Instilling Similarity Rating Model to Combat Malicious Raters

Ilung Pranata*, Rukshan I. Athauda
University of Newcastle, Australia NSW 2308, Australia.

* Corresponding author. Tel.: +61249218967; email: Ilung.Pranata@newcastle.edu.au
Manuscript submitted September 1, 2014; accepted April 22, 2015.
doi: 10.17706/ijcce.2015.4.5.298-308

Abstract: The rapidly growing popularity of auction and rating websites has given rise to an increase use of ratings in online environment. However, many of these websites do not take into consideration several issues in ratings such as malicious raters, different user perceptions, number of available ratings and others. Thus, this creates mistrust between online buyers and products/services that the websites offer. In this paper, we discuss various issues faced by the current online rating systems. We then propose a rating model, termed as CPR (Credible and Personalised Rating), to address some of these issues. We demonstrate how our rating model can be implemented in both centralized and decentralized architectures. In addition, several experiments are conducted to study the effectiveness of our rating model to combat malicious and bogus ratings.

Key words: Rating model, trust, reputation, e-commerce.

1. Introduction

Commercial websites that post customer ratings of businesses such as Yelp [1], TripAdvisor [2] has gained popularity in recent years. In addition, many online e-commerce websites and auction marketplaces like e-bay [3], amazon.com [4], and many others often provide rating mechanism for customers to rate their products. With the widespread adoption of rating websites that span from ratings of electronic products, restaurants, hotels to ratings of physicians and hospitals, customers have become habituated to view ratings and reviews before purchasing any product and/or service. According to the study conducted by Harris/VFM consulting firm on online booking websites [5], the star rating of properties/accommodations is among the most important factors in customers' mind when selecting one property over the other. The outcome of this study is in line with other studies in [6], [7] that suggest a similar outcome in their particular study area. This implies the importance of ratings in customers mind prior to purchasing product/service.

While ratings have been useful to customers, several concerns still exist. First, malicious/bogus ratings impede customers' trust on the rating system. The 5 star rating system is the commonly used rating system in websites that average the rating feedbacks provided by other users with no measure of user credibility in providing ratings. Second, the perception of each user's expectation and satisfaction varies and thus, their ratings also varies [8]. For example, in a hotel booking system, a customer is happy and rates a hotel highly just because of its clean room and delightful service. However another customer may consider other factors such as hotel's facilities, location, etc. when giving the rating. This demonstrates a need to measure the similarity of ratings provided by users.

Third, raters' bias and raters' lack of expertise in the rated products or services also impacts the effectiveness of ratings. For example, rating in medical services is prone to rater's lack of medical expertise to judge the quality of care provided [9], [10]. Fourth, the insufficiency of rating feedbacks used to derive product/service rating results incorrect quality representation of product/service. IS research in [11]. [12] shows that the majority of products in online marketplaces such as in Amazon.com [4] receive overwhelmingly positive ratings. The question arises of whether the product is simply outstanding or due to lack number of ratings and raters biases embezzling their ratings.

In this paper, we attempt to tackle the first two issues in online ratings by proposing *CPR*, a Credible and Personalised Rating model that mitigates the issues of malicious ratings and various users' perception. We show that our CPR model can be easily implemented in both centralised and decentralised environments. In addition, we also perform experiments to show the effectiveness of our proposed rating model in mitigating the malicious/bogus raters. The remainder of this paper is structured as follows: Section 2 provides a review on trust models found in the literature, Section 3 details our proposed rating model that takes into account the rating similarity between the users. Section 4 provides discussion on several attack models, and Section 5 details the setup of our simulation environment. This is followed by Section 6 that shows several experiments we conducted to study the effectiveness of the proposed rating model. Section 7 summarizes our present work and discusses future directions.

2. Related Work

While research that focuses on malicious ratings in e-commerce environment is limited, there exists several research studies on trust/reputation management. In general, literature review classifies trust/reputation mechanisms into two main architectures based on their implementation: centralised trust and decentralised trust. Centralised trust relies on centralised servers for collecting and computing the trust values while in decentralised trust, trust values are computed by individual nodes by collecting feedbacks from users/raters in the environment. Centralised trust has advantages of simple administration and data storage. In turn, this mechanism is most preferred by online marketplaces and rating websites. However, it has main drawbacks of the risk of a single point failure/compromise and high overhead on the servers. Several centralised trust model found in the literature are Peer to peer multi-dimensional trust model [13], DEco Arch [14], PathTrust [15] and Collaborative Filtering [16].

Decentralised trust allows each user to individually compute trust value by collecting feedbacks from other users. A user requests feedback values of another user to all users in the environment and compute the trust value upon receiving all users' feedbacks. Due to its decentralised approach, this mechanism does not face issues of single point failure and server overhead. However, its complexity, network overhead, and user overhead to request, aggregate and compute the trust values may deter its practicability in an e-commerce environment. A number of decentralised trust models with its unique trust algorithms were proposed in the literature mainly for peer-to-peer environment, such as TrustMe [17], PeerTrust [18], P2Prep [19] and EigenTrust [20]. Outside of the peer to peer environment, there also exist various trust and reputation systems for other environments, such as Travos [21] in mult-agent systems, Decoupled [22] for distributed networks and TIDE [8] for digital environments.

Various concerns exists on these decentralised trust models, such as not considering the credibility of users when providing ratings (i.e. in EigenTrust and TrustMe), not taking into consideration the different perceptions of users and its applicability in the online marketplaces and rating websites. Several e-commerce marketplaces such as E-Bay [3], Amazon [4], Android market [23] have used a centralised 5-star average rating system. However, this average rating system is prone to the issues of malicious ratings as discussed in the previous section. Additionally, this rating system does consider various user perceptions.

Due to these aforementioned reasons, we proposed a personalised rating model, termed as CPR (Credible and Personalised Rating), in this paper. Our rating model is outlined in detail in the next section.

3. Proposed Rating Model

In this section, we propose our rating model, termed as *CPR* (Credible and Personalised Rating). *CPR* aims to mitigate several issues in rating, mainly malicious raters and different rating perceptions.

3.1. Main Rating Formula

Suppose an online environment, i.e. e-commerce or auction websites, comprises *n* users (users are both consumers and service providers), denoted as $U = \{U_1,\dots,U_n\}$, and *k* products, denoted as $P = \{P_1,\dots,P_k\}$. Each user, U_i, is identified by a unique identity IDU_i in the environment. Each product, P_k, is identified by a unique identity IDP_k and may be associated to its owner/provider OP_k, where $OP_k \in U$. Each user U_i has provided ratings to *m* number of products at some stage, and therefore we denote its list of rated products as $P^i = \{P_1^i,\dots,P_m^i\}$, where $P^i \subseteq P$. In addition, U_i also maintains a list of product ratings that he/she has rated in the past, denoted as $R^i = \{R_1^i,\dots,R_m^i\}$. The total number of rated products and their ratings by a user U_i are denoted respectively as $|P^i|$ and $|R^i|$, which each equals to *m*. Note that if a user gives rating to a product, say P_k more than once, the average rating for product P_k is maintained in R_k^i.

Each product P_k in the environment may or may not have product raters, i.e. those users who have rated the product. Product raters are denoted as $PR_k = \{PR_k^i,\dots,PR_k^j\}$, where $PR_k^j \in U$ and thus PR_k^j has P^j and R^j. To compute the Credible and Personalised Rating (CPR) of a product P_k, we propose the following rating formula:

$$CPR_k^i = \sum_{a=1}^{j} \frac{Sim(U_i, PR_k^a)}{\sum_{a=1}^{j} Sim(U_i, PR_k^a)} \times R_k^a \tag{1}$$

where,

- CPR_k^i is the Credible and Personalised Rating of product P_k personalised to user U_i.
- R_k^a is the rating given by product rater PR_k^a to product P_k.
- $Sim(U_i, PR_k^a)$ is the similarity value between user U_i and PR_k^a. Similarity value will be discussed later.

In the presented formula [1], each rater's rating (R_k^i) is weighted by his/her similarity value with U_i. The highest the similarity value that each rater has the highest the impact of its rating on the product rating (CPR_k^i).

Steps taken by user U_i to compute CPR_k^i are as follows:

- A user U_i searches for a product P_k that he/she wants to purchase/download.
- U_i will request for P^j and R^j to all raters in PR_k. These values that will be used to compute the similarity between U_i and PR_k^j.
- Compute the similarity value for each product rater in PR_k. Computing similarity values will be discussed in the next section.
- Compute rating of product P_k using formula [1].

3.2. Measuring Similarity

To measure the credibility of a product rater (PR_k^j), we introduce the concept of similarity value. Similarity measures the closeness of past product ratings between a product rater PR_k^j and a user U_i. This value is introduced due to the differences in users' perception when rating a product, as discussed in Section 1. The similarity value in our model is in between 0 (dissimilar) and 1 (similar). The highest the similarity of PR_k^j, the highest the impact of its rating (R_k^j) on the product rating (CPR_k^i).

To compute the similarity value of a product rater PR_k^j, a user U_i first requests P^j and R^j from this rater. U_i then finds all products that both U_i and PR_k^j have rated in the past, i.e. $P^i \cap P^j$, as depicted in Fig. 1. For each $P_z \in P^i \cap P^j$, U_i finds the differences between its own product rating (R_z^i) and PR_k^j product rating (R_z^i).

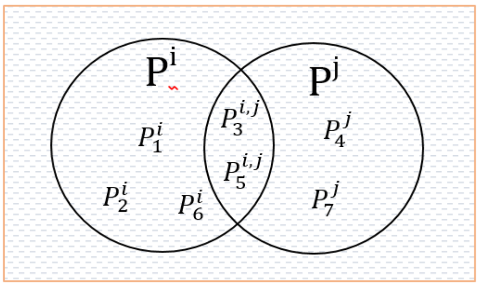

Fig. 1. Similar products that a user U_i and a product rater PR_k^j have rated in the past ($P^i \cap P^j$).

The similarity value is then derived by applying formula (2) below:

$$SIM(U_i, PR_k^j) = 1 - \left(\frac{\sum_{z=1}^{|p^i \cap p^j|} abs[R_z^i - R_z^j]}{|p^i \cap p^j|}\right) \tag{2}$$

where,

- $Sim(U_i, PR_k^j)$ is the similarity value between user U_i and product rater PR_k^j.
- $P^i \cap P^j$ is the total number of similar products that user U_i and rater PR_k^j has rated in the past.
- $abs[R_z^i - R_z^j]$ is the absolute value of differences between R_z^i and R_z^j.

For the situation where $P^i \cap P^j = \{\}$, then 0.5 default value (neither similar or dissimilar) is taken.

It is possible that user may have rated a particular product several times as he/she frequently uses this product. The number of ratings maintained for a particular product is represented by *t*. This means if *t* is set to 5, only the past 5 user's ratings on a particular product will be used in the similarity value computation. To reduce the complexity in product rating (R_z^i) computation, we simply use an average computation.

In a bootstrapping case where U_i is a new user in the environment who has not rated any product before, the similarity value of all product raters in PR_k is given a default value (δ = 0.5), neither similar nor dissimilar. Note that, giving the default similarity value δ to all product raters will not mitigate the issue of malicious and non-credible ratings in the computation as all malicious and non-malicious ratings are weighted equally. To solve this issue, the concept of Trust Circle [8] or pre-trusted peers [20] where a new user can manually assigns high similarity value to a set of high credible raters, including their friends or family members, can be introduced.

3.3. Implementation in Centralised and Decentralised Environment

Our rating model can be easily implemented in both centralised and decentralised environment. In a centralised architecture, rating data such as list of rated products (P^i), list of product ratings (R^i), and list of product raters (PR_k) can be stored in the central servers. Several data storage methods, such as files, database, etc. can be used in the centralised architecture. In addition, these servers are responsible to perform rating computation whenever requested by users.

In a fully decentralised architecture such as peer-to-peer (P2P) network, rating data must be stored in a decentralised manner as no central servers or super nodes are available. Thus, the onus is on each user/peer to maintain its own list of rated products (P^i) and their ratings (R^i) as shown in Fig. 2. Each user/peer also maintains t number of ratings for each product in its own data storage.

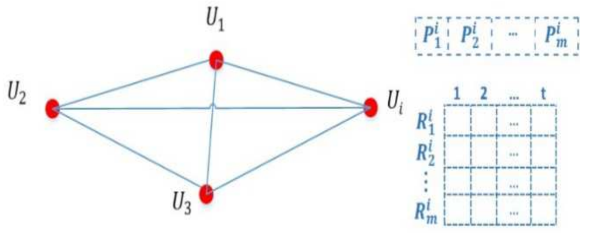

Fig. 2. Data storage for each user/peer in a de-centralised environment.

To compute a product rating in a de-centralised environment, We define a function SearchRatings(IDPk) that invokes the following:

- A user U_i will broadcast IDPk to $U_j \in U$, where $j = \{1, \ldots, |U|\}, j \neq$ OPk.
- Upon receiving IDPk, U_j will check if he/she has rated P_k. If U_j has rated P_k, he/she is subsequently categorised as the product rater ($U_j = PR_k^j$) and will return:

1) his/her P_k rating R_k^j,
2) rated product list (P^j), and
3) product rating list (R^j) to U_j.

- Once U_i collects all product raters' R_k^j, P^j and R^j, he/she will compute the similarity value of each product rater. If after a certain timeframe, a product rater does not reply to U_i request, this rater will be ignored from rating computation.

U_i will then compute the rating value for P_k.

4. Attack Vectors

In this sub-section, we discuss several attack vectors employed by malicious raters to subvert the rating algorithm. In addition, we also show how these attack vectors could be mitigated by our proposed rating model.

4.1 Rating Spoofing

In this attack, malicious rater attempts to modify his/her credibility by finding some vulnerabilities in the system to modify the similarity values. In our rating model, this attack is not effective as rater's similarity is not stored by the rater himself/herself but rather is determined by the user. The only way to spoof rater credibility is to modify users' similarity computation, which means breaking into the central servers if our rating system is centrally managed or to each user's system if de-centrally managed. As system security is not the focus of this paper, we assume that the central or distributed systems Table 1: Rating assignment for various user types are secured. Thus, malicious rater will not be able to spoof its own similarity value.

4.2 Sybil Attack

Sybil attack model [24] in which malicious user take advantage of low entry requirement to enter the environment. Malicious user waits until he/she provides some bogus ratings, deletes his/her account and reappears in the environment with a different account. Sybil attackers could prevent "good" raters to obtain trustworthy reputation as their bogus ratings outnumbered the credible ratings. With the initialisation of similarity default value (δ) for the new raters, this attack can be minimised. Suppose PR^{Sybil} re-register to the environment, this rater similarity value will be reinitialised to δ The default δ does not have a significant impact to the rating computation assuming $|PR_k^j|$ is large. Also, as PR^{Sybil} keep providing bogus ratings, his default similarity value will be reduced significantly. In addition to that, literature [20], [25], [26] has proposed several strategies, such as imposing cost of new ID creation or adding CAPTCHA will limit the impact of Sybil attackers.

Table 1. Simulation Setup

Parameter	Description	Default
nUser	# of users in the simulated environment.	100
nUserType	# of user types in the environment, i.e. honest and malicious.	2
nUserBehaviour	#of user rating behaviour model in the environment, i.e. strict and lenient.	2
nProduct	# of products in the simulated environment	1000
nProdType	# of product types in the environment, i.e. very good, good, soso, bad, very bad.	5
nCat	# of product categories in the environment.	10
nTrans	# of transactions for each simulation.	1000
nCycle	# of experiment cycles over which results are averaged.	5

5. Simulation Setup

This section details several experiments that we conducted to study the effectiveness of our proposed model in the simulation environment. Our experiments were conducted using the Reputation Management (RM) simulator [27]. As this simulator was primarily built for reviewing reputation systems in peer-to-peer (P2P) network, we modified this simulator so that it suits with our research objective, which is evaluating the effectiveness of our proposed CPR model in product rating. Several modifications made to the original RM simulator and the details of simulation setup are discussed below:

- Products are always centrally available with no clean-up percentage. The original RM simulator allows setting up the clean-up percentage as each peer in the P2P network is able to download and distribute a file. Clean up percentage shows the percentage of peers will clean the downloaded file to prohibit file distribution. As our research is solely focused on products rating, this percentage is removed as products are assumed to be always available.
- There are 10 product categories (nCat) set in the simulated environment. This is to mimic the real life rating website where people may search and rate for product in categories such as hotels, restaurants, beauty products, etc.
- Number of products setup in the simulated environment is 1000 products (nProduct) and these products are distributed equally to the 10 categories (nCat) such that each category has 10 products.
- Each product is categorised into the following type (nProdType): {"Very Good", "Good", "Soso", "Bad", "Very Bad"}. In the simulated environment, products are assigned in random order to these product types.
- There are two user types (nUserType) in the simulated environment: {Honest, Malicious}. Honest user always provide honest rating while malicious user always provide bogus ratings. The number of honest

and malicious users were varied in each simulation.

- The behaviour of user in providing ratings (nUserBehaviour) are categorised into the following: {Strict, Lenient}. Strict user gives a strict rating while lenient user gives a more lenient ratings. The number of strict and lenient users is divided equally in the simulated environment such that the number of strict and lenient users always equal to 50-50 for each user type.
- User rating mimics the 5-star rating system as rating ranges from 0 (one star: very bad) – 1 (5 stars: very good) with a step of 0.25 in between. In the simulated environment, various user's types and behaviours rates each product type differently, following the rules set in Table 2. For example, "honest" user with "strict" behaviour will rate "very good" product with rating of 1, "good" product with rating of 0.75, "soso" product with rating of 0.5, "bad" and "very bad" product with rating of 0, and so on. Such rating rules is introduced to mimic the real world rating system where users have different behaviours in providing ratings.

Table 2. Rating Assignment for Various User Types

User Type	**Behaviour Model**	**Product type**	**Ratings**
Honest	Strict	Very Good	1
		Good	0.75
		Soso	0.5
		Bad	0
		Very Bad	0
	Lenient	Very Good	1
		Good	1
		Soso	0.75
		Bad	0.25
		Very Bad	0
Malicious	Strict	Very Good	0
		Good	0
		Soso	0.25
		Bad	1
		Very Bad	1
	Lenient	Very Good	0
		Good	0.25
		Soso	0.5
		Bad	0.75
		Very Bad	0.75

- The collected experiment statistics are assessed in the following evaluation metric

$$\text{Eval.Metric}=\frac{\text{\# of Good products downloaded by Honest user}}{\text{\# of all products downloaded by Honest user}} \tag{3}$$

6. Simulation Results

6.1 Experiment on 25% Low Quality Products

We simulated an environment with default settings from Tables 1 and 2. In this experiment, we set the simulation environment with 25% low quality products, i.e. products with "bad" and "very bad" types as such the number of low quality products in the environment is 250 products. We progressively increased the number of malicious users such that the malicious users made up between 0% to 90% of all users in the environment. Note that, as discussed in section V, the number of strict and lenient users are divided equally in the two user types (i.e. honest and malicious user types). For each step of 15% malicious, we ran the simulation and obtained results depicted in Fig. 3.

In this experiment, we benchmark our proposed CPR model with the average 5-star algorithm (i.e. AverageRA) that is used prominently in many e-commerce websites. Fig. 3 shows that our proposed algorithm (red coloured line) performs above the average algorithms in all scenario. Note that after 60% of malicious users, the efficiency of average algorithm to deter malicious ratings has dropped significantly. This algorithm performs worse than when the environment does not have any rating algorithm (i.e. None – which randomly selects a product). This average algorithm weighs rater's feedback equally as such in the environment where malicious users are dominant, its efficiency to deter bogus ratings is significantly reduced.

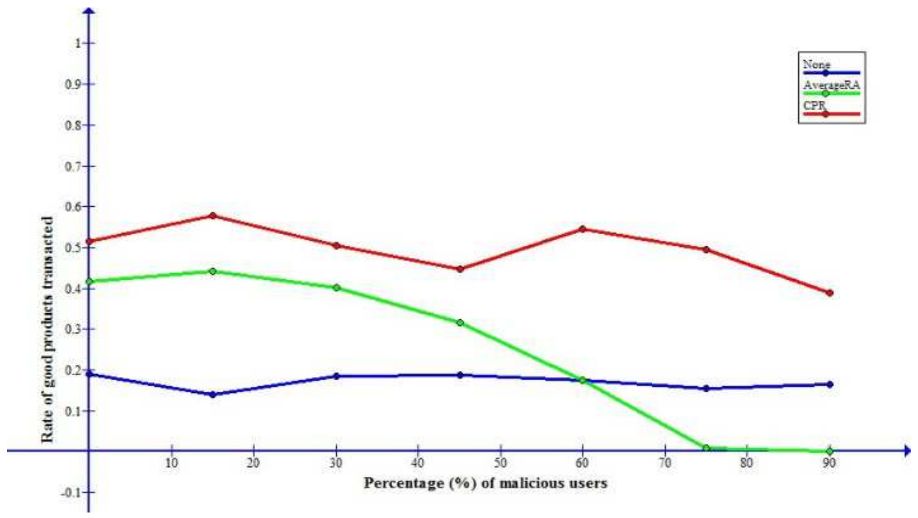

Fig. 3. Evaluation on 25% low quality products.

6.2 Experiment on 50% Low Quality Products

In this experiment, 50% of low quality products were simulated in the environment such that the number of low quality products reaches 500 products. We progressively increased the number of malicious users such that the malicious users made up between 0% to 90% of all users in the environment. The number of malicious users was incremented in steps of 15%. Fig. 4 shows the result of this experiment.

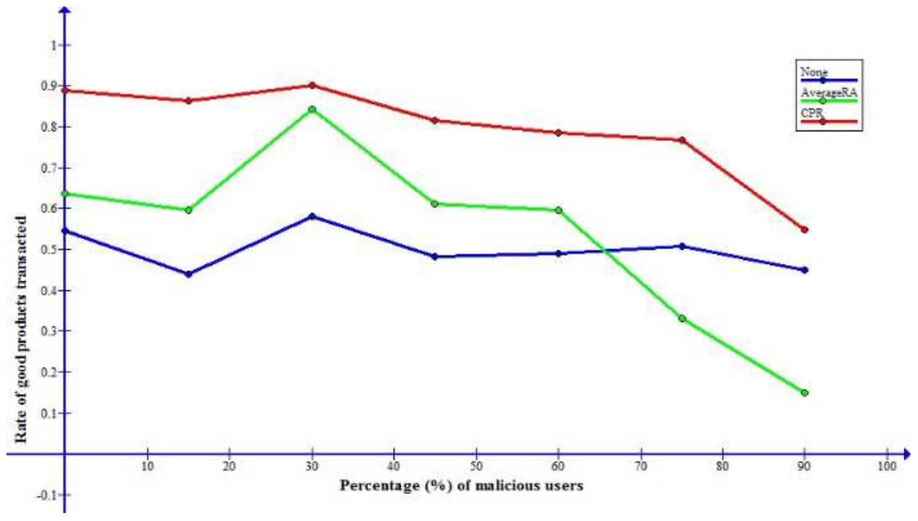

Fig. 4. Evaluation on 50% low quality products.

Similar to the previous experiment, we benchmark our proposed algorithm against the average 5-star algorithm (i.e. AverageRA). Our rating algorithm performs superbly in this experiment such that the percentage of good users transact good products is greater than 0.8 value in all scenarios. This means that, in 80% of total transactions performed, good users were able to purchase good quality products. The efficiency of the average 5-star algorithm reduces significantly as the percentage of malicious users increases.

6.3 Experiment on 75% Low Quality Products

We simulated an environment with default settings in Tables 1 and 2. We set the number of low quality products to 75% of all products. We start the number of malicious users from 0% and progressively increase this number to 90% in step of 15%. Statistics were collected for each step and were assessed with the evaluation metric. The results were depicted in Fig. 5.

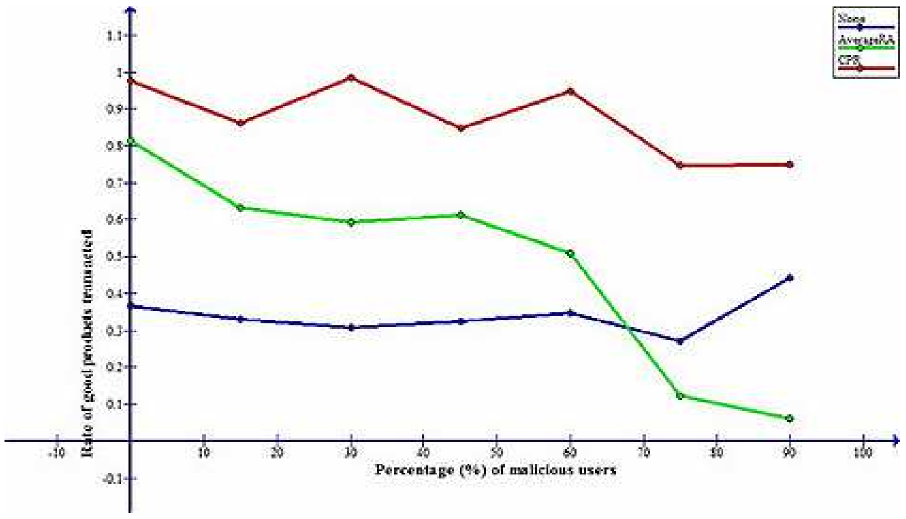

Fig. 5. Evaluation on 75% low quality products.

The results depicted in Fig. 5 are generally lower than the first two experiments. This is due to the environment is mostly filled by low quality products. With lower percentage of high quality products, the product options for users in each category are fewer. As a result, the success rate for good behaviour users in getting high quality products becomes lower. Note that our proposed algorithm still performs better than the average rating algorithm in this experiment. The average algorithm is punished badly as it weighs rater's feedback equally in spite of their credibility in providing the feedback.

7. Conclusion

The number of online consumers who use ratings to base their purchasing decision for a product/service in increasing. However, there exists several concerns when it comes to ratings in online environments, in particular the malicious ratings and different users' perceptions. These concerns are a common occurrence in auction and rating websites and they are not adequately addressed. This paper has proposed a rating model, termed as *CPR* (Credible and Personalised Rating) that takes into account the similarity between user and raters in the environment. Our experiment showed that *CPR* performs well in addressing these concerns. In addition, this rating model can be implemented in both centralized and decentralized environment. Future works include testing this rating model in the real auction/rating websites, surveying users' preference between the conventional average rating and the proposed rating model.

References

[1] Yelp. (2014, June 10). From http://www.yelp.com

[2] TripAdvisor. (2014, June 10). Form http://www.TripAdvisor.com

[3] Ebay. (2014, June 10). Form http://www.ebay.com/

[4] Amazon. (2014, 10 June). Form http://www.amazon.com/

[5] Boni, P. (2014, July 5). The distance between a hotel's keywords and bookings. From http://www.vfmii.com/media_content/howtoclose. pdf

[6] Bardach, N. S., Asteria-Peñaloza, R., Boscardin, W. J., & Dudley, R. A. (2012). The relationship between commercial website ratings and traditional hospital performance measures. *BMJ Quality and Safety Online First*, 1-8.

[7] Cosley, D., Lam, S. K., Albert, I., Konstan, J. A., & Riedl, J. (2003). Is seeing believing? How recommender interfaces affect users' opinions. *Proceedings of the CHI Conference on Human Factors in Computing Systems* (pp. 585-592). Fort Lauderdale, Florida.

[8] Pranata, I., Athauda, R., & Skinner, G. (2013). Modeling decentralized reputation-based trust for initial transactions in digital environments. *ACM Transactions on Internet Technology (TOIT)*, *12(3)*.

[9] Lagu, T., & Lindenauer, P. K. (2010). Putting the public back in public reporting of health care quality. *JAMA Network*, *304(15)*, 1711-1712.

[10] McCartney, M. (2009). Will doctor rating sites improve the quality of care? *BMJ Quality and Safety Online First*, *338*, 1033.

[11] Kadet, A. (2007). Rah-Rah ratings. *Smart Money Magazine*.

[12] Chevalier, J., & Mayzlin, D. (2006). The effect of word of mouth on sales: Online book reviews. Yale SOM Working Paper No's. ES-28 & MK-15.

[13] Ion, M., Danzi, A., Koshuntanski, H., & Telesca, L. (2008). A peer-to-peer multidimensional trust model for digital ecosystems. *Proceedings of the 2nd IEEE International Conference on Digital Ecosystems and Technologies* (pp. 461-469). Phitsanulok, Thailand.

[14] Schmidt, S., Steele, R., & Dillon, T. (2007). DEco Arch: Trust and reputation aware service brokering architecture in digital ecosystems. *Proceedings of the Inaugural IEEE International Conference on Digital Ecosystems and Technologies* (pp. 285-291). Cairns, Australia.

[15] Kerschbaum, F., Haller, J., Karabulut, Y., & Robinson, P. (2006). Pathtrust: A trust-based reputation service for virtual organization formation. *Proceedings of the 4th International Conference on Trust Management* (pp. 193-205).

[16] Massa, P., & Avesani, P. (2004). Trust-aware collaborative filtering for recommender systems. *Proceedings of the International Conference on Cooperative Information Systems* (pp. 492-508).

[17] Singh, A., & Liu, L., (2003). TrustMe: Anonymous management of trust relationships in decentralized P2P systems. *Proceedings of the 3rd International Conference on Peer-to-Peer Computing* (pp. 142-149).

[18] Xiong, L., & Liu, L. (2004). Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities. *IEEE Trans. Knowl. Data Eng.*, *16(7)*, 843-857.

[19] Damiani, E., & Vimercati, S. (2003). Managing and sharing servents' reputations in P2P systems. *IEEE Trans. Knowl. Data Engi*, *15*, 840-854.

[20] Kamvar, S., Scholsser, M., & Garcia-Molina, H. (2003). The eigentrust algorithm for reputation management in P2P networks. *Proceedings of the 12th International World Wide Web Conference* (pp. 640-651). Budapest, Hungary.

[21] Teacy, W. T. L., Patel, J., Jennings, N. R., & Luck, M. (2006). Travos: Trust and reputation in the context of inaccurate information sources. *J. Autonom. Agents Multi-Agent Syst.*, *12*.

[22] Swamynathan, G., Zhao, B. Y., Almeroth, K. C., & Zheng, H. (2007). Globally decoupled reputations for

large distributed networks. *Adv. Multimed*.

[23] Google. (June 10, 2014). Android Market. From https://play.google.com/store/apps

[24] Douceur, J. R., & Donath, J. S. (2002). The Sybil attack. *Proceedings of the 1st International Workshop on Peer-to-Peer Systems* (pp. 251-260).

[25] Yu, H., Kaminsky, M., Gibbons, P., & Flaxman, A. (2006). SybilGuard: Defending against Sybil attacks via social networks. *Proceedings of the ACM Annual Conference of the Special Interest Group on Data Communication* (pp. 267-278).

[26] Lesniewski-Laas, C., & Kaashoek, M. F. (2010). Whanau: A Sybil-proof distributed hash table. Presented at the 7th USENIX Conference on Networked Systems (NSDI'10).

[27] University-of-Pennsylvania. (March 5, 2014). TM/RM Simulator. From http://rtg.cis.upenn.edu/qtm/p2psim.php3

Ilung Pranata is a lecturer in IT (information technology) at the University of Newcastle, Australia. His research interests span from IT security, trust and reputation system, internet of things, information system to health informatics area. Dr. Pranata has published over 20 peer-reviewed research articles at various conferences and journals. His current research includes mitigating malicious behavior in rating system, usability and effectiveness of online rating interfaces, adopting technology to achieve weight lost goals, and collaborative online team learning.

Prior to working in academia, Dr. Pranata has worked for and provided consultancies mainly in banking and manufacturing sectors.

Rukshan Athauda is a lecturer at the School of Design, Communication and Information Technology (DCIT) at University of Newcastle, Australia. Dr. Athauda's research interests span from database systems, cloud computing, ICT education to ICT security. Dr. Athauda has published over 40 peer-reviewed research papers internationally.

Prior to working at University of Newcastle, Dr. Athauda has worked at Microsoft Corporation, USA, Sri Lanka Institute of Information Technology (SLIIT), Sri Lanka and High-Performance Database Research Centre (HPDRC) at Florida International University, USA.