DOI 10.12694/scpe.v16i3.1102
Scalable Computing: Practice and Experience ISSN 1895-1767
Volume 16, Number 3, pp. 289–301. http://www.scpe.org

CLASSIFYING AND FILTERING USERS BY SIMILARITY MEASURES FOR TRUST MANAGEMENT IN CLOUD ENVIRONMENT

FATIMA ZOHRA FILALI* AND BELABBAS YAGOUBI†

Abstract. Trust represents an important issue for adopting cloud services. A trust management framework is essentially, about user rating. Hence, correctly addressing user feedback and filtering out malicious rating is a main step in providing reliable services. In order to process their feedback and calculate a reliable trust degree. Thus, new opportunities can be offered for the establishment of a trust relationship among involved entities.

We propose a technique to process user rating by statistical methods. Then, we proceed to classify the users into different groups to detect malicious users. The users are grouped according to their rating by a k-means clustering technique, and the evaluation will show that the proposed solution gives better results than the traditional filtering solution.

Key words: Cloud, Trust Management, Similarity Measure, Filtering, Rating, Distance, Threats, Malicious, Fair, Unfair

AMS subject classifications. 68M14

1. Introduction. For the last years, Cloud services have grown to become an essential paradigm for both industry and academia, by allowing Cloud consumers to rent Computing, network, and storage resources. In that way, the consumers pay for their use of services without apprehensions about maintenance, management or cost.

In spite of all importance of Cloud Computing, most of the organizations are not making a trend of it, and its evolution has raised many concerns and was encountered by various obstacles. Security is one of the most crucial problems for this model, and the risks accompanying the deployment of services and applications are more important with the architecture of Cloud environment [1]. Moreover, in cloud computing environments, the user is dependent on the provider for various services. For many services, the user has to trust the provider for storing his data. Thus, a trust framework should be developed to allow trust establishment, interaction management or requirements share.

With the development of Cloud computing, we are heading towards more distributed and highly available infrastructures, which contribute significantly to the deployment of services anywhere, at anytime and for anyone. To provide reliable services in such infrastructures, we should consider both user's feedback and trust requirements within Cloud environment. Designing a trust management framework requires one to understand what service an entity needs, how to provide reliable feedback from different users, which services can be trusted, how categorize different users to assess feedback, etc.

In a previous work [2], we have proposed a trust model for Cloud computing, based on an opinion model for the subjective dimension and performance parameters for the objective dimension. The proposed model was validated by simulation with well-known trust model. Then, the trust model was integrated in a framework for trust management in Cloud Computing environment [3]. The designed framework identifies the metrics of performance to select the most suitable provider for performing service transaction and integrate it into trust rating process while filtering biased opinions with a simple bias function. This paper is a continuation of these works by improving the credibility evaluator to filter malicious raters.

Hence, this research describes the threats associated with trust management, especially for Cloud environment. Then, we categorize the users according to these threats into four classes. After that, we discuss the issue of filtering biased users, and the different statistical measures and distance. Next, we propose a method for computing the similarities between users. We also describe a mechanism to integrate a reward/punish service to the trust management framework. The next section, show experiments and results of the proposed solution. Finally, we discuss the future work.

2. Related Work. Cloud Security Alliance [4] have identified various security threats to cloud computing. In [5], the author classify several vulnerabilities and attacks that can be encountered in Cloud Computing, at

*Department of Computer Science, University of Oran1 Ahmed Ben Bella, Oran, Algeria (filalifz7@gmail.com).
†Department of Computer Science, University of Oran1 Ahmed Ben Bella, Oran, Algeria (byagoubi@gmail.com).

different security points such as : Abuse and Nefarious Use of Cloud, Insecure Interfaces and APIs, Malicious Insider, Virtualized Technology... However, these threats represent various aspects of security. To ensure the proper implementation of a trust management framework, specific threats must be considered. Hence, we focus on behaviour attacks in Cloud Computing.

For a trust management framework, a vulnerability is a weakness in the system that could be exploited to influence the recommendation and the trust of the provided services. Cloud computing is just as vulnerable as any other technology that uses the public internet for connectivity. The vulnerabilities include malicious attack, a man-in-the-middle attack, Sybil attack, denial-of-service attack, etc. Furthermore, Various types of rating attack against the trust management systems such as Bad mouthing attack, Feedback Collusion, Sybil attack, Reputation lag attack have been discussed in [6, 7, 8, 9, 10, 11].

In many researches, it has been demonstrated that users with false feedback have some common features. In [10], it has been identified that malicious users who have similar characteristics such as a higher request frequency to surpass honest users. As, users who try to increase or decrease a service popularity submit feedbacks frequently. They also tend to usually engage in minimum value transactions to meet the requirements of submitting a rating. Furthermore, malicious ratings tend to be either significantly lower or higher than the majority of the users [13]. Thus, a trust management system should have the ability to weigh the ratings of highly credible users more than those with a low credibility rating. There are several approaches that evaluate trustworthiness of users based on majority opinion, such as beta filtering feedback [14]. This approach works as long as the majority of ratings are not from a group of users that tend to falsify their ratings.

Another approach that uses beta probability density function to estimate the reputation of a provider as either bad or good is discussed in [15].This approach was later extended such that a feedback is considered to be fair if it falls in the range of lower and upper boundaries among all the ratings [9]. The limitation of this strategy is that users could collude as a group to manipulate the majority ratings. However, majority ratings scheme alone is not sufficient to accurately measure the trustworthiness of a user.

The authors [16] proposed models based on assumption that all customers in the system have provided feedbacks for a given period of time. For example, new users could be treated as bad users and their feedback will carry less weight in trust assessment.

Most of these approaches are based on similarity filtering technique such as [9, 14]. In this technique, the users with low similarity rating are considered as less truthful. These approaches have been proved very effective in term of filtering malicious users. Hence, we are going to use a similarity-based technique to proceed to a first phase which will consist at predicting missing values for rating services. We point out that the focus of our study is not computing the degree of trust but filter out malicious users to provide the trust management framework with reliable rating.

3. Cloud Computing trust threats. In this section, we will discuss some threats and vulnerabilities about trust management system. Various researchers [9, 11, 17] have discussed rating attacks and threats against trust management systems. We will present and describe the most common security threats applicable in the field of trust and reputation management over Cloud computing environments:

Cold-start problem: reference the issue where new services or new user recommendations meet difficulties to provide an adequate rating for the system. A service cannot be recommended unless it has a sufficient number of user ratings. As other users in the system tend to interact with high reputable services, the chance of a new service being selected for interaction is generally rare [18].

Malicious feedback: reference the issue where users report falsely their feedback, creating errors in the system. These malicious feedbacks can be either individual malicious where a unique user always provide false feedback for services or collective malicious when two of more users collectively boost a service reputation or conspire against a service provider. Hence, a trust management system must exclude unfair ratings. A common solution consists to use their statistical properties [19, 20].

Playbooks: a playbook consists to maximize the profit of a service according to certain criteria. For example, a provider can act honestly and provide quality services over a period to gain a high reputation score, then with his high reputation score providing low-quality services at a low production cost [21]. Thus, a trust management system must consider the property of auto-adjusting of the service rating over time.

Sybil attack: malicious users may acquire multiple identities. Each time it provides a false feedback in the

system, he replaced it with a new identity. For that propose, a good trust framework must integrate a reliable authentication service based not only on security but also on filtering out phantom identity. For our present work, we didn't consider this kind of attacks. From the discussed threats and vulnerabilities, we can observe that generally four groups of users can be categorized, either for a user or for a provider of the service. Hence, we assume four main group in our work:

1. Fair Positive (FP), when an entity is providing honest rating about a service with a high quality.
2. Fair Negative (FN), when an entity is providing honest rating about a service with a low quality.
3. Unfair Positive (UP), when an entity is providing malicious rating about a service with a high quality (providing unreasonably increased feedback assessments).
4. Unfair Negative, when an entity is providing malicious rating about a service with a low quality (providing low feedback assessments).

We also observe that the principal part of providing a reliable trust management system is to correctly identify the unfair rating. To achieve that it is essential to correctly identify in which groups the user is classified.

Hence, in this paper we propose to group users into four groups by using a learning technique such as k-means. The problem with that method is the missing value for many users. Consequently, before classifying groups we treat the problem of cold start by using a statistical method. Finally, as it has been pointed out in [22] we integrate a service for rewarding honest users groups and punish malicious groups.

4. Measuring rating similarities. The users of different services may provide their feedbacks about the services they consumed to present their satisfaction or dissatisfaction. Based on the provided feedbacks, the system re-evaluates trust rates for the services and service providers. However, the users may provide unfair and false feedback, so it is important to detect such ratings. Feedback filtering component is necessary to address the discussed threats. Some techniques to prevent unfair feedback have been presented in different studies [11, 12, 13, 14]. Although there are many algorithms to filter the unfair rating, the basic idea is to calculate different measures of similarity between users.

There are several similarities algorithms [23, 24] that have been used in filtering field Pearson correlation, cosine vector similarity, Euclidean distance and Minkowski distance. These measures can be effectively used to balance the prediction algorithm in the meaning of the ratings, therefore, to improve accuracy.

4.1. Pearson's correlation. Pearson's correlation measures the linear correlation between two sequences of ratings for the users x and the user y about the set of services m rated by both user x and user y.

$$Sim\_Pearson(x,y) = \frac{\sum_{i=1}^{m}(R_{ix} - \overline{R_{ix}})(R_{iy} - \overline{R_{iy}})}{\sqrt{\sum_{i=1}^{m}(R_{ix} - \overline{R_{ix}})^2 \sum_{i=1}^{m}(R_{iy} - \overline{R_{iy}})^2}} \tag{4.1}$$

where R_{ix} is the rating of the service i by the user x, $\overline{R_{ix}}$ is the average rating of user x, and R_{iy} is the rating of the service i by the user y, $\overline{R_{iy}}$ is the average rating of user y.

4.2. Cosine measure. The cosine measure looks at the angle between two sequences of ratings where a greater similarity imply smaller angle, as following formula:

$$Sim\_Cosine(x,y) = \frac{\sum_{i=1}^{m} R_{ix}R_{iy}}{\sqrt{\sum_{i=1}^{m} R_{ix}^2 \sum_{i=1}^{m} R_{iy}2}} \tag{4.2}$$

where R_{ix} is the rating of the service i by the user x, $\overline{R_{ix}}$ is the average rating of user x, and R_{iy} is the rating of the service i by the user y, $\overline{R_{iy}}$ is the average rating of user y.

4.3. Euclidean distance. A Euclidean distance represents the distance between two points in Euclidean space. We are going to use this distance to measure the distance between sequences of ratings for two users x and y about m services rated by both users.

$$Dis\_Euclidean(x,y) = \sqrt{\sum_{i=1}^{m}(R_{ix} - R_{iy})^2} \tag{4.3}$$

where R_{ix} is the rating of the service i by the user x, and R_{iy} is the rating of the service i by the user y.

4.4. Minkowski distance. The Minkowski distance can be considered as a generalization of both the Euclidean distance and the Manhattan distance. We used this metric to measure the distance between two sequences of ratings.

$$Dis\_Minkowski(x,y) = (\sum_{i=1}^{m}(R_{ix} - R_{iy})^p)^{\frac{1}{p}} \tag{4.4}$$

where R_{ix} is the rating of the service i by the user x, and R_{iy} is the rating of the service i by the user y.

4.5. Hamming Distance. The Distance of Hamming is used in information theory, to measure the difference between two set of strings or two sequence of bits. In another way, it measures the minimum number of errors that could have transformed one string into the other.

We are going to use this distance to measure the difference in two sequence of ratings, as described in the following formulas:

$$Dis\_Hamming(x,y) = \sum_{i=1}^{m}[R_{ix} \neq R_{iy}] \tag{4.5}$$

In this equation $Dis\_Hamming(x,y)$ is the Hamming distance between the user x and the user y, i is the index of the service rated for a total of n services. The Hamming distance gives the number of mismatches between the rating of user x and user y.

5. The proposed solution. For assess a reliable trust degree, it is important to filter unfair rating for the different services. Literature demonstrates that filtering methods successfully provide abundant evidence. However, there are some ways inadequate since the various filtering methods has cold start problems.

In this paper, we propose to classify users in groups. For this propose, we use the k-means clustering algorithm [25], to form user groups depending on the rating. However, the machine learning techniques suppose the presence of full knowledge about the user rating for different services. However, in a cloud environment and especially for a trust management system, these suppositions do not provide accurate results. Many of statistical literature deals with this case by replacing randomly the missing data, which will result in an inaccurate and biased estimator.

For that reason, we propose to pre-process the set of rating before proceeding to the classification, and since statistical techniques offer better results, we propose a hybrid technique based on the different measures discussed in the previous section.

We employ the neighbours to help identify users' classes. We find that the neighbour group mean deviation is small when the current service is objective, while it is large when the current service is subjective. Then, the resulting classes are subject to either recompensing for the fair groups or penalizing for the unfair groups.

5.1. Algorithm. We consider a set of users with fairly positive feedbacks, fairly negative feedbacks, unfairly positive feedbacks and unfairly negative feedbacks. For each user i, we have a sequence of Ni ratings about different services. The classification algorithm involves that the user rate all used services. Each user is represented by the term rating R_{ij} in the user list sequence, which contains the ratings have been expressed by a user i for the service j, as described in the following table.

Like discussed before, in a real environment it is impossible for a user to use each proposed service. Therefore, the presented table will contain many missing values for different service, which will result in an accurate or biased classification.

For that reason, we proceed to a pre-process phase where we predict the missing values by using the discussed statistical measures in the previous section. The resulting matrix values are then classified into four clusters using the k-means clustering algorithm [25]. We employ the resulting neighbors to help identify users' classes. We assume that the rating set whose mean is closer to the mean of neighbors' ratings may be from honest peer. Among these four sets, the rating set with the least group mean deviation may be the dishonest ones.

TABLE 5.1
Matrix of Users Rating

Users & Services	S_1	S_2	...	S_m
U_1	R_{11}	R_{21}	...	R_{m1}
U_2	R_{21}	R_{22}	...	R_{m2}
...	...	...	...	...
U_n	R_{n1}	R_{n2}	...	R_{nm}

Note that this solution is proposed in the context of trust-dependent assumption that a service rating for a user is dependent on provided quality of service. The procedures of the proposed solution are as follows.

Step 1: Select the neighbours for each user

Algorithm 1 select_neighbors (user x)

```
Begin
  for user y ≠ x do
    d ← Dis_Hamming(x, y)                          ▷ Formula 4.5
    if d < 0.5 then
      add y to neighbors_list
    end if
  end for
  return neighbors_list
End
```

Step 2: Calculate the similarities between users

Algorithm 2 get_similar_users (user x)

```
Begin
  for user y ≠ x do
    s1 ← Sim_Pearson(x, y)                         ▷ Formula 4.1
    s2 ← Sim_Cosine(x, y)                          ▷ Formula 4.2
    d1 ← Dis_Euclidean(x, y)                       ▷ Formula 4.3
    d2 ← Dis_Minkowski(x, y)                       ▷ Formula 4.4
    if (s1 ≥ 0.5)AND(s2 ≥ 0.5)AND(d1 ≤ 0.5)AND(d2 ≤ 0.5) then
      add y to similar_users_list
    end if
  end for
  return similar_users_list
End
```

Step 3: Compute the predicted missing rates for each user

Step 4: Classify users The resulting matrix values of rating are then classified into four clusters using the k-means clustering algorithm: FP, FN, UP, UN.

5.2. Self-Adjusting classification. Self-adjusting is an important issue where the user groups are established dynamically to reflect recent interactions and rating. The proposed solution would detect unfair users, changes in rating rates, and reforms new groups of users, by periodically performing clustering and detecting new groups of users.

5.3. Punishment and Reward. According to the resulted groups, a last step of punishment or recompense has to be performed. This punishing or rewarding mechanism works by decreasing or increasing, respectively, the weights assigned to each source of rating, which will depend on the user groups.

TABLE 6.2
Rating models

Filtering techniques	Users classes	Reference Classes FP	FN	UP	UN	Total Predicted	Accuracy % Acc.	MAE
Predicted classes	FP	**136**	2	10	1	149	91%	0.180
	FN	13	**146**	0	5	164	97%	0.147
	UP	1	0	**86**	10	97	86%	0.250
	UN	0	2	4	**84**	90	84%	0.220
	Total	150	150	100	100	500	90%	0.200
Overall Accuracy	452/500 = 90%							

with r_1, r_2, r_3, ..., r_n is the prediction of users' ratings, and c_1, c_2, c_3, ..., c_n is the corresponding real rating data set of users.

The metric RMSE is defined as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(r_i - c_i)^2}{n}} \tag{6.2}$$

The lower the MAE and RMSE, the more accurate the predictions would be.

In the same way, we used this two metrics to evaluate the accuracy of the classified groups.

To evaluate the prediction accuracy of our approach, we compare it with the two following approaches:

- Random: before proceeding to groups clustering, the missing values are generated randomly.
- Mean: the missing value for a user x about a service i is substituted by the value of rating mean for the user x.

6.3. Experiments. The simulation environment consists of cloud service users and cloud service providers. The simulation proceeds in simulation cycles. Each simulation cycle cloud users proceed to service rating corresponding to the model detailed in Table 6.2.
The results reported in this section were obtained assuming that 30% of the users are positive fair, 30% are positive unfair, 20% are positively biased, and 20% are negatively biased. In addition, we assumed that the number of ratings NR =10, with 1 being the lowest and 10 being the highest. (The selection of NR=10 is not significant, and any other values can be readily used).

6.3.1. Experiment 1. In this set of experiments, the total number of users is 500, and the total number of services is 25.

The results indicate that the proposed solution obtained high values for classification accuracy (84% - 97%) for each class of users (Table 6.2). The table shows that the overall classification accuracy of the proposed approach was 90%, which represents a high value.

For more model validation, we have conducted a repeated random sampling for 10 times for the proposed solution with the random filtering and the mean filtering. The experimental results of the accuracy is shown in Figure 6.1 for Fair Positive, Figure 6.2 for Fair Negative, Figure 6.3 for Unfair Positive, and Figure 6.4 for Unfair Negative. These figures shows that the proposed solution obtained the best classification accuracy representing for all the simulation sounds.

We also present the best, average and worst cases for each techniques, in Table 6.3.

From the results of the three groups of experiments, we can see that in the best case, our solution improves over Random Filtering and Mean filtering techniques respectively by 56%-15% in term of MAE and by 47%-29% in term of RMSE. In the worst case, the improvements increase to 61%-45% in term of MAE and 49%-29% in term of RMSE. This result means that our model has better robustness. In other words, it not only performs well in the best case but also overcomes the worst-case situations with slightly lower accuracy.

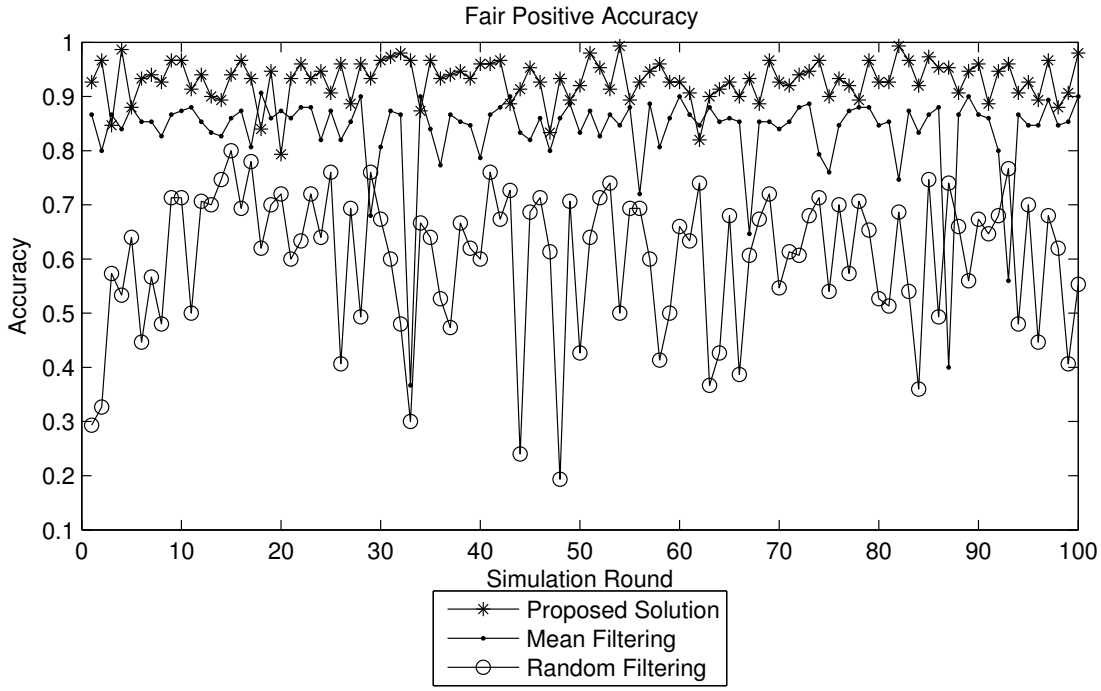

FIG. 6.1. *Accuracy of FP users*

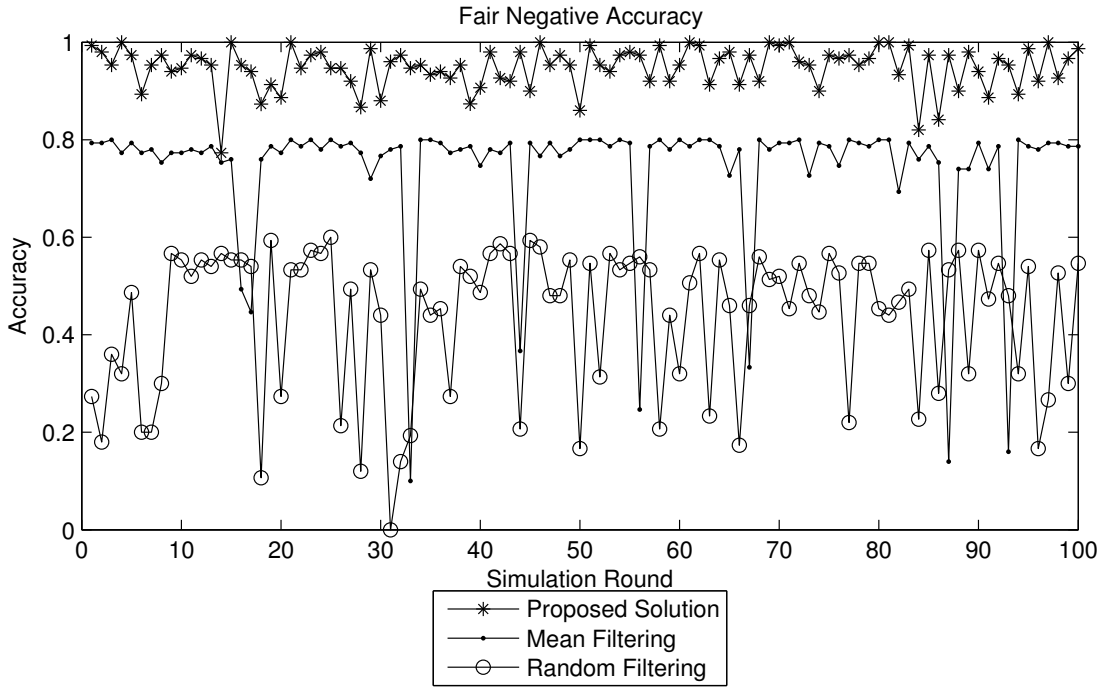

FIG. 6.2. *Accuracy of FN users*

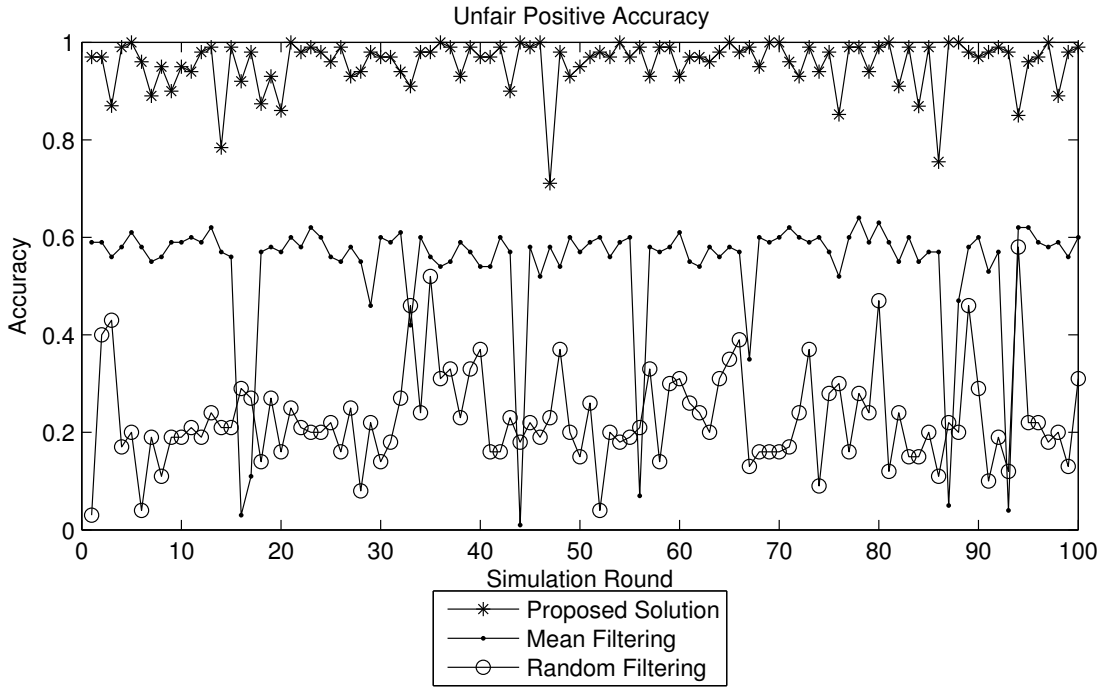

FIG. 6.3. *Accuracy of UP users*

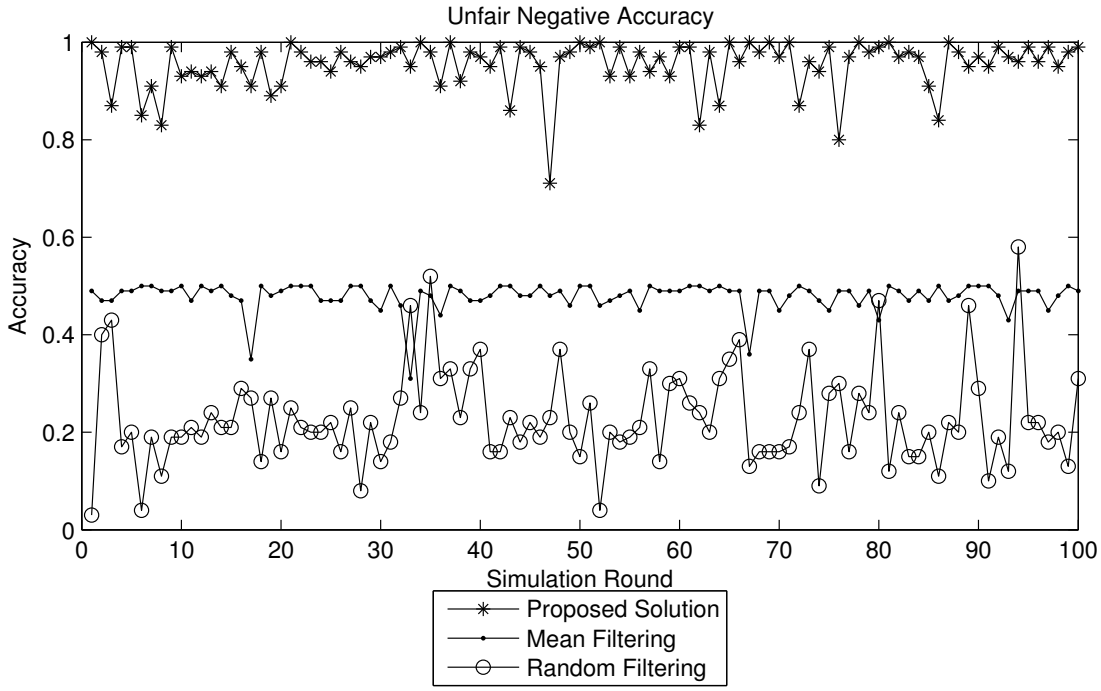

FIG. 6.4. *Accuracy of UN users*

TABLE 6.3
Accuracy of filtering techniques

Cases	Metrics	Proposed Filtering	Random Filtering	Mean Filtering
Best	MAE	**0.162**	0.365	0.190
	RMSE	**0.220**	0.419	0.311
Average	MAE	**0.181**	0.383	0.239
	RMSE	**0.245**	0.464	0.398
Worst	MAE	**0.193**	0.498	0.348
	RMSE	**0.297**	0.586	0.41

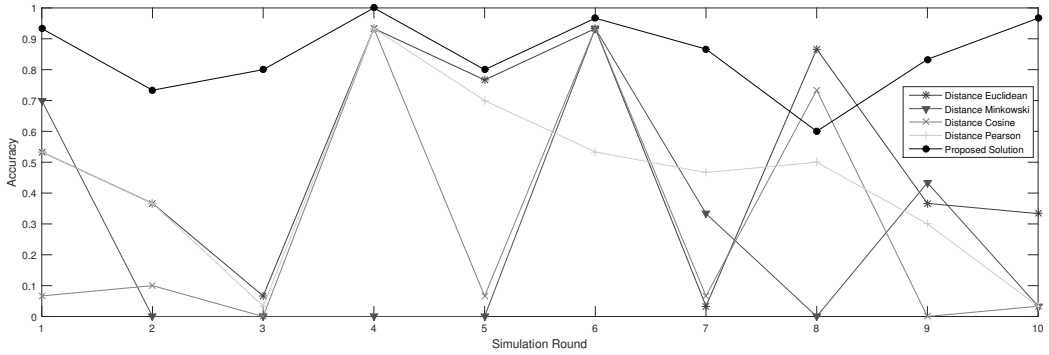

FIG. 6.5. *Accuracy of FP users*

6.3.2. Experiment 2. The results reported in this section were obtained assuming that the total number of users is 500, and the total number of services is 25.

To validate the proposed solution, we have conducted a repeated random sampling for 10 times for the proposed solution with the discussed solutions in section 4 namely Pearson's correlation, Cosine measure, Euclidean distance and Minkowski distance. the experiments were conducted for each measure separately with the same generated sampling.

The experimental results of the accuracy is show in Figure 6.5 for Fair Positive, Figure 6.6 for Fair Negative, Figure 6.7 for Unfair Positive, Figure 6.8 for Unfair Negative. The figure shows that the proposed solution obtained the best classification accuracy representing for all the simulation sounds.

From the results of the five groups of experiments, we can see that our solution obtained the best result in most of the cases while maintaing a definite stability. The result proves that the intersection of the resulted users rating from each measure permit to give better reliability, resulting in a better accuracy of the filtered users.

6.3.3. Experiment 3. In these experiments, we have tested the scalability of the proposed solution regarding the number of users and the number of services.

For the first experiment, we have conducted a repeated random sampling for 10 times with 250 users while increasing the number of the rated services. the experiment was started with 3 services, and increased each round by 5 services, as shown in Figure 6.9.

We can remark from the results that the accuracy of the reported values starts with a very low value which was 20% for the unfair positive group. However, the accuracy increases each round to achieve 100% for services between 23 and 28. This is due to the fact that the low number of services can't give a whole overview of the given rating. Then, with each round the accuracy increases and reaches an average of 80%-90% for services between 8 and 13 which remains very reliable and realistic situations.

The second experiment consists of a repeated random sampling for 10 times with 10 services while increasing

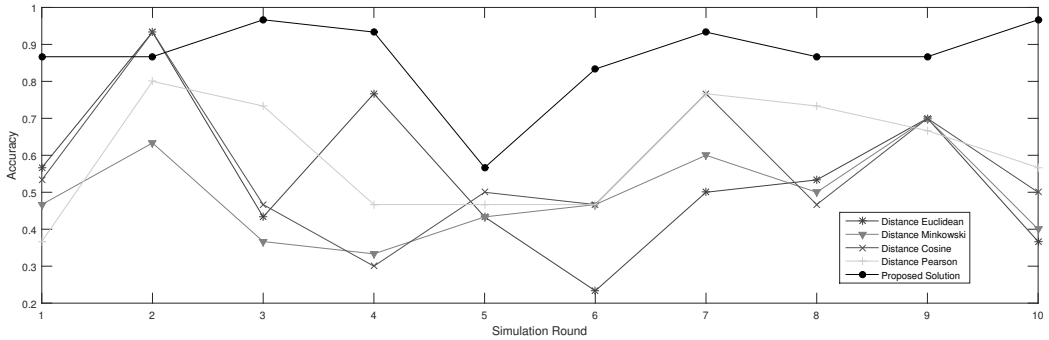

FIG. 6.6. *Accuracy of FN users*

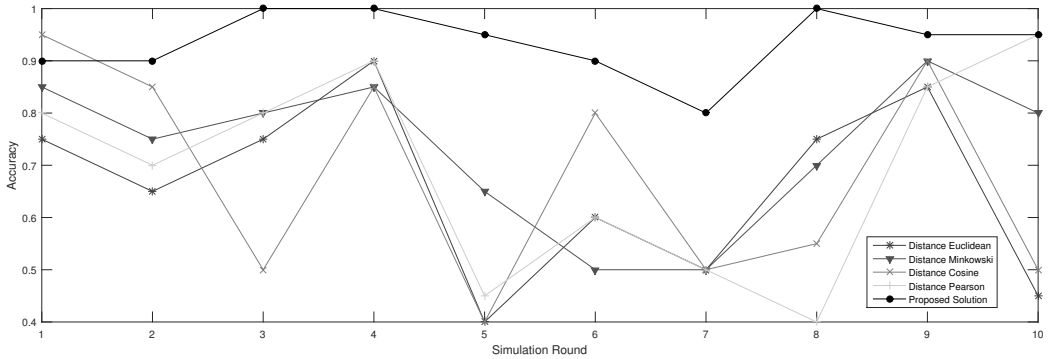

FIG. 6.7. *Accuracy of UP users*

the number of the user ratings. the experiment was started with 200 users, and increased each round by 50 users, as shown in 6.10. We can notice from the results that the accuracy remains stable for each group, which lies between 80% in the worst cases and 100% for the best cases. Hence, we can conclude that the increasing or decreasing of the users number doesn't alter the proposed solution and the results remain very accurate.

Consequently, we can conclude that the experimental results have demonstrated that the proposed technique significantly increase the predicting of the missing values for users rating, to proceed to an accurate clustering of the different groups. This is due to the following reasons. First, in our solution, take into account several measures of distance and similarities comprehensively. Second, the filtering technique would be an appropriate start to compute a reliable trust degree and assess the quality of the proposed services since the clustering is performed without any knowledge about the offered quality of service. Third, the integration of a module to reward or punish the different groups would result in a more accurate trust degree for a trust management system, likewise for the auto-adjusting classification.

7. Conclusion and Future Direction. To protect clouds, traditional security techniques such as encryption and authorization provide a solid basis, but they are insufficient when entities act maliciously over reputations and trust of different service. Trust as a security approach can fight against such threats by restricting malicious entities from participating in interactions and consequently offers a high trustworthiness cloud computing environment. If feedback and rating integrity concerns in trust management and service selection are not addressed sufficiently, the consequences may be severe.

This paper presents a novel classifying model that uses the rating provided by different users by grouping them into four different groups to filter the discussed user attacks. This model can be integrated into a trust management framework to obtain a more specific trust value of the same concept of services. As a future work,

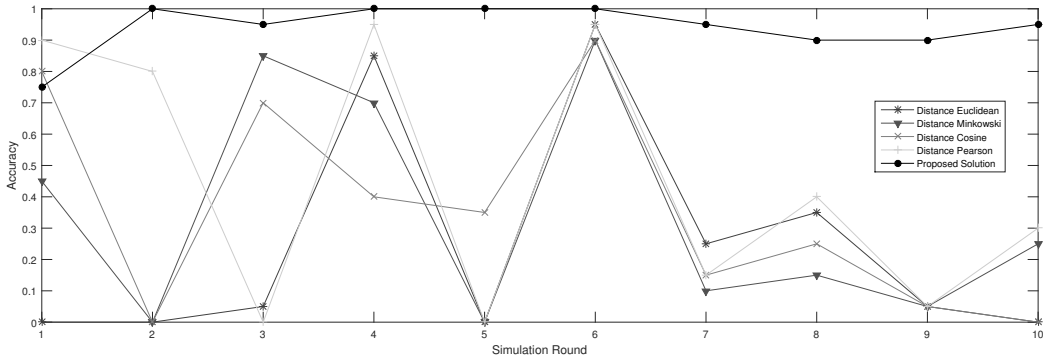

FIG. 6.8. *Accuracy of UN users*

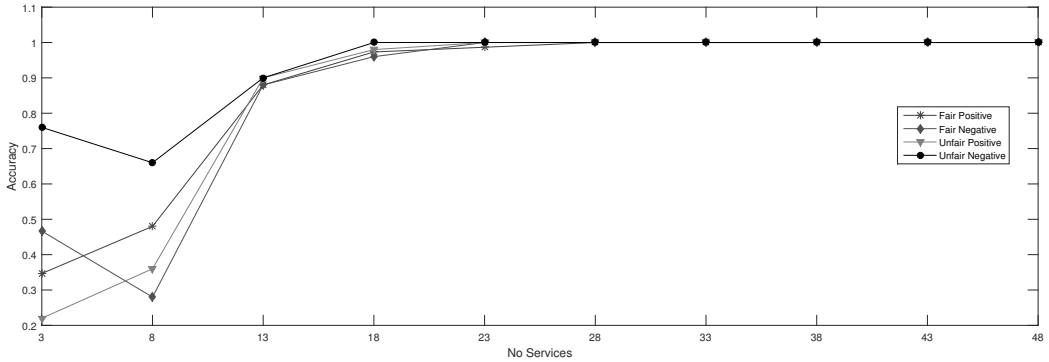

FIG. 6.9. *Services Scalability*

we will discuss design requirements and integration in a trust management system, as well as integrating an authentication service to deal with Sybil attack.

REFERENCES

[1] D. PUTHAL, B. P. S. SAHOO, S. MISHRA, AND S. SWAIN, *Cloud Computing Features, Issues and Challenges: A Big Picture*, no. Cine, 2015.

[2] F. Z. FILALI, B. YAGOUBI, *Global Trust: A Trust Model for Cloud Service Selection*, International Journal of Computer Network and Information Security, vol. 7, pp. 41-50, 2015.

[3] F. Z. FILALI, B. YAGOUBI, *A General Trust Management Framework for Provider Selection in Cloud Environment*, Unpublished conference paper at: 19th East European Conference on Advances in Databases and Information Systems (ADBIS 2015), September 8-11, 2015, Poitiers in France.

[4] G. ZHAO, C. RONG, M. G. JAATUN, AND F. E. SANDNES, *Deployment models: Towards eliminating security concerns from cloud computing*, Proc. 2010 Int. Conf. High Perform. Comput. Simulation, HPCS 2010, pp. 189-195, 2010.

[5] M. BAMIAH AND S. BROHI, *Seven deadly threats and vulnerabilities in cloud computing*, Int. J. Adv. Eng. Sci. & Techs., no. 9, pp. 87-90, 2011.

[6] Y. L. SUN, Z. HAN, W. YU, AND K. J. R. LIU, *A trust evaluation framework in distributed networks: Vulnerability analysis and defense against attacks*, Proc. IEEE INFOCOM 2006. 25TH IEEE Int. Conf. Comput. Commun., pp. 1-13, 2006.

[7] T. H. NOOR, Q. Z. SHENG, AND A. ALFAZI, *Reputation attacks detection for effective trust assessment among cloud services*, Proc. - 12th IEEE Int. Conf. Trust. Secur. Priv. Comput. Commun. Trust. 2013, pp. 469-476, 2013.

[8] W. FAN AND H. PERROS, *A reliability-based trust management mechanism for cloud services*, Proc. - 12th IEEE Int. Conf. Trust. Secur. Priv. Comput. Commun. Trust. 2013, pp. 1581-1586, 2013.

[9] A. JOSANG AND J. GOLBECK, *Challenges for robust trust and reputation systems*, 5th Int. Work. 5th Int. Workshop on Security and Trust Management, pp. 1-12, 2009.

[10] K. THIRUNARAYAN, P. ANANTHARAM, C. HENSON, AND A. SHETH, *Comparative trust management with applications: Bayesian*

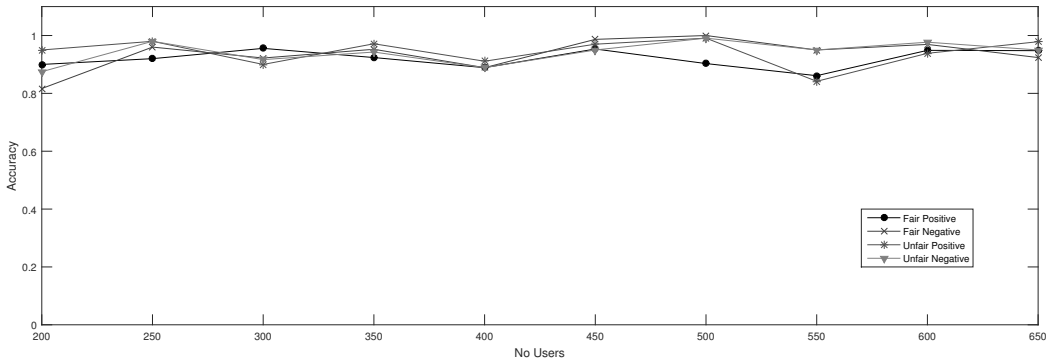

FIG. 6.10. *Users Scalability*

approaches emphasis, Futur. Gener. Comput. Syst., vol. 31, pp. 182-199, 2014.

[11] M. TAVAKOLIFARD AND K. C. ALMEROTH, *A Taxonomy to Express Open Challenges in Trust and Reputation Systems*, J. Commun., vol. 7, no. 7, pp. 538-551, Jul. 2012.

[12] J. TREVATHAN AND W. READ, *A simple shill bidding agent*, Proc. - Int. Conf. Inf. Technol. Gener. ITNG 2007, pp. 766-771, 2007.

[13] R. KERR AND R. COHEN, *Smart Cheaters Do Prosper: Defeating Trust and Reputation Systems*, Scenario, pp. 993-1000, 2009.

[14] A. WHITBY, A. JOSANG, AND J. INDULSKA, *Filtering out unfair ratings in bayesian reputation systems*, Chem. Biodivers., vol. 1, no. July, pp. 1829-1841, 2005.

[15] A. JOSANG AND W. QUATTROCIOCCHI, *Advanced features in Bayesian reputation systems*, Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 5695 LNCS, pp. 105-114, 2009.

[16] B. YU AND M. P. SINGH, *Detecting Deception in Reputation Management*, Proc. Second Int. Jt. Conf. Auton. agents & multiagent Syst., pp. 73-80, 2003.

[17] F. G. MARMOL AND G. M. PEREZ, *Security threats scenarios in trust and reputation models for distributed systems*, Comput. Secur., vol. 28, no. 7, pp. 545-556, Oct. 2009.

[18] H. J. AHN, *A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem*, Inf. Sci. (Ny)., vol. 178, pp. 37-51, 2008.

[19] D. CHRYSANTHOS, *Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior*, Proc. 2nd ACM Conf. Electron. Commer., pp. 150-157, 2000.

[20] M. CHEN AND J. P. SINGH, *Computing and using reputations for internet ratings*, Proc. 3rd ACM Conf. Electron. Commer. - EC '01, pp. 154-162, 2001.

[21] R. KERR AND R. COHEN, *Modeling Trust Using Transactional , Numerical Units*, PST 06 Proc. 2006 Int. Conf. Priv. Secur. Trust Bridg. Gap Between PST Technol. Bus. Serv., pp. 1-11, 2006.

[22] F. G. MARMOL, C. SORGE, O. UGUS, AND G. M. PEREZ, *WSANRep, WSAN reputation-based selection in open environments*, Wirel. Pers. Commun., vol. 68, pp. 921-937, 2013.

[23] J. S. LEE, C. H. JUN, J. LEE, AND S. KIM, *Classification-based collaborative filtering using market basket data*, Expert Syst. Appl., vol. 29, pp. 700-704, 2005.

[24] S. GONG, *An efficient collaborative recommendation algorithm based on item clustering*, Lect. Notes Electr. Eng., vol. 72 LNEE, pp. 381-387, 2010.

[25] J. G. J. GU, J. Z. J. ZHOU, AND X. C. X. CHEN, *An Enhancement of K-means Clustering Algorithm*, 2009 Int. Conf. Bus. Intell. Financ. Eng., vol. 2, no. 2, pp. 2-5, 2009.

[26] G. LEKAKOS AND G. M. GIAGLIS, *A hybrid approach for improving predictive accuracy of collaborative filtering algorithms*, User Model. User-adapt. Interact., vol. 17, pp. 5-40, 2007.

[27] Y. YAO, H. TONG, X. YAN, F. XU, AND J. LU, *MATRI: a multi-aspect and transitive trust inference model*, Proc. 22nd Int. Conference on World Wide Web, pp. 1467-1476, 2013.

Edited by: Dana Petcu
Received: Apr 9, 2015
Accepted: Aug 15, 2015