Efficient approach for Social Recommendations using Graphs on Neo4j

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

Master of Engineering

In

Software Engineering

Submitted By

Ashamdeep Kaur

(801631004)

Under the supervision of:

Dr. Rinkle Rani
Associate Professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY

PATIALA – 147004

JULY 2018

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, "Efficient approach for social recommendations using graphs on neo4j", in partial fulfilment of the requirements for the award of degree of Master of Engineering in *Software Engineering* submitted in Computer Science and Engineering Department of Thapar Institute of Engineering and Technology, Patiala is an authentic record of my own work carried out under the supervision of Dr. Rinkle Rani and refers other researcher's work which they 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.

Ashamdeep Kaur

(Ashamdeep Kaur)

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

Rinkle

(Dr. Rinkle Rani)
Associate Professor
CSED

ACKNOWLEDGEMENT

The successful completion of any task would be incomplete without acknowledging the people who made it possible and whose constant guidance and encouragement secured the success.

First of all I wish to acknowledge the benevolence of omnipotent Gog who gave me strength and courage to overcome all obstacles and showed me the silver lining in the dark clouds. With the profound sense of gratitude and heartiest regard, I express my sincere feelings of indebtedness to my guide **Dr. Rinkle Rani**, Associate Professor, Computer Science and Engineering Department, Thapar Institute of Engineering and Technology for her positive attitude and constant encouragement, keen interest, invaluable cooperation, generous attitude and above all her blessings. She has been a source of inspiration for me, I am grateful to **Dr. Maninder Singh**, Head of Department and **Rupali Bhardwaj**, P.G Coordinator, Computer Science and Engineering Department, Thapar Institute Of Engineering and Technology for the motivation and inspiration for the completion of this thesis. I will be failing in my duty if do not express my gratitude to **Dr. S.S. Bhatia**, Senior professor and Dean of Academic Affairs in the university for making provisions of infrastructure such as library facilities, computer labs equipped with internet facility, immensely useful for the learners to equip themselves with latest in the field.

Later but not the least I would like to express my heartfelt thanks to my parents and my friends who with their thought provoking views, veracity and whole hearted co-operation helped me in doing this thesis.

Ashandeep Kaur

ABSTRACT

Social networks are developing in number and size, with a huge number of client accounts and enormous amount of data. Explosive growth of data and digital information on internet has created a potential challenge for visitors to get coherent information which further hinders the user to access proper items of his interest on internet timely. . Recommendation Systems check the client's inclinations for proposing components to purchase or browse. They have turned out to be an essential applications in internet business and access to data that gives proposals that successfully diminish extensive data to the things that best address user’s issues and inclinations. Still sometimes we face a cold start problem and we do not get accurate recommendations. Another problem which is faced by basic recommender is sparsity problem where there is not enough data to extract recommendations.

We have proposed an extra favourable position of these systems in which collaborative filtering is used plus clients can encode more data about their relations than essentially say who they trust. Trust is the feature in which one user can explicitly state his trust on any user whose choice he likes.

Graphs are used to apply recommendations in the proposed approach. The graph database is an effective tool for handling relationships between entities of data model. Neo4j is used as graph database to implement the proposed algorithm as with the help of Neo4j we can manipulate graphs accordingly. Along with trust, also proposed different feature is transitivity in social networks through which we can get more accurate recommendations. Manipulated graph is called influenced graph in the whole research work. We have achieved sufficient results which prove this approach of transitivity between nodes is helpful for better recommendations.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1 INTRODUCTION

In past few decades computers have been serving as an extraordinary open doors for the development of society which indeed is growing with very fast pace making our daily life easy. Consistently we are utilizing them in our day to day lives. They have been developing at such a fast pace as people are now connected to each other because of computers- through our social networking profiles, conduct in talks and so forth [1]. Computers have manipulated our lives in several ways. Through all these social connections amongst users or with our virtual activities, behaviour of people changes eventually. Now it requires less time and efforts to send any message to anyone sitting anywhere compared to early times. So all these facts affect the technical and operational demand of computers and also of social networks. Social networks are receiving huge attention continuously in every perspective whether in business or in non-business areas. Commercialization stirred more extensive enthusiasm for recommender systems.

Explosive growth of data and digital information on internet has created a potential challenge for visitors to get coherent information which further hinders the user to access proper items of his interest on internet timely. Several crawlers or data repossession structures like DevilFinder , one famous data retrieval system google and Altavista resolved this problem in parts but personalisation (matching of user's requirement with items) and prioritization were still absent. In present day times, individuals are communicating in a wide range of routes, some by phone, some by E-mail and some by visiting networks. Consistently, individuals are utilizing the PC to converse with each other and social systems administration destinations are a major explanation behind this. Facebook, one of the biggest social networks on this planet, has been vital in giving individuals a chance to speak with each other simpler and many more social networking websites to communicate with each other. In the previous couple of years, in attendance was an enormous development of the communal webs. Such networks give the stage to users to express, share or talk about their thoughts, conclusions with their companions in social graph and furthermore speak with them [12]. Informal organizations are arranged both the expert and non-proficient customers. The professional social networks like

LinkedIn and Xing permit setting up the professional association between their clients and business cooperation. The web has grown exponentially in last few decades and which has resulted in blast of data and we have also seen exponential growth of mobile phones as well as social websites. As users have huge amount of information it becomes difficult for them to get the accurate data which is reliable as well as the availability of choices grows exponentially which further results in the issue outstands as Information overload. Rapid growth of internet poses a problem of abundance of data as well as users thus making users difficult to get accurate data according to their requirement. There is explicit as well as implicit data available on internet for which users require help and suggestions to get accurate data or matching and finding people on social networking websites. Matching people's choices with accurate data is done by recommender system. It has become so prevalent nowadays as clients do not prefer putting so much potential while browsing for accurate information; they are occupied with buying things rather than seeking accurate data. [5]. One of the traditional approaches for recommendations is Collaborative filtering. Collaborative filtering basically is in which individuals work together to help each other by rating the items and recording their feedback as a response to any product they buy or browse [2]. Recommender systems must have

(i) Foundation information, the information that the framework has beforehand the proposal progression starts,

(ii) Input information, the information that client must impart to the framework keeping in mind the end goal to create a proposal and

(iii) A calculation that joins foundation and information to land at its proposals.

1.1 Need of Recommender Systems:

The crucial need of recommender system arose when data on internet had explosive growth and user started to find it difficult to get relevant or accurate data on internet. Nothing is perfect neither is any recommender system. They just assist users in understanding or to easily get along with large digital collection in various domains and help them to access that data according to significant personal taste. The ultimate goal of recommender system is to match the customer with a correctly graded or rated that particular item. Another major factor of requirement of recommender system is money. Business tycoons are earning a lot with the help of recommender systems, otherwise why would Netflix have given one million dollar prize? If they had not been sure about the results they get from such competition that it would improve their overall revenue ten times. Ecommerce sites are putting lot of money into recommendations as if you would like to buy a cycle it would also show you helmet under the suggestions. They just try to stimulate the part of brain while you are shopping to make you feel happy when buying some stuff. Suggestions may appear like:

- Frequently bought together.
- Recommended for you.
- Because you have seen X you might also like Y.
- You will probably like this.
- Products related to this item.
- Customer who bought this item also bought.

1.2 Overview of Recommender System:

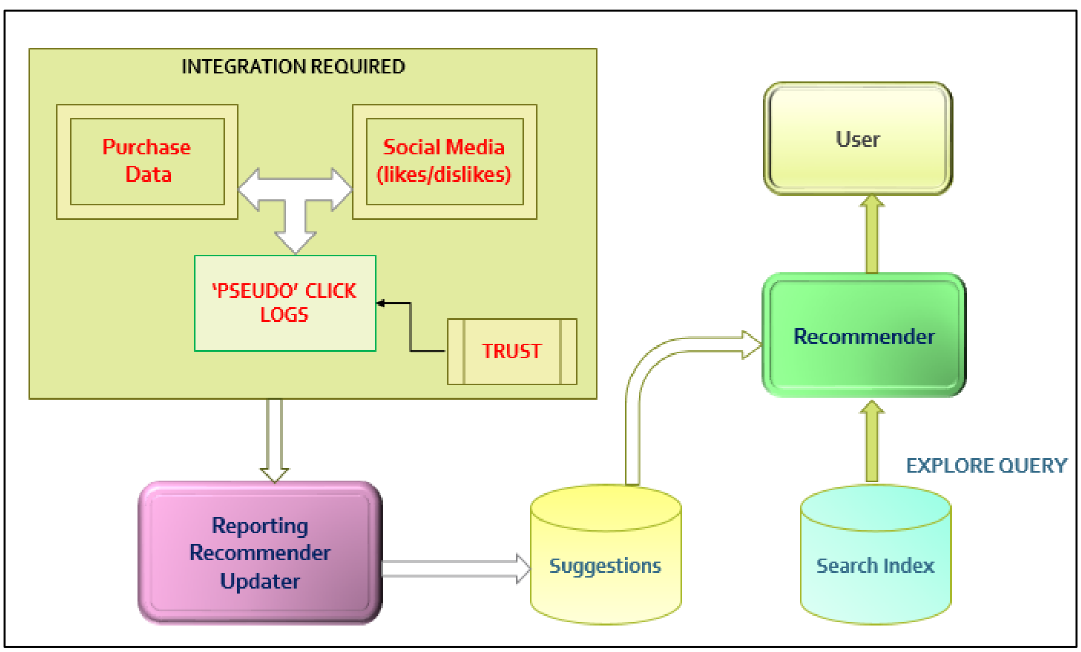

Figure 1: Working of Recommender System

Overwhelming amount of choices on internet makes a requirement to strain, prioritize and coherently deliver the information so that information overload problem can be alleviated.[30] Recommender systems as shown in figure 1 have been serving as the most ubiquitous feature for ecommerce websites not only because recommender systems have ability to help with personalised recommendation but also because of the profit they have been providing to the various ecommerce websites like amazon, Netflix, flipkart etc.[3] It has been clearly visible in amazon dataset that sales gets improved after implication of recommender system [29] . Recommendation system finds information designs or patterns in the informational collection by learning buyer's decisions and produces the results that co-identifies with their necessities and interests. To give more formal meaning of recommendation systems, let U_r be an arrangement of every single possible client, and let I_m be an arrangement of every probable item. In numerous online business websites, the space U_r and I_m can be vast. In perspective of recommendation system utility of items is delineated as rating. There is a matrix of user and items where relationships between user and items is represented and then calculated for further recommendations. User item matrix is shown below

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1	0	3	0	3	0
User 2	4	0	0	2	0
User 3	0	0	3	0	0
User 4	3	0	4	0	3
User 5	4	3	0	4	0

Figure 2: User-item matrix

As shown in figure 2 U_r 1 has rated I_m 2 as 3 and one user can rate any amount of items any positive number of rating. This matrix is constructed when people buy or rate items according to their preferences. 0 is taken as NULL as users have not given any rating to this particular item and by default system arranges that rating as 0.

1.3 Personalised Recommendations:

Personalised recommendations are those in which recommendations are given according to users rather than just recommending items generally. Personalised recommendations have improved the accuracy of recommendations in ecommerce websites and has served with various profits to the business. Personalization is a procedure that makes an applicable, individualized connection between two parties intended to improve the experience of the beneficiary. It utilizes knowledge in view of the beneficiary's personal data, and also conduct information about the activities of comparable people, to convey an affair that addresses particular needs and inclinations of choices accordingly.

1.4 Content Based Recommendation:

Content based or cognitive based filtering is in which user gets recommendation based on collation between item and user's previous history. It is different from collaborative based filtering as content based recommender systems are those who retrieve the information of user itself and filter the information. Basically in content based filtering the users get recommendations on the behalf of their previous history or the items they have preferred in the past. The client profile is spoken with similar terms and then developed by examining the depth of client's profile of things which have been seen by the client or have shown interest in buying them. It is also known as cognitive filtering. Most of the existing content based filtering techniques have their focus on recommending items who have information in textual form like books, news and documents. So it can be inferred that content based filtering works on contextual information. Content based filtering is one of the oldest technique used for recommendations. Content based approaches use a progression of discrete qualities of a thing keeping in mind the end goal to prescribe extra things with comparative properties about that particular item which means this filtering focuses on features of that item and make stack of items with same features and suggest that user accordingly. This kind of recommender system is exceptionally mainstream because of its accessibility and it can also be utilized to recommend things. For example sites, music, motion pictures, books, eateries and lodgings these algorithm is used to recommend. Steps one should consider while building content based filtering are:

1. Creating a profile for items with their attributes and their description of items which are to be recommended.
2. Create profiles for users with the same set of attributes.
3. Devise any utility task to generate the resemblance between user and item.
4. Provide customer with ranking of items with respect to similarity.
5. Contriving a strategy for refining the client profile based on feedback given by user.

1.5 Collaborative Filtering:

Another famous traditional approach for recommendations is Collaborative filtering, Collaborative filtering basically implies that individuals work together to help each other by filtering information according to their choices. In collaborative based systems user gets suggestions according to the people who have same choice as users. Like if one person has rated any item 4, collaborative filtering will find the similar users who have also rated that item 4 and with the help of Pearson coefficient or another techniques we can find similarity between users. In contrast to content based filtering, collaborative filtering rather than making recommendations in view of the comparability amongst client and thing, they prescribe things that clients of comparative inclinations have rated positively previously. The principle preferred standpoint of CF is that it doesn't have to know anything about the things/items beforehand in order to make recommendations. Recommendations are constructed solely in light of the similarity between users, implying that it is conceivable to prescribe or recommend exceptionally complex things also. Steps one should consider while making collaborative based filtering recommender are:

1. Clusters of relevant users are made.
2. In every cluster a ranked list of most popular item is made.
3. The item which is on top which user have not seen before is recommended.

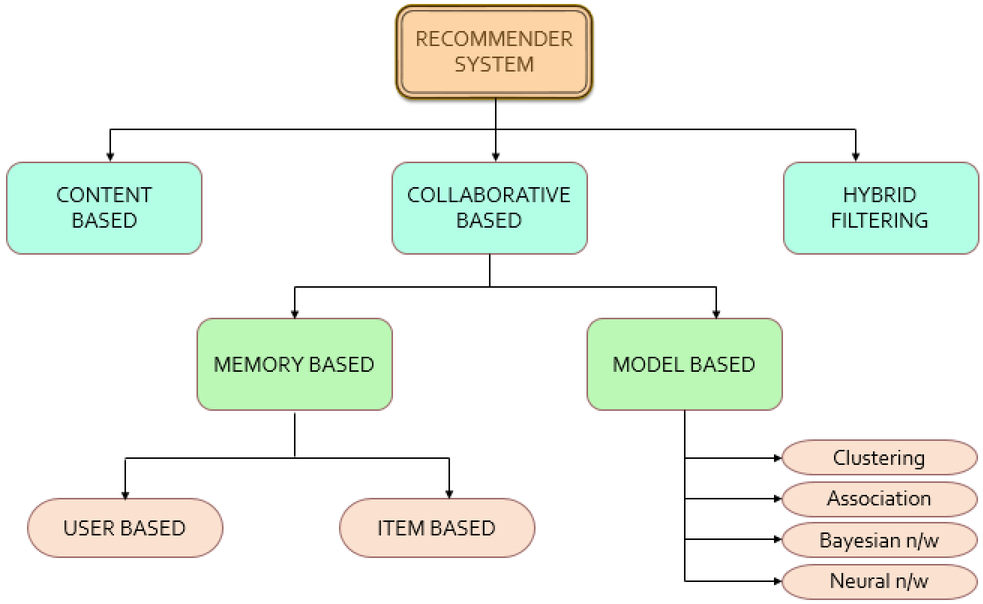

Figure 3: Types of Recommender System

Figure 3 shows various types of recommender system. There are two types of Collaborative Filtering methods which are used widely: Memory-based and Model-based. Memory based technique gives suggestions to target user based on ratings of similar users which is also known as user-based collaborative filtering and also gives suggestion based on items similar to that item which is also called item-based collaborative filtering. [31]

Model based technique trains a compact prototypical to make effective recommendations. In this a model is made considering various aspects of attributes and then trained based on the requirements of target users. Several training models have been investigated like clustering model [32], Bayesian hierarchical model [33], the aspects model [34] and the ranking models [35] but none of them encounters social network into account while performing.

1.6 Knowledge-Based Recommender Systems:

Knowledge-based recommender systems utilizes target client characterized inclinations and match them to the equivalent things. Such frameworks need to take care of same issues that emerge in Collaborative filtering (inadequate measure of information) or Content-based filtering (comparable thing does not really mean a right forecast). Knowledge-based frameworks are primarily reasonable for one-time costly buys – a computer, laptop or car. Client inclinations from past history of items bought may not exist, or they may be insufficient, or exceptionally old and obsolete. Two essential kinds of information based recommender frameworks are the *Case-based* and *Constraint-based* systems.

Case-based recommenders try to retrieve similar items using dissimilar sorts of resemblance measures. Constraint-based recommenders use unambiguously well-defined recommendation guidelines. [28]

1.7 Hybrid Based Recommender:

Hybrid as name suggests is combination of various filtering techniques in order to get more accurate and better performance with comparatively fewer drawbacks. Mostly collaborative filtering technique is combined with different frameworks. In command to alleviate the shortcomings confronted by described recommendation methods, several researchers have combined collaborative with content based filtering with different perspectives and degrees in order to get relevant recommendations. Burke [23] has classified six different hybridization techniques as follows:

1. Mixed
2. Switching
3. Cascade
4. Weighted
5. Feature combination
6. Meta-level

Hybrid recommender came into existence after evolution of various recommendation techniques. When all of recommendation techniques somewhere failed to connect to the bottom line of actual suggestions or recommendations then hybrid recommender was suggested. Because every technique has some pros and cons and therefore it was suggested by Burke[23] to mix them up to get all the positive points of recommender into one. Then these hybrid techniques shown in table were discussed that we should hybrid those techniques in which style. Hybrid recommender somewhere helped in reducing the errors in recommendations and was considered as successful invention. Table which is shown discuss various techniques through which recommendation techniques can be joined in any way according to the requirement of any website. Any hybridisation technique has some positive points together compared to individual technique.

Table 1: Demonstrates a portion of the mix strategies that have been utilized.

Hybridization technique	**Description**
Mixed	Recommendations from different recommenders are provided simultaneously.
Weighted	Score is computed from all the results provided by several recommendation techniques for recommended item.
Feature combination	All good features of different recommendation techniques are combined into one technique.
Switching	Switches between recommendation techniques on accordance with present scenario.
Cascade	Given recommendation is further filtered by another recommendation technique.
Meta level	Features are generated from one model as an input to second model.

Recommender system with trust values have been widely researched on which concludes that with the help of social trust user's preferences can be viewed alternatively other than ratings. User can follow any number of people's choices regardless of their similarity between each other. Trust aware recommendation is an enhanced version of recommendations where all the possibilities gets carried out. Yuan W [10] suggests that these trust networks are very compact networks where some arbitrary clients are socially associated to each other in this little separation, which further helps in implication of trust for better recommendation systems. On the other hand it has been proved that if we involve trust in our recommender system then it will definitely improve the final performance of results of recommendation. By integrating trust along with the collaborative filtering would work as icing on the cake. Rating prediction and item recommendation are two parts of recommender system on which recommender system works.

During early 1990s due to problem of growing internet and overload of data, recommender system came as an idea to researchers. First recommender was manual and that used to allow framework to tell users what other users are preferring. Such frameworks required the client's exertion and capacity to express their expectations. This recommender was based on collaborative filtering in which users are affected by other user's choice. [41]

2.1 Information Filtering:

Information filtering is the procedure permitting, beginning from an approaching volume of dynamic information, to concentrate and present the main information intriguing either a client or a gathering of clients having moderately comparable interests. The filtering framework makes a "forecast" about the handiness of the information to the client. This expectation depends on the "profile" of the client and prompts a basic leadership: "recommend" or "not recommend" information. [4 SEMINAR]

2.2 Classification Of Collaborative Filtering:

This technique is regularly assembled in two classes as shown above in figure. One is model based and memory based. Model based as name suggests first constructs a model out of the client-things collaboration database and after that uses this model to make recommendations to users. This first trains a model and then rectify the predictions. On the other hand memory based frameworks often called lazy frameworks the client-things ratings put away in the framework are straightforwardly used to anticipate ratings for new things without past preparing. In collaborative filtering we use similarity measures to compute similarity between users to provide them with suitable recommendations. Similarity measures which comes under memory based collaborative filter are explained below.

2.3 Similarity Measure:

Recommender frameworks confronts an issue of figuring closeness between clients (or things). The more comparative two clients are, the more probable it is that another thing loved by one of these clients will be preferred by the other. Diverse comparability measures (separations) can be connected to an issue, and the best decision relies upon the situation. There are several methods by which similarity can be measured. Some of those are Pearson Correlation Coefficient [42], probability-based similarity [t, u], Euclidean distance and Co-sine similarity [43]. Cosine similarity and Pearson Correlation Coefficient are the most widely used ones. The most common basic measure is **Euclidean distance**. It is computed as shown in equation 3

$$F(a, b) = \sqrt{\sum^{n} (a_c - b_c)^2} \qquad 3$$

Another commonly measured similarity measure is **Pearson correlation** which can be calculated as shown in equation 1.

$$P_c(U_\alpha, U_\beta) = \frac{\sum_{p=1}^{n}\left(\overline{G(\alpha)}p - G(\alpha)\right)\left(\overline{G(\beta)}p - G(\beta)\right)}{\sqrt{\sum_{p=1}^{n}(\overline{G(\alpha)_p} - G(\alpha))^2 \sum_{p=1}^{n}(\overline{G(\beta)_p} - G(\beta))^2}} \qquad (1)$$

Here, value of correlation calculated by Pearson formula varies from -1 to 1.

$G(\alpha)$ is final mean of all the ratings given by any particular user $\alpha, 0$

$G(\alpha)_p$ is rating which user α gave for any random product p,

$G(\beta)_p$ is rating which user β gave for any random product p,

$G(\beta)$ is final mean of all the ratings given by any particular user β. Here similarity is checked for all products between 1…n among users U_α and U_β.

Pearson relationship is helpful when we have to abstain from rating expansions. Envision we hereby have a motion picture recommender frameworks. We can expect two clients with same inclinations rate an arrangement of motion pictures also. However, in the event that one client is more \optimistic" her appraisals will be somewhat expanded as for the other. To reimburse that expansion and concentrate just on the standardized similitude between two client evaluations, we can utilize Pearson correlation.
Notice that Pearson does not conclude when one user give same rating to every item. That rating will be implied as NULL.

Cosine-based similarity calculates the cos of the angle for viewpoint which is formed by the alignment of rating factors. Resemblance amid user a and user b is computed as

$$F_{ab} = \frac{\sum_{c \in u} K_{ac} \,.\, K_{bc}}{\sqrt{\sum_{c \in u} (K_{ac})^2} \sqrt{\sum_{c \in u} (K_{bc})^2}}$$

2.4 Data Pre- Processing:

Pre-processing is preliminary processing of data which includes filtering scratching the appropriate data among large pile of data. Information is delegated as objects and their particular qualities where a trait is alluded as representative or property of a question. Different terms aimed at entity incorporate record, thing, point, test, instance or observation. An attribute may be likewise alluded to as a variable, field, feature or characteristic. Genuine information ordinarily should be pre-processed (e.g. purged, strained, transformed) while keeping in concern about the end goal to be utilized by machine learning procedures in the investigation step.

2.5 Trust Based Recommender:

Another trending recommendation technique is trust aware recommendation systems. Trust aware recommender frameworks are intelligent applications that make utilization of trust data which is one person trusting on another person explicitly and client individual information in social systems to give customized suggestions. Prior exploration in trust-aware frameworks it have appeared that trust-based frameworks have capacity of making exact forecasts combined by means of their vigour from various problems like overcoming shilling attacks which make them a healthier or better option as compared to that traditional recommender frameworks [50]. Here figure 4 shows input of trust and ratings and then simultaneously trust and ratings generate recommendations or predicted ratings.

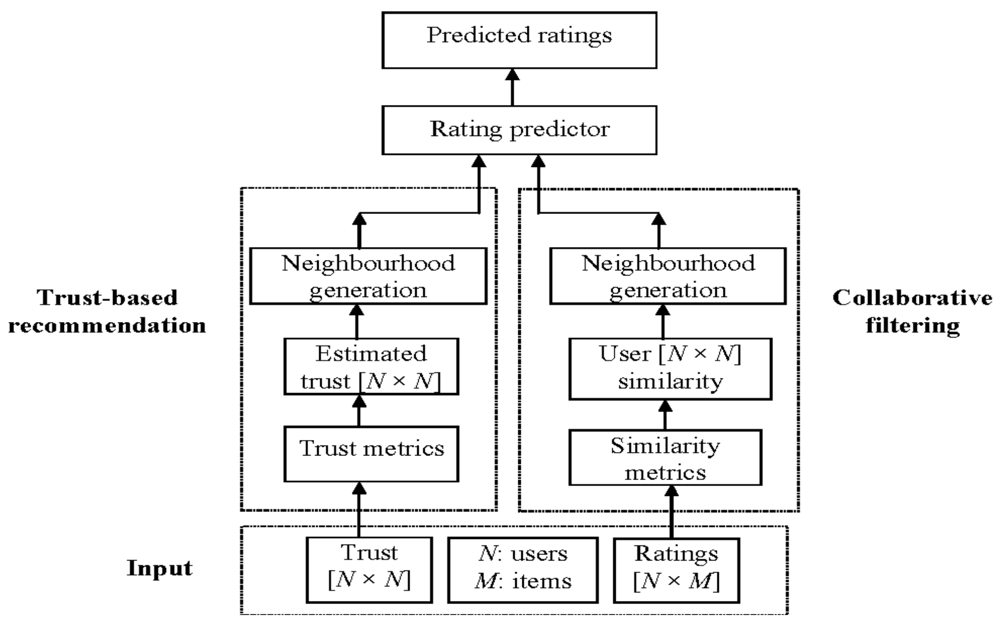

Figure 4: Architecture of Trust aware Recommender systems [41]

2.6 Social Trust Measurement:

When purchasing items, individuals for the most part see others' assessment, and counsel their companions or expert clients. Generally individuals have a tendency to acknowledge recommendations from individuals with cosy connections. In this examination, we endeavour to fabricate a trust estimation display in view of the client's informal organization and present a relating rating expectation calculation. [1]

2.7 Graph Database:

Nodes represent users and edges represent relationship among users. This is suitable database as it considers social relations. This uses a flexible graph model which can be scaled on various machines. Similar to these kind of applications are master data management, cloud and network management, security and access control, geospatial, social networks and recommendations and bioinformatics. Predominantly relational databases are not preferred for storing social relationships data as relational databases can be very complex because of queries, as it would require various joins therefore not preferred for deep traversing of relations.

One of the primary recommender systems was created by GroupsLens, an exploration researcher's group from the branch of software engineering and building at the University of Minnesota. Clients were requested to give a rating; the system additionally followed their other discernible activities. GroupLens at that point predicts future evaluations of news articles in light of the instinct that individuals who concurred in the past will most likely concur once more: if Alice and Bob both evaluated a ton of news articles so also before, at that point maybe Alice will like the following article that Bob enjoys too. Recommendations would then be able to be made in view of anticipated evaluations. Other recommender systems' improvement taken after for different spaces like for music recommendations Ringo was made, for recommending movies Bellcore was made and Jester was made to recommend jokes. [42]

He and Chu demonstrated that using influence of users can actually give better recommendation results than using only recommendation system which provides ratings based on similarity of users. Which has been proven on the dataset collected from yelp.com under an experiment. Upon conducting experiments trust network recommenders have shown its predominance over simple similarity based recommender system. [7].

Massa proposed a technique in which trust was measured and was taken into account for social recommendations which has shown the results better than considering similarity based recommendations [4]. Kim has also analysed various existing trust aware recommendation techniques and formulated their superiority in recommender systems [5]. If trust is used properly it can improve the results as user will get more accurate recommendations based on their own explicitly mentioned followings.

Influence of user on another user is also considered as an imperative part on directed social networks like microblogging websites [14, 15], which additionally influences clients preference while users buy any product or just assess an item. In study it has been proven that ratings on any new product given by influencers encourage other people whether rating is optimistic or pessimistic. Through this study it can be analysed that influence of users also affects the performance of recommendations in a positive manner.

Golbeck [8] demonstrated a new technique in which it aggregated all the ratings given by user's trusted neighbours in which trust is calculated in breadth first search style. He proposed this approach as TidalTrust. On the other hand GuoGuibing [9] proposed a technique in which user's rating profile is complemented by integrating ratings of trusted users by which effective recommendations are generated.

Table 2: Correlations of different existing strategies

Authors research	Prediction accuracy	Sparsity	Scalability	Social trust	Cold start
Than	✓	✓	✓	✓	×
Ma	✓	✓	✓	×	×
Hu	✓	×	×	✓	✓
GuoGuibing	✓	✓	×	×	✓
Tang	✓	×	✓	✓	×
Neto	✓	×	×	×	✓
Proposed system	✓	✓	✓	✓	✓

Robin burke compared various recommendation techniques and stated that all recommenders have some strengths and weakness [23]. By comparing various techniques

Below mentioned the comparisons of various research techniques in Table 3. Comparisons of various reputed authors and their research work is done and the outcomes are analysed and understood properly. This section helps in understanding the basics of recommendations and their needs and how those need are fulfilled previously by various researchers and how those can be improved.

Table 3: Comparison of various existing techniques

S.No	Author Name	Title	Year	Technique
1.	Mohamed Jemni and Mohamed Koutheaïr Khribi [17]	Hybrid Recommender System for E-Learning Personalization Based on Web Usage Mining Techniques and Information Retrieval	2012	User profiling and liberating un-necessary entities, association rules and collaborative filtering to recommend.
2.	Lamia Berkani1 and Omar Nouali[18]	Recommendation of learning resources and users using an aggregation-based approach	2014	Multilevel filtering on three basis: Content based, collaborative based and semantic based and then aggregated results.
3.	John K. Tarus and Zhendong Niu [19]	Hybrid recommender system for e-learning based on context awareness and sequential pattern mining	2015	Context awareness, Collaborative filtering and sequential pattern recognition was used to recommend.
4.	Ahmad A. Kardan, Solmaz Abbaspour and Fatemeh Hendijanifard [20]	A conceptual Hybrid Recommender System for E-learning Environments	2015	Collaborative tagging and conceptual maps

5.	Jia Zhou and Tiejian Luo[21]	A Novice Approach to Solve the Sparsity Problem in Collaborative Filtering	2010	Multiple imputations were used to solve sparsity, collaborative filtering and content filtering to recommend
6.	Jiliang Tang and Xia Hu[22]	Exploiting Local and Global Social Context for Recommendation	2013	LOCABAL was made in which local and global trust of surroundings was collected and then recommendations were made.
7	Robin Burke[23]	Hybrid Recommender Systems: Survey and Experiments	2003	Knowledge based, Demographic, CF, CB, Utility based were merged for hybridization with various styles like Switching, weighted, feature-combination, cascade and meta level
8.	Paolo Massa and Paolo Avesani [24]	Trust-aware Recommender Systems	2007	MoleTrust to compute local trust and PageRank to compute global trust simultaneously with user item ratings.
9.	Hao Ma, Tom Chao Zhou Michael, R. Lyu and Irwin King [25]	Improving Recommender Systems by Incorporating Social Contextual Information	2011	Social trust network, tags issued by users, tags associated with items and latent feature space is computed for social tags.

10.	Ziqi Wang, Yuwei Tan, Ming Zhang [26]	Graph-based Recommendation on Social Networks	2010	Tags and ratings were analysed in social networks and co tagging is considered to add similarity in relationships.
11.	Wang, Youwei, Weihui Dai, and Yufei Yuan [27]	Website browsing aid: A navigation graph-based recommendation system	2008	Navigation patterns of previous websites is considered to check the flow of users.
12.	Jan Škrášek[28]	Social Network Recommendation using Graph Databases	2015	Graphs are analysed and nodes are modified with user's interest and user's friend interest.

3.1 Complications and Challenges

In this section various issues and challenges are discussed which occurs in recommender system.

3.1.1 Sparsity:

This is another big issue which arises in social networks is sparsity of data. The accessibility of immense scope of information about the things of the inventory and the unwillingness of clients to rate various things which further makes a scattered profile in matrix further prompting fewer precise recommendations. To retrieve recommendation accessibility to matrix is required. Only few or none entries in matrix makes computationally hard for any filtering technique to generate accurate recommendations. Such matrix is known as sparse matrix.

3.1.2 Cold-Start Problem:

This issue arrives when user is fresh or new to the current recommender system, so when users are new to recommender system or to any website on which he is seeking suggestions, previously that user would not have rated any item so it will be impossible to find similar users. Thus this problem is called cold start problem, which is when a user is new to recommender system and he has not yet made any ratings entry for his choice of likes or dislikes through which the recommender system can infer some suggestions for that new user. In this kind of scenarios neither the choice of new user can be anticipated nor can the fresh brand new items be purchased or rated which leads to inaccurate recommendations [46] .This problem can be solved in some methods like if we probe all new users to rate some items according to his choice or asking new user to bluntly state their choice to aggregate the ratings accordingly. We can also use demographic location of user to know the state of his choice with respect to his surroundings. But all these solutions are not feasible in ecommerce website. New user would not like to waste time on explicitly defining his choice or to rate various items.

3.1.3 Latency Problem:

When any new-fangled piece is added to database that item is not recommended to anyone as no one has rated that item before and only items which rated repeatedly are shown in recommendations. This happens in collaborative filtering as it shows suggestions according to the rating provided by users to items. [38]

3.1.4 Grey Sheep:

This problem occurs in pure collaborative filtering where the person's choice is matching with no cluster and hence unable to receive benefit of recommendations. In this problem there are some people whose choice does not match with anyone and they find recommendations inaccurate. These user's whose choice is not matched with anyone are called grey sheep users. If we join collaborative with content based then this problem may be mitigated. [39]

3.1.5 Synonymy:

This is another major drawback of recommender system which is when one word has two different meanings or when an item is entitled with two non-similar names. Let's consider an instance, a collaborative filter which is based on memory will treat 'horror movie' and 'horror film' differently. As it saves the description of an item and variation of using synonym will degrade the performance of recommender system by getting confused between synonyms and not generating accurate recommendations. Since the item's content is not thoroughly considered so the latent link between items is not appraised and therefore new items are not recommended to anyone. [40]

3.2 Research Gaps:

1. There are a lot of approaches that have been proposed for recommender but still there is astute need of accurate recommendations for new users. When user is new and system has no knowledge of user's choice and previous history, then it is difficult to for existing recommenders to create recommendations.

2. There is a requirement of such recommender so that user can personally choose which particular person's recommendations he wants to see.
3. Typically recommender system generate ratings or suggestions based on user's similarity measure or items. There is need to consider more factors while generating recommendations.
4. Computing proposals of suggestions in recommender utilizing graphs is like the normal recommender frameworks approach. Sparsity problem also occurs in graphs when there are not enough relations to compute recommendations.

3.3 Objectives:

In the light of above discussed research gaps following objectives have been formulated.

1. To study and explore various recommendation tools and techniques available to generate predictions.
2. To compare various existing recommendation techniques and with pursue with the best technique to generate accurate recommendations.
3. To remove numerous redundancies in existing techniques and generate an efficient recommender.
4. To consider explicit trust of users apart from the traditional rating criteria to provide recommendations.

Recommendation Systems check the client's inclinations for proposing components to purchase or browse. They have turned out to be an essential applications in internet business and access to data that gives proposals that successfully diminish extensive data to the things that best address user's issues and inclinations. Still sometimes we face above mentioned problems and we do not get accurate recommendations. We have proposed an extra favourable position of these systems which is clients can encode more data about their relations than essentially say who they trust. Along with trust we have also proposed a transitivity in social networks through which we can get more accurate recommendations.

4.1 Basic Framework:

Recommender systems have been studied widely and different types of recommender systems have been came to existence. In previous recommender systems, algorithm used to take user as an independent user like it does not have any relation with any another user, but when social recommender cam to existence it did not think user as an independent entity, in fact it extracted social relations between users with the help of various algorithms. Various flaws were analysed in traditional recommenders in which users were taken as independent so social recommendation overcame this disadvantage. Website guests for the most part inspect site pages for very extraordinary reasons and under an assortment of settings.

According to this, visitors guest are very different from each other. Nonetheless, they do show numerous comparative interests inside specific gatherings of clients. When they visit similar websites with comparable targets, individuals from a similar gathering show some sort of similar behaviours. Along these lines, if the gathering participation can be distinguished by a few methods, fruitful proposal for recommendations can be made by referencing the conduct of individuals inside a similar network. When website is being browsed, important data for recognizing diverse users is covered in the log documents, where hints of clients are recorded.

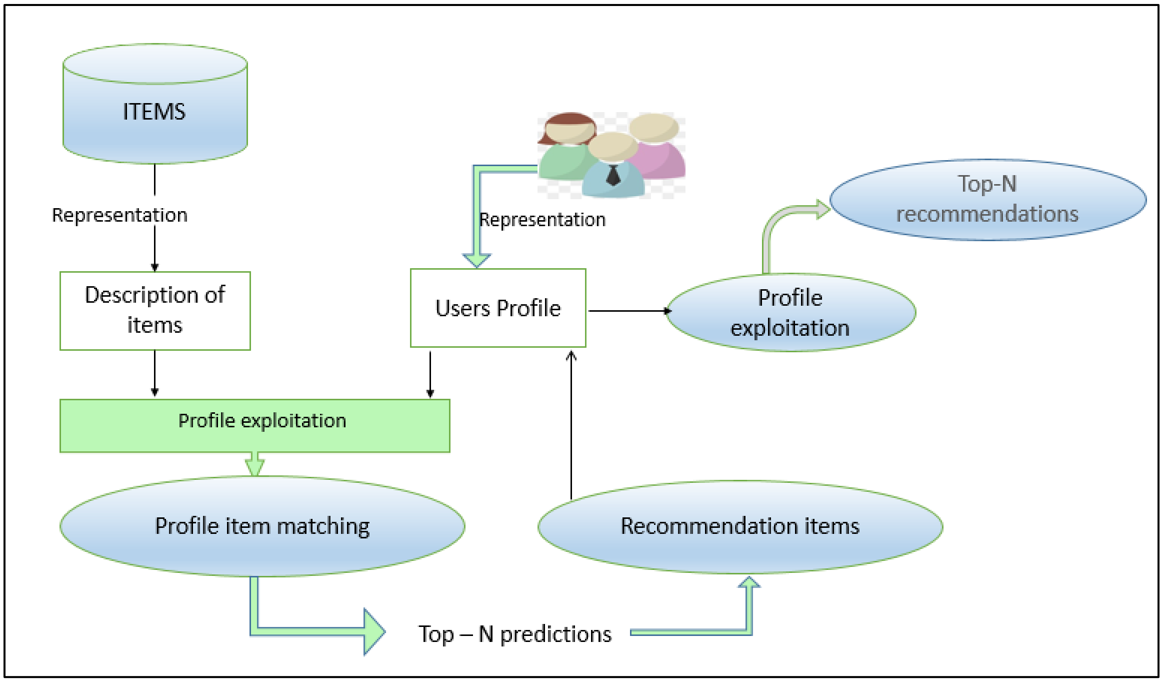

Figure 5: Generating Recommendations

Previously two types of filtering techniques were used for recommendations which were collaborative based filtering and content based filtering. Figure 5 shows how top-n recommendations are made. In content centred technique as name describes, clients used to get recommendation based on their previous history or what they bought or viewed in past?' users used to get choices based on their own behaviour only. The basic drawback of this filtering technique is privacy of users and independency of users. Another approach apart from content based filtering is called as collaborative based filtering, in which similarity of various users is computed like if one user have rated any item and any another user have also rated same rating to that product or item then according to this approach they both are similar and their choices are similar to each other. So if in future one user has rated any item positively that another user of similar choice will get recommendation of that product. Again the drawback of this approach was users were considered as an independent entity. Details are encoded in different types.

In user item matrix details of rating given by diversified users to items are specified. Generally we do not have much entries in this dataset as it is extensively big and user specifies only few ratings to some products of his interest only as user will not rate each and every item. At the point when dataset does not have much entries, the subsequent matrix is a scanty one because of less number of entries of ratings in the dataset [36]. Another problem which arrives is when user is new, so when users are new to recommender system or to any website on which he is seeking suggestions, previously that user would not have rated any item so it will be impossible to find similar users. Consequently this matter is called cold-start difficulty, which is when a user is new to recommender coordination and he has not yet made any ratings entry for his choice of likes or dislikes through which the recommender system can infer some suggestions for that new user. Our proposed technique in which hyperedge is used for influence among users control both of the existing problems which are sparsity of data in dataset and cold start problem. However apart from sparsity and cold start one another problem which traditional recommender faces is shortage of trust information as well as social relationships among users. So proposed technique ensures about mitigating this problem also. In proposed technique trust is used as an extra favourable feature for generating relevant suggestions and improving the sparsity of matrices.

4.2 Trust:

Trust is a general idea that can be connected to any specific situation. It assumes a vital part in a few trains, for example, humanism, brain research, PC sciences, recommender frameworks, and so on.

Psychology defined trust as “the individual is gone up against with a vague way, a way that can prompt an occasion apparent to be helpful which is considered as Va+ or to an occasion apparent to be destructive which is considered as Va-”.

On the other hand sociology defines trust as “a wager or assurance about the future unexpected activities of the trustee. This wager, or desire, is thought to be trust just on the off chance that it has some result upon the activity of the individual who makes the wager who is trustor”.

In computer sciences Goldbeck [8] simplifies trust as “a devotion of an entity to believe in the untroubled visit of the future actions of a different entity.”

4.2.1 Properties of Trust:

1. Trust is asymmetric *i.e* if user A trusts user B, that does not implies that user B also trusts A.
2. Trust is not distributive *i.e* if user A trusts user B and C together, that does not implies user A will trust them individually too.
3. Trust is not generic, which means if user trusts another person in computer science that does not mean user will trust him in medical field also.
4. Property of transitivity is uncertain. Some little works does not consider trust as transitive but majority of work considers that trust is transitive accordance with Josan and Pope [52]. Some researchers also say that trust is transitive and that transitivity requires some major constraints to be taken care of [53].

4.2.2 Trust Contextualization:

MoleTrust:

MoleTrust is a trust display proposed by Massa and Bhattacharjee [54] . In this model, Trust is characterized in unique way by binary number. The trust proliferation is an essential property in MoleTrust. It is expected that trust degree amongst X and Z isn't a similar trust degree amongst X and Y. The spread of Trust is communicated by transitivity. Truth be told, when client X trusts client Y, and Y trusts client Z, at that point X trust Z. Keeping in mind the end goal to settle the trust proliferation, creators characterized a spread having 4 as most extreme separation between two clients. MoleTrust predicts the trust an incentive between to clients X and Z by utilizing the accompanying recipe: Where d is the maximal separation of proliferation and n is the separation amongst X and Z. Utilizing transitivity,

$$trust = \begin{cases} (\mathrm{d} - \mathrm{n} + 1)/\mathrm{d}, & n \leq d \\ 0\,, & n > d \end{cases}$$

n = 2 on the grounds that, there is just a single middle of the road amongst X and Z.

While existing a few ways amongst X and Z, MoleTrust takes the most limited way and thinks of it as the best one which expands the trust esteem. Contrasted with shared separating, MoleTrust demonstrated his execution as far as suggestion precision. Be that as it may, the decision of the briefest way doesn't generally ensure the best execution. Since, we can locate a more extended way with more information about the objective.

TidalTrust:

Research proposed the model TidalTrust for informal organizations. It is devoted to prescribe films to clients. In this model, every client can assess films with rating of 1 to 5 stars. He can likewise assess his trust to another client with a size of discrete qualities in [1, 10].

Contrasted with MoleTrust, Massa and Bhattacharjee [54], in which the trust is represented as binary number, TidalTrust enables the client to express a steady trust to different clients. Nonetheless, this model didn't consider the trust contextualization. Likewise, it didn't consider neither the length of way nor the different conceivable ways.

Another trending recommendation technique is trust aware recommendation systems. Trust aware recommender frameworks are intelligent applications that make utilization of trust data which is one person trusting on another person explicitly and client individual information in social systems to give customized suggestions. Prior research in trust-aware frameworks have appeared that the capacity of trust-based frameworks to make exact forecasts combined with their vigour from shilling attacks make them a better option than traditional recommender frameworks. We have formulated a technique in which we use trust data and manipulate the graph with the influence i.e. one person's impact on another person's choice. The graph which is manipulated is specified as Influence graph in the whole work. Influence graph expands social trust an incentive between associated clients and this impact in recommending items in a compelling and proficient way with the idea of new concept in recommending which is transitivity. Trust is explicitly mentioned by the users on whom they have belief in. Generally, we compare the similarity of users to provide them the accurate and reliable recommendations. To ascertain how much users are similar to each another, Pearson correlation can be calculated.

$$P_c(U_{\alpha}, U_{\beta}) = \frac{\sum_{p=1}^{n}\left(\overline{G(\alpha)p} - G(\alpha)\right)\left(\overline{G(\beta)p} - G(\beta)\right)}{\sqrt{\sum_{p=1}^{n}(\overline{G(\alpha)}_p - G(\alpha))^2 \sum_{p=1}^{n}(\overline{G(\beta)}_p - G(\beta))^2}} \quad (1)$$

Here, value of correlation calculated by Pearson formula varies from -1 to 1.

G(α) is final mean of all the ratings given by any particular user *α,*

$G(\alpha)_p$ is rating which user α gave for any random product p,

$G(\beta)_p$ is rating which user β gave for any random product p,

*G(*β*)* is final mean of all the ratings given by any particular user β. Here similarity is checked for all products between 1…n among users U_α and U_β.

4.3 Influence Graph:

Let’s take an example, where let’s say one user p wants to buy a book, while looking for book he/she can get influenced by his trusted users and he may take their choices into account to buy a book. But what if that user has only one or two trusted neighbours and those trusted users might not have rated any book in past as they do not have any knowledge about books. So it will be impossible for the user p to get any suggestion in terms of book. So to user p for trust propagation the opinions or suggestions from his own trusted users are valuable [1, 6]. So it is worth to consider the ratings and opinions from some other users as well. But how can we connect another users to each other as they have shown their interest in only few users. So algorithm for transitivity was proposed to connect more users with each other. In such manner, a trust-aware recommender system that core interests a lot on trust (as opposed to rating) utility is likely to accomplish just minor picks up in proposal execution. Truth be told, the current trust-based models think about as it were the explicit influence of ratings. That is, the utility of ratings isn't very much exploited. We have implemented this transitivity approach through graph and called the end result graph as Influenced Graph (IG). In graph, nodes represent user and edges represent the relationship between users which are filled correspondingly in user-user matrix. In user-user matrix there are entries of users who have shown their trust on another user, so their value will be filled as 1 otherwise 0. In user-item matrix the entries will be of user’s rating about that particular item and value of rating will be shown. This IG improves the density of user-user matrix which is number of users who trust each other. Improving entries in matrix resolves the issue of sparsity and hence improves recommendations.

Recommendation systems consists of two types of system: System based on user-item matrix and system based on user-user interaction matrix which is trust matrix. Drawbacks of ‘only user-item’ matrix is that it does not consider social relations among users whereas drawbacks of ‘user –user’ matrix is that it is very sparse and only few entries are there but it considers social relations among users as user explicitly mention their trust on one another which makes system easy to recommend user accurately. IG considers both ‘user-item’ as well as ‘user-user’. Proposed technique overcomes both major problems of recommender system: cold start and sparsity. We have achieved sufficient results which prove this approach of transitivity between nodes is helpful for better recommendations. Experimental results show through MAE and RMSE that recommendations have been improved after this proposed approach.

4.4 Architecture:

In our proposed approach as shown in Fig 2. We have used Neo4j to manipulate social graphs. We have analysed our results or final recommendations by keeping some testing data separate to compare results with the manipulated results. There are various other graph databases also like another famous one is SNAP. Neo4j is more convenient and it is therefore used in proposed approach to manipulate graphs and extract important information from them. Neo4j can easily handle manipulation of graphs.

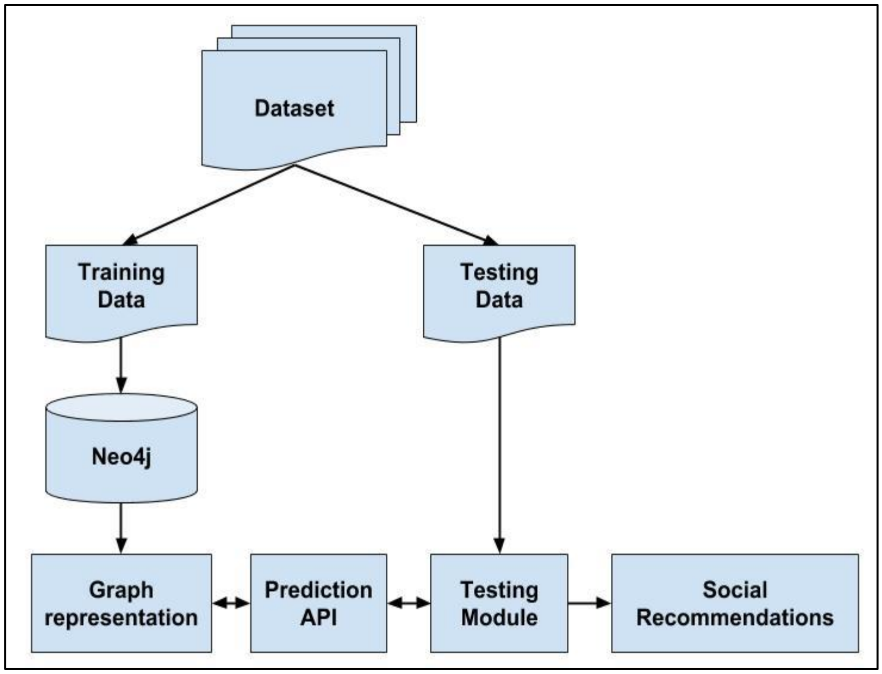

Figure 6: Flowchart of proposed technique on Neo4j

As shown in figure 6 Dataset is divided into two parts as training dataset and testing dataset to apply proposed algorithm and then compare it with testing dataset.

4.5 Graph Database:

Graph is data structure which is combination of edges and nodes. The graph database is an effective tool for handling relationships between entities of data model. Graph database also known as GDB which directly relates items in the store. Graph database handles the semantic queries with nodes and edges and their various properties. Graph database allows to store the direct relation among nodes and often retrieved with any operation. Graph database is a simple by design and allows the fast retrieval for complex hierarchal relations. Demonstrating objects and the connections amid those connections imply nearly no matter which can be spoken to in a relating graph. Nodes are those which have some attributes or properties and are sorted out through connections which correspondingly likewise have properties. A traversal explores a graph and recognizes ways which arrange nodes. A graph is a protest which contains nodes and connections.

A typical graph type upheld by most frameworks is the property graph. The term property graphs are further classified as credited, named and coordinated multi-graphs. Figure 7 explains the concept of graph, various entities and interaction among people and objects. The graph database is improved for the effective handling of very dense or complex interrelated datasets. An advantage to the multi graph is that it is the most complex usage in light of the fact that each other sort of graph comprises of subsets of the property graph execution.

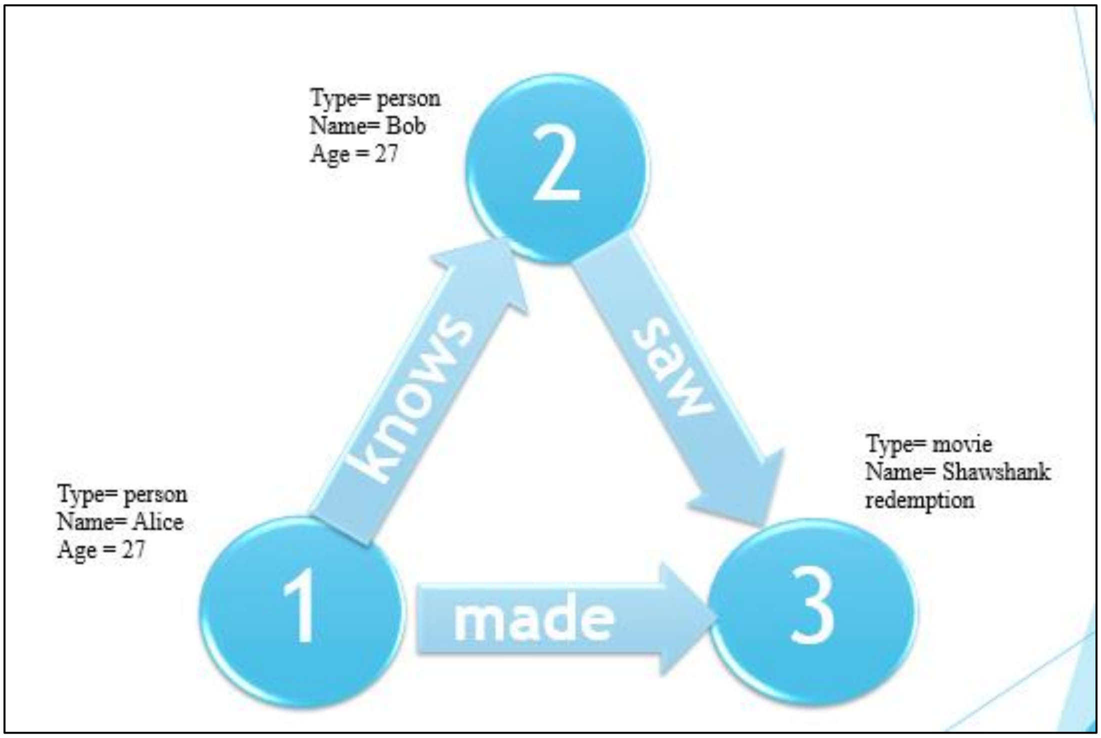

Figure 7: Visualization of graph with nodes and edges

This exceptionally powerful information display in which all hubs are associated by relations takes into consideration for the quick traversals beside the edges amongst vertices. A specific advantage is the way that traversals are confined plus they do not need to consider collections of random information.

4.6 Graph Storage vs. Graph Processing:

Despite the fact that they share an information show with the GDB (graph database) , graph processing structures are intended to settle an alternate kind of issue. This refinement of association of and online transaction processing (OLTP) and online analytical processing

(OLAP) is qualified. Frameworks of OLAP are like Apache Hama or one another famous framework is Google's Pregel, which further endeavour to quickly answer multidimensional analytical questions concentrating on elite processing of queries. Then again, the objective of OLTP frameworks, as Neo4j, is to encourage and oversee transaction situated inquiries. Pregel centres totally around the proficient dealing with various substantial graphs. This auxiliary make the possibility of utilization of graph built calculations which encourages the exceedingly equivalent tackling of issues identical to page-rank, bipartite coordinating plus one common greedy problem known as shortest path.

Interestingly, Neo4j endeavours to be the information backend for transaction based applications. Graph databases truly sparkle when working in regions where data about information interconnectivity or topology is essential. In such applications the relations amongst information and the information itself are as a rule at a similar level.

4.7 Social Graph:

The social graph use data over a scope of remembering the true objective to assess the associations between individuals. In customary social databases it is very testing to manage recursive information structures where it is considered that join is when each edge in the graph is traversed. Joins are costly on a RDBMS while a traversal over an edge costs practically nothing.

The social chart or graph use data over a scope of systems keeping in concern about the end objective to measure the connections between people. Figure 8 speaks to a little social diagram amongst companions and tracks the friendships, name, age, and most loved motion picture of every individual. Figure 8 gives a reasonable answer for finding the companions of Alice's companions. In customary old social databases it is very testing to manage recursive information structures where every traversal along an edge in the diagram would be a join. The companion of a companion issue is a case of an issue that would require far less work for a chart database when contrasted with RDBMS. Joins are considered costly on a relational DBMS although a traversal over an edge are practically nothing at cost.

The language for insertion and traversal is yet to be finalised or to be standardised by the world of graph databases. This absence of institutionalization has prompted inconceivably extraordinary usage and systems for information communication. The usage that are accessible range from the REST interface, Neo4j's Java API, DSL dialect named Gremlin and another called Cipher. Demon executes traversals through a framework called piping.

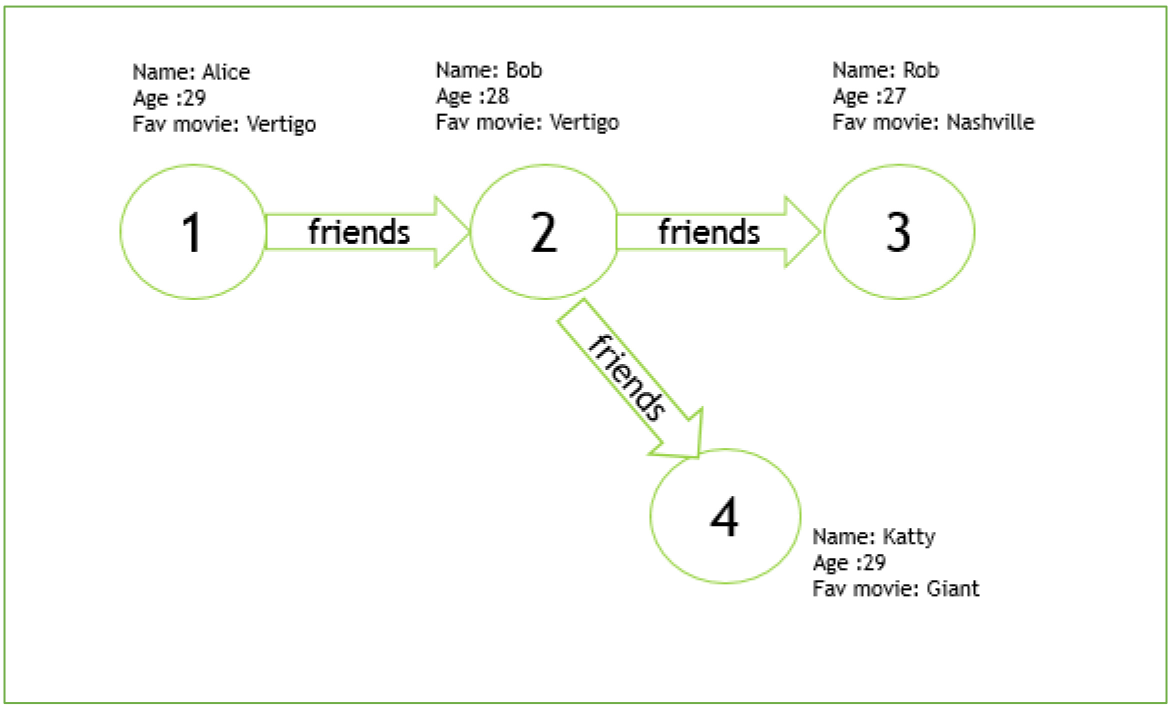

Figure 8: Relations among nodes in social graphs

Each piece of the query is a step to the next in a progressive fashion. However, there are various databases which attempts to be like SQL by utilising more keywords in system. One of those databases is Cypher [51]. There is an absence of consistency that expects one to take in all executions previously understanding what approach is most appropriate to the issue. Graph databases have realized another method for displaying and navigating interconnected information that is unparalleled in data stockpiling. With the approach of creation review frameworks, for example, Neo4j utilizing GDB issues can be tended to without falling back on a restricting execution on a RDBMS.

The GDB also known as graph database has a characteristic application for organic, semantic, organize and recommendation frameworks which have need of the kind of information demonstrate no one but they can offer. Neo4j is the most attractive and best fitted to the proposed algorithm where we need to manipulate our graph various times.

5.1 Neo4j:

It is a graph database server which is very famous and it is Cypher Query Language (CQL) which is written in one of the famous language java. There are various other graph databases also like another famous one is SNAP. Neo4j is more convenient and it is therefore used in proposed approach to manipulate graphs and extract important information from them. To use neo4j basics off neo4j should be learned like Spring DATA with neo4j, Neo4j CQL and its functions, etc.

This figure 9 shows how Neo4j graph database shows the connection between nodes and make clusters of similar nodes [47]. In our approach edges will represent relations and transitivity and nodes will represent users and items with different colors.

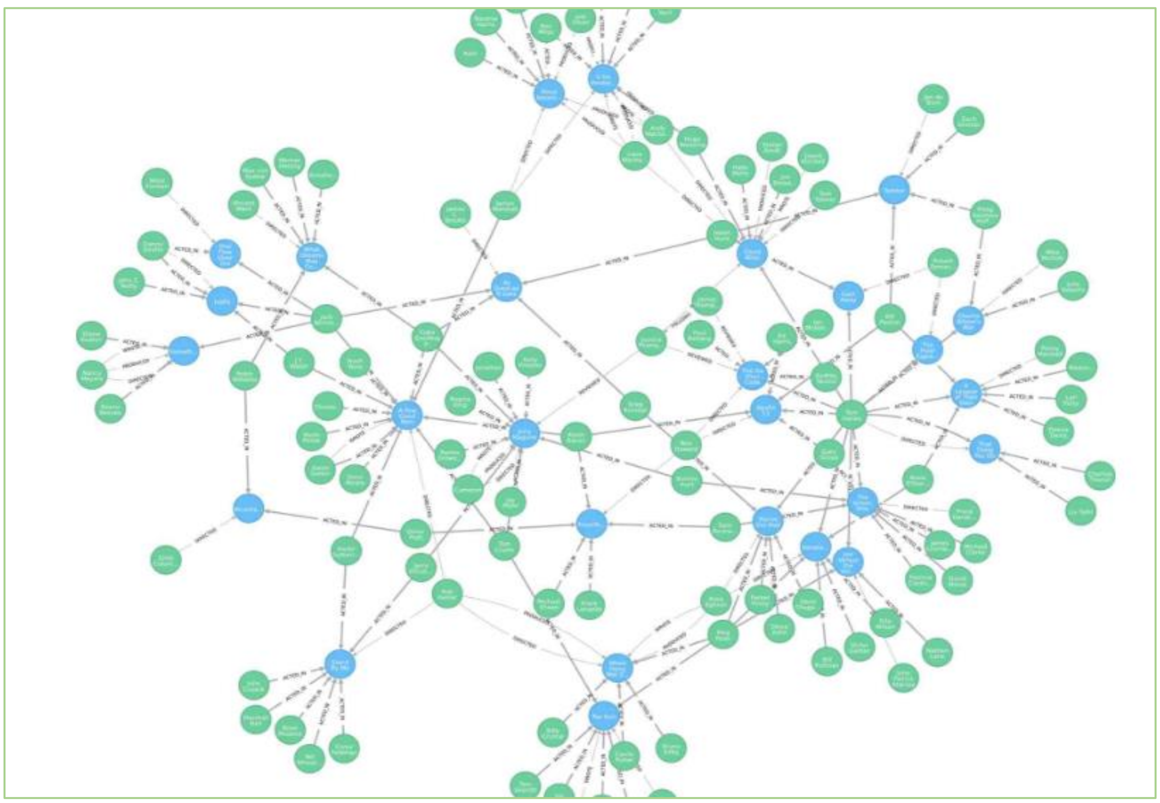

Figure 9: Representation of various nodes and connections among them in Neo4j

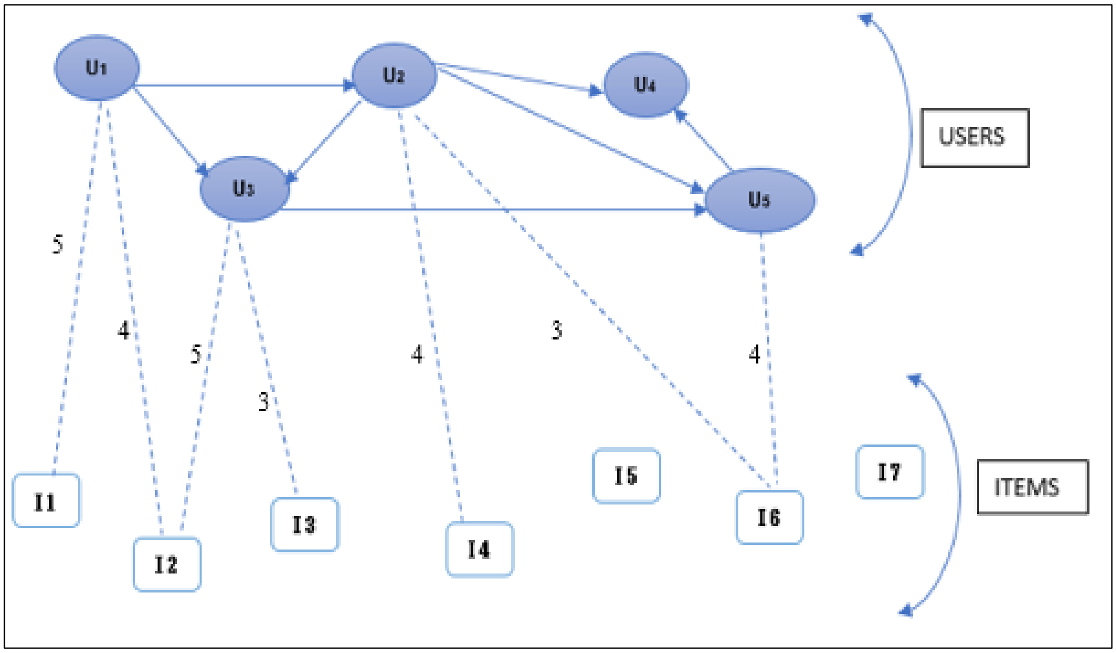

Figure 10: Connections among users and item's rating in dataset

To recommend anything socially there should be some connection between users and item's rating in dataset. There should be coordination among users and items in dataset to apply algorithm to recommend anything. As can be seen from figure 10, relationships among users and items in a socially connected network. Every blue node knows as user summons a few things and rates them. The coordinated strong line in figure 10 speaks to the accompanying connection among pair of users and on the other hand the dabbed line shows the rating connection amongst clients and products. The integer with the specked line shows the rating value given by the corresponding user for a specific item. At the point when any new or old user asks for recommendations, then recommendation calculations anticipate the already provided ratings to items by investigating the informal community and user-item rating information, at that point it further recommend items with high anticipated ratings.

Hyperedge is concept of transitivity in graphs. That transitive closure is recorded as social influence. With the idea of hyperedge social influence is manipulated in social graphs. We determine the nodes of a graph their associativity. A hyperedge associates different adjoining nodes in a graph with the help of transitivity [13]. Every node has some influence on some other node either in direct or indirect way at some extent in social graphs. This influence in social network changes the alignment of leadership of a user because of the influence of quick neighbours [14]. Based on this influence user can provide or get recommendation in this social networks with the usage of graphs.

Previously for social recommendations only neighbour nodes which are immediate to that user were analysed as an influence for providing recommendations. Here in our approach we have seen the connection of users through influence of each other with the help of transitivity which means if user A trusts user B and user B further trusts user C then there would be a connection between user A and C also. So user A will trust user C with the concept of hyperedge. In graphs transitive closure is the binary relationship to identify the reachability of nodes in social graphs. [11]

Rather than just considering similarity in collaborative filtering it would be better to consider social trust and influence as well. It will work as ice on cake if we consider all the combinations to recommend items to users. In mathematical equation users are in transitive closure if a relation R_n is defined on $a \; R_n \; b$ and $b \; R_n \; c$ then there is a relation among $a \; R_n \; c$. Matrix is among one of the forms in which this can be representated which contains information about the number of hops to reach from a to c [11]. To find the connectivity among users in transitive closure Warshall algorithm is used in this paper.

5.2 Py2Neo*:*

This is the driver which is used to connect neo4j and python. This is imported in the beginning of code as all the built in libraries are required in code which are imported through this driver. The library wraps the official driver including support for HTTP, a larger amount API, an OGM, administrator devices, an intuitive console, a Cipher lexer for Pygments and numerous different fancy odds and ends. Not at all like past discharges, Py2neo v4 never again requires HTTP and can work completely through Bolt. The py2neo.database is a package which contains classes and various functions which are required to connect with the server of Neo4j. The most imperative of these is the Graph class which speaks to a Neo4j diagram database example and gives access to an expansive bit of the most ordinarily utilized py2neo API. One interesting parameter of Neo4j is it only uses one graph per database [48] .The Graph class speaks to the diagram information storage room inside a Neo4j chart database. Association points of interest are given utilizing URIs as well as individual settings. Command through which driver connects neo4j with python is:

*from py2neo import **

Some of the other commands which are used in Neo4j are:

Database_name: This command returns the name of current database which neo4j is operating on.

Config: This command returns the parameters used in configuration of neo4j.

Default_graph: With this command the default graph appears in neo4j.

Uri: This returns the URI to which database is connected, etc.

5.3 Dataset:

We have used FilmTrust dataset (www.librec.net/datasets.html) where we have record of users with trust values. Filmtrust is dataset crawled from website of movies where ratings were specified as well for films. This data was collected in June 2011. Range of trusts in Filmtrust varies from 0.5-5.0. Table 4. Shows description of dataset.

Table 4: Dataset Description

Dataset	***Statistics***
Users	1508
Items	2071
Ratings	35,497
Density	1.14%

Social	***Statistics Dataset***
Trusters	609
Trustees	732
Trusts	1853

Format of dataset entries is:

1. Item Ratings (ratings.txt): [user-id, item-id, rating- value]

2. Trust Ratings (trust.txt): [user-id (trustor), user-id (trustee), trust-value]

5.4 Algorithm:

Input: User's set, item's set and user- user, user item matrix

Output: Improved social trust and recommendation.

Algorithm IG (U_i, I_j, K_{ij})

1. Initialize social trust relations with the help of user-item matrix and entries in user-user trust matrix.

2. Check relations of trusted users.

Say K= (k_{ij}) is a social trust matrix of users u_i and u_j. Value of k_{ij} is w_{ij} (weight or trust between users u_i and u_j) if there is trust between users otherwise m_{ij} is 0.

$$\begin{cases} k_{ij} = w_{ij} \text{ if } u_i \text{ have trust on } u_j \\ 0 \quad \text{otherwise} \end{cases}$$

3. Create directed edge graph G= {V, E} from matrix k_{ij}, where V is set of vertices and E is set of edges. Vertex v_i is connected to e_{ij} by an edge having weight w_{ij}.

4. Calculate MAE and RMSE

$$M_{AE} = \sum_{i=0}^{n}(P(u, i) - p(u, i)) / n$$

$$R_{MSE} = \sqrt{\sum^{n}(P(u, i) - p(u, i))2 / n}$$

$i = 0$

5. Calculate transitive closure of modified user-item matrix

$$M_m = M^1 \cup M^2 \cup \ldots \cup M^n$$

6. Manipulate graph after using concept of influence with hyperedge theory proposed from user-item recommendation.

Modified graph $G_m = (V_m, E_m)$ use modified social trust matrix M_m.

7. Using Neo4j library, create graph Gm from the proposed technique.

8. Calculate modified MAE and RMSE.

9. Compare existing technique metrics with proposed technique metrics.

10. Exit

This algorithm is checked on command prompt. To operate the code from this algorithm we have to first start the Neo4j service through administrator. After administrator allows the Neo4j service then the code is executed. The code is written in python and hence python driver is utilised to connect to neo4j database.

Two functions were computed first during implementation of proposed technique which were *GetFriends* and *GetRatings.* One attribute named Clear visuals is set to true in order to delete old visuals which are made in Neo4j database from previous execution of algorithm. One another attribute named GenerateVisuals is set to true in order to view the final output in form of graph that how nodes are connected to each other. One default prediction is given when someone has given no ratings. DefaultRating variable is set to NULL.

In this proposed algorithm first of all social trust relations are initiated from user-item and user-user matrix. After initialization of trust and ratings given by user to movies and relations of users among other users are checked through 'trust.txt' set. In this algorithm social graphs have been used in which they represent the correlation between users and items socially. Social graph is represented as G_r. It is mandatory for graphs to have some connection between nodes so that algorithm may apply to the dataset. When user will have social trust on any another user in graph G_r then its edge value is 1 if users are socially connected and 0 otherwise if they don't have any relationship. We first of all integrate all the ratings and find Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). After implication of hyperedge concept on nodes which have trust with each other, the transitive closure is calculated. Enhanced graph G_m is formed after transitivity between nodes. To check whether the results have been improved or not again MAE and RMSE are analysed for the enhanced graph. In our proposed algorithm it have been proved that using hyperedge concept has reduced the MAE and RMSE which means it is an effective approach.

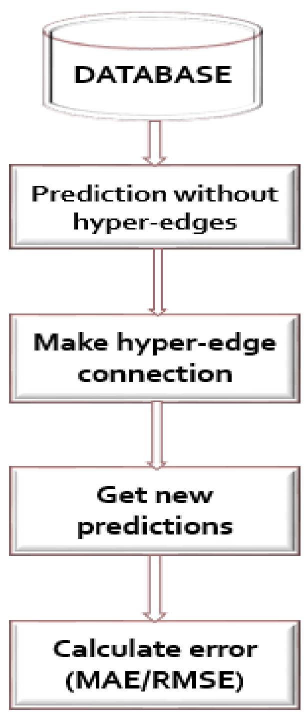

Figure 11: Working of proposed technique.

Figure 11 explains the whole process of algorithm, from database of User item ratings and another database of Trust ratings are given. First prediction without hyper-edges are calculated and then hyper-edges connections are made with the help of transitivity and further predictions are calculated and then compared with old predictions with the help of various metrics like MAE (Mean Absolute Error) and RMSE (Root Mean Square Error). Major drawback of recommender system which was sparsity of data *i.e.* when entries remains scanty in matrix to compute recommendations is eliminated. This problem is eliminated in this approach.

On the other hand one another constraint was there in algorithm which was only those connections were considered who had solid connections with each other. Not all the transitive closures are considered because it will degrade the performance and in some way all of the nodes will get connected to each other which will increase the complexity and will degrade the performance. Transitive closures who had strong connection with each other are considered only. Here it is supposed that near neighbours have more influence than far nodes.

5.5 Experimental Analysis:

We have evaluated our results and we have seen the change in recommendations. Using this approach we get better recommendations than traditional recommendation algorithms. Library which is used Neo4j library is used for analysis of social network represented in the form of graphs. We have modified user–user matrix by using hyperedge and transitive closure on Neo4j.Code was written in Python and Neo4j's python driver was used to connect python to Neo4j database.

Social trust is analysed between immediate neighbours, and using dynamic update of trust based on IPG, we have trained dataset for improving trust. We have evaluated our results through performance metrics like MAE and RMSE. We have selected in random 30% data from FilmTrust dataset for testing and 70% for training dataset to check the results of algorithm. Different evaluations were taken by taking different amount of training and testing dataset.

5.5.1 MAE (Mean Absolute Error)

This is the metric to evaluate the performance of recommender systems. Good recommender system should decrease its value as much as possible. In our approach it has been viewed that the value of MAE is decreased to few integer points. MAE is known as the best computing metric with which we can easily generate results of comparisons. Decreasing MAE means we have successfully reduced the error in our algorithm.

$$\text{MAE} = \sum_{i=0}^{n} (D\ (user, item) - d\ (user, item))\ /n \qquad (2)$$

Mean absolute error is calculated as shown in eq. 2 by the average of difference between *D(user, item)* that is the forecast of values in recommendation algorithm and *d(user, item)* which is actual prediction for n products. Bell has analysed that if we improve MAE to some extent it means we have conquered in obtaining good recommendation algorithm [16].

5.5.2 RMSE (Root Mean Square Error)

This is another metric used to calculate the accuracy of recommendation system or proposed recommender algorithm. In various studies it have been proved that RMSE is better at calculating the errors than MAE in terms of recommender system.

$$\text{RMSE} = \sqrt{\sum_{i=0}^{n} (D\ (user, item) - p\ (user, item))^{2}}\ /n \qquad (3)$$

In equation 3 RMSE is calculated by the square root of mean of square of difference between *D (user, item)* that is the forecast of values of ratings in recommender systems and *d(user, item)* which is actual prediction for n products .

Here in Fig 3. We can see old MAE and modified MAE. We can also notice modification of RMSE in the screenshot shown below of the results which we are getting after implementation of the proposed algorithm. It is clearly visible that MAE and RMSE have been improved by using this influence approach which shows that we get effective recommendations.

To check the changes in visuals in Neo4j, in figure 15 are the nodes which are mostly separated which means users are not much connected to each other and are independent at some extent. In this figure where some nodes are connected to each other are those nodes who have shown their trust on specific node and hence they are connected to each other. Here training size and testing size id divided into 70% and 30 % respectively. So this number of Read rating which is 35497 is divided into 70% and 30% and hence training size is 705 of read rating which comes as 24847 and testing data which is 30% of read rating comes to be 10650.

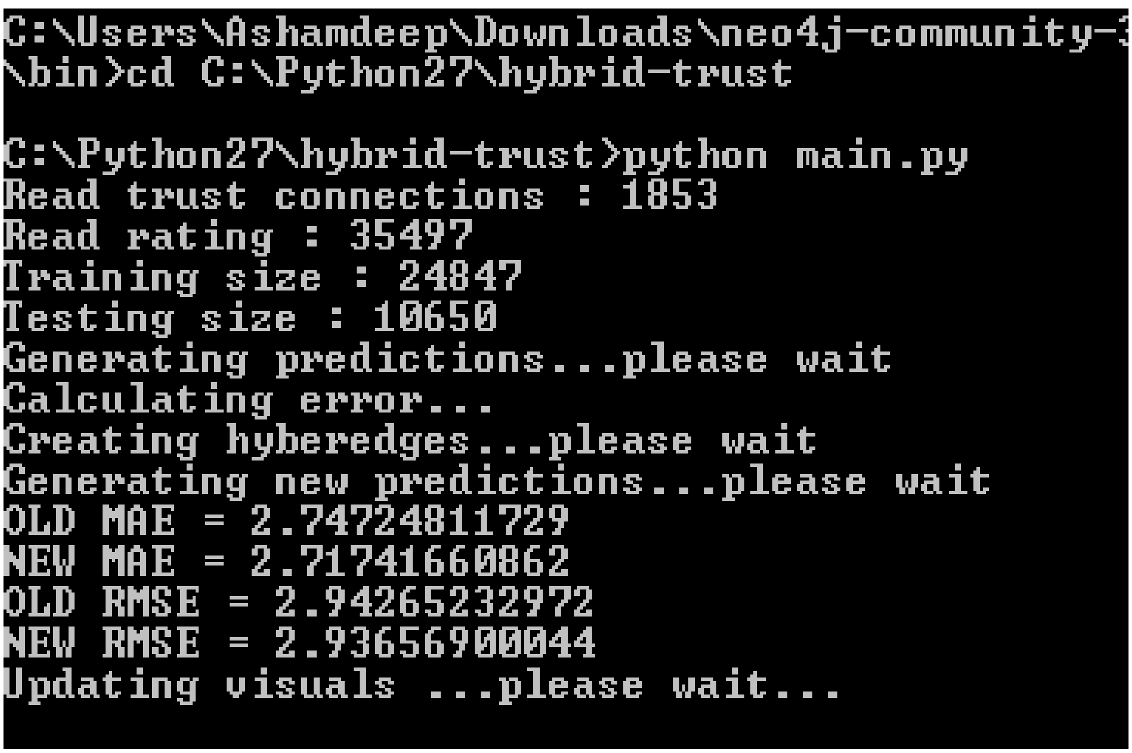

Figure 12: Screenshot of old MAE, RMSE and modified MAE, RMSE

This data is divided in order to compare the original result with final result which comes after application of this proposed algorithm.

In Fig 12 shown is screenshot of front page which opens up when we open Neo4j server after implementing algorithm. On top left corner is differentiation of nodes and relationship types. In node labels there comes three options as: *, items, users. Through this we can specify whether we want to see nodes of items only or users only or of both which is represented as *.

The blue nodes which are being shown are users here and edges between them is showing the relation among those users.

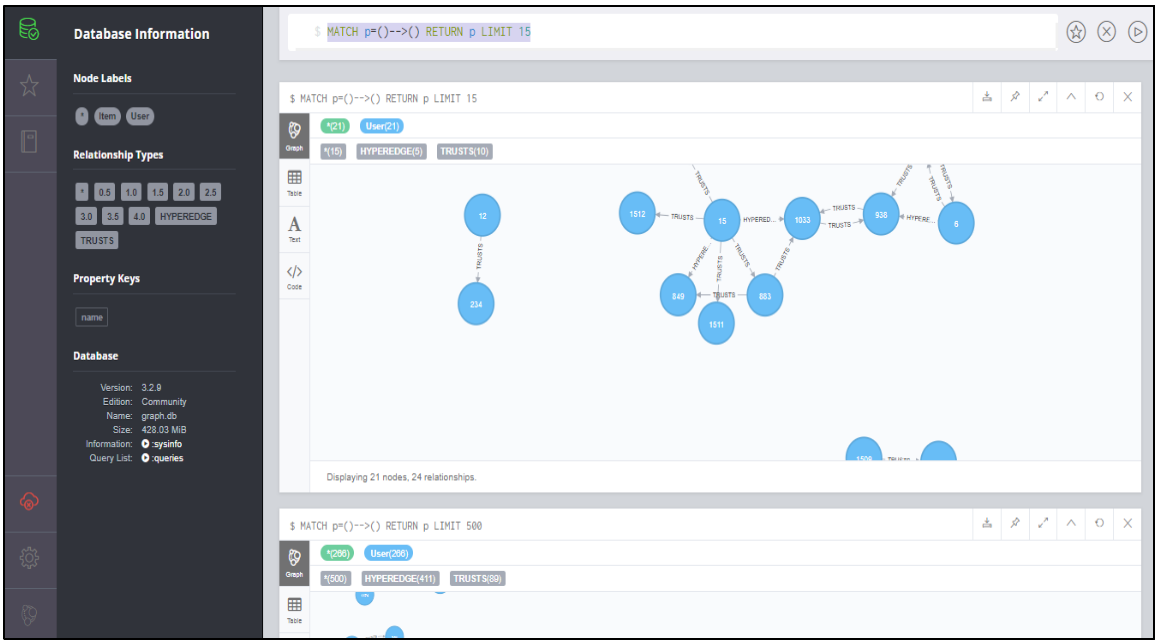

Figure 13: Homepage of neo4j server after implementation of algorithm.

Various relationship types are being shown here in figure 13. We have eleven options as *, hyperedge, trusts and ratings from 0.5-5. These range provides the user the variety to see among graphs and their connections.

Again by this we can specify according to our need which we want to view, whether hyperedges connections or trust connections or both of them or any particular rating. We can view that people who have given rating 4 have how many trust connections and how many hyperedges connections. Generally in our implementation we have used screenshot with regard to *. As we like to know how many trust connections have made how many hyperedges. On the other hand we can also put constraint on how many nodes we need or many connections we want to view.

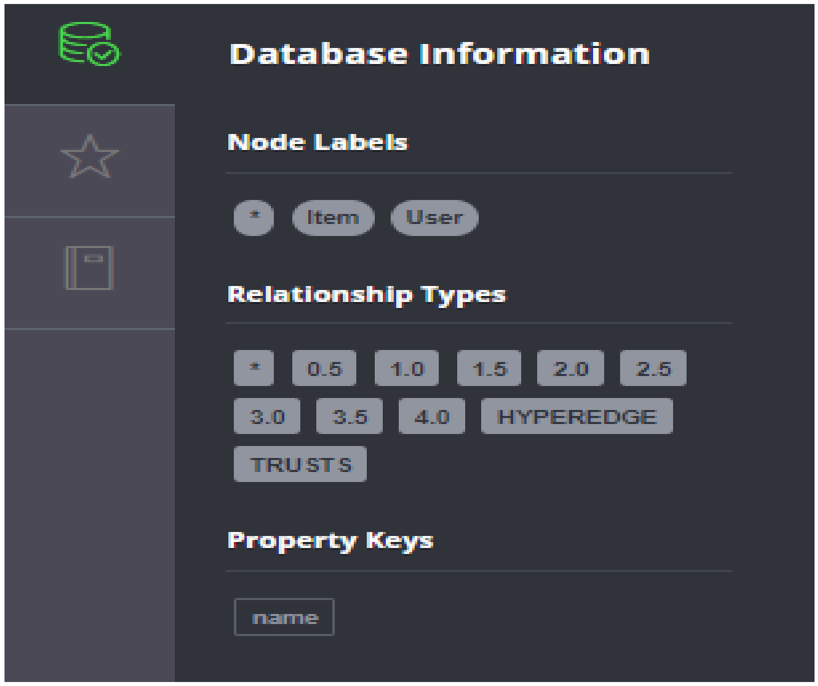

Figure 14: View of relationship types being shown on Neo4j.

As there are huge number of connections and nodes which makes the view more complex and hence less understandable, so we can put limit constraint on top where we execute the query. By this we can easily see the nodes and clusters made after the implementation of proposed algorithm. Query is written as,

(MATCH p= () --> () RETURN p LIMIT 150)

So by this we have put the constraint on relations that we only need to view 150 relations. This query can be seen in figure 14 on the top of graphs or where queries are executed. As shown in figure 16, these blue nodes represent users and some connections among them represents their relationship like which person trust which person or user. All nodes are scattered and have almost no link with each other. This is sparsity. We does not have much data to conclude the prediction or recommendation. The number written on blue nodes are the identities of users in database.

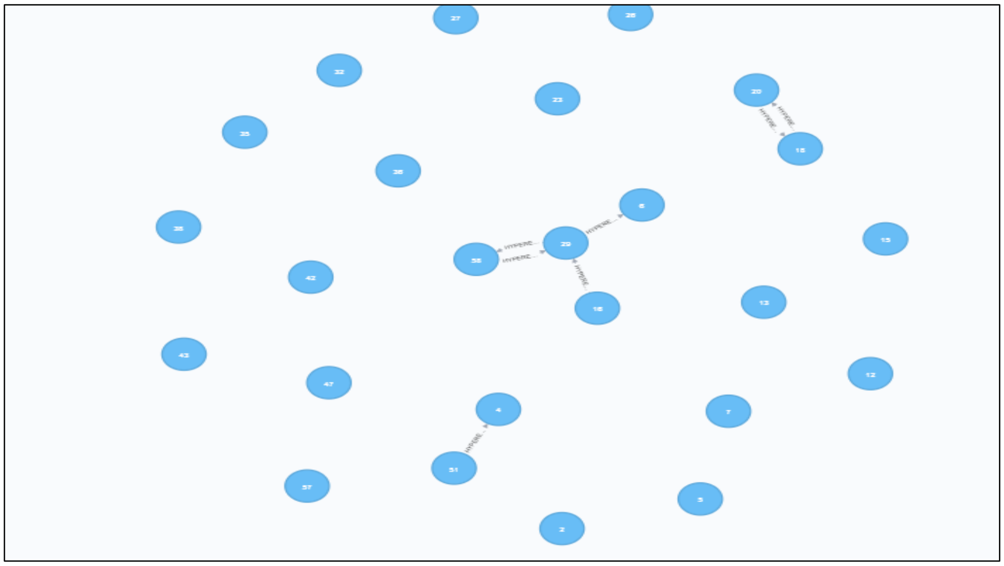

Figure 15: Independent nodes before application of proposed algorithm

After applying the proposed algorithm it can be seen in below in figure 16 the change where we have edges with integers between nodes which are also known as hyper edges which means various people/nodes are now connected to each other rather than only 1 degree connection with one trusted user. The number written on edges is the rating given by particular user to any item. And on some edges hyperedges is also written, which is called as transitive relation made between users. That is the real output of our proposed algorithm in this thesis.

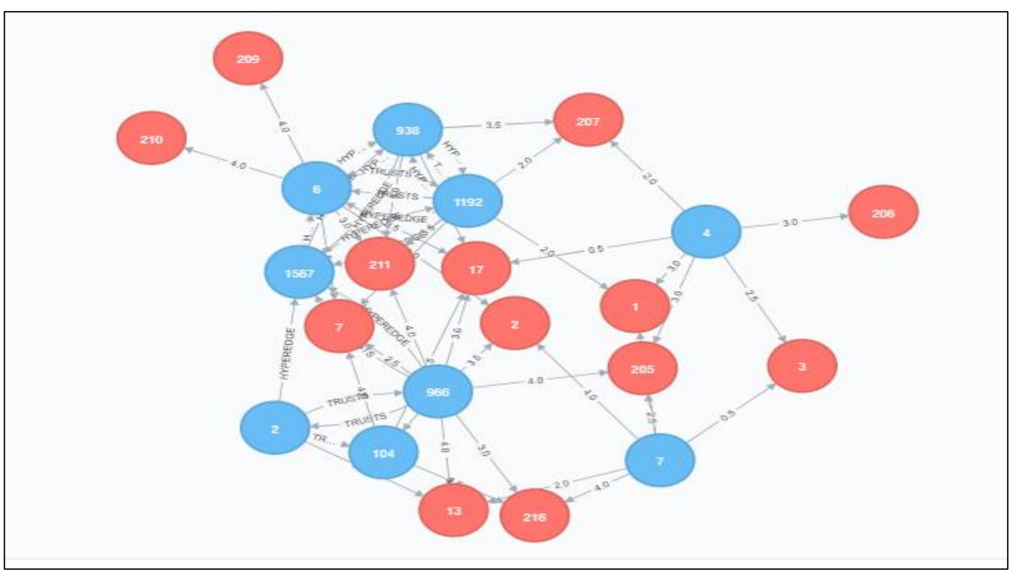

Figure 16: Screenshot of final result of nodes with hyperedges between users.

This further improves the accuracy of recommendations. In this fig red nodes are items and blue nodes are users and edges between them is relationship between them after application of algorithm. The number written on nodes are identities of items and users respectively. By implication of proposed algorithm we can clearly notice that the relationships have been increased and due to which it would be easy to predict for users and to generate accurate recommendations. As it is more likely that if our friend is showing some interest in any particular item, his trusted friend will also like it. So this relationship among users help in solving the problem of sparsity and cold start in recommendations.

Chapter 6 Conclusion and Future Work

6.1 CONCLUSION:

Graph databases have achieved another method for displaying and crossing interconnected information that is unparalleled in data stockpiling. With the coming of generation review frameworks, for example, Neo4j utilizing GDB issues can be tended to without falling back on a constraining execution on a RDBMS. The graph database has a characteristic application for arrange, semantic, organic and recommender frameworks that require the sort of information demonstrate no one but they can offer. Graph databases have achieved another method for demonstrating and crossing interconnected information that is unparalleled in data stockpiling. With the approach of creation review frameworks, for example, Neo4j issues can be tended to without falling back on a constraining execution on a RDBMS. The graph database has a characteristic application for arrange, semantic, organic and recommender frameworks that require the kind of information demonstrate no one but they can offer.

As a general rule, individuals have a tendency to acknowledge recommendations from dear companions, proficient individuals or comparative individuals. To adjust and take full preferred standpoint of these components. In this research, we present trust and client influence as upgrades for community oriented sifting based methodologies notwithstanding likeness, and propose another half and half suggestion calculation. This proposed technique overcomes the problems of traditional recommendations by using hyperedges concept on social graphs. Limitations of content and collaborative based filtering were cold start problem and sparsity issues.Interest and trust networks are represented using user–user matrix and user–item matrix respectively. We have increased the trust between users by which we have more entries in user- user matrix and in user item matrix which further results in vanishing that cold start and sparsity problem. Neo4j is used to analyse and manipulate social graphs. We have used FilmTrust dataset. Through experiments we have noticed that this proposed technique outperforms state of the art recommender system.

6.2 Future Work:

In future to enhance this technique more efficiently we can use social tagging or contextual information. Social tags appear bit attractive as they can improve the trust between users by giving more information in view of tags. We can also modify this algorithm by further allocating weights to the nodes in social graphs. In addition, we just utilize between client trust data in this article, yet in numerous online interpersonal organizations, distrust data is additionally expressed by numerous clients. Since a client trust highlight space may not be reliable with the relating client distrust highlight space, we can't just fuse the distrust data into our model. Later on, we have to research the accompanying two issues: regardless of whether distrust data is valuable to build the forecast quality, and how to consolidate this distrust data to acquire better-quality outcomes.

Moreover, while melding the social trust network data, we disregard the data diffusion or proliferation between clients. A more precise approach is where the diffusion procedure between clients is considered. Subsequently, we have to supplant/replace the social network matrix factorization with the social network diffusion forms. This thought will help reduce the information sparsity issue and will possibly build the prediction accuracy or exactness.

Ultimately, we either connect tags with clients or connect tags with items. As a matter of fact, we can plan a more broad system to join tags with clients and items at the same time. This thought will give more data than both of the proposed strategies, thus can additionally enhance the suggestion quality.

References

[1] Li Weimin, Zhengbo Ye, Minjun Xin and Qun Jin. “Social recommendation based on trust and influence in SNS environments.” Multimedia Tools and Applications 76(9), 11585-11602, 2010.

[2] Goldberg David, Nichols Dave and Terry. “Using collaborative filtering to weave an information tapestry” Communications of the ACM, 35(12), 61-70, 1992.

[3] Ray Sanjog, and Ambuj Mahanti. “Improving prediction accuracy in trust-aware recommender systems.” In proceedings of 43rd Hawaii International Conference on system sciences, IEEE, 1-9, 2010.

[4] Bonhard, Philip, and Martina Angela Sasse. “Knowing me, knowing you—Using profiles and social networking to improve recommender systems.” BT Technology Journal 24(3), 84-98, 2009.

[5] Kim Hee- Woong, Yunjie Xu and Sumeet Gupta. “Which is more important in Internet shopping, perceived price or trust?” Electronic Commerce Research and Applications 11(3), 241-252, 2012.

[6] Jamali Mohsen and Martin Ester. “A matrix factorization technique with trust propagation for recommendation in social networks." In Proceedings of the 4th ACM conference on Recommender systems, 135-142, 2010.

[7] HeJianming, and Wesley Chu. "A social network-based recommender system (SNRS)." Data mining for social network data. Springer US, 47-74, 2010.

[8] Golbeck, Jennifer. "Generating predictive movie recommendations from trust in social networks." In proceedings of International Conference on Trust Management, Springer, Berlin, Heidelberg, 93-104, 2006.

[9] Guo Guibing, Jie Zhang and Daniel Thalmann. "A simple but effective method to incorporate trusted neighbours in recommender systems." In proceedings of International Conference on User Modeling, Adaptation and Personalization. Springer, Berlin, Heidelberg, 114-125, 2012.

[10] Yuan W., Guan D., Lee, Y. K., Lee, S., and Hur, S. J. “Improved trust-aware recommender system using small-worldness of trust networks,” Knowledge-Based Systems, 23(3), 232-238, 2010.

[11] Bathla Gourav, Himanshu Aggarwal and Rinkle Rani, “A graph-based model to improve social trust and influence for social recommendation”, The Journal of Supercomputing, 1-19, 2017.

[12] Garg Neha and Rinkle Rani, "Analysis and visualization of Twitter data using k-means clustering". In Proceedings of the IEEE Intelligent Computing and Control Systems (ICICCS), 670-675, 2017.

[13] Roy Sanjukta and Balaraman Ravindran, "Measuring network centrality using hypergraphs". In Proceedings of the Second ACM IKDD Conference on Data Sciences ACM, 59-68, 2015.

[14] Cha Meeyoung, Hamed Haddadi, Fabricio Benevenuto, and P. Krishna Gummadi. "Measuring user influence in twitter: The million follower fallacy", Icwsm 10, 10-17, 2010.

[15] Kempe David, Jon Kleinberg and Eva Tardos. "Maximizing the spread of influence through a social network." In Proceedings of the 9th ACM SIGKDD international conference on Knowledge discovery and data mining, 137-146, 2003.

[16] Bell Robert, Yehuda Koren, and Chris Volinsky. "Modelling relationships at multiple scales to improve accuracy of large recommender systems." In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, 95-104, 2007.

[17] Khribi, Mohamed Koutheaïr, Mohamed Jemni and Olfa Nasraoui. "Automatic recommendations for e-learning personalization based on web usage mining techniques and information retrieval." In Advanced Learning Technologies, 2008. ICALT'08. Eighth IEEE International Conference on, 241-245, 2008.

[18] Zitouni, Hanane, Lamia Berkani, and Omar Nouali. "Recommendation of learning resources and users using an aggregation-based approach." In Advanced Information Systems for Enterprises (IWAISE), 2012 Second International Workshop, 57-63, 2012.

[19] Tarus, John K., Zhendong Niu, and Abdallah Yousif. "A hybrid knowledge-based recommender system for e-learning based on ontology and sequential pattern mining." Future Generation Computer Systems, 72(5) , 37-48, 2017.

[20] Kardan Ahmad, Fatemeh Hendijanifard, and Solmaz Abbaspour. "Ranking concept maps and tags to differentiate the subject experts in a collaborative e-learning environment." In proceedings of the 4th International Conference on Virtual Learning, 308-315, 2009.

[21] Zhou Jia and Tiejian Luo. "A novel approach to solve the sparsity problem in collaborative filtering." Networking, Sensing and Control (ICNSC), In proceedings of the International Conference, 165-170, 2010.

[22] Hu Xia "Exploiting social relations for sentiment analysis in microblogging." In Proceedings of the sixth ACM international conference on Web search and data mining. ACM, 537-546, 2013.

[23] Burke Robin. "Hybrid recommender systems: Survey and experiments." User modeling and user-adapted interaction, 12(4), 331-370, 2012.

[24] Massa Paolo and Paolo Avesani. "Trust-aware recommender systems." In proceedings of the 2007 ACM conference on Recommender systems, 17-24, 2007.

[25] Ma Hao, "Improving recommender systems by incorporating social contextual information." ACM Transactions on Information Systems (TOIS), 29(2), 9, 2011.

[26] Wang Ziqi, Yuwei Tan and Ming Zhang. "Graph-based recommendation on social networks." In proceedings of 12th International Web Conference (APWEB) Asia-Pacific, 116-122, 2010.

[27] Wang Youwei, Weihui Dai, and Yufei Yuan. "Website browsing aid: A navigation graph-based recommendation system." Decision Support Systems, 45(3), 387-400, 2008.

[28] Sharma A, Rani R. “KSRMF: Kernelized similarity based regularized matrix factorization framework for predicting anti-cancer drug responses.” Journal of Intelligent & Fuzzy Systems (Preprint) 1-2.

[29] Chen, Pei-Yu, Shin-yi Wu, and Jungsun Yoon. "The impact of online recommendations and consumer feedback on sales." In Proceedings of the ICIS 58, 711-723, 2004.

[30] Isinkaye Folajimi, and Ben Ojokoh. "Recommendation systems: Principles, methods and evaluation." Egyptian Informatics Journal, 16(3), 261-273, 2015.

[31] Linden, Greg, Brent Smith and Jeremy York. "Amazon.com recommendations: Item-to-item collaborative filtering." IEEE Internet computing 7(1), 76-80, 2003.

[32] Merialdo, Arnd Kohrs-Bernard. "Clustering for collaborative filtering applications." Intelligent Image Processing, Data Analysis & Information Retrieval 3, 199, 1999.

[33] Hofmann, Thomas. "Collaborative filtering via gaussian probabilistic latent semantic analysis." In proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, 259-266, 2003.

[34] Zhang Yi and Jonathan Koren. "Efficient bayesian hierarchical user modeling for recommendation system." In proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, 47-54, 2007.

[35] Liu Nathan and Qiang Yang. "Eigenrank: a ranking-oriented approach to collaborative filtering." In proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, 2008.

[36] Rex Douglas. "Quality in the technical performance of colonoscopy and the continuous quality improvement process for colonoscopy: recommendations of the US Multi-Society Task Force on Colorectal Cancer." The American journal of gastroenterology 97(6), 1296, 2002.

[37] Yu-Lan. “Research on embedded deployment of graphic database Neo4j” Modern Electronics Technique 22, 011, 2012.

[38] Khusro Shah, Zafar Ali and Irfan Ullah. "Recommender systems: Issues, challenges, and research opportunities." Information Science and Applications (ICISA) Springer, Singapore, 1179-1189, 2016.

[39] Gunes Ihsan. "Shilling attacks against recommender systems: a comprehensive survey." Artificial Intelligence Review 42(4), 767-799, 2014.

[40] Su Xiaoyuan and Taghi Khoshgoftaar. "A survey of collaborative filtering techniques." Advances in artificial intelligence 2009, 4.

[41] Schafer, Jon Ben. "Collaborative filtering recommender systems." The adaptive web. Springer, Berlin, Heidelberg, 291-324, 2007.

[42] Resnick Paul. "GroupLens: an open architecture for collaborative filtering of netnews." In Proceedings of the ACM conference on Computer supported cooperative work, 175-186, 1994.

[43] Chowdhury Gobinda “Introduction to modern information retrieval.” Facet publishing, 75, 2010.

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

[45] Sarwar Badrul, “Item-based collaborative filtering recommendation algorithms.” Proceedings of the 10th international conference on World Wide Web. ACM, 285-295, 2001.

[46] Adomavicius Gediminas, and Alexander Tuzhilin. "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions." IEEE transactions on knowledge and data engineering 17(6), 734-749, 2005.

[47] Webber Jim. "A programmatic introduction to neo4j." In Proceedings of the 3rd annual conference on Systems, programming, and applications: software for humanity. ACM, 217-218, 2012.

[48] Miller Jonas, Justin "Graph database applications and concepts with Neo4j." In Proceedings of the Southern Association for Information Systems Conference, Atlanta, GA, USA, 2324, 36, 2013.

[49] Deshpande, Mukund and George Karypis. "Item-based top-n recommendation algorithms." ACM Transactions on Information Systems (TOIS) 22(1), 143-177, 2004.

[50] Kim Dan, Donald Ferrin, and Raghav Rao. "A trust-based consumer decision-making model in electronic commerce: The role of trust, perceived risk, and their antecedents." Decision support systems 44(2), 544-564, 2008.

[51] Miller Justin. "Graph database applications and concepts with Neo4j." In Proceedings of the Southern Association for Information Systems Conference, Atlanta, GA, USA. 2324, 36, 2013.

[52] Jøsang Audun, Ross Hayward, and Simon Pope. "Trust network analysis with subjective logic." In Proceedings of the 29th Australasian Computer Science Conference Australian Computer Society, 49(3), 85-94, 2006.

[53] Christianson Bruce, and William S. Harbison. "Why isn't trust transitive?." International workshop on security protocols. Springer, Berlin, Heidelberg, 171-176, 1996.

[54] Massa Paolo, and Bobby Bhattacharjee. "Using trust in recommender systems: an experimental analysis." In proceedings of International Conference on Trust Management. Springer, Berlin, Heidelberg, 221-235, 2004.

[55] Selmi Afef, Zaki Brahmi and Mohamed Mohsen Gammoudi. "Trust-based Recommender Systems: An overview." In proceedings of 27th International Business Information Management Association, 2016.

List of Publications

Virk Asham and Rani Rinkle, "Efficient approach for social recommendations using graphs on Neo4j," In proceeding of IEEE International Conference on Inventive Research in Computing Applications (ICIRCA 2018), July 11-12, 2018, organized by RVS College of Engineering & Technology, Coimbatore. **[Accepted]**