HYBRID BEE COLONY TRUST MECHANISM IN RECOMMENDER SYSTEM

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

Master of Engineering
in
Computer Science and Engineering

Submitted By
Abhishek Kaleroun
(Roll No. 801232001)

Under the supervision of:
Dr. Shalini Batra
Asstt. Professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

June 2014

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, **"Hybrid Bee Colony and Trust mechanism in Recommender system"**, 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. Shalini Batra* 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.

Abhishek

Signature:

(Abhishek Kaleroun)

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

(Shalini Batra)

Assistant Professor,

CSED

Countersigned by

(Dr. Deepak Garg)
Head
Computer Science and Engineering Department
Thapar University
Patiala

14/7

(Dr. S. K. Mohapatra)
Dean (Academic Affairs)
Thapar University
Patiala

ACKNOWLEDGEMENT

No volume of words is enough to express my gratitude towards my guide **Dr. Shalini Batra**, Computer Science & Engineering Department, Thapar University, Patiala, who has been very concerned and has aided for all the materials essentials for the preparation of this thesis report. She has helped me to explore this vast topic in an organized manner and provided me all the ideas on how to work towards a research-oriented venture.

I am also thankful to **Dr. S. K. Mohapatra**, Dean of Academic Affairs, **Dr. Deepak Garg**, Head of Computer Science & Engineering Department and **Mr. Ashutosh Mishra**, P.G. Coordinator, for the motivation and inspiration that triggered me for the thesis work.

I would also like to thank the staff members and my colleagues who were always there at the need of hour and provided with all the help and facilities, which I required, for the completion of my thesis work.

Most importantly, I would like to thank my parents and the almighty for showing me the right direction out of the blue, to help me stay calm in the oddest of the times and keep moving even at times when there was no hope.

ABSTRACT

In the era of internet, there is a lot of information on the web. The information whether implicit or explicit is growing at an exponential rate, therefore perplexity in choosing the products and services has also soared up. Thus recommender system, an automated filtering mechanism, has been established to filter the information according to individual's behavior and preferences and provide best and accurate suggestions. Recommendation can be in the form of verbal reviews, reviews about a movie or a book in internet and newspaper, surveys, travel guides etc. Collaborative filtering is one of the most popular and mature techniques in recommender system and evaluates items on the basis of opinions of other people. But in spite of its adeptness, it still suffers from problems such as cold start, sparsity, scalability and is susceptible to attacks like grey sheep, shilling attacks etc. Thus a solution has been propounded, which involves the collaboration of artificial bee colony, a swarm intelligence method, and trusted graph mechanism with that of collaborative filtering. Swarm intelligence is an artificial intelligence technique to study the behavior of insects in distributed systems while trust is a measure of reliability of user based on its preferences and behavior with distinct context at a particular time period. Former possess the adaptation, self organization and distribution properties while latter removes the fake recommendations and develops faith in the system. Thus the hybrid of these techniques increases the accuracy and robustness of recommendations while eliminating the attacks prevailing in existing systems. The proposed framework has been compared with other existing recommender system approaches with different parameters and validated by using dataset of movies available online.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1
Introduction

1.1 Recommendation System

Any automated system that suggests/recommend or predict actively an item to purchase, invest or subscribe is known as recommender system. These systems filter the information and recommend the items to the user based upon their likes and dislikes [7].

In earlier times there were very few choices available, so people had very less or no options available for choosing an item. But in the 21st century, the choices have no limit [9]. If a person wants to buy some music, iTunes has over 11 million tracks to choose from. If someone wants to buy a book, Amazon has over 2 million books to choose from. So, now the major problem is to find the relevant stuff among the millions of products which are growing everyday at an exponential rate.

Hence a filtering mechanism is required which provides relevant information based on our choices, past preferences or demographic information. These suggestions must be quick and should cover all the choices user has. Therefore, recommender systems, which are intelligent software agents, have been introduced to provide solution for this problem [6].The most common and widely used examples of recommender systems are amazon.com, Netflix.com and so on. Following are the snapshots of amazon.com and netflix.com which show how recommendations take place.

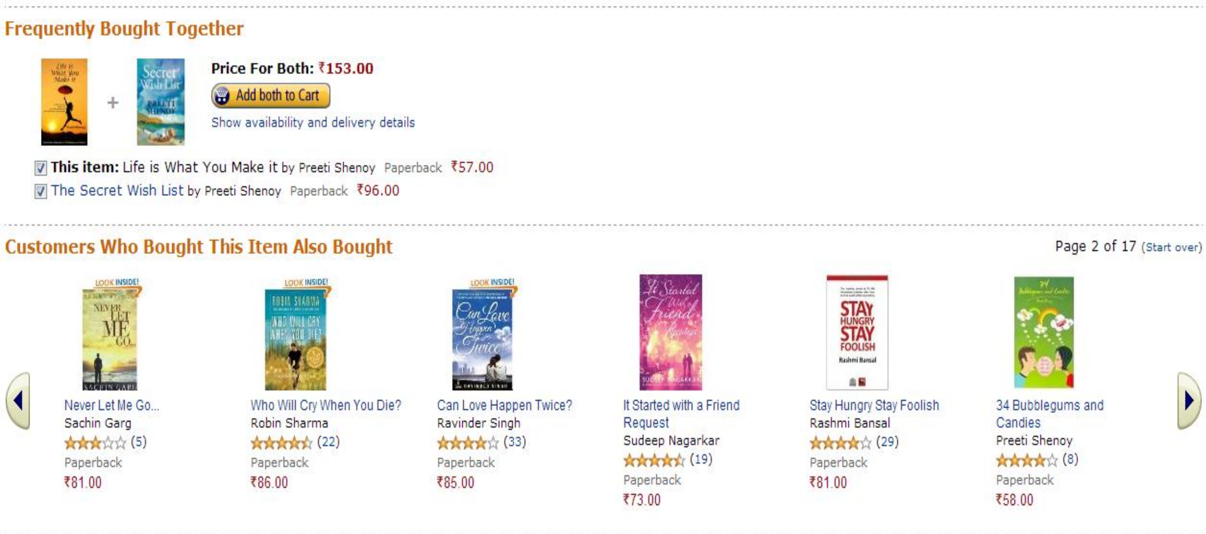

Figure 1.1: Recommendations from amazon.com

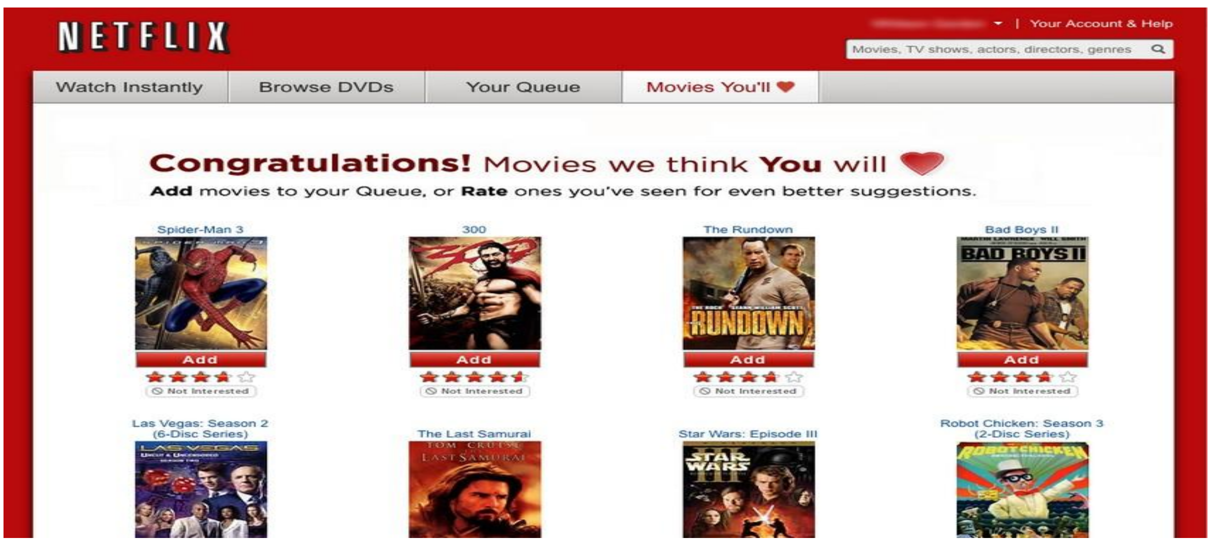

Figure 1.2: Recommendations from Netflix.com

Recommender systems have gained their importance in recent years, and are being used in a variety of applications. The popular applications include movies, books, research articles, music, news, social tags, search queries, and products in general [1,2,5]. Many recommender systems have been instituted for experts, restaurants, financial services, jokes, life insurances and twitter followers [2].

There are certain properties of a recommender system which decides how much the system is useful and correct [7]. Some of them are:

- Prediction Accuracy: The correct or accurate prediction is a very important property of a recommender system. Generating precise and useful suggestion for the user is the main concern of the recommender systems.
- Coverage: The degree or extent up to which the items or products are analyzed or covered for the recommendations is referred to as coverage. Recommender systems must have a high coverage to cover huge items for the user.
- Confidence: It refers to the recommender system's trust in the recommendation or suggestion generated. These automated systems must give recommendations with high confidence and must tell the level of confidence in each suggestion so that user may research the item before making any decision.
- Trust: The reliability or faith the user develops in the recommender system is denoted by trust. If the system provides reasonable recommendations to the user, then the trust of the user on the system will increase otherwise it will diminish.
- Novelty: The recommendations that are generated by the recommender system must be novel i.e. new to the users and should not be the one which the user has already chosen or rated.
- Serendipity: The recommendations which have been generated by the system must not be typically of one category which the user has selected in the past but should also recommend other less familiar types also. Recommender system also needs to balance serendipity with accuracy.
- Diversity: In some cases the recommendations of only the same type is not useful to the user. Recommender system must have the diversity in the suggestions for the users. It should not confine to single type but should cover maximum types.
- Robustness: It is the ability that how much a system is stable against the attacks from fake profiles. It must give accurate results even in the presence of the faux information in the system i.e. a skill to detect attacks and ignore them.
- Adaptivity: The recommender system must be flexible enough to adjust itself dynamically. It must change with the environment and the information given to it, so that it remains updated and can give the best suggestions.

- Scalability: Recommender system works on the large collections of items for large users. It must give quick results for the users and must not slow down with the increase in the users or the items.

Recommendation systems have been categorized into five major types according to their approach [8] for recommendation:

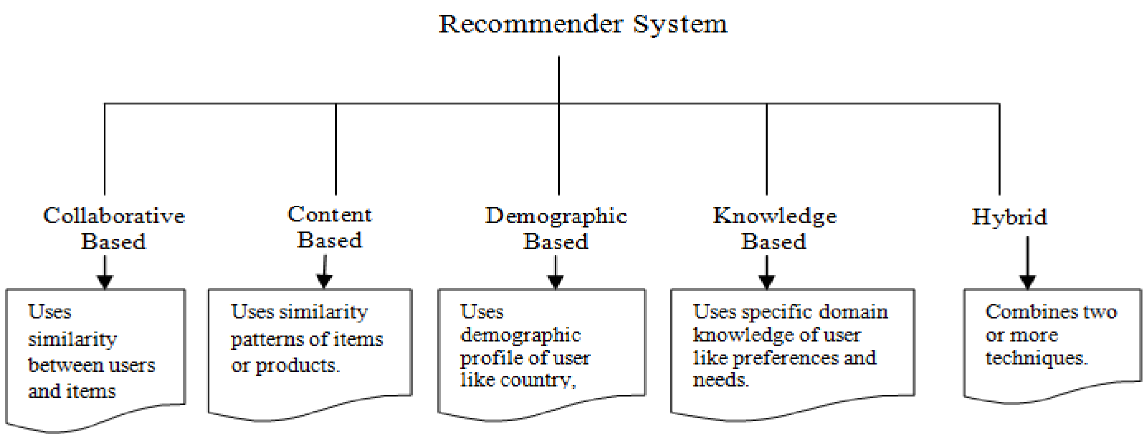

Figure 1.3: Different approaches of recommender system

- Collaborative based: This type of recommender systems uses the explicit information i.e. rating information of the user in the form of a user-item rating matrix for generating the recommendations. This system finds the users similar to the current user on the basis of rating history and using these neighbours the recommendations are generated [12].
- Content based: This approach implies upon the features associated with the items for generating the recommendations and uses the implicit information of the features of the items. The important features or characteristics of the item are collected and the items similar to the current item are selected for the recommendations. Therefore it is able to provide suggestions based on a particular feature depending upon the information provided by the user [3].
- Demographic based: Demography symbolizes section of people having common characteristics such as country, age, sex, marital status, class etc. Therefore the recommender system depends upon such factors for generating the

recommendations for user. Only those items which are relevant to the user at that point are recommended [40].

- Knowledge based: Knowledge based recommender system works on the principle of using user's needs and preferences and item knowledge for generating the recommendations. This approach contains explicit knowledge about the items how they satisfy the user needs [4].
- Hybrid: This branch is the combination of two or more approaches and is very useful for giving the suggestions to the users as it coalesces with other approaches to obtain useful information and generate high quality recommendations for the user [30].

The different recommender system approaches require different user inputs and uses different database for predicting the recommendations for the user [8]. Knowledge based recommender systems requires user's input or query as an input into the system and uses the domain knowledge and product database for giving suggestions. The content based recommender system take user's ratings as input and product database for predicting recommendations to its user. The collaborative based recommender system uses the user's rating as input and the user ratings database for its recommendation generating process. The demographic based recommendation system takes the user's demographics as its input and uses user ratings and user's demographics database for finding the predictions for the users. The recommender system techniques and their sources are described in fig. 1.4 :

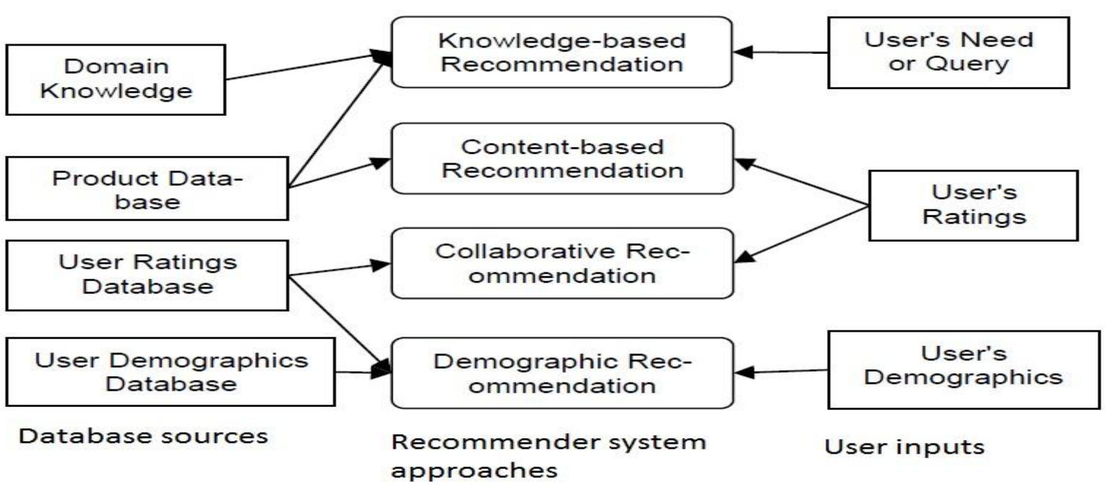

Figure 1.4: Inputs and sources of recommender system approaches

1.2. Collaborative Filtering

Collaborative filtering is the information filtering technique which works on collecting and analyzing the large amount of user's data. It tracks the user's behaviour and preferences and predicts what user may like, based on the similarity with other users [10]. It follows the simple principle that like-minded people have similar taste for giving recommendations. It filters data based on the collaboration of other users and utilizes the user-item matrix and therefore is the most productive technique [11]. It is based on the fact that if two users X and Y have rated n items similarly, or behave similarly in any environment than they will also rate or behave similarly on other items also [13].

Collaborative filtering conceptually follows three basic steps [11]:

- Build user-item rating matrix: First step is to build the user-item rating matrix. User-item rating matrix is the matrix in which rows represent the users and columns represents the items. The values of the matrix represent the ratings given by the particular user to the item. This matrix contains all the users, items and corresponding ratings provided user rates the items. The simple 4×4 user-item matrix is shown below:

$$\begin{array}{cc} & \begin{array}{cccc} \mathrm{I1} & \mathrm{I2} & \mathrm{I3} & \mathrm{I4} \end{array} \\ \begin{array}{c} \mathrm{U1} \\ \mathrm{U2} \\ \mathrm{U3} \\ \mathrm{U4} \end{array} & \left\{ \begin{array}{cccc} 4.0 & 3.5 & 0 & 4.5 \\ 3.0 & 0 & 2.5 & 4.0 \\ 5.0 & 2.0 & 0 & 3.5 \\ 2.0 & 4.0 & 4.5 & 0 \end{array} \right\} \end{array}$$

  Where U1, U2, U3 and U4 are users and I1, I2, I3 and I4 are items. The zero value in the user-item matrix shows that user did not rate that particular item.

- Select the nearest neighbours: Second step is to find the nearest neighbours using the similarity factor. The active user or item is taken and the nearest neighbours are calculated by finding the distance between them. Neighbours with minimum distance are selected. The distance between two users are calculated using one the following methods:
  - Manhattan distance: It is the simplest way of calculating the distance between two user which is calculated using the formula:

$$|x1 - x2| + |y1 - y2| \tag{1}$$

- Euclidean distance: This is the straight line distance which uses the concept of Pythagoras theorem. Euclidean distance is calculated using the formula:

$$\sqrt{(x1 - x2)^2 + (y1 - y2)^2} \tag{2}$$

- Pearson Correlation: Pearson correlation method is used for finding the similarity between the users. More the similarity between users less is the distance between them and vice versa. This correlation coefficient is calculated by using formula:

$$\frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2}\sqrt{\sum(y_i - \bar{y})^2}} \tag{3}$$

  Where x1, y1 are the ratings of user 1 and x2, y2 are the ratings of user 2, x_i, y_i are the ratings of user x, y on item 'i' and $\bar{x}$ is the average rating of user x.

- Generating recommendations: Items are selected from the neighbours selected in the above step and the best items from those are recommended to the user.

The various benefits that collaborative filtering possess are:-

- Subjective information: Collaborative Filtering uses subjective information of the items for the recommendations and therefore performs better to provide enhanced recommendations.
- Wide scope: Collaborative filtering provides suggestions based on the other user's preferences. So their scope is very wide as compared to others who give recommendations on the particular user preferences or type only. Hence there is lot of diversity and good coverage in case of collaborative filtering.
- Broader applications: These filtering approaches can recommend almost any types of items including books, movies and songs and is almost independent of the representation of the items being recommended.

Apart from all the beneficiaries, collaborative filtering provides, there are some drawbacks also which hamper the accuracy of recommendations [13]. Some of the disadvantages of the collaborative filtering are:

- Shilling attack: Shilling attack or profile injection attack is a very common attack in CF recognized nowadays. In this attack, various false or fake profiles give faux ratings which severely affect the recommender system. This attack is of two types i.e. push attack and nuke attack. When people give lots of positive ratings or comments to their own items or products just to increase its popularity and recommendations, this type of attack is known as push attack and when people give lots of negative ratings or comments for their competitor's items just to decrease its popularity and recommendations, this type of attack is known as nuke attack. This attack is hazardous and therefore must be identified and checked for the proper functioning of the system.

- Cold start problem: Cold start is also one of the most commonly encountered challenges of the collaborative filtering. It is also known as new user problem as it creates problem of generating recommendations for the new user. Collaborative filtering works on the principle of finding neighbours but without neighbours the suggestions cannot be given to the user. Since the user is novel, he/she must not have rated any items and thus do not have neighbours. So collaborative filtering fails to provide recommendations to them.
- Sparse matrix: A sparse matrix is a matrix which is primarily populated with zeroes. The '0' value in the user-item indicates that user did not rate the respective item. As the number of items are increasing at a very fast pace, the user cannot rate all the items. The number of items rated by user is very less as compared to the gigantic number of choices available to him. If the matrix is dense, the neighbours can be calculated with confidence and hence, the predictions will be more accurate. The neighbours of the active user cannot be calculated easily and accurately when the matrix is sparse because it becomes difficult to match the similarity precisely due to less number of neighbours and therefore severely affects the predictions. The sample of sparse user-item matrix is illustrated below:

$$\begin{array}{cc} & \begin{array}{cccc} I1 & I2 & I3 & I4 \end{array} \\ \begin{array}{c} U1 \\ U2 \\ U3 \\ U4 \end{array} & \left\{ \begin{array}{cccc} 0 & 3.0 & 0 & 0 \\ 0 & 0 & 4.5 & 0 \\ 2.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array} \right\} \end{array}$$

- Grey sheep problem: Recommendations for the Grey sheep users are most difficult to generate. Grey sheep users are those users whose opinion consistently does not match with any group. Due to this, neighbours for the grey sheep remain isolated in the groups and hence the predictions for them are very difficult. Thus collaborative filtering is not suitable for those users.
- Scalability: Scalability remains the big issue for the recommender systems. As the numbers of items are increasing exponentially, and pace of life is also increasing in the same pattern, people require results in a faster manner too. Thus the recommender system must be scalable enough i.e. it must not get slowed down with the increase in the data set.

1.3. Trust

Trust is the reliability or faith the user has in the system. It has become one of trending topic of research in many fields from the last few decades [15]. Trust in itself is very complex and multi-dimensional. It is relatively confusing and sometimes contradictory because of the various meanings of the trust in different disciplines [38]. There are lots of definitions given to the trust but none of them is complete and unambiguous.

Trust constitutes following properties [17]:-

- Subjective in nature: Trust is always subjective in nature i.e. it is based on the personal opinion, taste or mind.
- Context dependant: Trust depends on the context i.e. it varies with the changing circumstances. The value of trust is different for different context and situations.
- Exists with time: The value of trust exist with time. If A trust B in the past, it does not guarantee that A will trust B in the future.

- Dynamic in nature: Trust has the ability to change and adapt itself according to experience. It can be increased with the positive experience and can be decreased with the negative experience.
- Asymmetric: Asymmetric property gives the trust its directed nature i.e. it is always calculated for particular user. If A trust B, it does not implies that the B will trust A.
- Reflexive: The trust shows the property of reflexivity i.e. everyone have full trust on themselves.
- Directed: The trust is always unidirectional in nature thus the value of trust for A in B will be different from the value of trust for B in A.
- Non transitivity: Trust does not depend upon the relative behaviour of their neighbours thus ,if A has trust in B, and B has trust in C, it does not imply that A will also trust C.

Trust management aims at predicting the trustworthiness of the users based on their past records (i.e. ratings) and the behavior in an environment. By incorporating the trust, the recommendations generated will have higher confidence [16] and will be more optimized.

1.4. Swarm Intelligence

Insects, living in the colonies exhibit a common property of organizational skill without having centralized control. The interaction and behavior of these insects affects the whole colony. The way of living of these social insects, like honey bees, ants and termites in the colonies have inspired many information retrieval methods.

Swarm Intelligence is the self organized and collective behavior of these insects, natural or artificial, employed in work of optimizations technique. The inspiration of Swarm Intelligence comes from the biological systems in the environment or nature. To solve the optimization, combinatorial and communications problem, the use of social insects' metaphor has been increased now days. These systems typically consist of simple agents who interact with each other and with the environment without having any central control on them. They follow simple rules and the interactions between them make the whole system as an intelligent system.

The benefits of the swarm intelligence systems are:

- Adaptable: These systems are very adaptable in nature. They can easily adjust with the new situations and change according to the conditions.
- Evolvable: As these systems changes with the conditions, this quality of adaptability gives rise to the evolution of the system. Actually, the evolvable nature is the result of the adaptive quality of the swarm intelligence.
- Resilient: Swarm intelligence is a collective system i.e. it is made up of multiple nodes which are working in parallel. There is a lot of redundancy in the system which makes this system resilient.
- Novelty: Due to the adaptability, evolution and combinations of individual results, these systems generate new and novel results.

Swarm intelligence has applications in various domains such as robotics, routing problems, shortest path problems, other optimization problems, combinatorial problems etc. A lot of algorithms have been proposed based on the various social insects' behavior like ant colony algorithm based on pheromone strategy of ants, PSO on the grounds of changing velocity of the swarm particles in the group and bee colony algorithm based on the special dance property of the bees.

1.4.1 Bee Colony

Bee colony algorithm proposed in the year 2005, is the new introduction to the family of swarm intelligence algorithms. This algorithm simulates the food foraging behavior of honey bees. This algorithm uses neighborhood search with random search which was mainly used for optimization of problems. The communication in bee colony is done through the use of special dance. There are various terms used in bee colony algorithm such as:

- Beehive: This is a kind of home for the honey bees. It is the place from where they start searching for the food and communicates through dance after returning.

- Scout bees: The scout bees search for the food outside the beehive and returns to beehive. They communicate the food sources to other bees by doing a special dance in the beehive which tells the amount and directions of the food source to the other bees.
- Employed bees: The bees who are employed to the food sources are named as employed bees and are equal to the number of food sources. They go to the food sources and come back to the beehive and dance on the area for communicating their food sources. When their food source becomes scarce, they become scout bees and start searching for the new food source.
- Onlooker bees: As the name describes, the onlooker bees reside in the beehive and watch the dance of employed bees. They choose the food sources depending upon the dances of other bees and becomes employed bees.
- Nectar: It is the sugar liquid present mainly in the flowers which attracts the insects like bees. The scout bees check the amount of nectar present in the food sources and on this basis they select the food source.
- Waggle dance: Dance is the special property of the honey bees. They communicate through the dance. Waggle dance is the name given to the dance performed by the honey bees for communicating with the other bees. This dance tells the three things about the food source i.e. the direction of the food source, the distance of the food source from the hive and the quality of the food source (it depends upon the level of nectar present in the food source). If the distance of food source is less than 50 metres than the dance is known as “Round dance”.

Bee colony can extend themselves over a distance of more than 10 kms. In nature, the honey bees in order to get the food perform the following steps which show their distributed behavior:-

- Scout bees go for the search of the food source in multiple directions over a distance of more than 10 kms. They select the food source which contains the maximum amount of nectar in them.
- Basically the food source with more amount of nectar is visited by more number of bees and with less nectar will be visited by fewer bees.

- When the scout bees return to the beehive, those that found the food source which is above the certain threshold in terms of quality go to the floor and start the waggle dance.
- The waggle dance communicates the quality, distance and direction of the food source. This information helps the other onlooker bees to find the food sources and becomes the employed bees.
- The food source with high quality will be employed with the more bees. This process continuous till all the food sources becomes scarce.

This food foraging behavior of honey bees is used in many problems of the real world such as robotics, travelling salesman problem, clustering, data mining, job scheduling, statistical quality control, mechanical designs, etc. The special characteristics of bee colony algorithm like communication, self organization and dynamic nature can be used in recommender systems very efficiently for providing optimized suggestions.

1.5 Structure of the thesis

Below is the summary of the rest of the chapters of the thesis:

Chapter 2: Literature Survey. This chapter introduces all the related work and survey of the recommender system and its approaches.

Chapter 3: Problem Statement. In this chapter the gaps in the field of recommender system and motives of proposed approach is described.

Chapter 4: Proposed Work. Proposed methodology algorithm with proper example is explained in details in this chapter.

Chapter 5: Experimental Analysis. This chapter contains the description of the results and observations of the experimental analysis of proposed system.

Chapter 6: Conclusion and Future Scope. The whole work presented in thesis is summarized in this chapter and it also contains the directions for the future research of the work.

Chapter 2
Literature Survey

2.1 Evolution of Recommender System and Its Applications

Recommender Systems started in the mid 1990's with the work by Goldberg *et al.* [20]. Tapestry, the first recommender system used the concept of collaborative filtering in the mail system to filter the mails instead of the contents of the documents or mails. This system became very famous. From that day to the present, the recommender systems have been one of the trending topics of research. The "word of mouth" is the oldest version of the recommender systems. This is the method which is adopted by most of the people before they go to buy something and recommender system was built to automate this "word of mouth" [21]. Grouplens, a very well known project in the area of recommender system, is led by John, Joseph and Loren from the Minnesota University. Recommender System has been given separate research area since the exponential growth of the internet and the rise of the e-commerce solutions in the mid 90's and is very helpful in gaining higher profits.

There are many applications on which recommender systems proved very useful. Some of the examples of the recommender systems are:

- E-commerce: Recommender systems have been widely used in the e-commerce sites to promote their products and help the users to find more products to purchase. Various sites use different recommender systems according to their requirements as illustrated in [5]. Recommender systems help them increasing their profits and users by finding the products they want.
- Course Recommender: Recommender systems are also being used for the recommendations of the courses, students can take for completing their degree. O'Mahony and Smith [1] described that the students will be recommended with the best courses for them according to their needs and requirements to get the maximum in terms of knowledge, interest and course requirements.

- Planners: Recommender systems are used for planning the things. There are various applications which use recommender system for planning. Meetings can be planned using recommender system as illustrated in [22], wedding and tours can also be planned using the recommender systems.

2.2 Classification of Recommender System

In [8] Burke classified recommender systems into Collaborative, content based, demographic based, knowledge based and hybrid.

Collaborative Methods remains one of the most common and widely used methods in the field of recommender system. They were applied and evaluated extensively by grouplens [24] and in Ringo system. Most of the commercial sites like amazon.com, ebay.com, *etc.* use collaborative filtering due to its simplicity and good results. Breese *et al.* [23] have divided CF into three classes: Model-based methods, Memory-based methods and Hybrid methods.

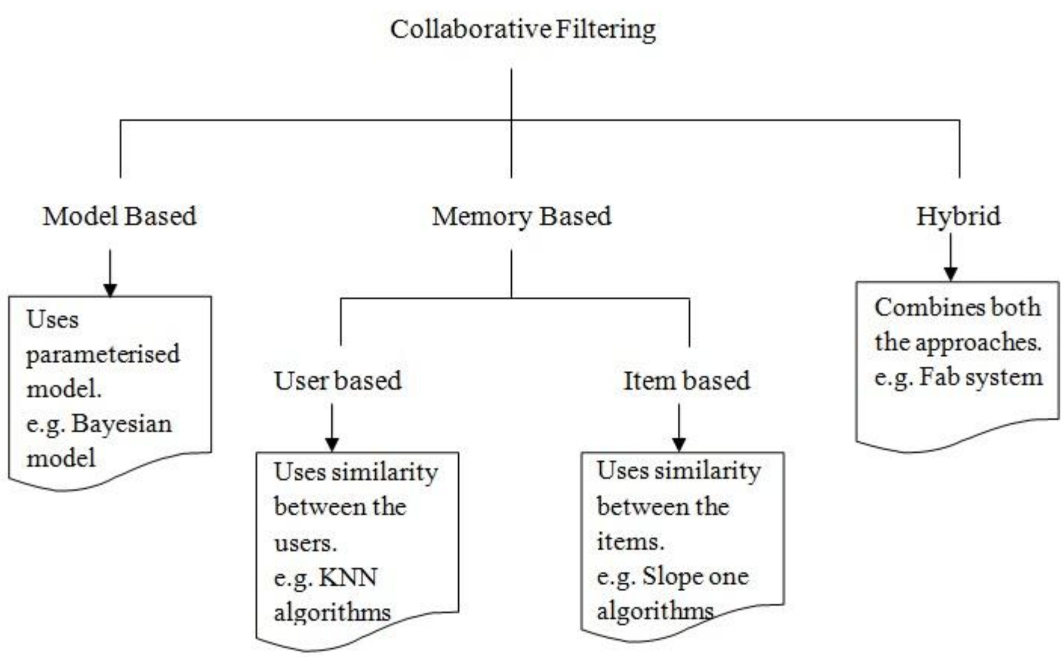

Figure 2.1: Classification of collaborative filtering

Memory based algorithms as demonstrated in [24] essentially makes predictions on the basis of the previously rated items by the users. Memory based algorithms has been bifurcated by Breese *et al.* [23] in two sub groups i.e. user-based and item-based. User based methods uses the similarity of different users for finding the neighbors. Various methods like Manhattan distance, Pearson coefficient as used by Sarwar *et al.* [27] are used for finding the similarity. KNN is the most common example of the user based systems which is used in [13]. Item based methods were introduced in [28] and use the similarity between the items instead of users as in the user based. The similarity between the items is calculated using the cosine similarity. The slope one algorithm is common example of item based methods [29] and predicts the ratings of the items through the other item's ratings.

The major shortcomings of the memory based CF are:

- Performance decreases if the data is sparse.
- New user cannot get the recommendations.
- Dependant on human ratings.
- Scalability is limited for large dataset.

A model based algorithm in [13] addresses the sparsity better and is also scalable. It uses the rating matrix to fit in a parameterized model to give the recommendations e.g. Bayesian classifiers [7]. The most common example of the model based methods are clustering CF, Bayesian belief nets CF and latent semantic analysis CF.

The drawbacks of the model based CF are:

- Expensive to build models.
- There is a trade-off between the recommendation accuracy and the scalability.
- Lack diversity.

Hybrid collaborative filtering [30] combines both the memory based and model based methods to generate the more efficient algorithm. One of the common examples of hybrid CF algorithm is Fab recommender system. But there are some disadvantages prevalent in this approach too [13]. Some of these are:

- Increased complexity.
- Expensive to implement.
- Lot of information required.

A content based recommender system employs the use of keywords (content or the description) of the items and thus the matching is done on the basis of keywords only [41], thus it is capable of recommending new items. The most famous internet radio system, Pandora.com uses the concept of content based recommender system. It uses the music genome project which evaluates a song using almost 400 attributes. Other famous examples are rotten tomatoes, internet movie database and so on.

But this class of recommender system also has loopholes. Some of them [41] are:

- The type and number of features of the items are fixed, so there is limited analysis of the content.
- They always recommend the similar items which the user already rated or liked. Hence the problem of serendipity arises.
- They cannot detect the unexpected.
- Unable to provide predictions for the new users.

Demographic based systems [40] use the demographic information like age, sex, education, country for identifying the type of the user and making the predictions to the user. Matrimonial sites, naukri.com are some of the usual examples of demographic based systems.

Demographic based systems also deal with some weaknesses which are:

- They require much information about the user before giving any predictions. Less information may cause bad recommendations.
- These systems are very specific. They only give same kind of recommendations.
- Highly dependent on user inputs.

Knowledge based systems [4] uses the knowledge about the users and items to generate the recommendations. This system works on the inferences about the user's preferences and predicts items that satisfy user needs [8]. The first use of the knowledge based system was in Entree system, a restaurant recommender system. This system makes recommendations of the restaurants in different cities which are similar to those the user likes or knows.

But knowledge based system also have some drawbacks as:

- It requires large amount of information.
- The accuracy of the prediction is not very good.
- It cannot predict recommendations for the users with no or less information.
- It requires human interaction to provide all the details or information about themselves.

Hybrid recommender systems are the systems which combines two or more approaches of the recommender systems to conceal the weaknesses of the approaches. In [8] various combinations of the above techniques have been made to build the better recommender system. One such example is of Netflix.com. It recommends movies by having a look on the searching and watching patterns of the users (i.e. collaborative filtering) and also offers movies that have some common characteristics with the movie watched by the users (i.e. content based).

Shortcomings of the hybrid systems are:

- Increased complexity.
- Expensive implementation.
- Context dependent.

2.3 Combining different approaches with collaborative filtering

Collaborative filtering is mostly implemented in commercial sites but different categories of attacks are hindrance in the way to get valid and perfect results. The profile injection or shilling attack [39] acts as an obstacle in getting accurate recommendations. There are also various issues like cold start problem, grey sheep problem and sparse user-item matrix in [13] which are a big concern for the recommender system.

Due to these existing problems and to improve the performance of CF technique, various new approaches have been developed over the years. To solve the problems of collaborative filtering, [35] matrix factorization technique is utilized with item based collaborative filtering to generate the recommendations for the television programs. This method produces good result but is expensive in terms of computations. Another technique which mixes the collaborative filtering with the genetic algorithms has been discussed in [26] is not able to generate results quickly. The collaborative filtering has also been merged with the famous k-means algorithm in [36] where the users are divided into k clusters based on their interest and then use a voting method for generating the recommendations for the users. This method is more accurate and fast then the traditional collaborative filtering but it also suffers from the problem of cold start and is not able to handle the outliers.

2.3.1 Incorporation of swarm intelligence with collaborative filtering

Various swarm intelligence algorithms are also been clubbed with the collaborative filtering techniques to improve the predictions. Swarm intelligence uses the biological nature of social insects and their properties of decentralization and self-organization. The various studies of different swarm intelligence principles are described in [18] by Garnier. These swarm intelligence techniques are used in the different recommender systems like particle swarm optimization in [25]. It is an adaptive approach which finely tunes the profile matching algorithm of the recommender system. This algorithm gives good results in comparison to the traditional recommender system. Camazine and Sneyd [31] presented the new system called case based recommender system (CASIS) which uses the metaphor from social insects' colonies. This system learns the preferences of the user and then automatically generates the predictions for them. Ant colony is also used in the recommender system for generating the course recommendations for the students by [37]. The generation of the final grades of the students, represented in the form of graph is generated upon the completion of the course.

2.3.2 Introduction of Trust in Recommender System

Introduction of trust in the recommender system is the huge advancement in the area of recommender system. The concept of trust in the web and social networks was first introduced in the [38] by Lausen and Zeigler. Trust factor increases the confidence in the recommendations generated and helps to check the issues like shilling attacks in the recommender system. The trust in the recommender was used by O'Donovan and Smyth[33], this system improves the quality of recommendations of the collaborative filtering by adding the factor of a trustworthy user.

As illustrated in [15,38] trust is a 2-D graph which shows the value of trust for every users with other users they are connected in the graph with. In [15], a novel method is proposed to reduce the computations required in the step of generating the trust. It reduces the whole large set of users into the small subset of user with which the active user interacts and thus subjugates the computational overhead. The random walk method was introduced by Jamali[16] for combining the trust with the collaborative filtering. This method also deals with the sparsity of the dataset.

Trust was initially considered as a probability value i.e. its value lies between 0 and 1 (0 means no trust and 1 means full trust between the users). A method has been proposed by Wang in [15] which takes probability and certainty as two dimensions of the trust. Trust is calculated through various factors but in most of the cases the trust is calculated as the combination of two factors i.e. degree of similarity which shows how much similar the users are and level of confidence which shows the amount of confidence the user has in another user. These two factors are projected by Bedi and Sharma [37] for the calculation of trust in the recommender systems. These factors are very effective for the estimation of trust; the value of trust is calculated by taking the harmonic mean of the two factors. By using these methods, the value of the trust can be incorporated in the recommender systems, through which attacks like profile injection, which utilizes the fake profile can be removed because the recommendations are always generated through trustworthy users.

Chapter 3

Problem Statement

Collaborative filtering is a widely used technique because of its simplicity in implementation and good results. The incorporation of trust mechanism in the collaborative filtering increases the confidence of the recommendations of the systems and also helps the user ensure that the recommendations are generated only through the trustworthy users. It thus eliminates the chances of the faux recommendations through fake profiles.

Since there is no single approach which is able to handle all the problems confronted by the recommender system, in many cases, combinations of various approaches have been used for generating precise recommendations. But, sometimes even combinations of approaches are unable to solve all the problems. Collaborative filtering and clustering techniques like K-means have been clubbed together to improve speed but is still prone to attacks and sparsity problem. Collaborative filtering and swarm intelligence methods like ant colony algorithms; hybrid genetic algorithm methods improve accuracy but are not able to solve the problem of scalability. Trust mechanism is incorporated in collaborative filtering to remove the fake profile attacks and increase the confidence in the recommendations but it fails to address the scalability and sparsity of dataset.
But the suggested hybrid approach of collaborative filtering with swarm intelligence and trust mechanism has the ability to provide solutions to the maximum number of issues recommender systems are facing. It generates optimized and enhanced recommendations and makes the system reliable and intelligent.

The proposed approach aims to improve and enhance the productivity and efficiency of recommender systems. The work mainly focuses on the flaws prevailing in the conventional recommender systems and builds a system which eliminates most of the vulnerabilities.

The proposed framework is a hybrid of collaborative filtering technique with swarm intelligence method i.e. bee colony algorithm and trust mechanism. The method addresses the problems which majority of current recommender systems are facing like cold start problem, sparse dataset, scalability, shilling attacks or profile injection attacks and diversity in the predictions and provides enhanced and optimized results.

3.1 Objectives

The main objectives of the proposed approach are:

- To propose some generic solution of providing recommendations for items which are not known to the users.
- To generate unbiased suggestions to the users which are not affected by the bogus ratings of the fake profiles.
- To build confidence in the recommendations by generating the recommendations from the trustworthy users leading to faith building of the users in the system by giving genuine suggestions.
- To predict fast recommendations for the users even if the dataset is very large and ever increasing rapidly, making the system dynamic and trustworthy.

3.2. Validation and methodology of the proposed system

To fulfill the above outlined objectives the methodology followed is:

- With the introduction of graphs, neighbors of the user can be found very easily and through them novel recommendations can be generated; new users can also get recommendations generated from this system in spite of the fact that they haven't rated any item. They will get the recommendations through the most trusted user i.e. the user who is trusted by most of the other users present in the whole system and thus, removes the problem of cold start.

- Embedding swarm intelligence algorithms i.e. bee colony algorithms with the collaborative filtering helps in improving the results because of their properties like decentralization and self-organization. Bee colony's random search method and the special dance property helps in finding the neighbors and updating the trust in an efficient manner.

- Trust mechanism is introduced with the help of trust graph which shows the user's trust or confidence in every other user i.e. how much he/she trusts the other users in the graph. It ensures the recommendations are genuine i.e. free from fake and false predictions.

- Swarm intelligence ensures the quick and easy method of finding the predictions. The adaptability feature of the swarm intelligence helps in continuously updating the trust graph. It also gives efficient method of generating the recommendations from the neighbors. Performance of the system is not affected by the increasing number of users and items because for every user only connected users are checked which makes the system scalable.

Implementation results prove that the intended system has the capability to handle many of the vulnerabilities in the existing recommender systems in an effective and proficient manner.

Chapter 4
Proposed Methodology

4.1. Introduction

Any single approach is not proficient enough to generating the précised and optimized recommendations for the user. Therefore an amalgamation of different approaches is needed to provide better results in less time. In this section, brief explanation of the proposed work is given. The proposed methodology is a combination of:

- Collaborative filtering technique.
- Trust mechanism through the graphs.
- Random search method of bee colony in the trust graphs.
- Special dance property of honey bee for communicating between users and updating the trust.

In the proposed method the trust graph for each user with other users is first created using the confidence and the similarity between the users. This graph once generated is continuously updated. The recommendations are made by using the Artificial Bee Colony technique which simulates the food foraging behavior of the honey bees and the trust is updated through the special dance of honey bees.

The proposed approach works in two phases:

- Phase 1-Directed trust graph construction (Static/offline phase)
- Phase 2-Providing recommendation to the active user (Dynamic/online phase)

The Static or offline phase consists of the steps which are executed only once initially before the actual recommendation generating process begins and these steps are never repeated in the lifetime of the algorithm. These steps construct the directed trust graph which shows the trust value for every user with other users.

Dynamic or online phase consists of steps which will be executing continuously throughout the lifetime of the algorithm. The trust value is updated continuously in the trust graph built in the static phase due to change in behavior of the user in the system.

Fig. 4.1 provides the complete flow chart of the mechanism followed in the proposed system.

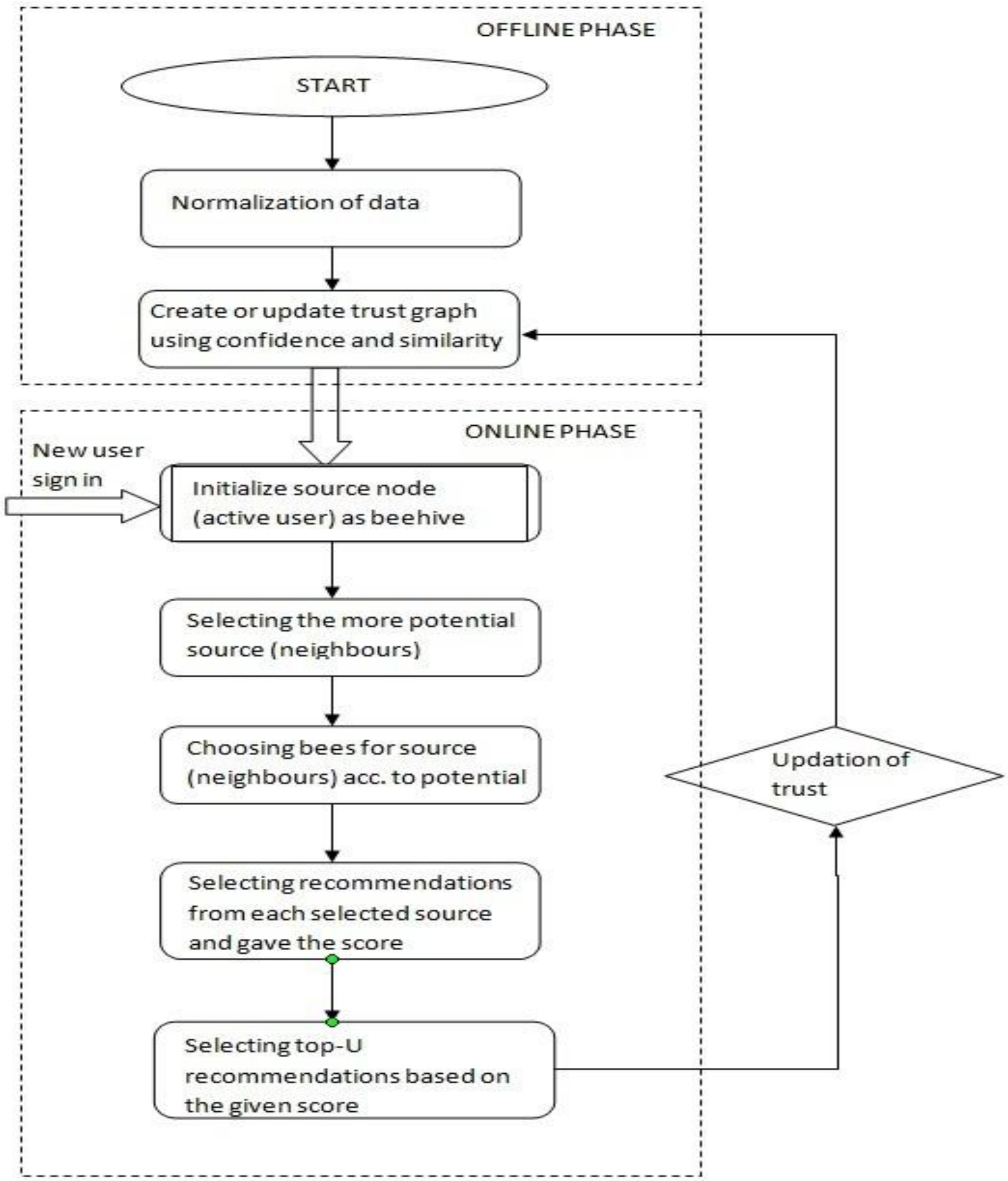

Figure 4.1: Flowchart of proposed system.

4.2. Detailed Explanation of the algorithm

4.2.1 Phase1-Directed trust graph construction

This phase, also known as pre-processing phase, requires a directed trust graph to be built for each user utilizing the user-item rating matrix and information about trusted users. This phase consists of following steps:-

1. Normalization of data - The data used by any recommender system are usually stored in user-item rating matrix. The columns represent the items and rows represent the users. This matrix contains the ratings of users for the items. These ratings may be real or discrete ranging in different scales. So for easy calculations, normalization is done to a standard value range i.e. 0-1 or 1-5 etc.

   The normalized value 'y' is calculated as:

   $$y = \frac{x - minimum\ \ value}{maximum\ \ value - minimum\ \ value} \tag{4}$$

   Where 'x' is the original rating of the data and 'y' is the normalized rating.

2. Trust Graph - Trust for each user is calculated with respect to all the other users. This trust value is stored in the form of graph G= (V, E), where V represents the users and E represents the edges as shown in figure 4.2. The weight on each edge represents the intensity of trust among the users and the value of trust lies between 0 and 1. This value increases or decreases with every positive or negative response. Trust is updated using the special dance strategy of honey bees (discussed in next subsection 4.2.2). The value of trust is calculated using the confidence and similarity between the users as shown below:

   - Confidence in user: This value shows how much confidence an active user has in other user. The value of confidence for user 'u' in user 'v' is :

     $$Conf(u,v) \quad = \frac{no.of\ \ items\ \ rated\ \ by\ u\ and\ \ v\ both}{no.of\ \ items\ \ rated\ \ by\ \ u} \tag{5}$$

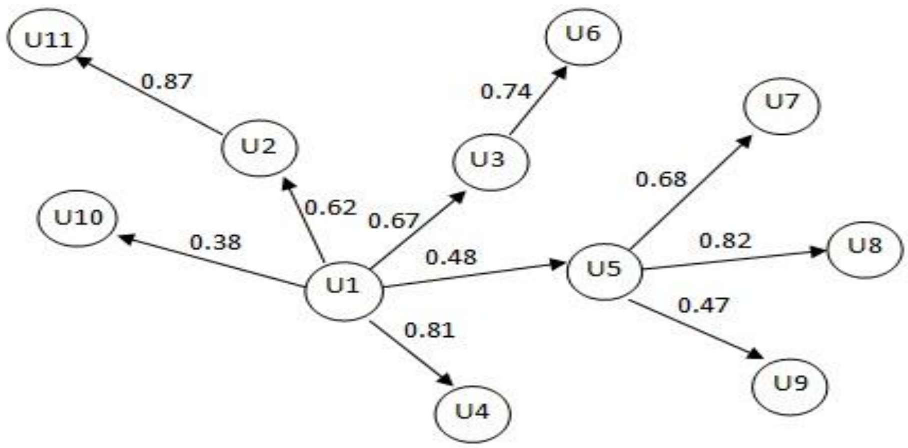

Figure 4.2: Directed trust graph

- Similarity: The degree of similarity between users shows how similar they are on the basis of ratings they provide. Here Pearson coefficient has been used for calculating the similarity between the two users 'u' and 'v' as shown below:

$$sim(u,v) = \begin{Bmatrix} P_{u,v} = \frac{\sum(r_{u,i}-\overline{r_u})(r_{v,i}-\overline{r_v})}{\sigma_u \sigma_v} & if\ P_{u,v} > 0 \\ 0 & else \end{Bmatrix} \quad (6)$$

Where $r_{u,i}$ is the rating of user U for item i, $\overline{r_u}$ is the average rating of user U and σ_u is the standard deviation of the ratings of user U.

Now trust ($\tau_{u,v}$) is calculated by taking harmonic mean of equation (5) and (6) which is denoted by:-

$$\tau_{u,v} = \frac{2 \times sim\,(u,v) \times conf\ (u,v)}{sim\,(u,v) + conf\ (u,v)} \quad (7)$$

The benefit of using harmonic mean is that the value of trust is high only if both the factors are high.

EXAMPLE: Consider an example for calculating the value of trust between the users U1 and U2 from the graph as shown in the above diagram. Let user U1 has rated 20 items and U2 has rated 25 items out of which 12 items are similar. So, confidence of U1 on U2 is:

Confidence = 12/20

i.e. 0.6 is the degree of confidence of user U1 on user U2.

The similarity between the users is calculated through the Pearson correlation coefficient method. Let the similarity between the user U1 and user U2 comes out to be 0.66.

Then, the value of trust between the users U1 and U2 is calculated by taking the harmonic mean of both the values i.e.

Trust = (2 × 0.6 × 0.66) / (0.6 + 0.66)

So, trust between users U1 and U2 is 0.6286 as mentioned in the above diagram.

4.2.2 Phase 2-Providing recommendation to the active user

The recommendations are made by using the Artificial Bee Colony technique which simulates the food foraging behavior of the honey bees and thus the trust is updated through the special dance of honey bees.

This phase constitutes of following steps:-

1. Initializing source node (user) as a beehive - The active user or source node S is initialized as a beehive for which the bees will bring the nectar or actually provide the recommendations.
2. Select K-potential sources (K-nearest neighbours) – In this step, K scout bees are sent for finding the potential sources (neighbours) from where actually recommendations can be generated. Scout bees are the honey bees which are send to discover the potential source for other unemployed bees. This detection is performed using following steps:
   - Create 'K' scout bees which are equal to the number of neighbours required for recommendations. Each bee tracks the number of nodes visited by it.
   - Each bee located at node 'i' will move towards node 'j' depending upon the trust between the node 'i' and 'j' and level of connectedness from S i.e.

     $$Prob_{i,j}^{k}(t) = (\tau_{i,j} \times \eta_{i,j}) \quad (8)$$

     Where $\eta_{i,j} = \frac{1}{level}$ from S.
   - If the potential of the node is greater than threshold value (α) i.e.

     If $(Prob^{k}_{i,j}(t) - \alpha) > 0$, then the node is selected by the bee otherwise it is rejected by the bee. Here α is constant and depends on the user-item matrix.

- If selected sources are less than K, then move to next level.

3. Choose bees according to their potential for the selected sources - The scout bees perform the special dance known as *waggle dance* to communicate about the source and its potentials. The numbers of bees are selected through this waggle dance. The duration of this dance indicates the potential of the source and hence requires more number of bees.
   - The formula for duration of this dance or number of bees selected for particular source is given as:

   $$Number_{i,j} = \frac{\tau_{i,j} \times U}{\tau_{max}} \tag{9}$$

   Where $Number_{i,j}$ is the number of bees required by selected source 'j' from active user 'i'

   $\tau_{i,j}$ is the trust of active user 'i' on user 'j' and τ_{max} is the maximum trust of active user 'i' respectively. 'U' is the total number of recommendations to be generated for 'i'.

4. Fetch recommendations from each source- The unemployed bees are employed to the each selected source equal to the number which has been calculated in the above step. Each bee will fetch the item from the selected source which has not yet been rated by the active user. All these bees with recommended items are collected and evaluated on the basis of score value. The score value depends on:
   - The ratings which the selected user (source node) provide to the item.
   - The trust between the selected user (source node) and the active user.

   i.e. $\text{Score}_i = \text{rating}_i \times \text{trust}\ (\tau) \tag{10}$

   Where $score_i$ are the score of the item 'i', $rating_i$ denotes the rating given by selected user to item 'i' respectively and τ represents the trust between the active user and selected user.

   Based upon the given score, the best bee is selected i.e. bee with item having best score while others are rejected.

5. Generate recommendations for active user- The recommendation for active user is generated by selecting the top-U items based on the score from the list of above mentioned items.

EXAMPLE: Consider the earlier example to display the steps 1-5 followed in phase 2. If recommendations are required to be generated for the user U1, then

- User U1will be initialized as the beehive.
- Let the value of k be 4 (nearest neighbours) and value of α is 0.5 (minimum threshold value for trust), then the selected neighbours will be U2, U3, U4 from the level 1 and U11 from the level 2. This is explained in the fig. 4.3.

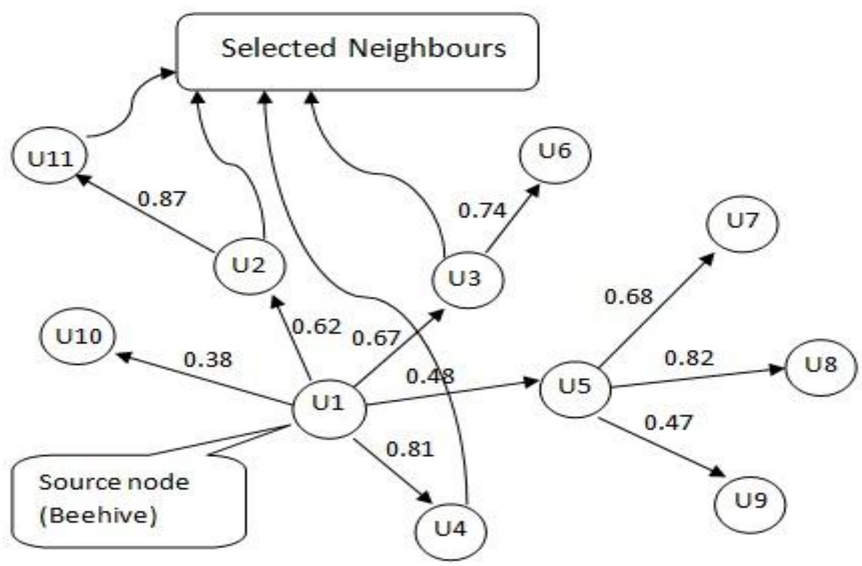

Figure 4.3: Directed trust graph showing source node and neighbours

- Choose the number of items that should be fetched from the selected neighbours. Let the total number of recommendations to be generated for the user U1 is 10 and the maximum value of trust among all the neighbours is 0.81 for the level 1. Therefore the value of this "number" for different users is calculated as:

  Number (U2) = integer ((0.62/0.81) × 10)

  i.e. number for U2 is 7

  Number (U3) = integer ((0.67/0.81) × 10)

  i.e. number for U3 is 8

  Number (U4) = integer ((0.81/0.81) × 10)

  i.e. number for U4 is 10

Number (U11) = integer ((0.62/(0.81 × 2)) × 10)

i.e. number for U11 is 3

These number counts are the maximum number of items that is to be fetched from the respective neighbours.

- All the items (i.e. 28) are stored in the array and a score is calculated for every respective item. The value of score represents the importance of that item for the course of recommendations. It is calculated by multiplying the rating of the item given by the respective user and the trust of that user with active user. Suppose item I1 is rated by user U2 as 4.5 and I2 is rated by user U3 as 4.0, the respective score for the items will be:

  Score (I1) = (4.5) × 0.62 i.e. 2.79

  Score (I2) = (4.0) × 0.67 i.e. 2.68

  After getting the score of all the items present in the array, next step is to sort the items according to the score and store them in the new array.

- Now finally the top U items i.e. 10 items are recommended to the active user.

6. Update trust graph- Trust being a dynamic quantity, must be updated continuously. Trust graph is updated each time the recommendations are made. If the recommendations are good, then the trust is increased and if it is not worthy then it is decreased. The trust is updated through the special dance of honey bees. The honey bee whose item is selected by active user will perform a typical dance namely UP-dance and honey bees whose items are not selected by active user will perform DOWN-dance. There is a small constant μ, which is used for updating the trust value in the graph. The updation strategy is explained below:-

- UP-dance: This dance is done by the bee whose item is selected by the user. Further two options are available: whether the user only visits the item or rates the item as well.

  ➢ When the user only visits the item, the value of trust (τ_{new}) will become:

$$\tau_{new} = (1 + \left(\mu \times \frac{count}{level}\right)) \times \tau_{old} \tag{11}$$

  where τ_{new} and τ_{old} are the new and old trust value respectively, μ is the fixed constant, count is the number of recommendations generated by this particular user for active user and level represents the level of connectedness with active user.

Using this formula, the trust of user having greater potential of generating the recommendations will be increased and enhanced more than those having less potential.

- When the user rates the item also, the new value of trust (τ_{new}) will become:

$$\tau_{\text{new}} = \left(1 + \left(\mu \times \frac{\text{count}}{\text{level}}\right)\right) \times \tau_{\text{old}} + \frac{\Delta \text{Q}}{\text{level}} \tag{12}$$

$$\Delta Q = \tau_{old} \times \frac{(rating - average\ \ rating\ )}{maximum\ \ rating} \tag{13}$$

Where rating is the rating given by active user to item, average rating subjects to the average of the ratings which can be given to items, maximum rating denotes the maximum rating which can be given to the item.

For example, if the scale of 1-5 is considered i.e. 1: very bad, 3: average and 5: very good. Here if the user gives rating more than 3 (average) then ΔQ will be positive i.e. the user has visited the item and liked it, if user rating is less than 3 (average) then ΔQ will be negative i.e. user has visited the item but disliked it.

- DOWN-dance: This dance is performed by the bees whose item is not selected by the user. This will decrease the trust between the users. The new value of trust (τ_{new}) becomes:

$$\tau_{new} = \begin{Bmatrix} \left(1 - \left(\frac{\mu}{level}\right)\right) \times \tau_{old} & if\ count = 0 \\ \left(1 - \left(\frac{\mu}{count \times level}\right)\right) \times \tau_{old} & else \end{Bmatrix} \tag{14}$$

where, μ is the fixed constant, count is the number of recommendations generated by this particular user for active user and level refers to the level of connectedness with active user.

The formula described in Eq (14) decreases the trust by fewer amounts from the users having more potential in generating the recommendations but by greater extent from the rest of the users.

So by using these formulae, the trust of users who provide better recommendations is enhanced while the trust of users who do not provide good recommendations is reduced.

EXAMPLE: Consider the previous example. After the items are recommended to the user, the user may check one of the items or may even rate them. Now the value of trust must be updated for the user whose recommended item is selected and also for those users whose recommended items are not liked or selected by the user.

We will consider the different cases one by one. Let the number of items in the final recommendation (i.e. count) from the user U2 is 3, from the user U3 is 2, from user U4 is 4 and from user U11 is 1.

Case 1: Let the active user U1 visits the recommendations generated from U2. The UP dance case of the bee colony is adhered for updating the trust. The updated value is equal to:

$\tau_{new} = (1 + 0.1 \times 3/1) \times 0.62$ i.e. 0.80

Case 2: Let the active user U1 also rates the recommendations generated from U2 as 4.0. Now again UP dance will be followed for updating its trust. In this dataset the average value is taken to be 3.0, maximum is 5.0 and minimum is 0.0. Thus the revised value is calculated as:

$\tau_{new} = ((1 + (0.1 \times 3/1)) + (4.0 - 3.0)/5.0) \times 0.62$ i.e. 0.901

Case 3: Now the trust of active user U1 for all other users whose recommendations are not rated and not even visited will be decreased. Here the policy of DOWN dance of the bee colony will be abided. The updated values of the trust of user U1 for the other user is equal to:

τ_{new} (U3) = $((1 - (0.1/ 2\times1)) \times 0.67$ i.e. 0.62

τ_{new} (U4) = $((1 - (0.1/ 4\times1)) \times 0.81$ i.e. 0.78

τ_{new} (U5) = $((1 - (0.1/ 1)) \times 0.48$ i.e. 0.43

The complete directed trust graph after the updation is described in fig. 4.4:

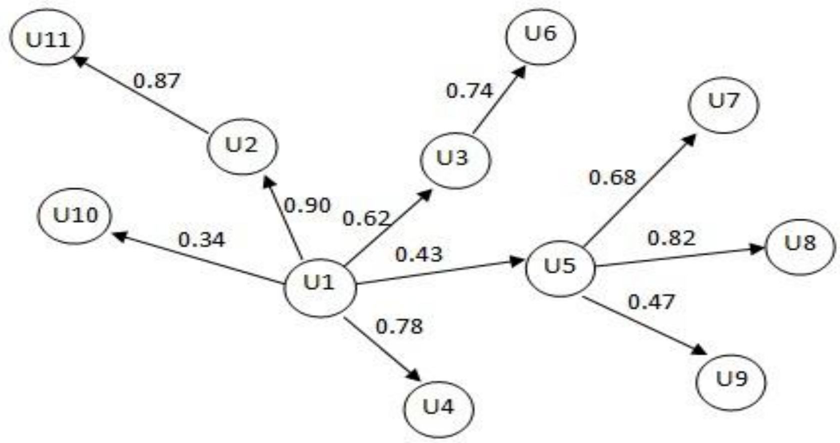

Figure 4.4: Directed trust graph after updation

Chapter 5
Experimental Analysis

The experimental analysis has been performed in Python 2.7 on Microsoft windows operating system. This analysis was executed on the Movielens dataset which is provided online by the Grouplens. This dataset is available online on http://grouplens.org/datasets/movielens/.

The Movielens dataset used has following features:

- It contains over 10, 000, 00 ratings from 943 user to 1682 movies
- Each user has rated minimum 20 movies.
- Every user ratings are on the scale of 5 (i.e. 1-5, 1means dislike and 5 means perfectly like).
- It has different files containing the rating information, user's demographic information, categories of the items and information of user, item and ratings.

5.1. Evaluation Parameters

The parameters on the basis of which results have been evaluated are Precision and Recall, common parameters adopted in various fields for evaluating the results. For measuring their value, the top-N items for which user's rating already exists are predicted and then evaluated for their correctness. The dataset is divided into training set and test set. The common formulae for Precision and Recall are:

$$Recall = \frac{Size\ of\ hit\ test}{Size\ of\ test\ set} \tag{15}$$

$$Precision = \frac{Size\ of\ hit\ test}{Size\ of\ top - N\ set} \tag{16}$$

Where, size of hit set is the number of items present in test set and the top-N recommendations generated for user.

Size of test set is number of items which are used as test set.

And size of top-N set is the number of items recommended for user.

The Recall value increases with the increase in number of recommendations produced as the size of hit set will be incremented but the value of Precision will decrease as the portion of denominator will increase more than the numerator. Another important parameter is used which combines both these metrics known as F-measure or F1 score. Its value is given by:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{17}$$

5.2. Results

The experimental analysis has been done on different recommender system approaches like KNN approach (traditional collaborative filtering technique using nearest neighbours), Slope one approach (Item based approach) and our proposed approach. The top 35 recommendations (i.e. top-N=35) are generated from 5 neighbors (i.e. K=5) and the test set constitutes 40 items. The results obtained from the implementation of the proposed system (before and after trust updation) have been provided in the form of comparative analysis with traditional approaches through Table 1.

Table 1. Analysis of results from traditional approaches and proposed approach.

	KNN approach	Slope one approach	Proposed approach (before updation)	Proposed approach (after trust updation)
Recall	32.25	28.75	31.56	32.75
Precision	37.14	32.66	39.25	42.68
F1	34.5	30.58	34.99	37.06

The result of the table I shows that proposed approach outperform the traditional collaborative filtering system (KNN approach) and slope one approach. The trust graph updates with every positive feedback and decreases with every negative feedback through the special UP and DOWN dance of the bee colony, thus after performing some iteration, the trust graph will be updated and the performance of the system increases with the iterations as shown by the results. The bar graph of the results of the table I is shown in fig 5.1.

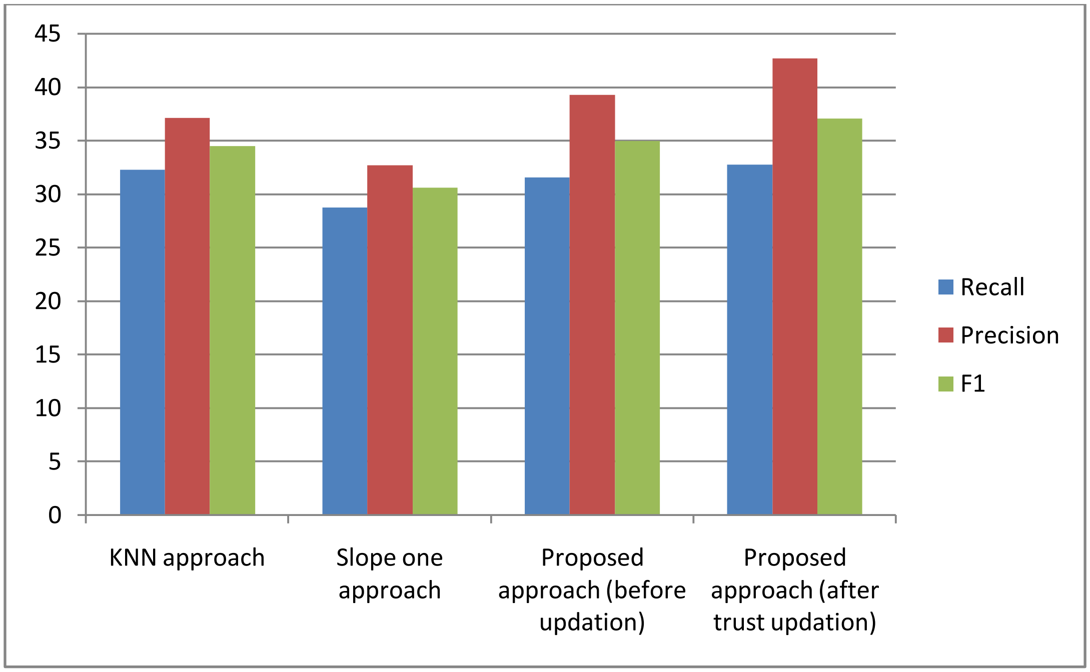

Figure 5.1: Bar graph of comparison of different approaches

The above proposed approach eliminates the following vulnerabilities in the system:-

1. Cold Start Problem- In the given system, the users who don't have any neighbours, obtain the recommendations from the most trusted user of the group, which is found by aggregating the trust value on each edge of users and thus the user with maximum value will be selected.
2. Shilling attack- Here, the recommendations are generated only through the trusted neighbours. So the possibility of fake recommendations is eradicated.
3. Sparse data-Now, the user may get recommendations from multiple levels i.e. gathering recommendations from trusted neighbourhood of active user's trusted neighbour. Hence recommender system's accuracy increases.
4. Scalability-. In the proposed approach, the trust graph is prepared only once and beforehand. The new user is added to the graph and his/her trust is updated according to his preferences. Also during recommendations, only particular user's edges are evaluated and updated which have been already maintained. Hence, computation is minimized as there is no need of finding the similarity.

5.3. Observations

Parameters such as scalability, effect of sparsity, diversity in results, profile injection (shilling attack), cold start problem, coverage and continuous updation of systems have been used to compare the above proposed approach with the famous recommender system approaches. The observation provides results in the form of yes or high if it has the corresponding parameter or follows it, no if it does not have, medium if it may or may not affect the approach depending upon situation and partial if it affects the approach slightly. The following observations are observed on studying these approaches:

Table 2: Comparison of proposed approach with other approaches of recommender system.

	CF (KNN approach)	*Item-based (slope one approach)*	*Swarm intelligence approaches*	*Trust based approach*	*Proposed approach*
Scalability	*Low*	*Low*	*Good*	*Good*	*High*
Effect of sparsity	*High*	*Low*	*Partial*	*Low*	*Low*
Diversity of results	*Yes*	*Less*	*Yes*	*Yes*	*Yes*
Attacks effect (profile injection)	*Yes*	*Yes*	*Partial*	*No*	*No*
Cold start problem	*Yes*	*Yes*	*Yes*	*No*	*No*
Coverage	*Low*	*High*	*Medium*	*Medium*	*Medium*
Confidence	*Low*	*Low*	*Medium*	*High*	*High*
Continuously updated	*No*	*No*	*Yes*	*No*	*Yes*

As shown in Table 2, the proposed approach is more scalable and has mild effect of sparsity as compared to any of the conventional approaches currently being used.

Problems like cold start and profile injection attacks which have bad consequences on the efficiency of recommender systems have been resolved. The confidence and coverage comes out to be high and more prominent and with every incoming recommendation the system gets continuously updated.

Based on the results achieved in Table 1 and parameters used for comparisons in Table 2, it is clearly evident that the proposed system as a whole works much better and demarcates itself from the traditional approaches existing.

5.4. Illustration of implementation steps

The steps followed in the system are:

1. Foremost the user has to login to the system with their prescibed user-id and password.

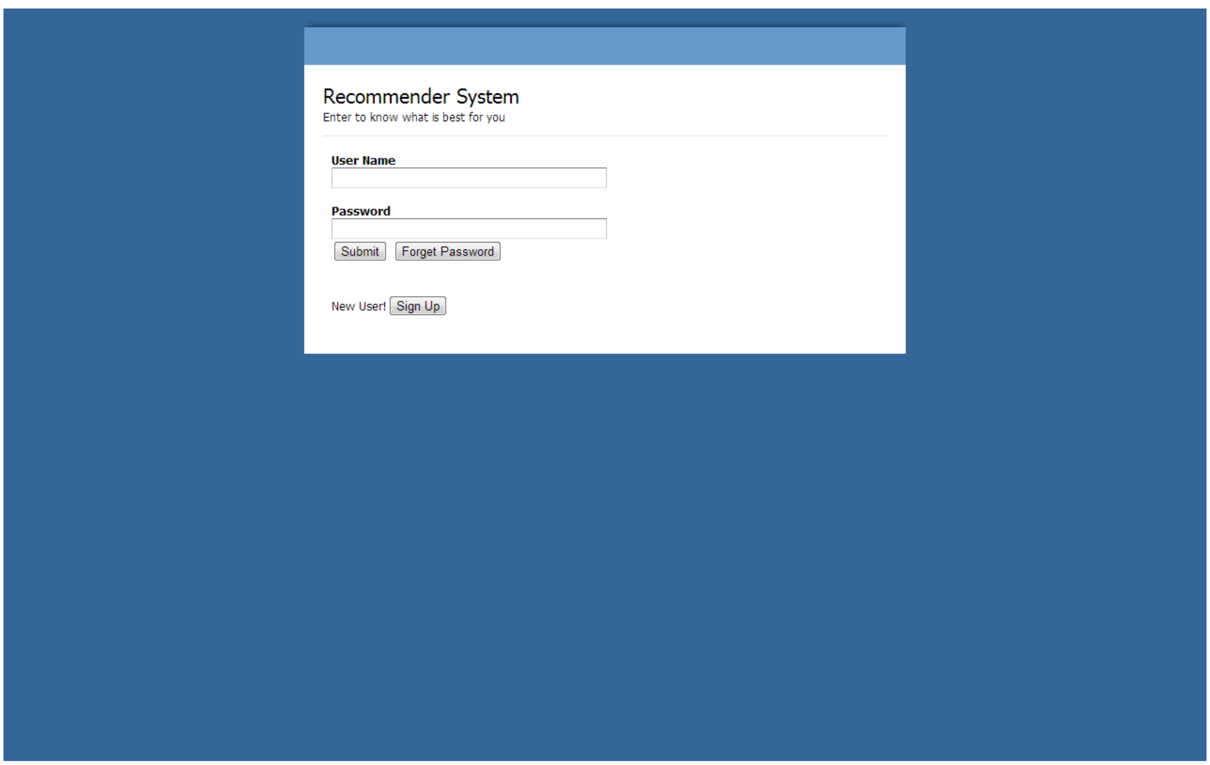

Figure 5.2: login page of the system

2. If the user-id and password does not match with the data present in the database, then the system will show the error message as shown in the fig.

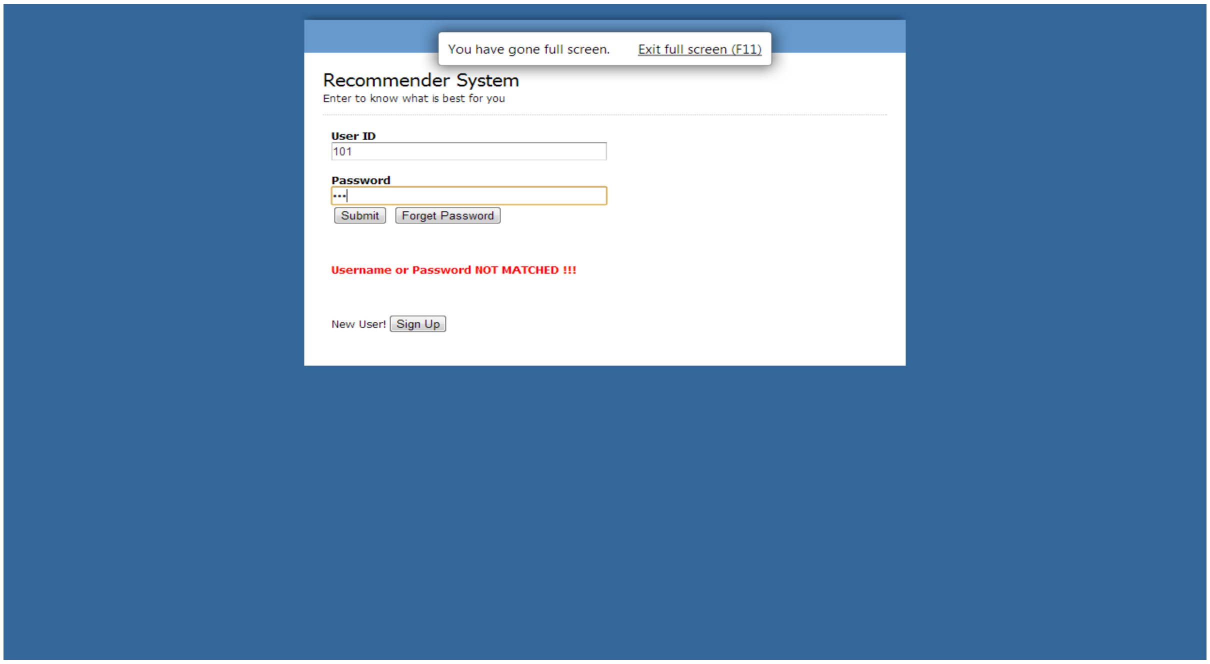

Figure 5.3: Invalid id or password

3. The database authenticates the user id and password and user obtains the privileges to access the site and obtain recommendations about the movies according to their choices. Fig. 5.4 shows the various recommendations for the user.

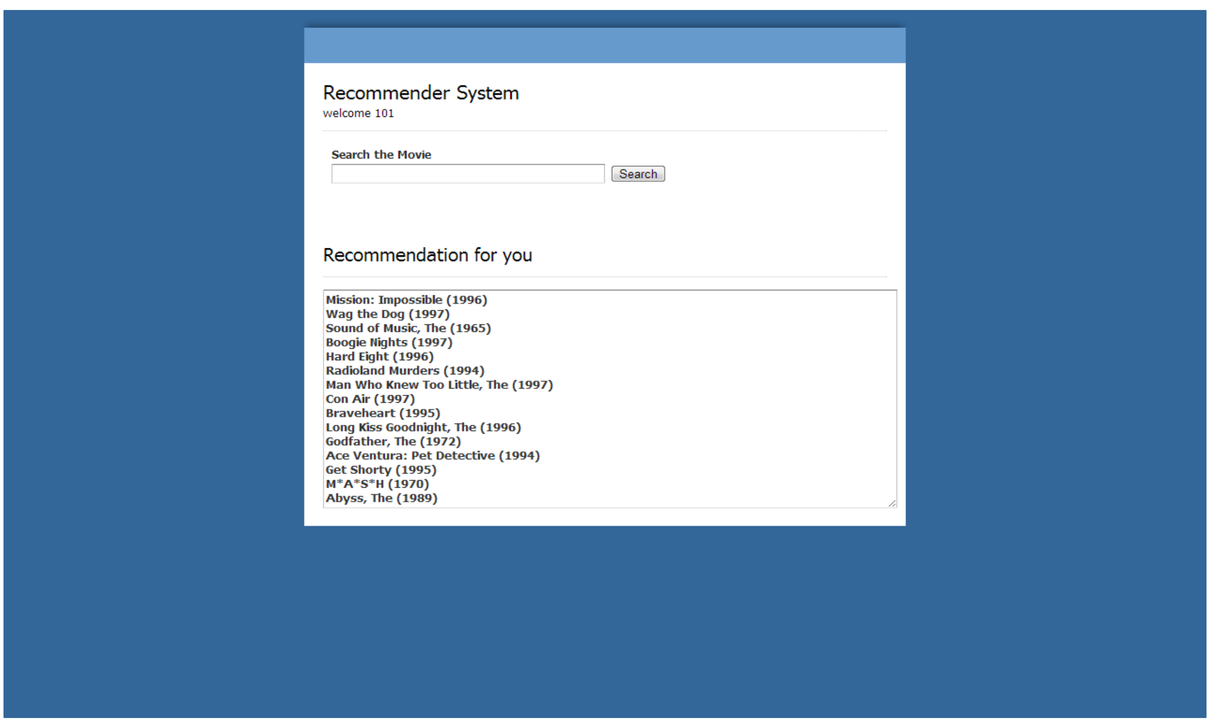

Figure 5.4: Generated recommendations

4. Fig. 5.5 shows the user opts for an item from the set of recommendations generated for him based on his preferences and then the enter button is pressed.

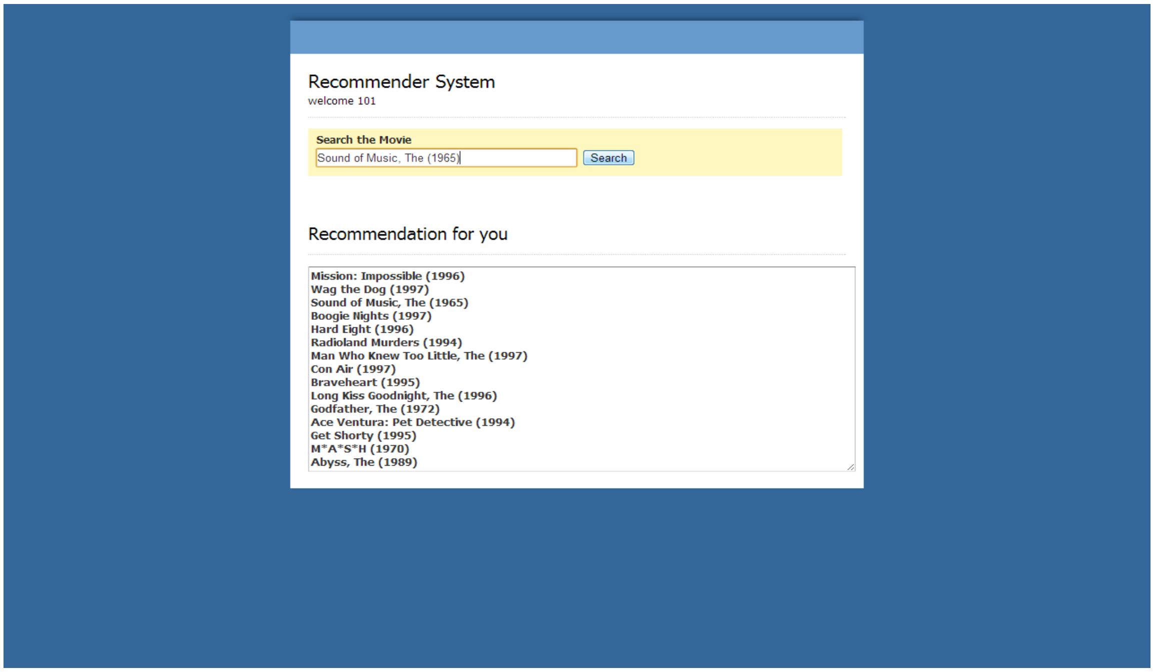

Figure 5.5: Selecting items

5. After the recommended item is chosen, the user is queried to rate the item and the user submits the rating for the item in the system.

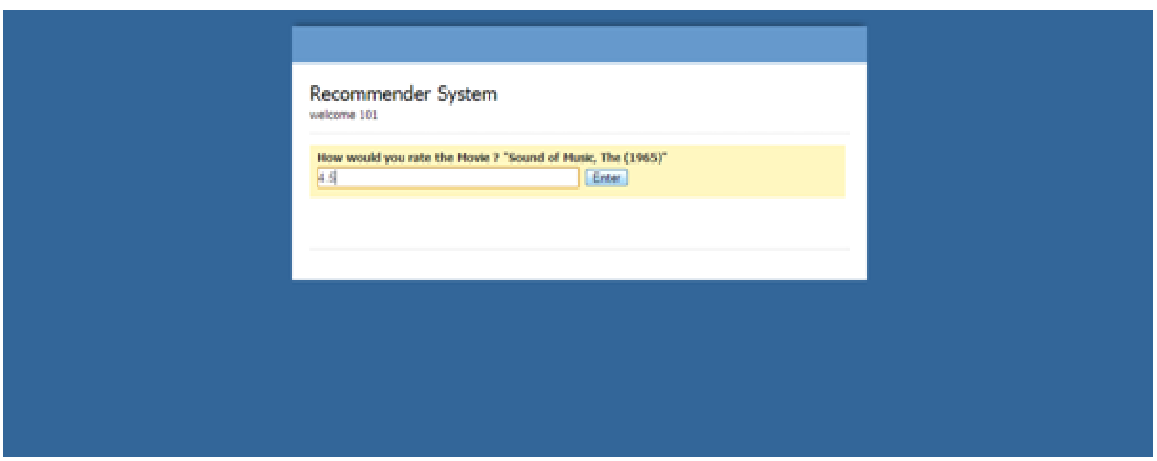

Figure 5.6: Rating the item

6. Once the item is rated, the user's trust graph will be updated and the recommendations are then generated through the new values of trust graph for the user.

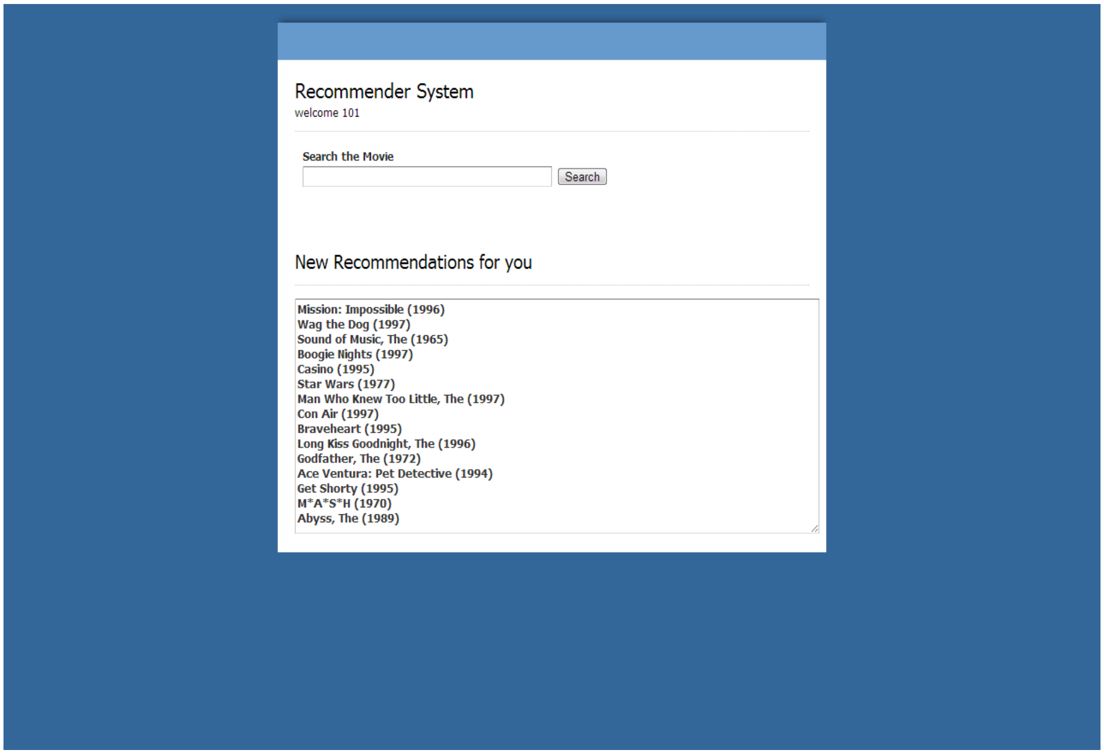

Figure 5.7: New recommendation

7. If the user is new to the system, then he/she can register himself/herself at the system by clicking on the signup button. Fig. describe the form appears after clicking it.

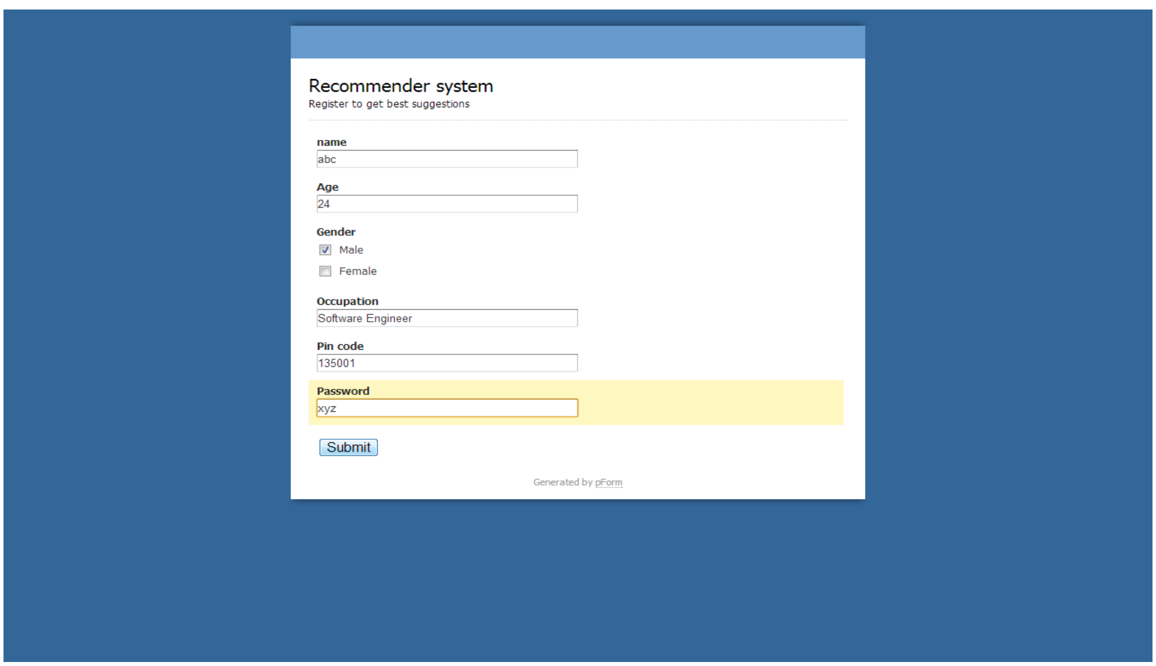

Figure 5.8: Registering new user

8. The user, after filling all the required fields, submits the registration form and gets successfully enrolled.

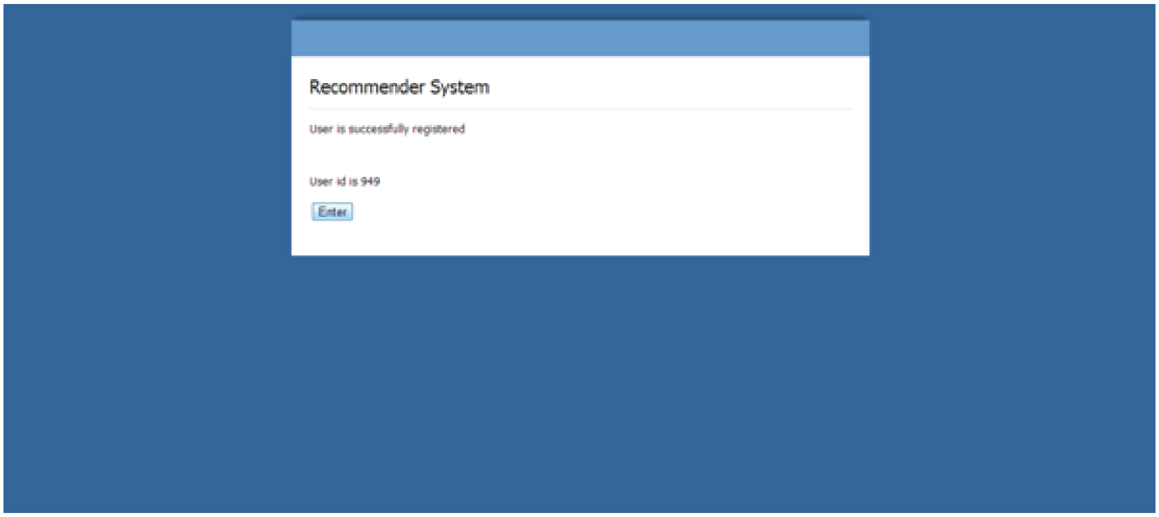

Figure 5.9: Successful registration

9. After registration, the new user access the system with unique ID, and the recommendation for him is also generated in the system as illustrated in fig.

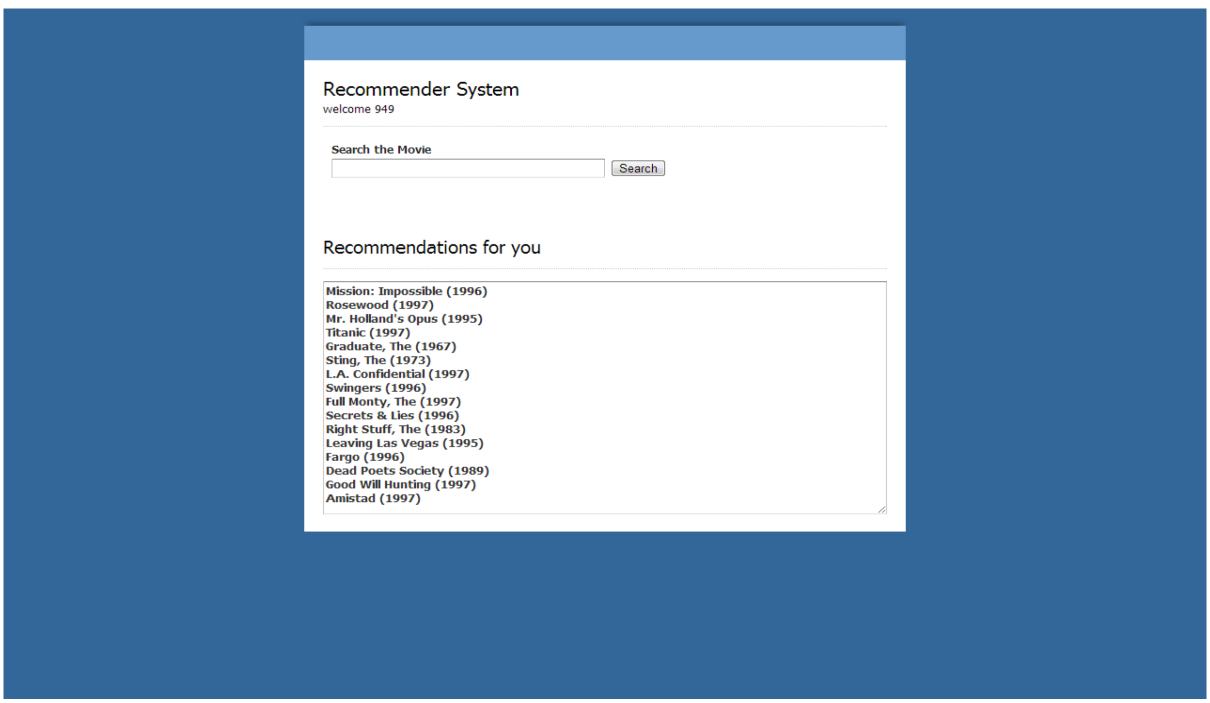

Figure 5.10: Recommendations to new user

Chapter 6

Conclusion and Future Scope

6.1. Conclusion

The proposed approach has the ability to eliminate the problems endured by collaborative filtering and render enhanced recommendations. It removes one of the most problematic issue of recommender system i.e. attacks through fake profiles. The trust involved in the system allows only trusted users to give recommendations thus eliminating the fake recommendations which solves shilling attack problem.

Another major problem of recommender systems is making predictions for new users. The most trusted person in the system will act as the default neighbors for new users who do not have any neighbors, thus solves the cold-start problem.

The trust graph used in the system is prepared beforehand and only once. This trust value is updated continuously using the special dance property of honey bees. Only the active user's trust values are used and updated during recommendations, thus computation is decreased. New user also can be added to the graph without causing many updation or computations, hence it is easily scalable.

Bee colony allows easy and efficient predictions for the users and the dance property of bee (UP and DOWN dance) colony helps in updation of the directed trust graph which makes the system dynamic and increases its proficiency.

The recommendations can be generated from higher levels also i.e. from trusted neighborhood of trusted users, hence it is good in case of sparse matrix and the coverage is also high. Thus this system provide enhanced recommendations which improves user satisfaction as they have trust in the recommendations which helps them making better decisions.

6.2 Summary of contribution

The proposed system put forward solutions to most of the problems and issues of the recommender systems which other recommender systems are not able to eliminate and generate enhanced and fair recommendations.

This system prepares a hybrid combination of swarm intelligence and trust mechanism with collaborative filtering which removes most of the problems which single system is unable to deal with. This system is of generic type not context dependent i.e. it can be used in any field for generating the recommendations.

First time users are getting the recommendations through this system with the help of most trusted user. The recommendations are generated very precisely through the neighbors who are picked using swarm intelligence. Fake profiles cannot hamper the accuracy of this system because these profiles are being discarded as non trusted profiles for recommendations. This system handles sparsity of dataset by generating recommendations through higher levels also and adjusting the factors accordingly. The issue of scalability does not slow down the speed of the system as it can handle increase in user and items.

Thus the recommender system which can handle all these problems and still predict the efficient suggestions for the users is a significant improvement. This improvement can be useful in improving the recommender systems of any field and helps in generating more accurate results which definitely provide ease to the user in choosing the items and also increase in the profit.

6.3. Future Scope

The accuracy and efficiency of this system depends upon the finding of the neighbors which are very close to the active user. These neighbors are calculated through the distance measure which uses the rating as an only factor of calculating the distance between the users. These ratings can be ambiguous as it depends upon the user's perception of the item. One user may like the item on some factor while the second user like the same item due to other factor e.g. one may like the movie because of its story while other may like the same movie because of acting and not on the basis of story. Thus these two persons cannot behave similarly.

These ratings are only one dimensional but for more accuracy multi-dimensional rating systems are required which can distinguish people on the basis of which they rate the items. The multi-dimensional and fuzzy system should be used for reducing the ambiguity and becoming surer in finding the similarity between the user and thus generating best suggestions.

References

[1] M. P. O. Mahony and B. Smyth, “A Recommender System for On-line Course Enrolment : An Initial Study,” pp. 133–136.

[2] J. Bobadilla, F. Serradilla, and a. Hernando, “Collaborative filtering adapted to recommender systems of e-learning,” *Knowledge-Based Syst.*, vol. 22, no. 4, pp. 261–265, May 2009.

[3] MJ Pazzani, D Billsus, Pazzani, "Content-based recommendation systems." *The adaptive web*. Springer Berlin Heidelberg, pp. 325-341, 2007.

[4] R. Burke. "Knowledge-based recommender systems," *Encyclopedia of library and information systems* 69, no. Supplement 32, pp. 175-186, 2000.

[5] J. Ben Schafer, J. Konstan, and J. Riedl, “Recommender Systems in E-Commerce,” pp. 158–166, 1999 In Proceedings of the 1st ACM conference on Electronic commerce, pp. 158-166. ACM, 1999.

[6] K. Choi, D. Yoo, G. Kim, and Y. Suh, “A hybrid online-product recommendation system: Combining implicit rating-based collaborative filtering and sequential pattern analysis,” *Electron. Commer. Res. Appl.*, vol. 11, no. 4, pp. 309–317, Jul. 2012.

[7] F Ricci, L Rokach, B Shapira,” Introduction to recommender systems handbook,” Springer US, 2011.

[8] R. Burke, “Hybrid Web Recommender Systems,” pp. 377–408, Springer Berlin Heidelberg, 2007.

[9] P. Resnick, H. R. Varian, and G. Editors, “ecommender Systems mmende tems,” Communications of the ACM vol. 40, no. 3, pp. 56–58, 1997.

[10] V Formoso, F Cacheda, V Carneiro,” Algorithms for Efficient Collaborative Filtering. In Efficiency Issues in Information Retrieval Workshop”, p. 17, 2008.

[11] P Han, B Xie, F Yang, R Shen, "A scalable P2P recommender system based on distributed collaborative filtering." *Expert systems with applications* 27, no. 2, pp. 203-210, 2004.

[12] J. Lee, M. Sun, and G. Lebanon, “A comparative study of collaborative filtering algorithms,” *arXiv Prepr. arXiv1205.3193*, pp. 1–27, 2012

[13] X. Su and T. M. Khoshgoftaar, “A Survey of Collaborative Filtering Techniques,” *Adv. Artif. Intell.*, vol. 2009, no. Section 3, pp. 1–19, 2009.

[14] J. Sobecki, "Ant colony metaphor applied in user interface recommendation," *New Generation Computing* 26, no. 3, pp. 277-293, 2008.

[15] J. Chandra, I. Scholtes, N. Ganguly, and F. Schweitzer, “A Tunable Mechanism for Identifying Trusted Nodes in Large Scale Distributed Networks,” *2012 IEEE 11th Int. Conf. Trust. Secur. Priv. Comput. Commun.*, pp. 722–729, Jun. 2012.

[16] M. Jamali, “TrustWalker : A Random Walk Model for Combining Trust-based and Item-based Recommendation.” In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, pp. 397-406, 2009.

[17] J. Golbeck, “Computing with Trust: Definition, Properties, and Algorithms,” *2006 Secur. Work.*, IEEE, pp. 1–7, Aug. 2006.

[18] S. Garnier, J. Gautrais, and G. Theraulaz, “The biological principles of swarm intelligence,” *Swarm Intell.*, vol. 1, no. 1, pp. 3–31, Jul. 2007.

[19] D. Karaboga and B. Basturk, “On the performance of artificial bee colony (ABC) algorithm,” *Appl. Soft Comput.*, vol. 8, no. 1, pp. 687–697, Jan. 2008.

[20] D. Goldberg, D. Nichols, and B. M. Oki, “Using to weave an Information,” Communications of the ACM 35, no. 12, pp. 61-70, 1992.

[21] U. Shardanandand and P. Maes, "Social information filtering: algorithms for automating “word of mouth”." In *Proceedings of the SIGCHI conference on Human factors in computing systems,* ACM Press/Addison-Wesley Publishing Co., pp. 210-217, 1995.

[22] S. Macho, M. Torrens, and B. Faltings, "A multi-agent recommender system for planning meetings." In *Workshop on Agent-based recommender systems*, 2000.

[23] JS Breese, D Heckerman, C Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," In *Proceedings of the Fourteenth*

conference on Uncertainty in artificial intelligence, Morgan Kaufmann Publishers Inc., pp. 43-52, 1998.

[24] P. Resnick , N. Iacovou, M. Suchak, P. Bergstrom, and John Riedl, "GroupLens: an open architecture for collaborative filtering of netnews." In*Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pp. 175-186, 1994.

[25] S. Ujjin and P. J. Bentley, "Particle swarm optimization recommender system," *Proc. 2003 IEEE Swarm Intell. Symp. SIS'03 (Cat. No.03EX706)*, IEEE, pp. 124–131, 2003.

[26] J. Bobadilla, F. Ortega, A. Hernando, and J. Alcalá, "Improving collaborative filtering recommender system results and performance using genetic algorithms," *Knowledge-Based Syst.*, vol. 24, no. 8, pp. 1310–1316, Dec. 2011.

[27] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "*Application of dimensionality reduction in recommender system-a case study"* No. TR-00-043. Minnesota Univ Minneapolis Dept of Computer Science, 2000.

[28] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms." In *Proceedings of the 10th international conference on World Wide Web*, ACM, pp. 285-295, 2001.

[29] D. Lemire and A. Maclachlan, "Slope One Predictors for Online Rating-Based Collaborative Filtering." In *SDM*, vol. 5, pp. 1-5. 2005.

[30] Y. Li, L. Lu, and L. Xuefeng, "A hybrid collaborative filtering method for multiple-interests and multiple-content recommendation in E-Commerce."*Expert Systems with Applications* 28, no. 1, pp. 67-77, 2005.

[31] F. Lorenzi, D. Scherer, and A. L. C. Bazzan, "Case-based recommender system inspired by social insects." 2005.

[32] R. Andersen, A. Flaxman, C. Borgs, J. Chayes, and U. Feige, "Trust-based recommendation systems : an axiomatic approach Categories and Subject Descriptors," pp. 199-208. ACM, 2008.

[33] J. O'Donovan and B. Smyth, "Trust in recommender systems," *Proc. 10th Int. Conf. Intell. user interfaces - IUI '05*, p. 167, ACM, pp. 167-174, 2005.

[34] P. Resnick, P. Bergstrom, and J. Riedl, "GroupLens : An Open Architecture for Collaborative Filtering of Netnews," In Proceedings of the 1994 ACM conference on Computer supported cooperative work, ACM, pp. 175-186, 1994.

[35] A. B. Barragáns-Martínez, E. Costa-Montenegro, J. C. Burguillo, M. Rey-López, F. a. Mikic-Fonte, and A. Peleteiro, "A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition," *Inf. Sci. (Ny).*, vol. 180, no. 22, pp. 4290–4311, Nov. 2010.

[36] G. M. Dakhel, "A New Collaborative Filtering Algorithm Using K-means Clustering and Neighbors ' Voting," In Hybrid Intelligent Systems (HIS), 2011 11th International Conference IEEE, pp. 179-184, 2011.

[37] P. Bedi and R. Sharma, "Trust based recommender system using ant colony for trust computation." *Expert Systems with Applications* 39, no. 1 pp.1183-1190, 2012.

[38] C.N. Ziegler and G. Lausen, "Propagation models for trust and distrust in social networks." *Information Systems Frontiers* 7, no. 4-5, pp. 337-358, 2005.

[39] K.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.

[40] M.J. Pazzani, "A framework for collaborative, content-based and demographic filtering." *Artificial Intelligence Review* 13, no. 5-6, pp. 393-408, 1999.

[41] P. Lops, M. de Gemmis, and G. Semeraro, "Content-based recommender systems: State of the art and trends." In *Recommender Systems Handbook*, Springer US, pp. 73-105, 2011.

List of Publications

[1] A. Kaleroun and S. Batra, "'Hybrid Artificial Bee Colony with Trust Mechanism in Recommender System" In Third International Conference on Advances in Computing, Communications and Informatics (ICACCI-2014) IEEE, 2014. [Accepted]

[2] A. Kaleroun and S. Batra, "Comparison of KNN and Slope one Algorithm in Sparse Environment" In International Journal of Advanced Research in Computer Science and Software Engineering Vol. 4 Issue 4, April 2014. [Accepted]

[3] A. Kaleroun and S. Batra, "' Collaborating Trust and Item-prediction with Ant Colony for Recommendation" In Seventh International Conference on Contemporary Computing (IC3)-2014 IEEE, 2014. [Communicating]