall and precision are the evaluation metrics used for evaluating the performance of the classifier. Classifier and its result was generated using WEKA data mining tool. The conclusion has been made that love-hate attack and segment attack are difficult to detect at low filler sizes.

Zhang and Zhou proposed a detection model by including ensemble technique and back propagation neural network and ensemble technique that improves the low precision of existing supervised approaches [21].

Shilling attack can also be detected using unsupervised or semi-supervised techniques. Lee and Zhu developed an attack detection technique by using clustering algorithms and multidimensional scaling but it is not useful for recognizing attack with small filler sizes [22]. In order to improve the power of the item based CF algorithm, a new collaborative filtering technique has been proposed by building various user models and DBSCAN clustering technique is used to detect the malicious users [23]. A new unsupervised attack detection approach i.e. RD-TIA [24]. A new detection attribute, DegSim' which succeeded in detecting segment and other group attacks was also introduced. Dhimmar and Chauhan used ECLARANS and PAM clustering algorithms to detect the spam users and proved that former algorithm has higher accuracy than later [25].

In addition to supervised and unsupervised techniques, there are semi-supervised approaches that can be used to detect the attackers. Semi-SAD detector has been introduced by applying semi-supervised techniques [26]. It uses unlabeled profiles for improving the performance of detection. Zhang examined that topic level recommendation algorithm based on trust is more secure [27]. It incorporates topic oriented trust model into CF algorithms under average attack and he concluded that this CF algorithm has more stability than standard kNN approach under average attack. Zhang proposed an average hybrid and bandwagon attack model, analyzed their effectiveness against trust-based recommendation algorithm and showed that the proposed hybrid attack model has more impact on recommendations generated by system than other attack models [28].

Donovan and Smyth introduced trust based models in CF with the aim to improve the accuracy [29]. They concluded that even trust based models are more vulnerable to shilling attack. In a later research, they modified the trust building process and solved this problem, which reduces the prediction shift by 75 percent as compared to classic CF algorithm. Calculating the similarity between the users is difficult because of sparse data in the matrix of ratings, so a trust metric is required to solve this problem. Avesani and Massa introduced a robust CF algorithm based on "web of trust" metric [30].

The advantages and drawbacks of the collaborative filtering techniques are described in Table 2.5.

Table 2.5 Summary of collaborative filtering techniques

CF categories	Techniques	Advantages	Drawbacks
Memory based CF	- User based CF - Item based CF	- Implementation is easy. - Does not consider content of the items - New data can be added easily.	-depends on user's ratings. - when data is sparse, its performance decreases.
Model based CF	- Clustering CF - MDP based CF - Latent semantic CF - Bayesian belief CF - CF using dimensionality reduction	- improve prediction performance. - better address the scalability, sparsity and other problems.	-trade-off between scalability and prediction performance. -building of model is expensive.
Hybrid	-combination of memory based and model based CF algorithms. For example, Personality Diagnosis	-improve prediction performance. -overcome the drawbacks of content based and CF systems.	- cost and complexity increases. -require external information

CHAPTER 3

PROBLEM STATEMENT

Recommender systems play an important role in choosing relevant items from millions of available products. Collaborative filtering approach is used to construct a recommender system because of its simplicity and efficient performance. But the systems based on collaborative filtering have security issues. They are highly prone to shilling attacks because it depends on the feedback of the customers. The fictitious users who are indistinguishable from genuine users enter the system. They create the unscrupulous profiles using various attack models and inject them into the database of the system. This may alter the result of the recommender system. As a result, the items that are irrelevant to the current user might get recommended to him. This may degrade the performance of the system. Particularly, we are interested in attacks which can be inserted into the system's database with less knowledge of the rating distribution.

The main objective of this thesis is to deal with the shilling attack. Following issues need to be addressed:

i. To test the stability and robustness of the recommender system after the shilling attack.
ii. To check the possibility for a malicious user to mount various types of attacks.
iii. To identify the profiles of the attackers.
iv. To test the performance of the classifiers used to detect the fictitious profiles.
v. To improve the accuracy of the predictive models used to detect the attack profiles.

CHAPTER 4
PROPOSED METHODOLOGY AND TECHNOLOGIES USED

4.1 Introduction to Methodology

The attack profiles are created using various attack models discussed in chapter 2. Then similarities between users and between items are computed using Pearson's correlation in user based collaborative filtering and item based collaborative filtering based systems respectively. The ratings are predicted for target users using a standard algorithm known as k-nearest neighbor algorithm [11]. The effectiveness of the attack is measured via prediction shift. The attack which would have high impact on the system would have higher probability of recommending pushed items to the target user.

There is a need to detect the attack profiles. Data preprocessing is a beginning stage of any data mining task [16]. Classification and clustering are two important techniques which are used to find hidden pattern in a dataset. Classification approach is used when the class is already defined in the dataset. Clustering technique is used when the class to which instances belong is not defined. In this, data is partitioned into meaningful subclasses known as clusters. The objects in a cluster have more similarity than the objects in different clusters have. In machine learning, classification is known as supervised learning and clustering is called as unsupervised learning.

Clustering techniques are of three types: hierarchical methods, density based methods and partitioning methods [23]. Hierarchical methods partition a dataset into a particular hierarchy on the basis of some criteria. Density based techniques such as DBSCAN and OPTICS search for the dense clusters which are partitioned by noise. K-means is the most commonly used partition method. Various classification techniques are Naive Bayes, Random Forest, J48 decision tree, Support Vector Machine, ZeroR and many more. The classification techniques perform differently for different datasets.

4.2 Architecture of Proposed Work

a) To measure the stability of the system, the following steps are followed.

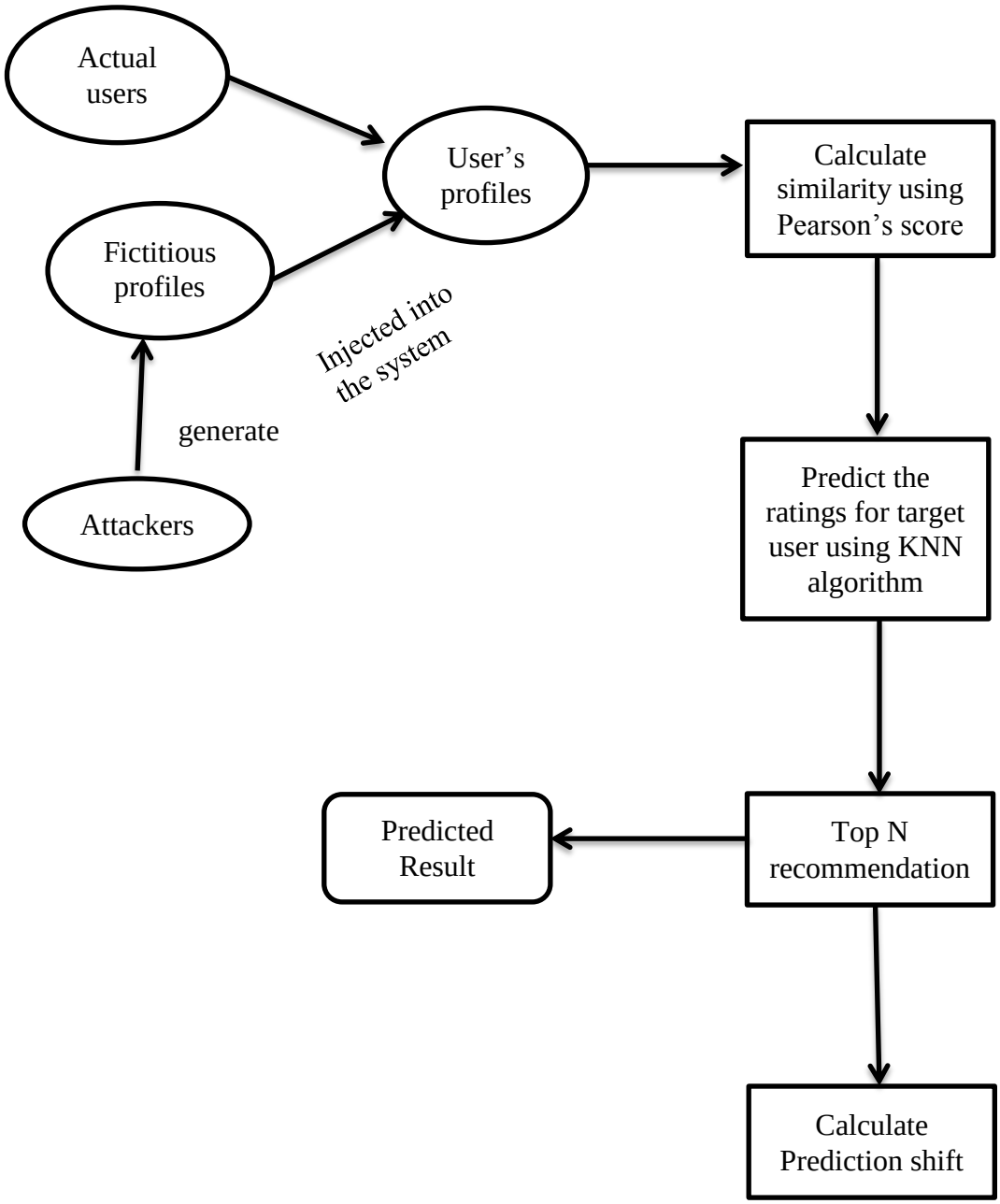

Figure 4.1 Steps to calculate the effectiveness of the attacks

b) To detect the malicious profiles from the user's profiles present in the database of the system, the following steps are followed:

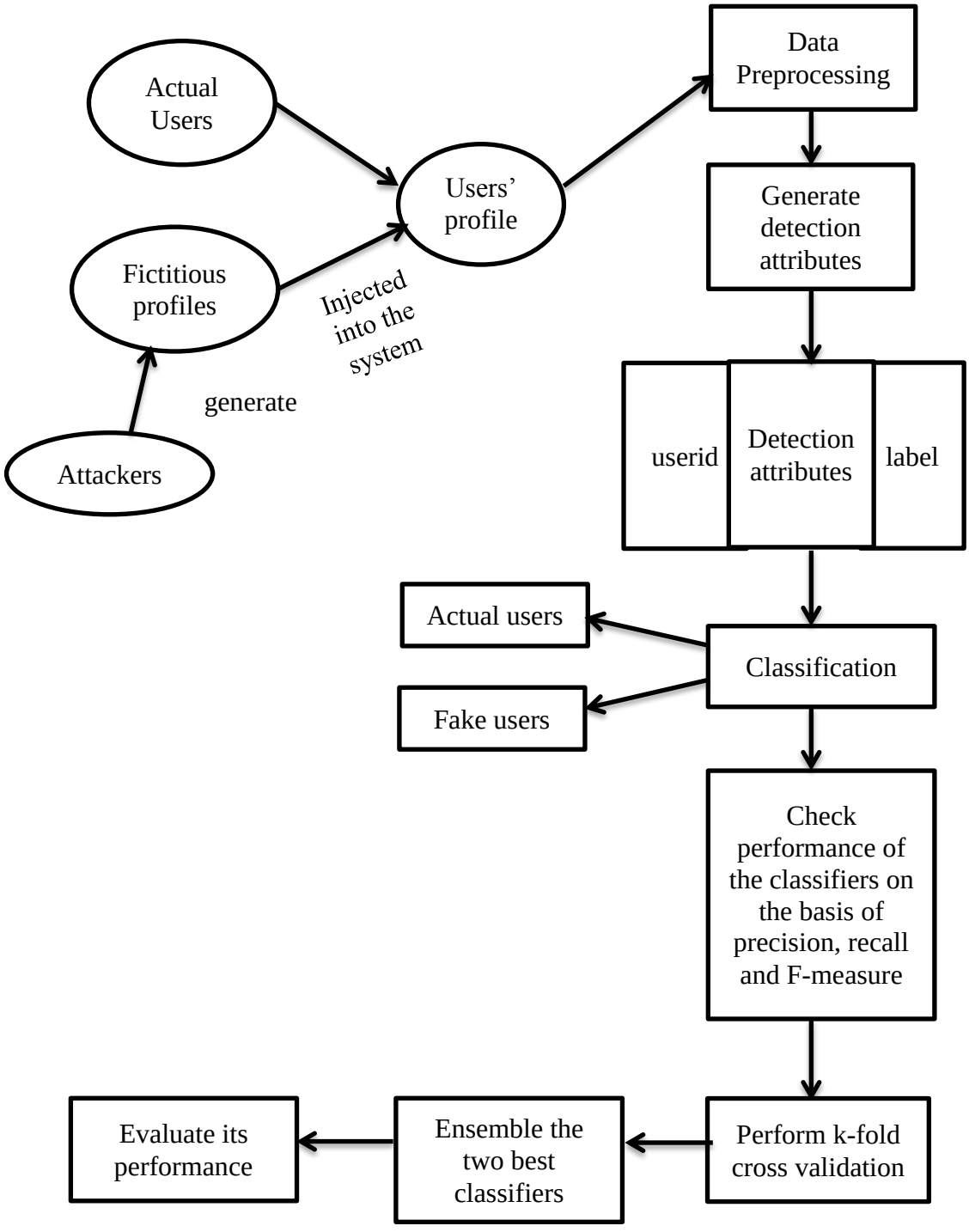

Figure 4.2 Steps to classify attacker's profile

4.3 Technologies Used

Python 3.4 is used to find the predicted value of the ratings for the target users and Waikato Environment for Knowledge Analysis (WEKA) has been used to identify the malicious profiles from the system's database.

4.3.1 Python 3.4

- Python is an open source programming language.
- Guido van Rossum, a programmer introduced it in 1991.
- It is named after the television show Monty Python's Flying Circus.
- Python is an interpreted language i.e. it does not need to be compiled to run.
- Python is a good programming language for beginners as it is easy to understand as compared to other programming languages. PyCharm 5 (Community Edition) IDE is used to write the code in python.
- It is a high-level language. It means a programmers focus on what to do instead of how to do.
- It takes less time to write program in python as compared to other programming languages.
- The standard library of the python consists of many functions which come with it when it is installed. Many other libraries can be downloaded from the internet which makes it a powerful language.
- The important thing which should always be taken care of is the indentation.
- Python does dynamic variable assignment which means programmer does not need to specify the type of the variables due to which these variables can be reused.
- It is used for game programming, web development, desktop GUIs, network programming etc.
- There are many versions of python are available such as, Python 2.7, 3.4, 3.5, 3.6. Python 2.x is the legacy and python 3.x is the present and future of the language.
- Python 3.4 consists of many improvements and bug fixes than python 2.x.

- The drawback of Python is that it is slower than the compiled languages such as C programming language, because it does not directly run machine code.

4.3.2 WEKA Data Mining Tool

- WEKA is a data analysis tool written in Java, developed at the University of Waikato, New Zealand.
- It is used by machine learning and data mining researchers.
- It is an open source software and is available free of cost.
- WEKA is platform independent tool and can be easily installed on any operating system such as Mac, Windows, and Linux etc.
- It is easy to use because of its graphical user interfaces.
- It is a collection of data preprocessing and modelling techniques.
- The interfaces for WEKA are *Explorer, Experimenter, Knowledge flow, simple CLI* as shown in figure 4.3.

Figure 4.3 Interface of WEKA

- Explorer is the main interface which includes data preprocessing, classification, clustering etc.

- Experimenter is a powerful tool using which we can design our own experiments of existing algorithms on datasets, run the experiments and analyse the results.
- Knowledge flow and CLI have same functionalities as WEKA Explorer. The difference is CLI is a command line interface.
- It is up-to-date tool. The new algorithms of machine learning and data mining are added as soon as they found to be important.
- Many free of cost tutorials are available on the internet.

CHAPTER 5

IMPLEMENTATIONAND EXPERIMENTAL RESULTS

5.1 Introduction to Dataset

MovieLens-100k dataset has been used in the experiments which contain 1,00,000 ratings rated by 943 viewers on 1682 movies. The integer value of ratings varied from 1 to 5 where 1 is assigned to most disliked items and 5 to most liked items. The dataset contains those users who rated at least 20 movies.

5.2 Experimental Setup

50 movies are selected from the dataset such that rating distribution for these movies is similar to the overall distribution of ratings. 63 users are taken for testing, showing the overall distribution of users as per the ratings provided. Attack is performed on each movie individually. The attack profiles are created using various attack models at different attack size and filler size. Attack size refers to the number of attack profiles added into the database i.e. suppose there are 1000 users and 100 items in the system then 1% attack size means 100 fake profiles are added into the system. 10% filler size means each attacker gives rating to 10 items. Then the similarities between the users are calculated using Pearson's correlation and ratings are predicted using kNN algorithm. In kNN algorithm, neighborhood size, k=20 has been used in the experiments performed. The attack models, filler size and attack size that have been used in the experiments are given below:

- *Attack models*: random, average, bandwagon and segment attack.
- *Filler size:* percentage of filler items, fixed at 50%.
- *Attack size:* 5%, 10%, 15%, 20%, 25%.

The details of attack profiles are described below:

- I_T*:* 10 unpopular items were selected randomly and assigned the maximum rating i.e. r_{max}=5. Unpopular items are those items which are not liked by most of the users.
- I_F*:* Set of filler items, random ratings are assigned to items which centered around standard deviation=1.1, mean=3.6

- I_S: Selected items which are used only in bandwagon attack. 10 most frequently rated items were assigned to r_{max}, i.e. r_{max}=5.

In case of segment attack, a set of segmented users and items are identified. Five popular horror movies are selected from the dataset as one segment. These horror movies are: Alien, Frighteners, Heavy Metal, The puppet Masters, From Dusk Till Dawn. Users, who gave ratings greater than 3 to any three of these five horror movies, are chosen. Then the combinations of those three movies that have minimum 30 users are taken. 10 users are chosen randomly and the results are averaged. Then the attack profiles are generated which are put into the system and the predictions are generated. To measure the stability of the system, prediction shift is used.

In order to maintain the accuracy of the recommender system, it is required to detect the fake profiles from the system. The detection of attack profiles is described in section 5.7.

5.3 Measuring the Effectiveness of Attack

The robustness of the system determines how it performs before and after the attack and how attacks affect the recommendations generated by the system. The stability measures the shift in the ratings of the pushed item before and after the attack. The stability of the system and effectiveness of the attack is measured via prediction shift.

Prediction shift: The main goal of malicious user in "push" attack is that the targeted items should have higher chances of being recommended to the target users after mounting the attack than before the attack. To compute the stability of the system, prediction shift can be used [11]. Let I denotes the target items set and U be the set of target users. $\Delta_{a,j}$ denotes prediction shift for each pair of user and item (a,j). It can be calculated as $\Delta_{a,j}=p'_{a,j}-p_{a,j}$, where p' and p denote the predicted ratings after and before the attack, respectively. The attack has been implemented successfully if the value of $\Delta_{a,j}$ is positive. For an item j over all users, average prediction shift can be evaluated as:

$$\Delta_j = \frac{\sum_{a \in U}(p'_{a,j} - p_{a,j})}{|U|} \qquad \ldots\ldots\ldots\ldots\ldots\ldots (16)$$

Likewise, for all tested items, average shift in prediction can be calculated as:

$$\bar{\Delta} = \frac{\sum_{j \in I} \Delta_j}{|I|} \qquad \ldots\ldots\ldots\ldots\ldots\ldots (17)$$

Higher the value of prediction shift, higher is the chance of recommending the pushed item to the target user. But it is not true in the case, when the target item has very low scores.

Random and average attacks are effective against user based CF but they have less impact on item based CF [31]. Here random, average and bandwagon attack are performed at different attack size and filler size fixed at 50%. Figure 5.1 and Figure 5.2, show that the average attack performs very well in both item based CF and user based CF. Prediction shift of average attack is highest in almost all cases, making average model most effective and is difficult to detect. But, average attack is difficult to implement because it is a high knowledge attack i.e. it requires knowledge about rating distribution of the system. Also, it can be concluded that, bandwagon attack which requires less knowledge about the system, is comparable to average attack.

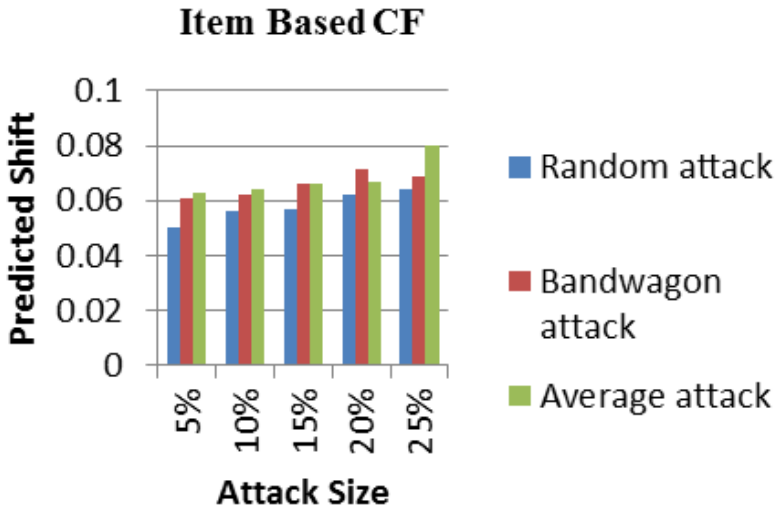

Figure 5.1 Attack models in item based CF.

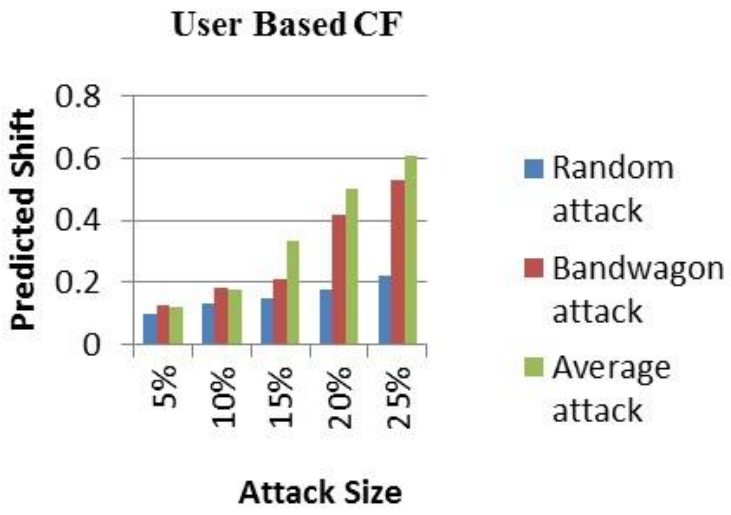

Figure 5.2 Attack models in user based CF.

Prior study demonstrated that these attacks have less impact on item based CF. Figure 5.3 shows that the average attack is the strongest attack as the value of prediction shift of average attack is higher in case of item based CF than that of user based CF. Therefore, former CF algorithm has more security than later.

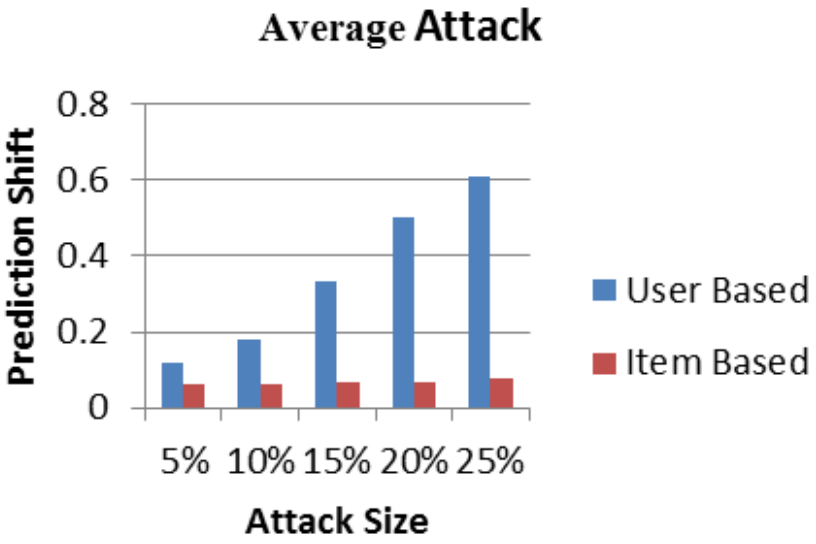

Figure 5.3 Average attack in item based and user based CF.

Prediction shift of in-segment users and all users in item basis algorithm and user basis CF algorithm is shown in figure 5.4 and figure 5.5 respectively, which represents that segment attack is less effective against all users than in-segment users in both the cases.

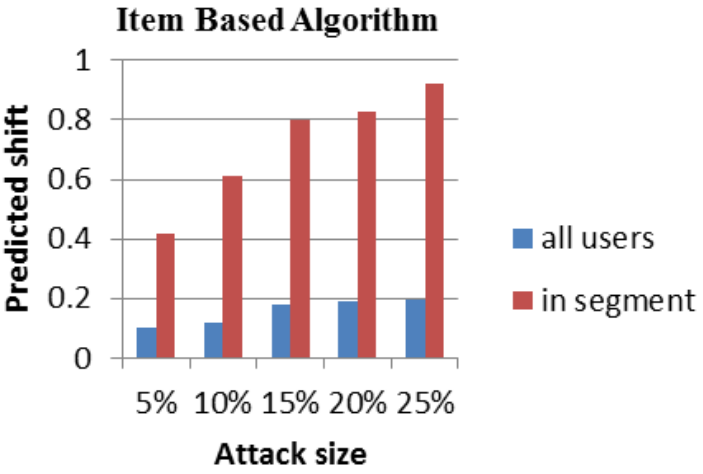

Figure 5.4 Segment attack in item based CF.

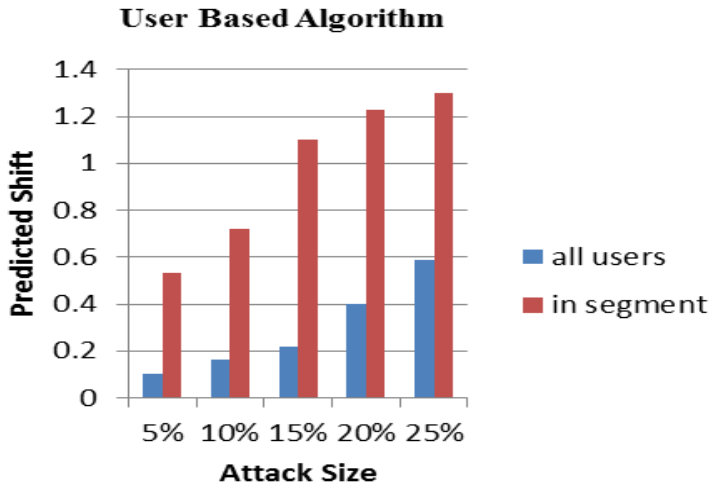

Figure 5.5 Segment attack in user based algorithm.

The segment attack for all users is less powerful than average attack as shown in figure 5.6. Also, figure 5.7 confirms that in-segment attack in user based CF is more effective than that attack in item based CF. It can be concluded that segment attack is the strongest attack, since it is most effective against item based CF. It can be said that when attack is mounted using segment attack model then there is a high probability that pushed item might be recommended to a group of users too which attacker belongs.

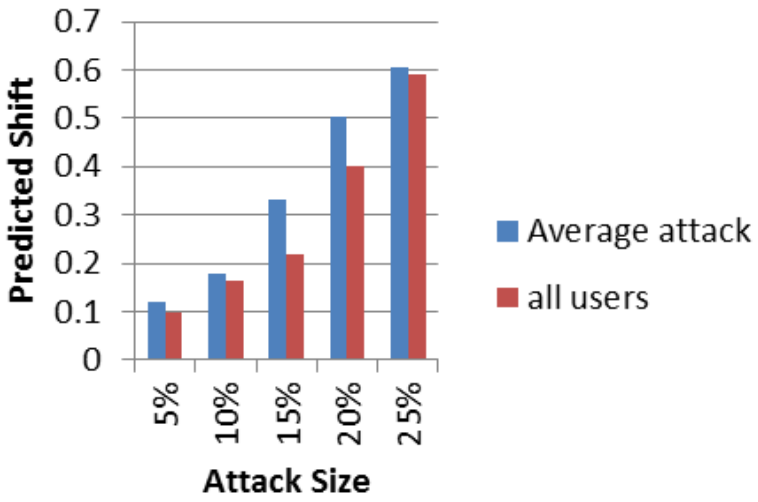

Figure 5.6 Comparison in user based algorithm.

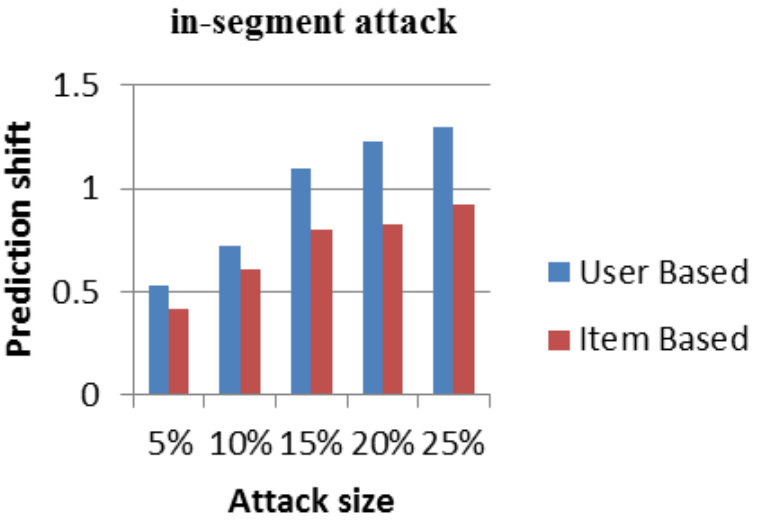

Figure 5.7 Comparison of in-segment.

5.4 Attack Profile detection

5.4.1 Classification of attack profiles

A training set has been used to train a classifier in order to differentiate malicious profiles from the authentic profiles. Then two types of detection attributes have been created i.e. generic and model specific attributes. Generic attributes are generated by considering a profile as a whole whereas model specific attributes are created to find the features of a particular attack model. The performance of five classification algorithms: Random forest (RF), Naive bayes (NB), J48, ZeroR, Radial basis function

network (RBF n/w) has been compared. k-fold cross validation technique is used to evaluate these predictive models.

5.4.2 Attack Detection Attributes

In the experiments, MovieLens dataset has been used which contains userid (UID), movieid (MID) and ratings (R). Then the fake profiles are inserted into the actual dataset using various attack models. The structure of the dataset is shown in Table 5.1.

Table 5.1 Structure of the movielens dataset

UID	MID	R

Some characteristics are required based on which the user profiles can be classified as fake or genuine. After generating the attributes the dataset will look like as shown in Table 5.2.

Table 5.2 Structure of the dataset after preprocessing

$Attribute_1$	$Attribute_2$	$Attribute_3$	…	$Attribute_n$	Label

These attributes are categorized as generic or model specific attributes [33]. Generic attributes are basic metrics for all profiles. Type specific attributes are those attributes which are used to distinguish fake profiles based on the features of particular attack model. Model specific attributes are more effective than generic attributes.

5.4.2.1 Generic Attributes

The generic attributes depend on the aspect that the malicious profiles will have different statistical signature than that of genuine profiles. The difference is because of two reasons: rating assigned to the target items and rating distribution among the filler items. Various general attributes have been introduced by Chirita for detecting anonymous profiles [33]. Some of them are discussed below:

i) Length Variance (LengthVar):

This attribute depends on the count of the ratings in a given profile. It calculates the variation in the given profile's length from the mean length in the system. It can be calculated as:

$$\boldsymbol{LengthVar_a} = \frac{|\#\boldsymbol{r_a} - \overline{\#\boldsymbol{r}}|}{\sum_{\boldsymbol{a}\in\boldsymbol{U}}(\#\boldsymbol{r_a} - \overline{\#\boldsymbol{r}})^2} \quad \ldots\ldots\ldots\ldots\ldots\ldots$$

(18)

where, U is a universal set of users in the system, $\#r_a$ *is number of ratings for user* a.

ii) Weighted degree of agreement (WDA):

It captures the total of the variation in ratings of profile and mean rating of the particular item divided by the frequency of item's rating. It can be determined as follows:

$$WDA_a = \sum_{x=0}^{M_a} \frac{|r_{a,x} - \bar{r}_x|}{R_x} \quad \ldots\ldots\ldots\ldots\ldots\ldots (19)$$

where, M_a specifies the count of items given by user a, $r_{a,x}$ is the rating that user a gave to some item x. R_x is total count of ratings given to item x. $\bar{r}_x$ denotes the mean rating of an item x.

iii) Rating deviation from mean agreement (RDMA):

It identifies the fictitious profiles from a large number of genuine profiles by measuring the profile's mean deviation per item divided by the reciprocal of the total number of ratings of that item. It can be computed by using equation (20):

$$RDMA_a = \frac{\sum_{x=0}^{M_a} \frac{|r_{a,x} - \bar{r}_x|}{R_x}}{M_a} \quad \ldots\ldots\ldots\ldots\ldots\ldots (20)$$

iv) Degree of similarity with top neighbors (DegSim):

Previous studies have hypothesized that there is a high possibility that malicious profiles would have higher similarity with their top 30 closest neighbors than genuine users. This attribute depends on the mean similarity of the profile's k nearest

neighbors. The similarity between the users' profiles is calculated using equation (21). DegSim can be computed as follows:

$$\boldsymbol{DegSim_b = \frac{\sum_{b \in neighbors(a)} S_{a,b}}{l}}$$

…….(21)

v) Weighted deviation from mean agreement (WDMA):

It is one of the variations of RDMA attribute. This attribute highly focuses on the deviation of ratings for sparse items. There is a difference in the denominator that it uses the square of the count of ratings given to an item. It is calculated using equation (22).

$$\boldsymbol{WDMA_a = \frac{\sum_{x=0}^{M_a} \frac{|r_{a,x} - \bar{r}_x|}{R_x^2}}{M_a}} \quad \text{…….. (22)}$$

5.4.2.2 Model Specific Attributes

Previous studies have shown that only generic attributes are not sufficient in differentiating the fake profiles and original users especially when the number of filler items is small. So, generic attributes are augmented with the attributes of particular attack type [33]. This attributes intent to identify the unique signature of a specific attack model. They partition each user's profile with the motive of maximizing the similarity of the profile to the one generated by the attack model. Each profile is partitioned into two sets i.e. $p_{a,t}$ and $p_{a,f}$. The set $p_{a,t}$ consists of all the items with maximum ratings in case of push attacks and with minimum ratings in case of nuke attacks. The set $p_{a,f}$ contains all the other ratings. These attributes are:

i) Mean Variance (MeanVar):

It is used to detect average attacks in the system. It calculates the mean variance between the overall mean and the filler items. It is computed as follows: $\boldsymbol{p_{a,t}}$ denotes the set of ratings of target i.e. $\boldsymbol{r_{a,i}}$=r_{max} and $\boldsymbol{p_{a,f}}$=$\boldsymbol{p_a - p_{a,t}}$.

$$MeanVar_a = \frac{\sum_{i \in p_{a,f}} (r_{a,i} - \bar{r}_a)^2}{|p_{a,f}|} \quad \text{…….. (23)}$$

where, $\boldsymbol{p_a}$ denotes a profile of user a, $\boldsymbol{p_{a,t}}$ denotes the set of ratings of target i.e. $\boldsymbol{r_{a,i}}$=r_{max} and $\boldsymbol{p_{a,f}}$=$\boldsymbol{p_a - p_{a,t}}$.

ii) *Filler Mean Target Difference (FMTD):*

This metric detects the bandwagon attack profiles. The difference in the items' ratings in $\boldsymbol{p_{a,t}}$ set and the item's ratings in $\boldsymbol{p_{a,f}}$ maximizes the effectiveness of the attack. This attribute is computed as:

$$FMTD_a = \left|\left(\frac{\sum_{i \in p_{a,t}} r_{a,i}}{|p_{a,t}|}\right) - \left(\frac{\sum_{j \in p_{a,f}} r_{a,j}}{|p_{a,f}|}\right)\right| \quad \ldots\ldots\ldots\ldots\ldots (24)$$

5.5 Evaluation Metrics

Recall, precision and f-measure are the metrics that have been used to measure the performance of classifiers [32]. Recall means the fraction of appropriate cases that are retrieved whereas precision is the fraction of retrieved cases that are appropriate. F-measure is the combination of precision and recall. Precision, recall and F-measure can be calculated using equation (8), (9) and (10) respectively described in chapter 1.

5.6 Ensemble approach for profile classification

There are three popular ensemble methods: Boosting, Bagging and Blending that can be used to improve the performance of the existing classifiers. Boosting is a method used to ensemble the models that begin with a base classifier which is trained on training dataset [34]. Then another classifier is generated behind it that focuses on those instances in the training dataset in which first classifier goes wrong. Bagging is another ensemble method. It is also known as bootstrap aggregating. It creates various samples of the training dataset, generates classifier for each sample and merges the results of these multiple classifier using majority voting or averaged method. Blending is the ensemble approach in which multiple different algorithms are created on training dataset. Then a meta classifier is generated which memorize how to take the predictions of these classifiers and make correct predictions on test data. This method is also known as stacking.

5.7 Experimental Setup for Detecting Attack Profiles

In order to detect the attack profiles, the training set is created by selecting the set of profiles from system's database that doesn't contains malicious profiles and is labeled as *genuine*. Then a mixture of attacker's data at several attack sizes ranged from 5% to 25% and filler sizes ranged from 20% to 50% are pushed into this training set and are labeled as *fake*. Detection attributes for each profile in the training set are generated as described in section 5.4.2. Classifiers: RF, NB, J48, ZeroR, RBF n/w are trained using training dataset in WEKA. Then k fold cross validation technique is performed to estimate the predictive models. Their performances are analyzed on the basis of recall, precision (PR), and F-measure (F-m). These classifiers perform differently for different datasets and for considered datasets it was found that NB and RF are the best performers. Then these models are combined using majority vote method and a new integrated model is built.

5.7.1 Procedure for detecting the shilling profiles

1. Click on Weka Explorer -> preprocess -> open file. Select the training dataset. First the dataset was selected with 5% attack size and 50% filler size.

2. Now, for the classification of fake profiles click on classify tab -> choose -> classifiers -> rules -> ZeroR. Cross validation was selected equal to 10 folds and the label was selected on the basis of which classification is to be done. Similarly, other classifiers can be chosen from the classifiers menu.

3. Click on Start. The result for this classifier is shown in figure 5.8.

Similarly, the above steps are repeated for other classifiers at the combination of various attack sizes and filler sizes.

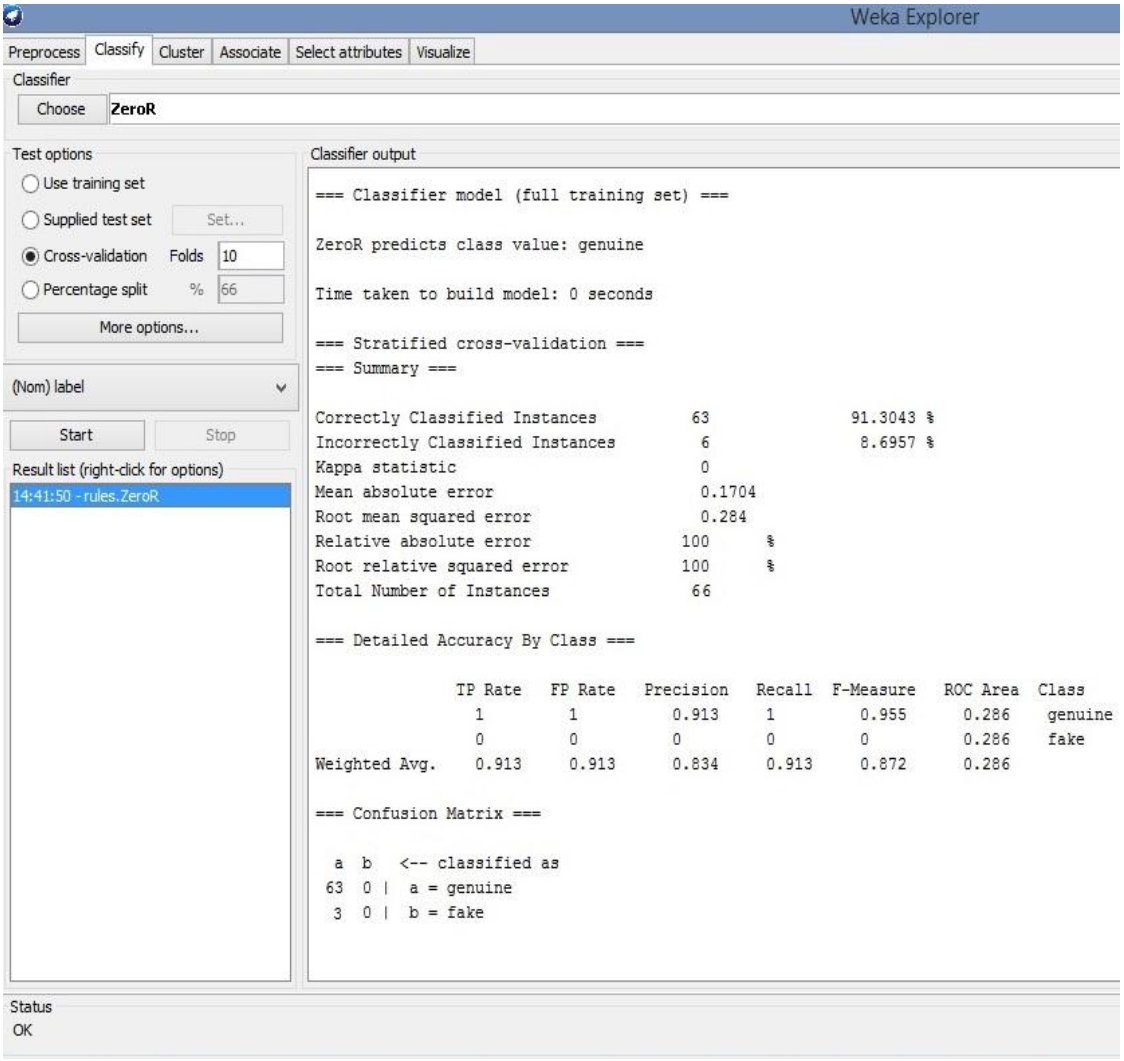

Figure 5.8 Result of performance of ZeroR classifier

5.7.2 Procedure to Ensemble the Classifiers

1. Click on classify tab -> choose -> classifiers -> meta -> vote. Right click on vote classifier name -> show properties and then a dialog box will open. Click on classifiers -> choose -> classifiers -> Bayes -> Naive Bayes -> click add. To add another classifier, click on choose -> classifiers -> trees -> Random forest -> add -> click on X. Now, set combination rule by selecting majority voting from the drop down menu as shown in figure 5.9.

2. Now, click on start. The result of integrated model is shown in figure 5.10.

Performance results of the models on attack sizes: 5%, 10%, 15%, 20% and filler size of 50% for bandwagon and average attack are shown in Table 5.3 and 5.4 respectively. Now, performance is analyzed on 25% attack size and 20%, 30%, 40%, 50% filler sizes for bandwagon and average attacks. The results are presented in Table 5.5 and 5.6 respectively.

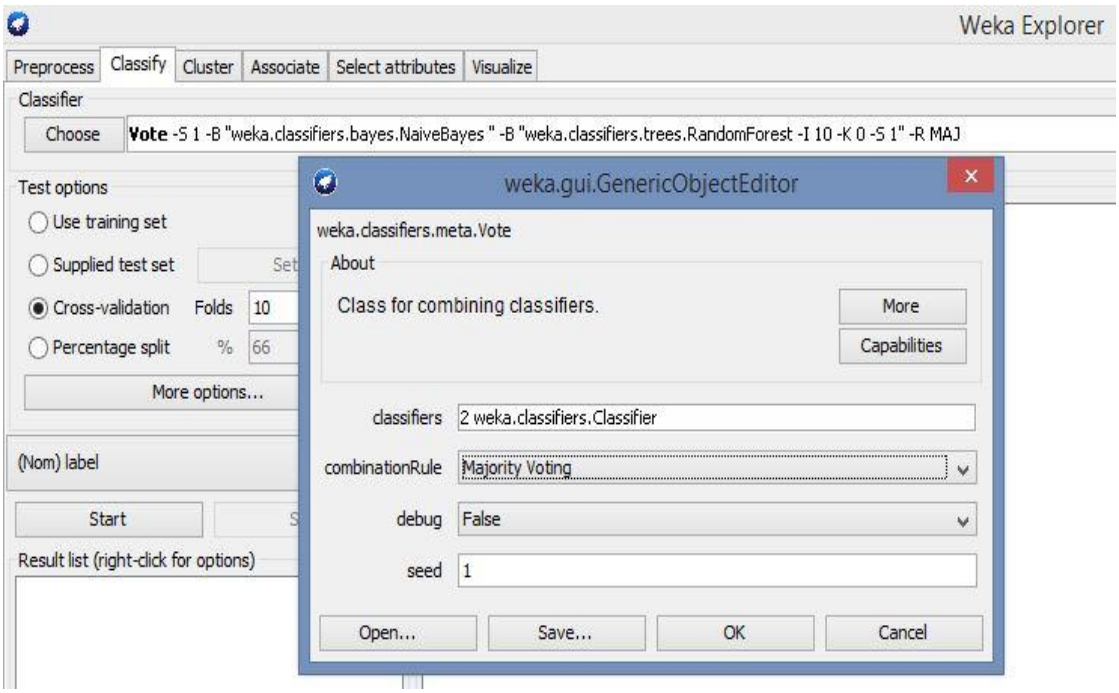

Figure 5.9 Setting the properties of the classifier

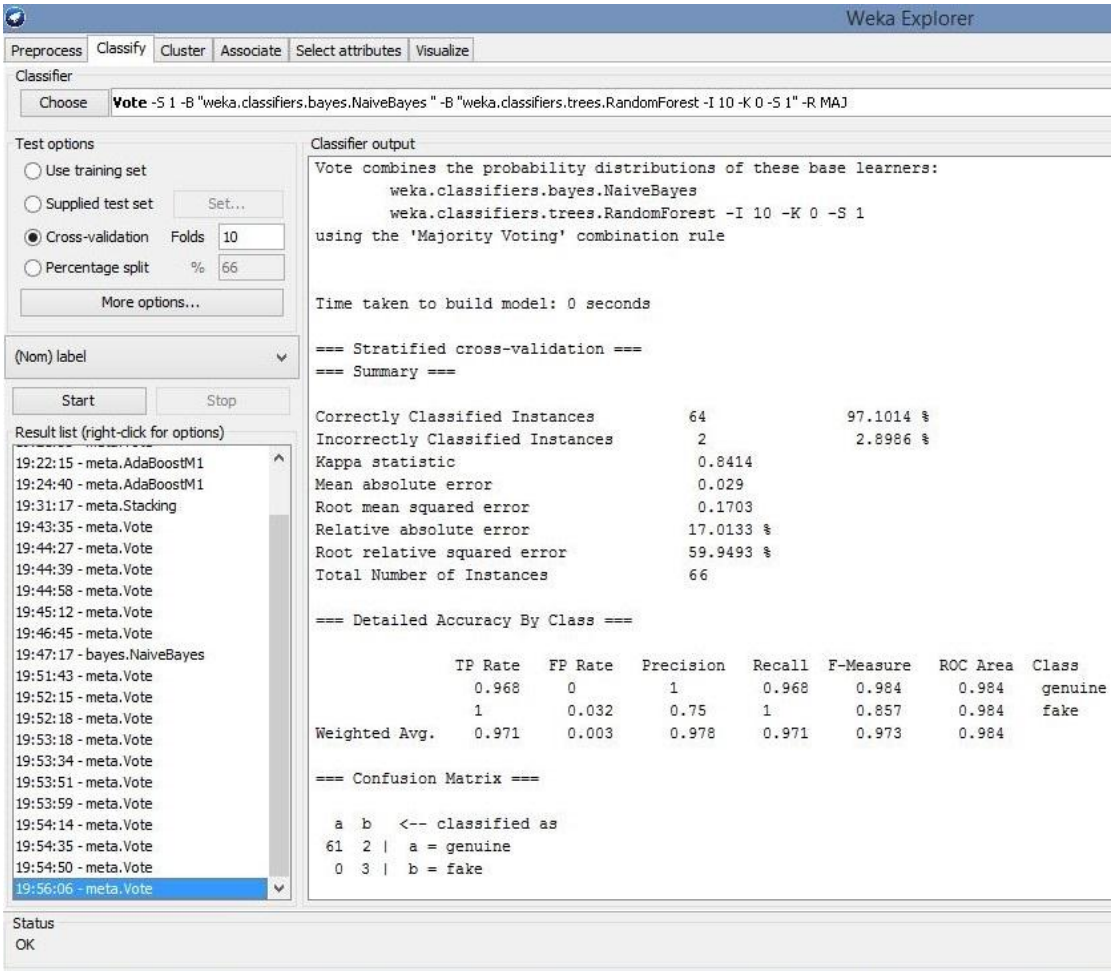

Figure 5.10 Result of integrated (NB+RF) model

Table 5.3 Performance Analysis of Models for Bandwagon Attack at 50% Filler Size.

Attack Size	5%			10%			15%			20%		
Models	**PR**	**Recall**	**F-m**	**PR**	**Recall**	**F-m**	**PR**	**Recall**	**F-m**	**PR**	**Recall**	**F-m**
RBF n/w	0.972	0.972	0.963	0.952	0.967	0.962	0.948	0.949	0.948	0.891	0.893	0.896
J48	0.949	0.946	0.945	0.949	0.948	0.957	0.956	0.955	0.956	0.952	0.953	0.956
ZeroR	0.821	0.875	0.862	0.726	0.731	0.759	0.801	0.813	0.816	0.823	0.821	0.827
NB	0.976	0.979	0.98	0.981	0.98	0.976	0.98	0.979	0.976	0.970	0.971	0.971
RF	0.952	0.961	0.954	0.972	0.973	0.968	0.979	0.978	0.978	0.988	0.985	0.986
Integrated (NB+RF)	0.972	0.963	0.965	0.981	0.98	0.975	0.981	0.984	0.982	0.987	0.986	0.986

Table 5.4 Performance Analysis of Models for Average Attack at 50% Filler Size.

Attack Size	5%			10%			15%			20%		
Models	**PR**	**Recall**	**F-m**	**PR**	**Recall**	**F-m**	**PR**	**Recall**	**F-m**	**PR**	**Recall**	**F-m**
RBF n/w	0.986	0.986	0.985	0.978	0.986	0.984	0.96	0.954	0.923	0.892	0.813	0.924
J48	0.952	0.942	0.946	0.941	0.943	0.95	0.942	0.946	0.957	0.947	0.951	0.92
ZeroR	0.834	0.913	0.872	0.83	0.826	0.862	0.766	0.875	0.817	0.792	0.791	0.793
NB	0.988	0.986	0.986	0.975	0.92	0.941	0.971	0.973	0.968	0.974	0.968	0.971
RF	0.96	0.957	0.958	0.986	0.984	0.974	0.987	0.986	0.986	0.981	0.982	0.983
Integrated (NB+RF)	0.978	0.971	0.973	0.98	0.986	0.985	0.988	0.981	0.973	0.971	0.975	0.976

Table 5.5 Performance Analysis of Models for Bandwagon Attack At 25% Attack Size.

Filler Size	20%			30%			40%			50%		
Models	**PR**	**Recall**	**F-m**	**PR**	**Recall**	**F-m**	**PR**	**Recall**	**F-m**	**PR**	**Recall**	**F-m**
RBF n/w	0.89	0.892	0.893	0.902	0.906	0.904	0.921	0.924	0.923	0.935	0.933	0.934
J48	0.965	0.962	0.962	0.959	0.953	0.956	0.949	0.946	0.947	0.937	0.931	0.935
ZeroR	0.846	0.85	0.852	0.861	0.864	0.861	0.842	0.845	0.847	0.832	0.835	0.836
NB	0.978	0.976	0.975	0.983	0.984	0.984	0.976	0.975	0.978	0.98	0.982	0.981
RF	0.968	0.967	0.968	0.965	0.964	0.964	0.97	0.978	0.978	0.981	0.983	0.983
Integrated (NB+RF)	0.978	0.976	0.97	0.970	0.973	0.975	0.981	0.981	0.979	0.973	0.976	0.976

Table 5.6 Performance Analysis of Models for Average Attack at 25% Attack Size.

Filler Size	20%			30%			40%			50%		
Models	**PR**	**Recall**	**F-m**	**PR**	**Recall**	**F-m**	**PR**	**Recall**	**F-m**	**PR**	**Recall**	**F-m**
RBF n/w	0.885	0.886	0.883	0.87	0.872	0.873	0.865	0.867	0.865	0.875	0.873	0.874
J48	0.95	0.952	0.954	0.961	0.965	0.95	0.956	0.955	0.953	0.946	0.948	0.949
ZeroR	0.81	0.813	0.816	0.847	0.846	0.842	0.802	0.803	0.803	0.791	0.79	0.786
NB	0.962	0.962	0.965	0.97	0.972	0.971	0.98	0.979	0.978	0.972	0.975	0.973
RF	0.972	0.971	0.974	0.964	0.964	0.965	0.974	0.976	0.976	0.98	0.982	0.983
Integrated (NB+RF)	0.974	0.973	0.973	0.973	0.976	0.975	0.978	0.979	0.978	0.981	0.983	0.984

The advantage of the integrated model is that it will give good performance in almost all the cases. Figure 5.11 shows result of k-fold cross validation for average attack at different values of k.

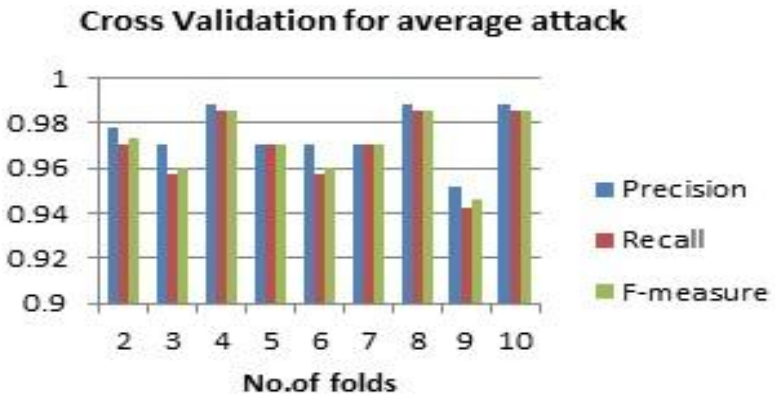

Figure 5.11 k fold cross validation at 25% attack size and 50% filler size

5.7.3 Procedure for improving the accuracy of classifiers

1. Click on Experimenter -> Setup -> New. Select cross-validation as Experiment type and 10 as number of folds and mark classification radio button.
2. In dataset part, click add new button and open the training dataset.
3. In algorithms portion, click add new button -> choose -> classifiers -> trees -> random forest -> OK.
4. Now, ensemble the RF model using boosting method. Click on add new -> choose -> classifiers -> meta -> AdaBoostM1. A dialog box will open, then choose the classifier by clicking classifier -> choose -> trees -> random forest -> OK.
5. Similarly ensemble the model using bagging. Click on add new -> choose -> classifiers -> meta -> bagging. Now, select the classifier -> trees -> random forest -> OK.

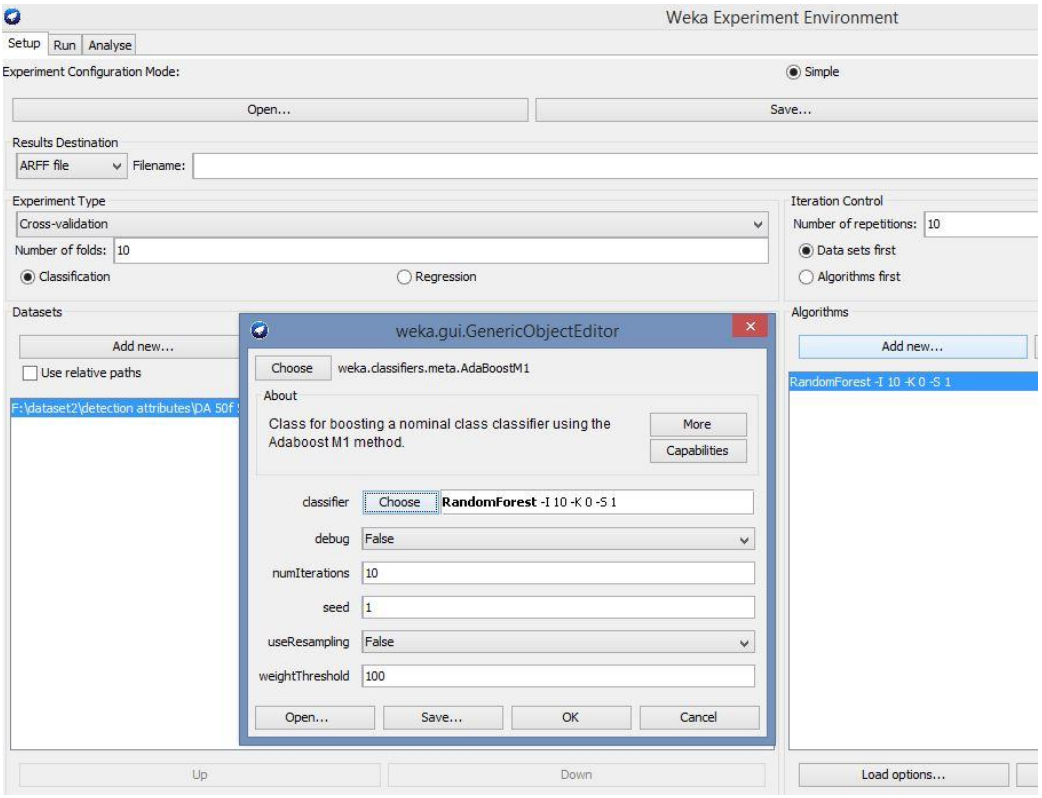

Figure 5.12 Ensemble the Random Forest model using Boosting method

6. Now, model is integrated using stacking method. Click Add new ->choose -> classifiers -> meta -> stacking. Select random forest as metaclassifier as shown in figure 5.13.

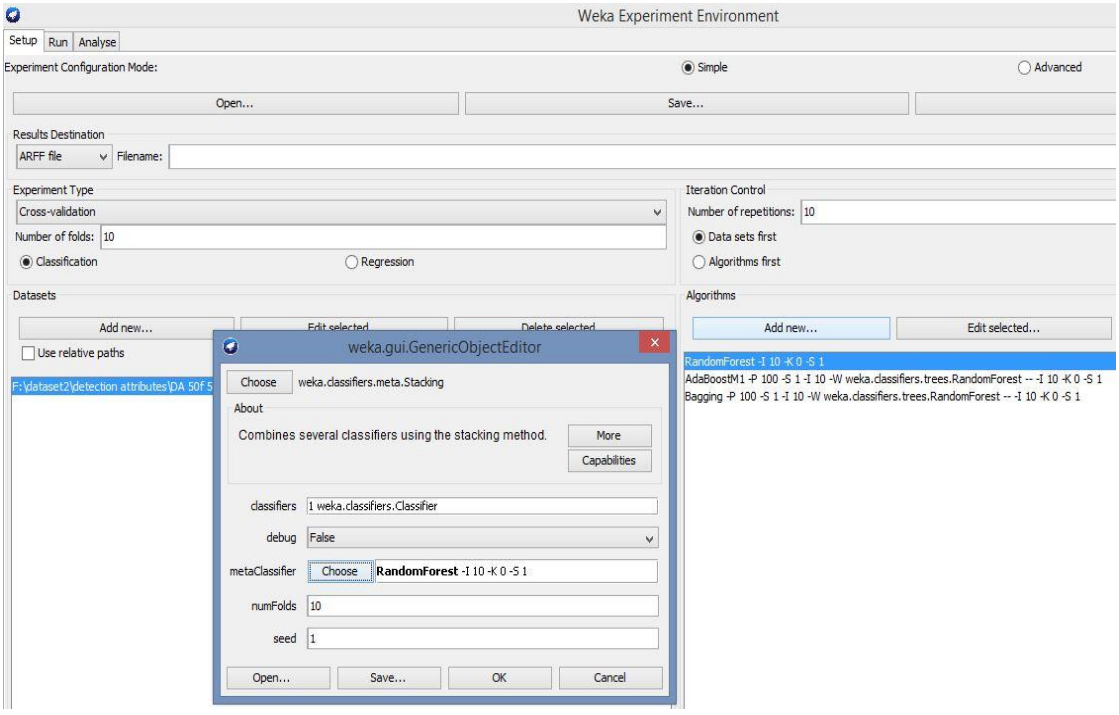

Figure 5.13 Ensemble the Random Forest model using Stacking

7. Click on run tab -> Start.

8. To analyze the results, click on analyze tab. The experiment results analysis panel will get open. Click on experiment available on the top right side of the panel. In configure test, set percent_correct as comparison field and click the perform test button. The accuracy is shown in figure 5.14.

It was observed that boosted RF has 99.86% accuracy whereas simple RF has 99.71% accuracy. It can be observed that there is a * in front of the accuracy of boosted RF which means that this algorithm is meaningful. Accuracy without * is meaningless. Hence boosted RF is more accurate than simple RF. The above steps can be performed for other classifiers also to improve their accuracy.

```
Tester:     weka.experiment.PairedCorrectedTTester
Analysing:  Percent_correct
Datasets:   1
Resultsets: 4
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       14/5/16 5:49 PM


Dataset                   (1) trees.RandomFo | (2) meta.AdaBoo (3) meta.Bagging (4) meta.Stackin
-------------------------------------------------------------------------------------------------
'all attributes'         (100)   99.71(2.01) |   99.86(1.43) *   100.00(0.00)     100.00(0.00)
-------------------------------------------------------------------------------------------------
                                      (v/ /*) |         (0/0/1)          (0/1/0)          (0/1/0)


Key:
(1) trees.RandomForest '-I 10 -K 0 -S 1' -2260823972777004705
(2) meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W trees.RandomForest -- -I 10 -K 0 -S 1' -737810780893311
(3) meta.Bagging '-P 100 -S 1 -I 10 -W trees.RandomForest -- -I 10 -K 0 -S 1' -5178288489778728847
(4) meta.Stacking '-X 10 -M \"trees.RandomForest -I 10 -K 0 -S 1\" -S 1 -B \"trees.RandomForest -
```

Figure 5.14 Improved accuracy of ensemble Random Forest model

CHAPTER 6
CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this thesis work, focus is on various components in the field of attacks against recommender systems based on collaborative filtering. It has been investigated that the attack profiles can be generated and injected into the database even with the limited knowledge of the system. It has also been examined that the average attack has highest impact on the recommender system. But, it is less effective against item based algorithm and also requires more knowledge about the system. The segment attack has also been studied and the results show that segment attack effects the item based algorithm to a degree that other attacks are not. But, it has more impact on user based than item based collaborative filtering algorithm. Therefore, it can be concluded that item based collaborative filtering recommender systems have higher security than user based collaborative filtering systems.

Detection of shillers is a key component for robust recommender system. Therefore, training sets are created by generating generic and model specific attributes, various classification models are demonstrated to distinguish the attack profiles. Their performance is analyzed using precision, recall and F-measure and it is identified that Naive Bayes and Random Forest models are best performers. When integrated using majority voting approach, k-fold cross validation proved that the new model outperforms in most of the cases. It was also determined that integrated random forest model is more accurate than simple one.

6.2 Future Work

As a future work, hybrid models can be built to inject anonymous profiles into the system and more metrics can be considered to measure the stability of the system. More detection attributes can be used to detect the malicious profiles. More than two models can be integrated for more accurate results and their accuracy can be improved using ensemble methods. To detect the attack profiles, clustering approach can be implemented and the malicious profiles can be removed from the system.

REFERENCES

[1] F. Ricci, "Travel recommender systems", *IEEE Intelligent Systems17*, no.6, pp. 55-57, 2002.

[2] S. Ojo, S. Ngwira and K. Zuva, "A Survey of Recommender Systems Techniques, Challenges", *International Journal of Emerging Technology and Advanced Engineering*, vol. 2, no. 11, pp. 81-173, 2012.

[3] M. Ekstrand, J. Riedl and J. Konstan, "Collaborative Filtering Recommender Systems", *Foundations and Trends in Human-Computer Interaction*, vol. 4, no.2 pp. 44-54, 2011.

[4] http://www.cs.carleton.edu/cs_comps/0607/recommend/recommender/memorybased.html

[5] http://www.cs.carleton.edu/cs_comps/0607/recommend/recommender/modelbased.html

[6] S.K. Verma, N. Mittal and B Agarwal , "Hybrid Recommender System based on Fuzzy Clustering and Collaborative Filtering", *In Proceedings of the 4th International Conference on Computer and Communication Technology*, 2013.

[7] S. Lam and J. Riedl, "Shilling Recommender Systems for Fun and Profit", *In Proceedings of the 13th international conference on World Wide Web,ACM*, pp. 393-402, 2004.

[8] J. Herlocker, J. Konstan, A. Borchers and J. Riedl, "An algorithmic framework for performing collaborative filtering", *In Proceedings of the 22nd ACM Conference on Research and Development in Information Retrieval (SIGIR'99)*, 1999.

[9] B. Sarwar, G. Karypis, J. Konstan and J. Riedl, "Item-Based Collaborative Filtering Recommendation", *ACM, Hong Kong*, pp. 285-295, 2001.

[10] G. Linden, B. Smith and J. York, "Amazon.com recommendations: item-to-item collaborative filtering", *IEEE Internet Computing*, vol. 7, no. 1, pp. 76-80, 2003.

[11] B. Mobasher, R. Burke, R. Bhaumik and C. Williams, "Effective Attack Models for Shilling Item-Based Collaborative Filtering Systems", *In Proceedings of the 2005 WebKDD Workshop, Chicago, Illinois*, 2005.

[12] M. Mahony, N. Hurley and C. Silvestre, "Recommender Systems: Attack Types and Strategies", *American Association for Artificial Intelligence*, pp. 334-339, 2005.

[13] B. Mobasher, R. Burke, R. Bhaumik and J. Sandvig, "Attacks and Remedies in Collaborative Recommendation", *IEEE Intelligent. System.*, vol. 22, no. 3, pp. 56-63, 2007.

[14] M. O'Mahony, N. Hurley, N. Kushmerick and G. Silvestre, "Collaborative recommendation: A robustness analysis", *ACM Transactions on Internet Technology*, pp. 344-377, 2004.

[15] R. Burke, B. Mobasher, R. Bhaumik and C. Williams, "Segment-based injection attacks against collaborative filtering recommender systems", *In Proceedings of the Fifth IEEE International Conference on Data Mining*, 2005.

[16] A. Kumar, D. Garg and P. Rana, "Ensemble Approach to Detect Profile Injection Attack in Recommender System", *In Proceedings of the Fourth IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI-2015)*, pp. 1734-1740, 2015.

[17] R. Burke, B. Mobasher, C. Williams and R. Bhaumik, "Classification Features for Attack Detection in Collaborative Recommender Systems", *In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 542-547, 2006.

[18] Mark Orr, "Introduction to Radial Basis Function Networks".

[19] P. Chirita, W. Nejdl and C. Zamfir, "Preventing shilling attacks in online recommender systems", *In WIDM '05: Proceedings of the 7th annual ACM International workshop on Web information and data management*, pp. 67-74, 2005.

[20] C. Williams, B. Mobasher and R. Burke, "Defending recommender systems: detection of profile injection attacks", *SOCA*, vol. 1, no. 3, pp. 157-170, 2007.

[21] F. Zhang, Q. Zhou, "Ensemble detection model for profile injection attacks in collaborative recommender systems based on BP neural network", *IET Information Security*, vol. 9, no. 1, pp. 24-31, 2015.

[22] J. Lee and D. Zhu, "Shilling Attack Detection—A New Approach for a Trustworthy Recommender System", *INFORMS Journal on Computing*, vol. 24, no. 1, pp. 117-131, 2012.

[23] M. Gao, B. Ling, Q. Yuan, Q. Xiong and L. Yang, "A Robust Collaborative Filtering Approach Based on User Relationships for Recommendation Systems", *Mathematical Problems in Engineering*, vol. 2014, pp. 1-8, 2014.

[24] W. Zhou, J. Wen, Y. Koh, Q. Xiong, M. Gao, G. Dobbie and S. Alam, "Shilling Attacks Detection in Recommender Systems Based on Target Item Analysis", *PLOS ONE*, vol. 10, no. 7, pp. e0130968, 2015.

[25] J. Dhimmar and R. Chauhan, "An accuracy Improvement of Detection of Profile-Injection Attacks in Recommender Systems using Outlier Analysis", *International Journal of Computer Applications*, vol. 122, no. 10, pp. 22-27, 2015.

[26] Z. Wu, J. Wu, J. Cao and D. Tao, "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*, pp. 985-993, 2011.

[27] F. Zhang, "Average Shilling Attack against Trust-Based Recommender Systems", *2009 International Conference on Information Management, Innovation Management and Industrial Engineering*, pp. 588-591, 2009.

[28] F. Zhang, "Analysis of Bandwagon and Average Hybrid Attack Model against Trust-based Recommender Systems", *Fifth International Conference on management of e-Commerce and e-Government*, pp. 269-273, 2011.

[29] J. O'Donovan and B. Smyth, "Trust in Recommender Systems", *IUI, Association for Computing Machinery, New York, NY, USA*, pp.168-174, 2005.

[30] P. Massa and P. Avesani, "Trust-aware recommender systems", *In Proceedings of the 1st ACM Conference on Recommender Systems (RecSys '07)*, pp. 17-24, 2007.

[31] G. Karypis, "Evaluation of item-based top-n recommendation algorithms ", *In Proceedings of the tenth international conference on Information and knowledge management*, pp. 247-254. ACM,2001.

[32] C. Williams, R. Bhaumik, R.Burke and B. Mobasher, "The impact of attack profile classification on the robustness of collaborative recommendation", *WEBKDD'06, USA*, ACM, pp. 343-352, 2006.

[33] C. Williams and B. Mobasher, "Profile Injection Attack Detection for Securing Collaborative Recommender Systems", *DePaul University CTI Technical report*, pp. 1-47, 2006.

[34] F.Zhang, "A meta-learning-based approach for detecting profile injection attacks in collaborative recommender systems", Journal of Computers, vol. 7, no. 1, pp. 226- 234, 2012.

LIST OF PUBLICATIONS

1. Parneet Kaur and Shivani Goel, "Shilling Attack Detection in Recommender Systems using Classification Techniques",in *International Journal of Engineering Applied Sciences and Technology*, vol. 1, no. 7, pp. 147-152, 2016. [Published]

2. Parneet Kaur and Shivani Goel, "Shilling Attack Models in Recommender System",in *International Conference on Inventive Computation Technologies (ICICT 2016), IEEE,* August 26-27, Coimbatore. [Accepted] to be held on August 26-27,2016 at Hotel Arcadia, Coimbatore, Tamilnadu, India.

VIDEO LINK

https://www.youtube.com/channel/UCa0_OxOK2Eqz4MC-BWGACSQ

PLAGIARISM REPORT

ORIGINALITY REPORT

8%	5%	7%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	maya.cs.depaul.edu Internet Source	1%
2	facweb.cti.depaul.edu Internet Source	<1%
3	ceur-ws.org Internet Source	<1%
4	Kumar, Ashish, Deepak Garg, and Prashant Singh Rana. "Ensemble approach to detect	<1%