 CMC, vol.59, no.3, pp.805-815, 2019

A Neural Network-Based Trust Management System for Edge Devices in Peer-to-Peer Networks

Alanoud Alhussain[1], Heba Kurdi[1,*] and Lina Altoaimy[2]

Abstract: Edge devices in Internet of Things (IoT) applications can form peers to communicate in peer-to-peer (P2P) networks over P2P protocols. Using P2P networks ensures scalability and removes the need for centralized management. However, due to the open nature of P2P networks, they often suffer from the existence of malicious peers, especially malicious peers that unite in groups to raise each other's ratings. This compromises users' safety and makes them lose their confidence about the files or services they are receiving. To address these challenges, we propose a neural network-based algorithm, which uses the advantages of a machine learning algorithm to identify whether or not a peer is malicious. In this paper, a neural network (NN) was chosen as the machine learning algorithm due to its efficiency in classification. The experiments showed that the NNTrust algorithm is more effective and has a higher potential of reducing the number of invalid files and increasing success rates than other well-known trust management systems.

Keywords: Trust management, neural networks, peer to peer, machine learning, edge devices.

1 Introduction

Internet of things (IoT) is a newly emerging technology that connects enormous numbers of different devices (edge devices) and allows them to exchange and share data as well as other resources. As the number of connected devices is growing exponentially, there has become a need to have a more reliable and scalable distributed system with open architectures to efficiently handle the growing demand placed on IoT applications [Steffenel, Pinheiro, Peres et al. (2018)].

A peer-to-peer (P2P) network, which is one of the most widely used networks for sharing files [Bradai, Abbasi, Landa et al. (2014)], has the opportunity to improve the direction of IoT and enhance its applications.

P2P consists of a collection of peers without a centralized control, which makes it more flexible and encourages dynamic and rich communications [Bashmal, Almulifi and Kurdi (2017)]. Thus, it allows connected devices to form peers and facilitates the communication among them [Xie, Yuan, Zhou et al. (2018)]. However, the open nature

[1] Computer Science Department, King Saud University, Riyadh, 11451, Saudi Arabia.

[2] Information Technology Department, King Saud University, Riyadh, 11451, Saudi Arabia.

[*] Corresponding Author: Heba Kurdi. Email: hkurdi@ksue.edu.sa.

of a P2P network means that it also has multiple security threats, which make it highly vulnerable to attacks from different types of peers [Xie, Yuan, Zhou et al. (2018)] such as malicious peers and free-riders. For instance, malicious peers are able to upload inauthentic files to compromise the network [Chuang (2017)]. They may also collaborate with other malicious peers to improve their reputations and interpolate good peers [Fan, Liu, Li et al. (2017)]. Therefore, it is crucial to identify malicious peers and isolate them from networks, thus allowing good peers to share resources without the fear of such malicious behaviors [Kurdi (2015)]. Many mechanisms have been introduced to solve the problems caused by malicious peers [Hawa, Al-Zubi, Darabkh et al. (2017)]. Reputation and trust management are examples of well-known and powerful mechanisms that reduce the effect of malicious behaviors in P2P networks. Despite their efficiency, existing reputation management algorithms face difficulties in identifying malicious groups [Alkharji, Kurdi, Altamimi et al. (2017)]. Malicious groups are intelligent models of malicious peers that form groups and mislead good peers [Alhussain and Kurdi (2018)]. They do this by giving malicious peers high reputation values or by being inconsistent about the authenticity of the files they provide [Kurdi, Alfaries, Al-Anazi et al. (2018)].

Trust management systems are crucial in all environments that involve exchanging and sharing data or services between different entities, including edge computing, cloud computing, and P2P network environments. Number of reputation systems have been introduced in the P2P network, which is considered a main precursor for edge computing [Lopez, Montresor, Epema et al. (2015)]

The EigenTrust algorithm [Kamvar, Schlosser and Garcia-Molina (2003)] is a well-established reputation algorithm; it has many reputation systems that appear to enhance it in order to eliminate the need for pre-trusted peers, as seen in Kurdi [Kurdi (2015)]. This study hypothesizes that the honest peer is the one with the maximum reputation value. A similar goal was discussed in the trust mirroring that was used in Shirgahi et al. [Shirgahi, Mohsenzadeh and Javadi (2017)] to predict trust in social networks without the need for global trust information. However, a limitation existed in that study: the trust value could not be calculated unless direct trust existed between the two nodes.

The inverse of the PageRank (PR) algorithm used in Lee [Lee (2016)] is the idea that if one can recognize peers as dishonest, he can reasonably filter out their dishonest friend peers from the environment, which is the opposite of the approach taken with most reputation systems, such as subjective logic [Jsang, Hayward and Pope (2006)] and its variants [Kurdi, Alfaries, Al-Anazi et al. (2018); Kurdi, Alshayban, Altoaimy et al. 2018)].

Despite that the previous mentioned schemes have improved its efficiency, there is still a need to identify more complicated types of malicious peers, such as collective malicious peers.

To the best of our knowledge, no studies have combined the advantages of traditional trust algorithms and machine learning. Therefore, this paper introduces the NNTrust algorithm for the purpose of reputation management in P2P networks. It focuses on recognizing malicious peers using a neural network algorithm that analyzes peers' transaction histories. In addition, we use a well-controlled experimental framework to evaluate NNTrust against other benchmark algorithms.

The remainder of this paper is organized as follows. Section 2 introduces the proposed NNTrust algorithm. Sections 3 and 4 discuss the evaluation plan and the experimental results. Finally, Section 5 summarizes and concludes the paper.

2 System design

This section describes the design and architecture of the proposed NNTrust system, which determines trustworthiness among interacting edge devices (peers) to enhance the robustness of the file-sharing network in a P2P environment. The NNTrust system uses EigenTrust to compute local and global trust values among peers and an NN to predict malicious behaviors. The architecture of the NNTrust system, shown in Fig. 1, consists of the following components:

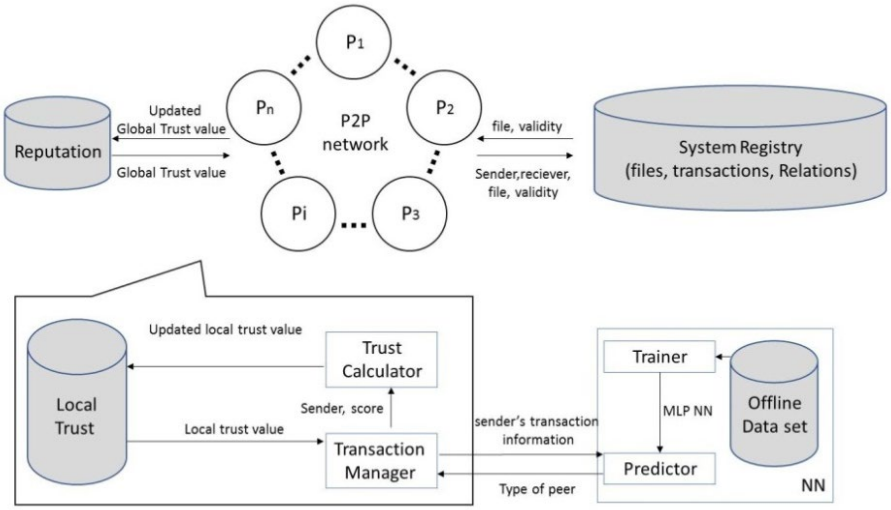

Figure 1: System architecture

1. System registry:

This database contains a list of the files and their owners. It arranges the file requests as a queue of transaction requests, and it links each receiver to the sender in a relationship. It also maintains all files and dynamically updates their statuses.

2. Reputation database management system:

This database stores and updates the reputation matrix calculated using EigenTrust.

3. Peers:

Each peer has the following three components:

3.1. Trust calculator: It receives the sender and its score from the transaction manager to compute the trust matrix, and it sends the updates to the local trust database.

3.2. Trust database: It contains the matrices of trust for all peers in the system and the history of the peer's transactions itself. It also receives requests from the trust calculator to store or update the matrix and the to-whom list after each transaction.

3.3. Transaction manager: It contains the list of files and accepts the requests for the files.

It finds the best senders and sends their information to the NN to predict whether the senders are malicious. In addition, it rates and normalizes the received files. It also sends the scores of rated senders and their files to the trust calculator.

4. Neural network (NN):

The NN contains the following:

4.1. The trainer: It takes the offline dataset and constructs the layers of the NN to teach the network how to classify each peer. The trained network is used later by the predictor to predict the type of sender.

4.2. The predictor: It receives the peer's information, analyzes it to extract the peer's features, and enters it into the NN to predict the type of provider. The transaction manager, based on the type of provider, will decide to either accept or reject its offer.

3 Evaluation methodology

This paper introduces a very well-controlled evaluation framework in a P2P network model. All the control variables and performance measures were obtained from the application field, which is file sharing to increase the level of assurance that no specific strategy in the selection is favored over any other strategies. Two performance measures were considered:

1. The percentage of invalid files exchanged by good peers, which needs to be minimized.
2. The success rate, which represents the number of valid files received by good peers over the number of transactions attempted by good peers as:

$$\text{Success rate} = \frac{\text{number of valid files received by good peers}}{\text{umber of transactions attempted by good peers}} \quad (1)$$

3.1 Dataset

The dataset was constructed by simulating a P2P network using TM-SIM [West, Kannan, Lee et al. (2010)]. The network contains 5,000 users, and 100,000 files, and 100,000 transactions. Regarding users, there are 2,500 well-behaved users, including 500 pre-trusted ones. The network also contains five different types of malicious peers, which are purely malicious users, feedback-skewing users, malignancy-providing users, disguised malicious users, and sybil attack users, in order to teach the machine all possible scenarios of malicious behavior. Each type of malicious peer has 500 peers that represent its behavior. Tab. 1 summarizes the dataset characteristics.

The dataset contains 5,000 instances and nine numeric attributes: 1) positive ratings from a good peer, 2) negative ratings from a good peer, 3) positive ratings from a malicious peer, 4) negative ratings from a malicious peer, 5) positive ratings for a good peer, 6) negative ratings for a good peer, 7) positive ratings for a malicious peer, 8) negative ratings for a malicious peer, and 9) transactions. Peers are classified using the class attribute, which is either 0 or 1, where 0 represents a good user and 1 represents a malicious user.

Table 1: Dataset characteristics

Area	Computer network	No. of attribute	9
Attributes type	Numeric	Type of class	Binary
No. of instances	5000	No. of types	2
Associated tasks	Classification	Attributes characteristics	Integer
No. of missing values	0	No. of transactions	100000
No. of good peers	2500	No. of malicious peers	2500

4 Experimental setup

In an approach similar to that used in Kurdi et al. [Kurdi (2015); Lu, Wang, Xie et al. 2016)], the number of peers and files were varied to represent a meaningful sample of P2P environments. There were two sets of experiments, as summarized in Tab. 2.

Table 2: Experimental settings

	1st set			2nd set		
	Ex. 1.1	Ex. 1.2	Ex. 1.3	Ex. 2.1	Ex. 2.2	Ex. 2.3
No. of users	200	500	800	500	500	500
No. of files	500	500	500	200	500	800
No. of malicious peers	50	125	200	50	150	250

As in Kurdi et al. [Kurdi, Alshayban, Altoaimy et al. (2018)], the peer models included pre-trusted, good, and malicious peers. The pre-trusted peers had high trust values. In this paper, we considered two types of malicious peers: purely single malicious peers and purely collective malicious peers. We also selected three benchmark trust systems to compare and evaluate the performance of the proposed NNTrust system. These were: 1) EigenTrust [Kamvar, Schlosser and Gareia-Molina (2003)]. 2) Incremental EigenTrust [West, Kannan and Lee (2010)] and 3) the base case, None, in which there was no reputation management system and each peer randomly chose its file provider.

5 Results and discussion

A summary of the results of running 100,000 transactions for the two sets of experiments is shown in Figs. 2-9. As discussed previously, we used the percentage of invalid files and the success rates to evaluate the efficiency of NNTrust.

Figs. 2 and 3 illustrate the relationship between the percentage of invalid files exchanged by good peers and the number of files in the system. On the other hand, Figs. 4 and 5

show the relationship between the percentage of invalid services exchanged by good peers and the number of peers in the network. From the figures, we can see that for both malicious types, NNTrust had more success decreasing the percentage of invalid files than other benchmark systems.

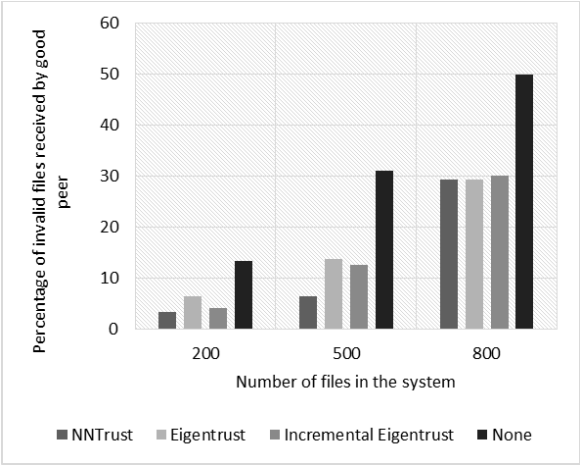

Figure 2: Percentage of invalid files when different numbers of files are considered (purely single malicious peers)

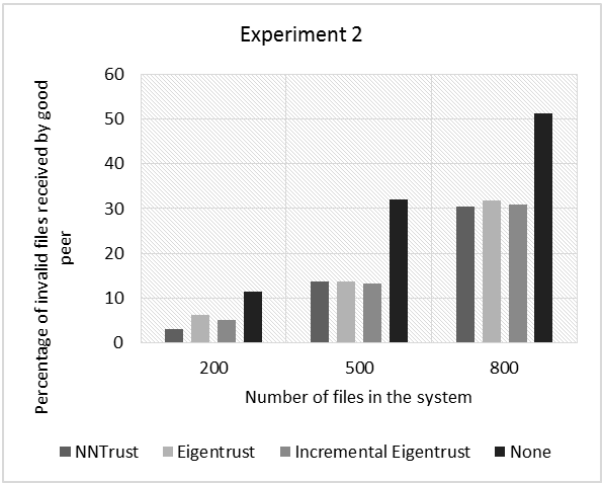

Figure 3: Percentage of invalid files when different numbers of files are considered (purely collective malicious)

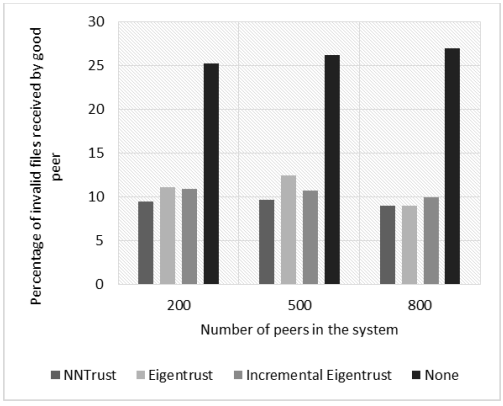

Figure 4: Percentage of invalid files when different numbers of peers are considered (purely single malicious)

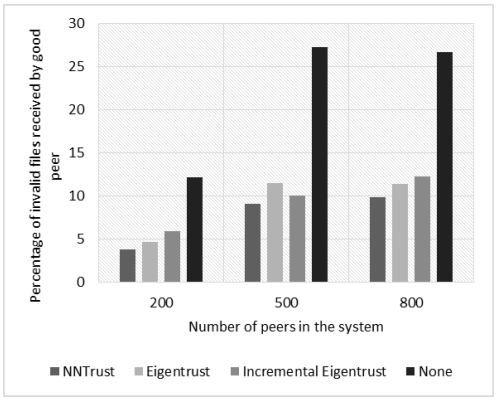

Figure 5: Percentage of invalid files when different numbers of peers are considered (purely collective malicious)

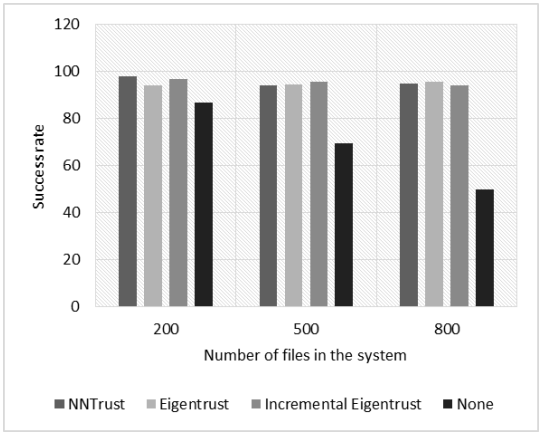

Figure 6: Success rate when different numbers of files are considered (purely single malicious)

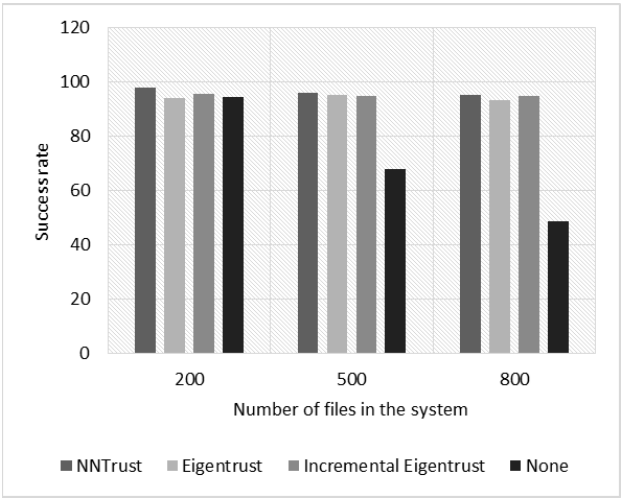

Figure 7: Success rate when different numbers of files are considered (purely collective malicious)

In Figs. 6 and 7, the success rate is plotted against the number of files in the system. Although the success rate of NNTrust dropped slightly in the case of 500 files and purely single malicious peers, its performance increased with increased number of files in the system. The reason behind this is that NNTrust analyzed larger log histories that increased as the number of transactions increased, and thus, improved its performance. In Figs. 8 and 9 the success rate is plotted against the number of peers in the network. We can see that for all scenarios, NNTrust had a higher success rate than other benchmark systems.

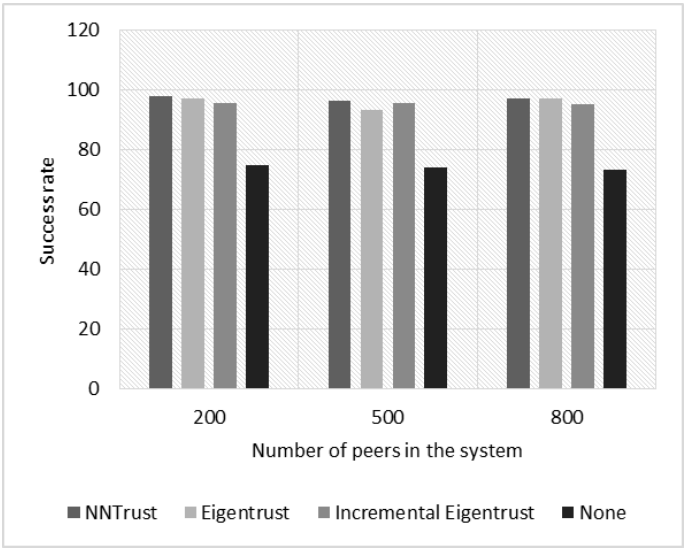

Figure 8: Success rate when different numbers of peers are considered (purely single malicious)

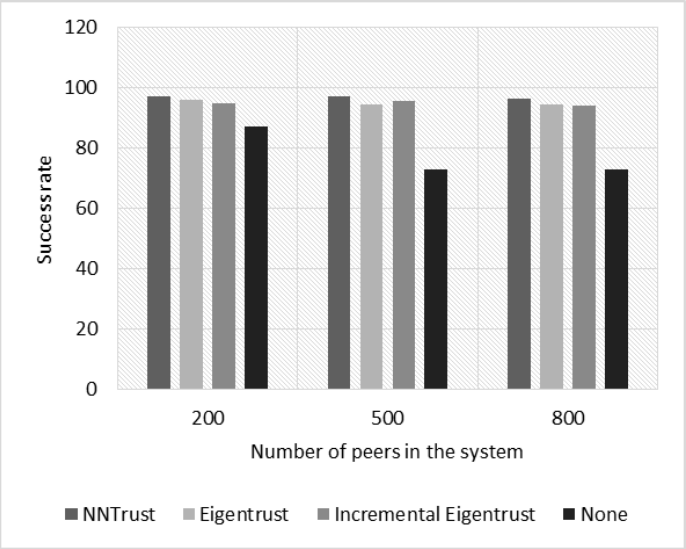

Figure 9: Success rate when different numbers of peers are considered (purely collective malicious)

6 Conclusion

In this paper, we proposed the use of historical data to keep track of all transactions that had been processed by edge devices in P2P networks. In the simulation setup, we compared the proposed NNTrust system with EigenTrust, Incremental EigenTrust, and the base case, None. We also evaluated the performance in terms of the percentage of invalid files and success rates of good peers. The results showed that NNTrust outperformed the other systems, minimizing the percentage of invalid files and maximizing the success rate of good peers. In our future work, we plan to produce a new dataset according to different patterns of malicious behaviors.

Acknowledgement: This research was supported by a grant from the Research Center of the Center for Female Scientific and Medical Colleges Deanship of Scientific Research, King Saud University.

References

Alhussain, A.; Kurdi, H. (2018): EERP: an enhanced eigentrust algorithm for reputation management in peer-to-peer networks. *Procedia Computer Science*, vol. 141, pp. 490-495.

Alkharji, S.; Kurdi, H.; Altamimi, R.; Aloboud, E. (2017): AuthenticPeer++: a trust management system for P2P networks. *European Modelling Symposium*.

Bashmal, L.; Almulifi, A.; Kurdi, H. (2017): Hybrid resource discovery algorithms for unstructured peer-to-peer networks. *Procedia Computer Science*, vol. 109, pp. 289-296.

Bradai, A.; Abbasi, U.; Landa, R.; Ahmed, T. (2014): An efficient playout smoothing mechanism for layered streaming in P2P networks. *Peer-to-Peer Networking and Applications*, vol. 7, no. 2, pp. 101-117.

Chuang, Y. T. (2017): Protecting against malicious and selective forwarding attacks for P2P search & retrieval system. *Peer-to-Peer Networking and Applications*, vol. 10, no. 4, pp. 1079-1100.

Fan, X.; Liu, L.; Li, M.; Su, Z. (2017): Grouptrust: dependable trust management. *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 4, pp. 1076-1090.

Hawa, M.; Al-Zubi, R.; Darabkh, K. A.; Al-Sukkar, G. (2017): Adaptive approach to restraining content pollution in peer-to-peer networks. *Information Systems Frontiers*, vol. 19, no. 6, pp. 1373-1390.

Jsang, A.; Hayward, R.; Pope, S. (2006): Trust Network Analysis with Subjective Logic. *Proceedings of the 29th Australasian Computer Science Conference,* Australian Computer Society Inc., Australia.

Kamvar, S.; Schlosser, M.; Garcia-Molina, H. (2003): The eigentrust algorithm for reputation management in P2P networks. *Proceedings of the 12th International Conference on World Wide Web*.

Kurdi, H.; Alfaries, A.; Al-Anazi, A.; Alkharji, S.; Addegaither, M. et al. (2018): A lightweight trust management algorithm based on subjective logic for interconnected cloud computing environments. *Journal of Supercomputing*, pp. 1-21.

Kurdi, H.; Alshayban, B.; Altoaimy, L.; Alsalamah, S. (2018): TrustyFeer: a subjective logic trust model for smart city peer-to-peer federated clouds. *Wireless Communications and Mobile Computing*, vol. 2018, no. 6, pp. 1-13.

Kurdi, H. A. (2015): HonestPeer: an enhanced eigentrust algorithm for reputation management in P2P systems. *Journal of King Saud University-Computer and Information Sciences*, vol. 27, no. 3, pp. 315-322.

Lee, G. M. (2016): RpR: a trust computation model for social internet of things. *IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress.*

Li, X.; Gao, Q.; Wu, L.; Sun, X.; Deng, S. (2018): EnhanEigen: a new comprehensive trust model for peer-to-peer network. *Proceedings of 2017 Chinese Intelligent Automation Conference.*

Lopez, P.; Montresor, A.; Epema, D.; Datta, A.; Higashino, T. et al. (2015): Edge-centric computing: vision and challenges. *ACM SIGCOMM Computer Communication Review*.

Lu, K.; Wang, J.; Xie, L.; Zhen, Q.; Li, M. (2016): An eigentrust-based hybrid trust model in P2P file sharing networks. *Procedia Computer Science*, vol. 94, pp. 366-371.

Shirgahi, H.; Mohsenzadeh, M.; Javadi, H. H. S. (2017): A new method of trust mirroring estimation based on social networks parameters by fuzzy system. *International Journal of Machine Learning and Cybernetics*, vol. 9, no. 7, pp. 1153-1168.

Steffenel, L. A.; Pinheiro, M. K.; Peres, L. V.; Pinheiro, D. K. (2018): Strategies to implement edge computing in a P2P pervasive grid. *International Journal of Information Technologies and Systems Approach*, vol. 11, no. 1, pp. 1-15.

West, A.; Kannan, S.; Lee, I.; Sokolsky, O. (2010): An evaluation framework for reputation management systems. *Trust Modeling and Management in Digital Environments: From Social Concept to System Development*.

Xie, X.; Yuan, T.; Zhou, X.; Cheng, X. (2018): Research on trust model in container-based cloud service. *Computers, Materials & Continua*, vol. 56, no. 2, pp. 273-283.