Exploiting Trust and Suspicion for Real-time Attack Recognition in Recommender Applications

Ebrahim Bagheri and Ali A. Ghorbani
Faculty of Computer Science,
University of New Brunswick
Fredericton, N.B., Canada
{e.bagheri, ghorbani}@unb.ca

Abstract. As is widely practiced in real world societies, fraud and deception are also ubiquitous in the virtual world. Tracking and detecting such malicious activities in the cyber space is much more challenging due to veiled identities and imperfect knowledge of the environment. Recommender systems are one of the most attractive applications widely used for helping users find their interests from a wide range of interesting choices that makes them highly vulnerable to malicious attacks. In this paper we propose a three dimensional trust based filtering model that detects noise and attacks on recommender systems through calculating three major factors: Importance, Frequency, and Quality. The results obtained from our experiments show that the proposed approach is capable of correctly detecting noise and attack and is hence able to decrease the absolute error of the predicted item rating value.

1 Introduction

The rapid growth of online virtual communities has resulted in new types of collaboration and communication between the members of such societies. The main purpose of these interactions revolves around information sharing and data distribution. Any single person can disseminate his preferred information in the cyber world. The open atmosphere provides suitable grounds for free flow of information and an equal opportunity for every one to express or convey their knowledge, information, beliefs, or ideas. People can even exploit this opportunity as a means for marketing their products as has been practiced in e-commerce. With

no doubt the Internet, as the major medium resembling the cyber space has been overly populated with tremendous amounts of information from different sources. It would be hence a tiresome or even at times impossible attempt to find the appropriate information in the right time. Recommender systems are one of the most attractive applications widely used for helping users find their interests from a wide range of interesting choices [1].

One of the major worries in uncontrolled information society is the aspects of privacy and security. Security in cyberspace can have two very different aspects. Its first face that seems more obvious is restricting the use of the shared information only to authorized users. In order for users to become authorized, it is most likely that the information source or owner has to explicitly or implicitly grant the access. This type of security is referred to as hard security. Hard security; however, is not the only security requirement in such a setting. Protecting the users from malicious sources of information is also a challenging task. Since information is freely distributed by any person without proving its credibility, shielding users from spiteful information sources is highly desirable [2]. Johnson [3] states that 90% of the children encounter pornography online while doing their homework which elucidates the need for protecting children from deceit in the wild, undiscovered, and unregulated frontier called the Internet.

Fraud and deception are not only related to virtual communities, but also pervasive in real societies and actual life [4]. Different attempts have been made to create a methodology for detecting deceit, fraud, slander, and cheat with respect to different contexts [5, 6]. Zhao et al [7] believe that deception in a multiagent environment can be classified into three major categories. In the first category the agents are not sincere in expressing their abilities. This type of deception is called Agent Ability Declaration Deception. In the second category, Agent Information Deception, the agent spreads false information to mislead others or disguise reality. In an Imposter Deception an agent spawns many fake agents to interact with others to broadcast rumor or a special thought in the agent society.

From a formal logic point of view; Sun et al [4] have stated that an agent is a combination of knowledge and inference. For instance suppose a_i is an agent, therefore it will have a knowledge base K_{ai} and a set of reasoning methods R_{ai}. Exploiting such definition allows us to define three circumstances that deceit would occur. It would either be an expression of knowledge base contradiction (*i*), reasoning opposition (*ii*) or both (*iii*), which have been named Knowledge base Deception, Inference based Deception, and Hybrid Deception, respectively.

- $K_{ai} \neq K_{aj}$ and $R_{ai} = R_{aj}$ (*i*)
- $K_{ai} = K_{aj}$ and $R_{ai} \neq R_{aj}$ (*ii*)
- $K_{ai} \neq K_{aj}$ and $R_{ai} \neq R_{aj}$ (*iii*)

In an e-commerce environment deceit can be employed to defame rivals. False information or partially true facts can be spread out by biased buyers or sellers to defame a specific product or seller. Bitting and Ghorbani [8] propose a defamation protection model based on the concept of reputation. In their model, whenever a

transaction takes place between two parties, a buyer and a seller, the buyer can become suspicious of the information provided to him. If the received quotes cause his perception of some other seller (or sellers) to change to a significant enough degree, that quote is deemed suspicious. Similarly, if any of the quoted prices differ significantly from what the buyer believes to be real, the information is taken as an indication for becoming suspicious to that seller. If the buyer is suspicious to the information received from a specific seller, he can call for a consensus. Reaching confidence based on the conclusion of the consensus that defamation has taken place, the buyer can decrease the seller's reputation. Participants with low reputation value are overlooked in this model; therefore different parties try to promote their social face by providing truthful information.

Electronic commerce systems need to suggest the most relevant set of items to the users to increase their sales and customer satisfaction. Recommender systems can serve this purpose by exploiting the opinions of the community to aid individual members effectively identify their appropriate needs from an overwhelming set of choices [9]. Content based and collaborative filtering are the two most widely used methods that are employed in different recommender systems. Each of these methods suffers from different problems. Content based filtering recommends a set of items that are conceptually closest to the items that have been previously selected by the user. One of the deficiencies of this model is that it requires a correct human aided classification and proper ontological categorization of all items. Since this categorization procedure is human centric, it is time consuming and error prune [10]. There are also cases in which items cannot be clearly classified into specific categories. Jokes are a clear example of such instances [11].

It is not only the content based filtering that experiences certain difficulties, but collaborative filtering has also its own deficiencies. Collaborative filtering provides recommendation to the end users through inter-user rating pattern similarities. The cold start problem, recommendation sparseness, and attack vulnerability are the major issues in this class of recommender systems. Cold start refers to the fact that since new comers have not rated sufficiently enough number of items, the recommender algorithm is unable to direct appropriate recommendations at the user. This results in a poor recommendation list for the people with fewer ratings. As is the case for many recommender systems, when there are only a few people with the similar rating patterns to the current user, poor recommendations are given that is a consequence of the sparseness problem. Collaborative filtering algorithms are also vulnerable to malicious attacks. By attacks we mean that malicious users can insert unfaithful ratings to deceive others. This can be a tool for people to advertise their own products while degrading other people's goods.

In collaborative filtering based recommender systems users provide ratings for four specific reasons: improve their profile, express themselves, help others, or influence others [12]. The first group of people believe that their contribution to the system will benefit them through receiving much more accurate recommendations. A user within the second class however, provides rating to express himself in the community; while in the third group, people tend to assist others make the right decisions. On the contrary to these three groups of users, the fourth group tries to

influence the recommender system's behavior by providing unfaithful ratings. Their ratings may aim at pushing an item's conceptual coordinates in a well-connected position in the virtual correlation space that the recommender system would recommend the item to many other users. Nuke attacks may also be pursued to devalue products of other competitors. Some of the users in the fourth category only have the intention to harm the recommender system itself. This type of attack will affect the overall behavior of the recommender system and be undertaken for fun or defaming the recommender system amongst many other recommender applications.

An attack can be analyzed from many different points of view [13]. It can be firstly analyzed from the intention aspect to see whether it is aiming to push or nuke a set of items or is it aiming at the recommender system as a whole. The target of the attack should also be considered. An attack may aim specific users or items. Any guided attack requires some level of knowledge and expertise which is very much algorithm dependent and needs information of the rating datasets. Some of this information may be collected from the interfaces of recommender systems that provide the average rating of every specific item. It is also important to increase the cost of attack in a recommender system so that fewer people are willing to launch an attack. Social costs are paid through idea elicitation and reputation endangerment [14]. Monetary costs have also been applied in e-commerce systems such as eBay [15] that giving ratings requires a user to have at least one financial transaction. In such situations, users prefer not to waste their rating chances for defaming others.

O'Mahony et al [16] have proposed a model to detect natural and malicious noise in a dataset of recommender systems. In this approach they exploit the *Mean Absolute Error* (*MAE*) between the actual and the predicted rating as the consistency measure. Any rating that falls below a given threshold (φ) is deemed to be classified as one of the before mentioned noises. Let $r_{u,v}$ be a rating value, $p_{u,v}$ be the predicted rating for the user-item pair, and r_{min}/r_{max} be the minimum and maximum permissible ratings. Consistency *c* is calculated using Equation 1.

$$c_{u,v} = \frac{\left| r_{u,v-} p_{u,v} \right|}{r_{\max} - r_{\min}} \quad (1)$$

$$c_{u,v} > \varphi \quad (2)$$

In this paper we propose a layered model for detecting noise in a recommender dataset. The most important feature of the algorithm is that it is performed online and during the recommender system execution. As a new rating is provided in the systems a trust value is ascribed to the rating. Trust is formalized in this context through three major constituent elements:

1. Importance (ζ),
2. Frequency (γ),
3. Quality (λ).

Importance (ζ) measures the degree of conformance between the asserted rating value for *item j* in the given rating and the overall trend towards rating the same item in all previous interactions from all other users. This factor focuses on the social aspect of trust and has been incorporated into the model to reflect the real world fact that ratings which fall far from the general trend of rating in the history of a specific item should not heavily affect the rating behavior of the recommender algorithm. Frequency, γ, determines how often a user participates in the activities of the community. This factor implicitly encompasses both the longevity and interaction roles [17] of a user in a recommender system. This constituent element of the formalized trust value targets the behavior of the user that has asserted the rating. Quality (λ) is also the other component of the proposed trust model that addresses the excellence degree of a user's past behavior and interaction with regard to the current recommender system. We formalize the trust value ascribed to every rating asserted by a user through a 3-Tuple T = (ζ, γ, λ). In this way, and with the help of signal processing theory each rating in the recommender system can be quantified as a signal. Any of the signals that have an altitude lower than the average trend of the overall signal altitudes that is calculated by the autoregressive moving average (ARMA) model is regarded as *Suspicious*. By suspicious we mean that it is considered as a distrustful rating. Any suspicious rating that descends below the standard deviation of the overall signal altitude will then be regarded as attack or natural noise and will be discarded from the dataset.

The rest of the paper is organized as follows. In the next section we will analyze the structure of the proposed trust model for detecting noise and malicious attacks. In Section 3 the structure of the employed datasets for evaluating the model, different types of attacks and simulation results have been provided. The paper then concludes in Section 4.

2. Trust Formalization for Noise Filtering

Any recommender system can be a target for malicious activity. Although malicious activity causes serious worries for the accuracy and the ability of a recommender system in giving precise and at the same time useful recommendations, but natural noise is also the other factor that may affect the functionality of the recommender system. Hill et al [18] have shown that users may provide inconsistent ratings for the same item at different points of time. For this reason, a specific rating cannot undoubtedly be classified as malicious or attack, and hence punish the corresponding user for unfaithful recommender system manipulation, since it may well be a natural noise that has occurred due to the misalignment of the user with his normal behavior at that certain time.

It would also be unfair to basically cluster the set of ratings related to a specific item and consider the outliers as noise or attack. Although this approach seems to give good insight into how different ratings are spread out for a particular item, but cannot be exploited as a sole factor in the detection procedure. For example other factors such as the asserting user's past behavior, his overall contribution in prior interactions, and longevity may compensate for this poor rating and even make it the

decisive rating in certain circumstances. Suppose that an attacker has launched an Imposter Deception attack on the recommender system. In this attack he creates numerous imposters to nuke item χ. All these imposters would hence rate item χ with very low rating. If user υ, that has a high reputation value based on his previous interaction, rates item χ with a high value, the detection mechanism which is based on a simple clustering technique will be easily misleaded to decide that the rating provided by user υ is either noise or malicious attack (Figure 1). This misdetection has two major repercussions which are Incorrect User Devaluation, and False Rating Disposal. In incorrect user devaluation a user with loyal rating will be devaluated because of incorrect attack detection. Disposing the correct rating values under the suppression caused by imposters can further disable the recommender system from giving suitable ratings. The major risk that threatens the recommender system as an effect of these two side effects is that the recommender system itself will assist the imposters by devaluating reliable users and disposing correct ratings.

In our proposed methodology we exploit a three dimensional factor for detecting natural noise or malicious activity in a recommender system dataset. Whenever a new rating is entered into the system by a specific user, the rating is analyzed in a real-time fashion. The rating is then tagged with a trust value showing how much confidence the system has on the new rating. The lower the trust value is, the more the system will be suspicious of the rating as being noise or attack. As it can be seen in Equation 3, suspicion has a inverse relationship with trust.

$$Suspicion = (Trust)^{-1} \qquad (3)$$

We have applied an adaptive threshold for filtering suspicious ratings. This means that not all suspicious ratings are disposed, but only those who fall lower than the threshold would be deleted. The reason for why we have applied a threshold instead of deleting suspicious ratings is the fact that some degree of uncertainty exists in the decision making process. Josang et al [19] state that due to a system's imperfect knowledge, it would be unreasonable to think that every opinion is strictly classified into belief or disbelief (or in our case trust or distrust); hence uncertainty should also be taken into account. Lack of evidence, vague user rating process, external factors affecting the user's behavior and many other factors can contribute in establishing uncertainty and lead us to devising a more conservative filtering approach.

To track each user's behavior in the system, an implicit reputation ascription process has also been devised. Reputation is a distributed, socially ascribed, and collective belief of the society towards the stand point of a single person within the context of that society [20]. For this reason we exploit user reputation values as one of the dimensions of rating trust ascription. The employed reputation management model is centralized and handled by the trust management process. A user with higher reputation would have a privilege over other users and has the chance to affect the overall ratings in the system.

Trust has been formalized as a three dimensional vector in the proposed malicious attack detection strategy. It consists of Importance (ζ), Frequency (γ), and

Quality (λ). Unlike Frequency, and Quality, that address some of the features of the user who has expressed the rating, Importance is directly related to the rating itself. It compares the altitude of the generated signal by the rating with the expected altitude. The weaker the signal is, the less it would have the ability to manipulate the system status. For instance if the recommender system has reached a stable rating for a given item, a very powerful input signal is required to interrupt the equilibrium. Algorithm 1 shows the overall behavior of our proposed filtering module.

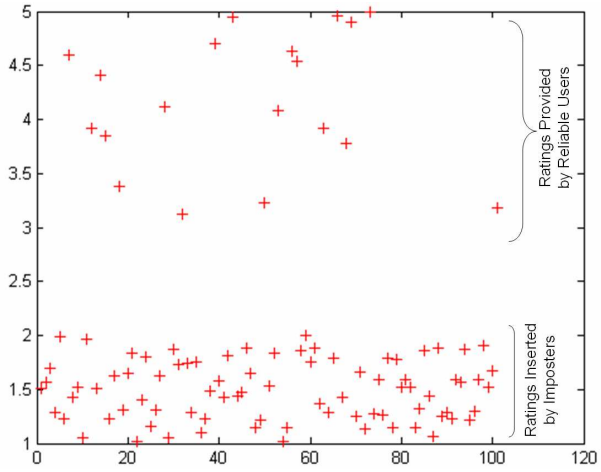


Fig. 1. The colony of ratings inserted by imposters easily deceives a naive attack detection algorithm

2.1. Importance

Importance calculates the potency of the input rating as a normalized signal. To determine the altitude of the input signal every rating is normalized in the first stage. In the normalization phase the input rating $rating_{i,j}$ that has been expressed by user i for rating item j will be compared with the previous rating behavior of user i. It is obvious that the ratings provided by each user for a specific item cannot simply be compared. For example if user υ_1 and υ_2 rate the same item χ with 2, and 5, respectively, in a 10 scale rating scheme, these ratings cannot be simply used to infer that $\upsilon 1$ has a lower belief to χ compared with υ_2. For this reason we normalize the rating value based on the prior rating behavior of the user.

$$NormalizedRate_{i,j} = \frac{rating_{i,j} - \frac{\sum_{k=1}^{n} rating_{i,k}}{n}}{\sqrt{\frac{1}{n}\sum_{k=1}^{n}(Rating_{i,k} - \overline{Rating_i})^2}} \quad (4)$$

In Equation 4, *n* represents the number of items that have been previously rated by user *i*. Having calculated the normalized value of the input signal, we plot the overall rating trend in rating item *j*. In this phase the ratings that have been given to item *j* from the start of its life will be considered. However ratings that have an older age will have a lower effect. Using this trend and with the application of the Autoregressive and Moving Average (ARMA) model (see Equation 5), we will estimate a possible rating value for this stage.

```
While Running(RS)

  If Received (Rating)

   ζ=Importance(Rating->Rate,Rating->Item)
   γ=Frequency(Rating->User,RS->History(User))
   λ=Quality(Rating->User,Rating->Date)

    // The trust value is calculated based on
    // ζ, γ, λ
    Trust = f (ζ, γ, λ)

    //  Weaker  input  signals  than  what  is
    // expected will be filtered
    if (Trust < (ExpectedTrust – Threshold) )
        FilterRating (Rating)
     End

  End //end if
End  // end while
```

Algorithm 1. Overall Behavior of the Proposed Filtering Module

Given a time series consisting of the ratings for a specific item, the ARMA model will provide the basic tools for understanding and predicting future values in this series. The ARMA model consists of two parts, an autoregressive (AR) part and a moving average or (MA) part. The model is usually referred to as the ARMA (*p*, *q*) model where *p* is the order of the autoregressive part and *q* is the order of the moving average. We employ ARMA (2, 1) in our experiments. The predicted value through ARMA will show the importance of the input signal that the system expects from a faithful user regardless of his rating behavior (since signals show normalized ratings).

$$X_t = \varepsilon_t + \sum_{i=1}^{p} \phi_i X_{t-i} + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} \quad (5)$$

In Equation 5, ε_t is the error term, while the first and second summations calculate the AR and MA parts of the ARMA model, respectively.

The predicted signal altitude will then be used as the center of a Gaussian distribution like function (see Equation 6) to decrease the value of the input signals that are far from the predicted value. The altitude of the input signal calculated by Equation 6 will represent the Importance of the current rating.

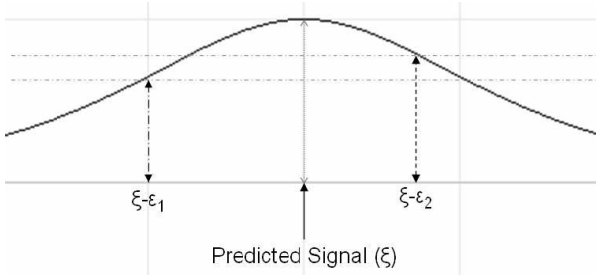


Fig. 2. ξ Shows the predicted signal value calculated by the ARMA model

$$\zeta = \frac{2}{e^{\Theta(\Delta\xi)} + e^{-\Theta(\Delta\xi)}} \quad (6)$$

$$\Delta\xi = \xi - \varepsilon \quad (7)$$

In Equation 6, Θ is a constant regulating factor that controls the gradient of the importance function. ξ and ε represent predicted signal value and the input signal in Equation 7, respectively.

2.2. Frequency

Frequency (γ) determines how often a user participates in the rating process in a recommender system. This factor implicitly encompasses and verifies both the longevity and interaction role fulfillment of the user. The more rates are contributed to the recommender system, the more successful it will be. Therefore the ratings of the users that have provided more ratings in the system should be valued more than other negligent users. Respecting these users will also have another benefit by guarding their ratings from deceitful attacks of imposters. Since imposters start an attack without any prior interaction with the system, the proposed algorithm will not value their ratings as much as it does for more frequent raters. There are cases where the imposters commence their attack by providing fair ratings for a few items so that

they gain enough reputation in the system to enable them to attack a specific item later on. In this case other factors of the trust model will contribute to the attack prediction process.

The frequency of a user participating in the activities of a recommender system is calculated through the ratio of signals (ratings) that he has recently emitted into the system with regard to all input signals. An aging factor (β) has been employed to value the users that have a continuous rating behavior. $\Psi_{i,t}$ shows the number of contributions of user *i* at time *t*.

$$\lim_{x\to\infty}\int_0^x \left(\frac{1}{1+e^x}\right) = 1 \tag{8}$$

$$\Phi_i = \sum_{t=1}^{now} \frac{\Psi_{i,t}}{1+e^{(now-t)\times\beta}} \tag{9}$$

$$\gamma_j = \frac{\Phi_i}{\frac{\sum_{j\in|\{users\}|}\Phi_j}{|\{users\}|}} \tag{10}$$

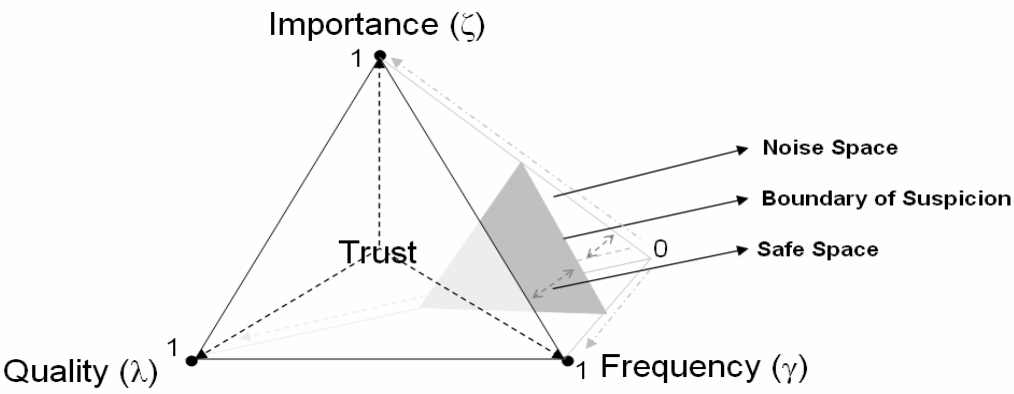


Fig. 3. The proposed model exploits a three dimensional trust value

2.3. Quality

Quality refers to the degree of excellence of a user in his rating history compared with the rest of the users in the recommender system. The calculation of this factor is achieved through counting the number of positive ratings (the ratings that the system has detected as clean) to the total number of his ratings compared with the behavior of others.

$$\alpha_i = \sum_{t=1}^{now} \frac{1}{1+e^{(now-t)\times\beta}} \times \frac{|CleanRatings_i|}{|Ratings_i|} \qquad (11)$$

To find out the general trend between the users as to what percentage of their rating contains noise; we follow a similar approach to Figure 2 and Equation 6. In this way a value is calculated that shows that a specific degree of noise in the rating is legitimate. This value is based on both the current user's past behavior and the other users' previous rating performance. If the current input signal contains more noise than the expected rate, it would be assigned a lower quality value.

The proposed trust model is a three dimensional concept that comprises Importance, Frequency, and Quality as its building blocks. Figure 3 clearly depicts the notion of Trust and Suspicion and their relationship with the three introduced factors. As the value of each factor decreases the trust value also diminishes and reaches towards the *Boundary of Suspicion*. We name the area between complete trust and the boundary of suspicion as *Safe Space*. The ratings that have a trust value in this area will be regarded as clean; however ratings with trust values in the *Noise Space* will be regarded as noise or malicious attack.

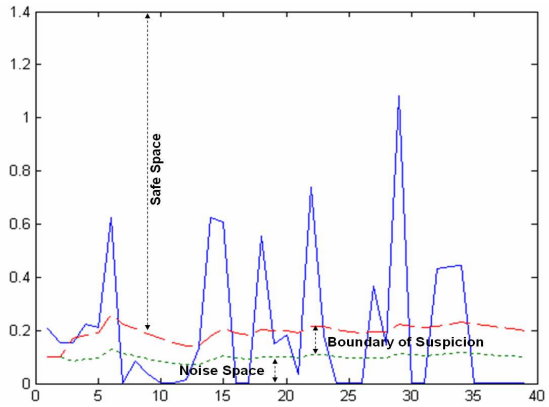


Fig. 4. A sample trust signal for a specific item (The item has 39 ratings in the dataset)

To specify the exact placement of the boundary of suspicion we employ an adaptive approach. In this approach we use the ARMA (2, 1) model again, to predict the next tolerable trust value. We also apply some degree of tolerance which is based on the standard deviation of the overall trust values calculated from the input signals for a specific item. As Figure 4 depicts, stronger signals have higher altitudes that makes them more trustworthy and less suspicious of being noise or attack. Other signals that have a lower altitude are suspicious of being noise or attack; but are tolerated. The last set of signals that fall below the boundary of suspicion are tagged as noise or attack and are hence filtered out.

We currently do not devalue the signals that fall in the boundary of suspicion, but further research can be conducted to see the effect of applying a fading factor to such signals.

3. Experiments and Results

In this section we will initially analyze the dataset that we have employed for our simulations. Different types of attacks that have been launched against the dataset in different periods of time will also be explained. The improvements achieved through the application of the trust model have also been depicted that are based on the Absolute Error (AE) between the predicted and the actual rating.

We have calculated the final trust value by building a vector (Equation 12) from the attributes of the trust model: Importance, Quality, and Frequency. Two sample trust vectors are shown in Figure 5.

$$Trust = vector(\varsigma, \lambda, \gamma) \qquad (12)$$

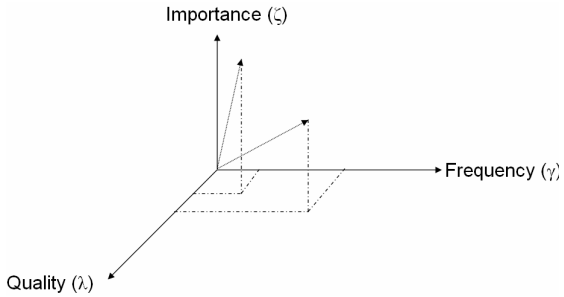
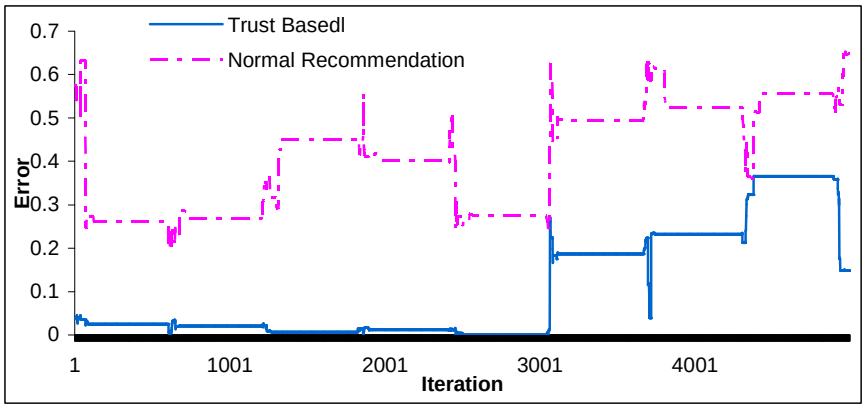


Fig. 5. The trust values as three dimensional vectors

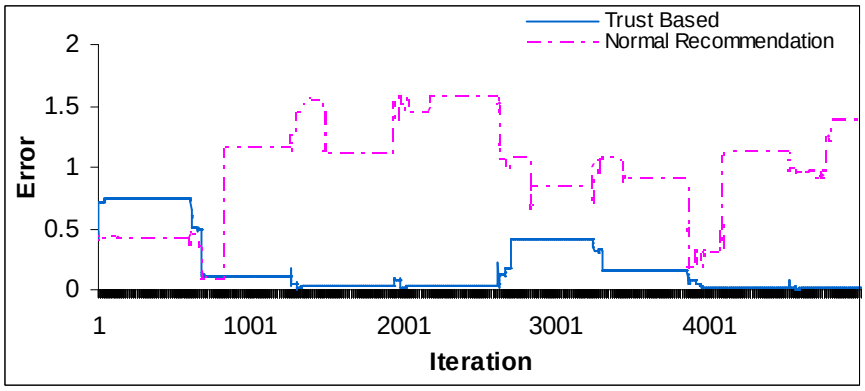
3.1. Dataset

There are several recommender system datasets freely available on the web such as EachMovie and MovieLens. The Eachmovie dataset consists of 72,916 users that have provided 2,811,983 ratings for 1,682 movies. The MovieLens dataset is a smaller dataset that comprises 100,000 ratings from 943 users for 1,682 movies. In our simulations we generated a sample dataset consisting of 12,000 ratings for 300 items by 60 users over 20 days. The ratings were on a scale of {1, 2… 5}. Our initial experimentations were conducted based on this dataset since we were doubtful that the previously introduced datasets may themselves contain noisy data. For this reason and because of their probable internal noise (or even malicious attack data that may be the result of attacks launched against these datasets at the time of their preparation) we decided to generate a new dataset for our simulations. In this way we are able to analyze the behavior of our proposed model under different attack strategies without having to worry about unknown noise that may affect the behavior of the algorithm.

The users in our generated dataset are categorized into 6 main classes. Each user depending on its class and the certain condition that he is in will show a specific behavior. Some users tend to rate the items they encounter with a high rating (class1) while the others prefer to give lower ratings (class 2). The rest of the user classes (classes 3 through to 6) conceptually differentiate between the items and rate each category of items in a different manner (e.g. a class of users may rate philosophical books with a high rating while they rate mathematic books very low.).



(a)

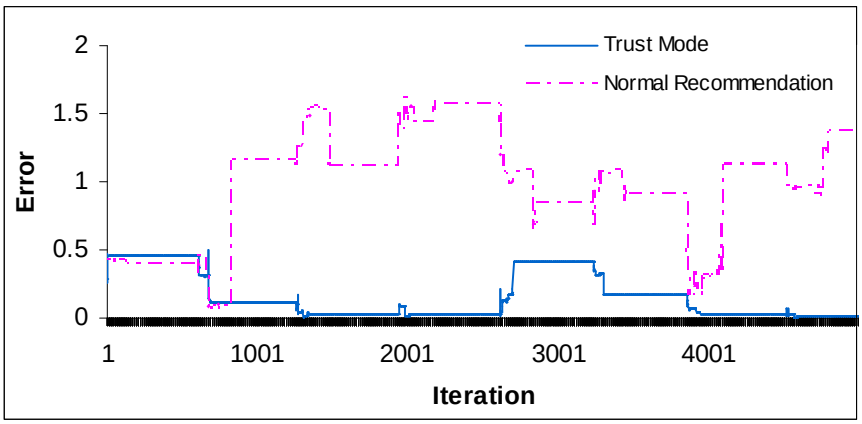


(b)

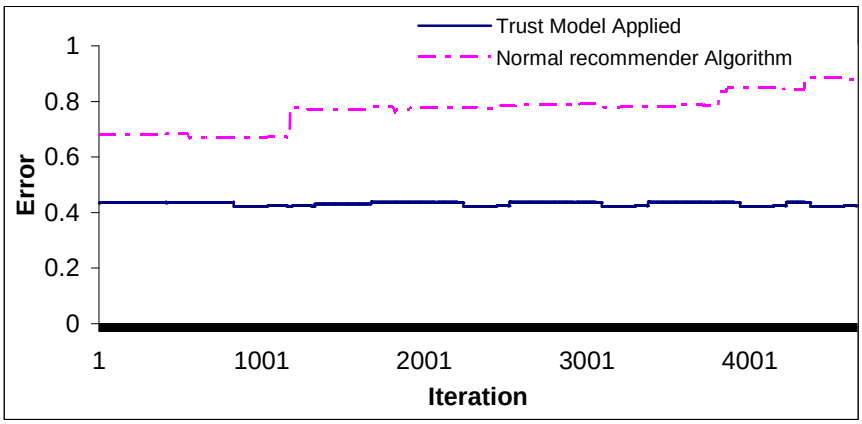
3.2. Attack Strategies and Evaluation

O'Mahony et al [16] have introduced various attack strategies on recommender system datasets from which we have adopted four: Natural Noise, Random Attack, Probe Attack, and AverageBot. In the simulations conducted with the natural noise strategy we did not add any extra ratings into the dataset, and the algorithm was applied to the dataset in a temporal manner. Recommendations were made in each iteration for a random item, and the difference between the real rating value assigned by the user and the predicted value by the same recommendation algorithm [21], but

with the application of the trust based filtering model were calculated. The recommendation error of each method, with and without noise detection, is shown in Figure 6(a). The proposed filtering method shows a much better performance compared with its counterpart.



(c)



(d)

Fig. 6. Graphs from (a) to (d) show the Absolute Error of the recommender application

The random attack strategy is the simplest type of attack that we consider. In this strategy *m-1* items are selected at random from the item set. These items are rated in a normal fashion, while one other item is either rated as r_{max} or r_{min} based on the average rating that the other users have ascribed to the item (Figure 6(b)). The popular attack attempts to ruin the attraction of the most popular items within the

recommender dataset. These items are good candidates for attacks since they are likely to be in a neighborhood of many other items and users, in this way damage to such an item can propagate to others that results in decreasing the cost of an attack (Figure 6(c)). The last type of attack that we undertake is the AverageBot attack. In this strategy the attack profile consists of all the items in the systems (or in our case a small portion of it). The attacked item receives r_{min} or r_{max}, while the other items receive a random rate on a normal distribution with the mean equal to the average rating of the item being rated and the standard deviation of all items in the dataset (Figure 6(d)).

4. Conclusions

Recommender systems are very attractive for malicious activity and vulnerable to attack. There are three major sources of threat intimidating recommender systems. The first source of such threats is the inconsistency of user's behavior in providing reliable and steady ratings. Although this type of risk causes concerns, but malicious activities that aim to nuke or push a certain item or groups of users arouse much more serious worries. In this paper we have proposed a three dimensional trust model comprising Importance, Frequency, and Quality to distinguish between noisy and clean ratings in a dataset of a recommender system. The model has a dynamic nature and analyzes incoming ratings in a real-time fashion. The results show great improvement from the perspective of reducing the absolute error between the real ratings and the predicted ratings. We would like to analyze the behavior of the proposed model on other datasets to understand its behavior under various conditions. It would also be provoking to measure the time complexity of the recommender system with the application of the proposed trust based filtering algorithm.

5. Reference

1. Rashid, A.M., Karypis, G., and Riedl, J., Influence in Ratings-Based Recommender Systems: An Algorithm-Independent Approach. SIAM International Conference on Data Mining, 2005.
2. Golbeck, J., Hendler, J.A., Accuracy of Metrics for Inferring Trust and Reputation in Semantic Web-Based Social Networks. EKAW 2004.
3. Johnson, S., Keep Your Kids Safe on the Internet, McGraw-Hill Osborne Media, 2004.
4. Sun, Z., Finnie, G., "Experience Based Reasoning for Recognising Fraud and Deception," Fourth International Conference on Hybrid Intelligent Systems (HIS'04), 2004.
5. Cristiano Castelfranchi, Yao-Hua Tan, The Role of Trust and Deception in Virtual Societies, International Journal of Electronic Commerce, Volume 6, Number 3 / Spring 2002.

6. Schillo, M., and Funk, P., Who can you trust: Dealing with deception. In Proceedings of the Autonomous Agents Workshop on Deception, Fraud and Trust in Agent Societies, 1999.
7. Zhao, S., Jiang, G., Huang, T., Yang, X., "The Deception Detection and Restraint in Multi-agent System," 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'05), 2005.
8. Bitting, E., Ghorbani A., Protecting e-commerce agents from defamation. Electronic Commerce Research and Applications 3(1): 21-38, 2004.
9. Schafer, J. B., Konstan, J., and Riedi, J., Recommender systems in e-commerce. In Proceedings of the 1st ACM Conference on Electronic Commerce, 1999.
10. Massa, P., and Avesani, P., Trust-aware collaborative filtering for recommender systems. To Appear in: Proceedings of International Conference on Cooperative Information Systems, 2004.
11. Goldberg, L., Roeder, T., Gupta, D., Perkins, C., Eigentaste: A Constant Time Collaborative Filtering Algorithm, Information Retrieval, Volume 4, Issue 2, Jul 2001.
12. Herlocker, J. L., Konstan, J. A., Terveen, L. G., and Riedl, J. T., Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.* 22, 1, 2004.
13. Lam, S. K. and Riedl, J., Shilling recommender systems for fun and profit. In Proceedings of the 13th international Conference on World Wide Web, 2004.
14. Donath, J., and Boyd, D., Public displays of connection, BT Technology Journal 22(4):pp. 71-82, 2004.
15. Resnick, P., and Zeckhauser, R., Trust Among Strangers in Internet Transactions: Empirical Analysis of eBay's Reputation System. The Economics of the Internet and E-Commerce. Michael R. Baye, editor. Volume 11 of Advances in Applied Microeconomics. Amsterdam, pages 127- 157, Elsevier Science, 2002.
16. P.O'Mahony, M., J. Hurley, N., Silvestre, N., Detecting Noise in Recommender System Databases, IUI'06, 2006.
17. Carter, J., Bitting, E., Ghorbani, A., Reputation Formalization for an Information–Sharing Multi–Agent System, Computational Intelligence 18 (4), pages 515-534, 2002.
18. Hill, W., Stead, L., Rosenstein, M., and Furnas, G. 1995. Recommending and evaluating choices in a virtual community of use. In Proceedings of the SIGCHI Conference on Human Factors in Computing System, 1995.
19. Josang, A., Modeling Trust in Information Security. PhD thesis, Norwegian University of Science and Technology, 1997.
20. Jøsang, A., Ismail, R., and Boyd, C., A Survey of Trust and Reputation Systems for Online Service Provision, Decision Support Systems, 2005.
21. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J., GroupLens: An Open Architecture for Collaborative Filtering of Netnews, Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work, 1994.