sciendo

BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **21**, No 4
Sofia • 2021 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2021-0040

Linear Regression Trust Management System for IoT Systems

Ananda Kumar Subramanian[1], *Aritra Samanta*[1], *Sasmithaa Manickam*[1], *Abhinav Kumar*[1], *Stavros Shiaeles*[2], *Anand Mahendran*[3]

[1]*School of Computer Science and Engineering, Vellore Institute of Technology, India*
[2]*Cyber Security and Resilience Research Group, School of Computing, University of Portsmouth, UK*
[3]*Post-Doctoral Research Fellow, Laboratory of Theoretical Computer Science,HSE University, Moscow, Russia*
E-mails: aritrasamanta18@gmail.com sasmithaamanickam@gmail.com abhinav.kumar2017a@vitstudent.ac.in s.anandakumar@vit.ac.in stavros.shiaeles@port.ac.uk amahendran@hse.ru

***Abstract*:** *This paper aims at creating a new Trust Management System* (*TMS*) *for a system of nodes. Various systems already exist which only use a simple function to calculate the trust value of a node. In the age of artificial intelligence the need for learning ability in an Internet of Things* (*IoT*) *system arises. Malicious nodes are a recurring issue and there still has not been a fully effective way to detect them beforehand. In IoT systems, a malicious node is detected after a transaction has occurred with the node. To this end, this paper explores how Artificial Intelligence* (*AI*), *and specifically Linear Regression* (*LR*), *could be utilised to predict a malicious node in order to minimise the damage in the IoT ecosystem. Moreover, the paper compares Linear regression over other AI-based TMS, showing the efficiency and efficacy of the method to predict and identify a malicious node.*

***Keywords*:** *Trust management, Internet of Things* (*IoT*), *Linear Regression, node analysis, wireless sensor networks.*

1. Introduction

With the advent of Wireless Sensor Networks (WSNs), the mechanism behind the communication among nodes changed from being supervised and/or predetermined to unsupervised and self-analysing systems. This change not only brought forward many benefits such as better routing protocols and better security but also raised many challenges. An unsupervised system means that all the alterations in the node system must be observed and accounted for by the system itself and then make the system change the path of communication among nodes. The problem was to create an algorithm that would help self-sustain a system of unattended nodes. Now, with the newer architectures such as Internet of Things (IoT), things have only grown more and more complex. Internet of Things focuses on all the different heterogeneous

networks in today's environment which bring out challenges of managing security and reliability.

Trust Management is necessary to identify both "trustable" nodes and malicious nodes based on certain parameters. In other words, an ideal trust management system can handle most, if not all, of the challenges presented by a system of nodes. A Trust Management system can be used in many different parts of a system of nodes and can have different factors influencing the trust score of a system based on the case of use. For example, a TMS in an IoT system that deals with confidential user data will have a higher weight for security and privacy than an agriculture-based IoT system which may require higher weights for reliability and accuracy.

After studying the Trust Management Systems (TMS) currently in use, studies which will be discussed later in detail, we found out that there is a significant scope for improvement. In previous designs, only one function is used to assess the trust level of a connection. It is found to be beneficial if historic data about the system is used for the assessment of trustworthiness. Although there are other works that have used past data, all of the data is grouped into one metric hence do not use the heterogeneity of the data. Emerging IoT technologies require the trustworthiness of a system to be accurate hence heterogeneity of historic data is required.

In recent years, many trust management systems have been developed that have high computational requirements. Concepts such as neural networks and fuzzy logic [1, 2] have been used to calculate the trust in an ever-changing node ecosystem. Although these methods have high accuracy in their predictive capabilities, they require a large ammount of resources to run. Since it is impractical, in many cases, for every single node to run a complex algorithm, say a deep neural network, the system architects are forced to centralize the node system. If centralized, the nodes instruction comes from a single control system which then negatively effects the efficiency of the system and work load of individual nodes. In order to decentralize the system, each node must be capable of calculating the trust factor of its peers without the help of a control system. Our system has successfully implemented the decentralized method by using an algorithm that can be run by the resource constrained nodes in a system.

Another outcome of our proposal is the higher efficiency of detecting malicious nodes. Our proposed algorithm is less complex and requires less computation than other TMS algorithms yet does not perform worse than the other algorithms in the case of malicious node detection. Instead of classifying a node as malicious just after one use, it uses a novel way to classify a node as "potentially malicious", therefore not affecting the range of the node system by not removing nodes from its system. Our system aims at being used in a variety of situations unlike many other which are more specialized, such as a vectored approach to P2P systems [3], a novel language-oriented approach to healthcare Trust management [4] and using game-theory to calculate trust in a complex system [5]. We have not put any constraints to the use case of this model, thereby allowing any industry to adopt it and use their respective policies in the system.

A survey [6] of various trust management systems, has explored the variety of properties involved in trust management in order to find its impact on trust

relationships. A Proposal of objectives for a holistic IoT trust management system has been given emphasizing their supporting layers involved in vertical trust management, which is very essential in establishing a trustworthy IoT system. Many studies [7] solely classify trust-based models on five dimensions based on a design which is considered for computation of trust values are a composition of trust nodes, aggregation of trust-based nodes, updating of trust values, the formation of trust propagation of trust for easier classification of many IoT computational trust-based models. Researchers identify all the advantages and disadvantages of the already existing IoT computational models and predict the effectiveness of each model against malicious attacks on nodes of an IoT system of networks.

The structure of this paper is as follows. In Section 2 we have the related works followed by the proposed method section and implementation where we introduce our Trust method using Linear Regression in order to address the problem of predictive and prescriptive analysis over the trust in a system. Finally, we present our findings and conclusion in Sections 5 and 6, respectively.

In summary, this paper contributes to the world of Internet of Things by creating a novel Trust management system that uses Linear regression for analysing the trust scores of neighboring nodes. The novelty lies in the decentralised approach this model takes and the low resource requirement. This paper can be taken and used as the foundation to create a complete routing protocol based on trust management. The current most favoured routing protocol for wireless sensor networks is the IPv6 RPL (Routing Protocol for Low-Power and Lossy Networks) which uses DODAG (Destination Oriented Directed Acyclic Graph) graphs for high energy efficiency during running. This protocol is very proactive and can quickly create the ideal topology for a resource constrained network. But this routing protocol does not take into consideration the trust parameters of each node. Using our linear regression trust management system, the nodes can create a route to the control system without the interference of the control system or admins.

2. Literature review

Apart from the basic concepts of building trust-based management systems on IoT systems, certain researchers develop adaptive protocols for trust management for many social IoT systems, and thereby it can be an application to the management of services in IoT systems. These social interaction systems are made available with a protocol that can be distributed and each node will be updated with its trust value only when it encounters interaction events with social relationships with other nodes [8].

Further from developing models for trust management, certain studies provide systems for trust-aware access control which considers the trust values of the account devices to make decisions regarding authorization. These trust-based models consider four main parameters such as reputation, social relationships, quality of service, and other security aspects in order to evaluate trust values. This model makes use of the fuzzy control systems which depend on the four main parameters and these

are solely quantified on the historical data of the trust values of the nodes under the system design [9].

Present-day Trust Management Systems (TMS) assess the trustworthiness of a system using a single function [10]. This system is outdated and inefficient. Main problems include inaccuracy, inability to learn from past events, and inability to analyse heterogeneous groups of data. To overcome this problem the proposed TMS utilizes context-aware and multiservice models. The context-aware system acquires "trust scores" from historical data to predict the changes in the trustworthiness of nodes in a network. The test scores are dynamic which means the scores change based on trust level data models. As there are different functions for each node to execute so different trust scores are assigned to each function of the node, so the system chooses which node to access based on the status of the current node for the function it is required by comparing their trust scores.

The context-aware model has five phases, namely information gathering, entity selection, transaction, reward/punish, learning. In information-gathering it accesses its knowledge base, then in entity selection, chooses the node. It then completes the transaction. Based on the result of the transaction it reduces or increases the trust scores of the nodes. It then learns the confidence values of the nodes and is used in the information gathering step again as it is a recurrent process. It is the duty of the system to correlate the trustworthiness to the node for the specific reason it has been used. The process of entity selection is very complex and is divided into five steps. Step 1 involves restricting the set of nodes by selecting potential candidates. Next, it compares the reports of previous confidence values of the nodes and creates a set of reports to referring from. In the third step, it computes the weights of each retained reports in the second step. It then computes the trust values for each node. Finally, it provides the node with the best rating to the system. After the transaction, it evaluates the success of the transaction and adds the result to the knowledge base. It also learns its behaviour when it is targeted by a class of attacks in order to learn and withstand these attacks [10].

The system used the following formula to calculate the trust value:

$$T_i = \frac{1}{\sum_{j=1}^{n} \omega_{R_{ij}}} \times \sum_{j=1}^{n} \left(\omega_{R_{ij}} \cdot \mathrm{QR}_j \cdot N_j \right), \quad (1)$$

QR_j (Quality of Recommendation of the node j having issued the report R_{ij} about the proxy P_i) is the trustworthiness score assigned to a witness node depending on the accuracy of its past reports. It ranges between 0 and 1, 1 representing a very trustworthy node and a node reporting the opposite of the actual service quality; $\omega_{R_{ij}}$ is the weighting factor computed [10].

Understanding the growth of IoT and various cloud computing and big data concepts, it is clear that security among nodes in a network is a growing concern. A research paper focuses on the concept of a Brain-inspired Trust based model to ensure secure data transmission streams in various applications in the Neuroscience industry. This Trust Management Model calculates the behavioral trust of joint nodes and the consecutive data trust values using the ANFIS node analysis model along with a weighted addictive methodology. Several simulations have been done including NS2 simulations which depict that this model is much better than the previously existing

Trust management algorithms such as Fuzzy Inference System (FIS) and Neuro Fuzzy based Trust Management System (NFTM). It is clear that in the future, great concepts of Deep learning and Bayesian statistics concepts to prepare more trustworthy Trust management systems will be predominant in the upcoming future [11]. Another research paper mainly focuses on an IoT-based Trust model which comes under a scalable trust-based system that can be applied to many IoT devices [12]. In order to realize the practicality of this proposed method, four other algorithms have been tested along with it. This includes the main algorithm which mainly focuses on removing the outliers which occur among the trust values in order to avoid any IoT-based security attacks on the nodes. The second algorithm focuses on the development of an efficient mechanism to form IoT clusters with great trust. The third one includes the trust-based migration of a node from one location of the cluster to another and the final algorithm includes the analysis of the currently present cluster nodes of IoT according to their trust values to determine which clusters should be joined. Future work is all about developing this project on a major scale to be applied in real-time IoT applications for secure communications among nodes [12]. A comparative study of the different trust models and parameters used in IoT Wireless Sensor is shown in Table 1.

Table 1. Different trust models

Trust models used	Parameters				
	Network strain	Social relationship	Change response	QoS	Credibility
Fuzzy [1]	High	Yes	Very fast	Yes	Yes
Bayesian [13]	High	Yes	Very fast	Yes	Yes
Static Weighted Sum [14]	Less	No	Slow	Yes	Yes
Event Driven [15]	Less	No	Fast	Yes	No
Delay tolerant [16]	Less	No	Slow	Yes	No
Similarity [18]	Less	Yes	Slow	No	No
Centrality [19]	Less	Yes	Slow	No	No
Service Reputation [20]	Less	No	Fast	Yes	Yes
Capability [6]	Less	No	Slow	Yes	Yes

Using five parameters, Table 1 compares the various already existing trust management systems which are discussed in the literature survey. There have been various other trust models created in recent times. One model was a recursive approach to the well-known Bayesian model which uses a new Gaussian system for calculating the trust in wireless sensor networks [13]. There are models that use fuzzy logic. One such model uses fuzzy logic for detecting malicious nodes in IoT [22] and another uses it for trust management in semantic P2P grids [23]. With the emergence of cloud technologies, trust models for identity management have been made, for example one, which uses a dynamic mechanism [24], and one, which uses a blockchain-based mechanism for cloud identity management [25]. Some models use neural networks, like distributed probabilistic neural networks [26] and artificial neural network routing algorithms based on the trust of the nodes in a system [27]. Trust models dealing with M2M applications [28] and P2P-based applications [29]

have also emerged, which analyze the security issues within these applications and deals with the threats.

It can be stated from the above analysis that trust management done using a single function is unable to predict and analyze the trust in a network. Therefore, a model must be made that can do both predictive and prescriptive analysis over the trust in a system, which is the aim of this research work.

3. Proposed methodology

The proposed Trust Management system uses Linear Regression in the Things layer of an IoT system as the algorithm used to predict future trust values of every node. The "Things" layer consists of the various devices, in this case, sensors used in an IoT system. Linear regression is a form of regression analysis that assumes that the data points are linearly distributed. In this system, for every node, each trust parameter value for each iteration is plotted against the iteration number, i.e., the *n*-th time that particular node was used. The algorithm creates a best fit line of the form

$$y = \beta_0 + \beta_1 * x, \tag{2}$$

where *y* is the trust parameter value and *x* is the iteration number. After creating the best fit line using *n* consecutive trust parameter values we can predict the trust parameter value of the (*n*+1)-th iteration by substituting the value of *x* in the equation by *n*+1. The justification behind using linear regression is that LR is a simple yet effective algorithm. It does not require much computational power nor storage space. The program can output in mere bytes, hence linear regression is very efficient.

Considering the fact that our model is built for nodes with constraints in power and memory, the ideal routing protocol is the IPv6 RPL (Routing Protocol for Low-Power and Lossy Networks) which uses Destination oriented directed acyclic graphs for low power consumption during routing. This protocol is very proactive and can quickly create the ideal topology for a resource-constrained network.

Since our research is not focused on the trust parameters themselves but instead the predictive capability of an IoT framework we have used existing trust parameters for our model. From the research papers discussed above we have identified five parameters which are the most widely used and are enough to calculate the total trust score of a node. The parameters are Availability, Integrity, Security, Honesty, and Privacy. Although we do not necessitate the use of the following methods to calculate the values of the parameters, we consider them to be ideal in our proposed system based on their efficiency and accuracy. Availability is given a binary value of either "0" or "1" based on whether or not the node was available. The availability is determined using an existing research work based on blockchain technology, which determines the device or node failure in an IoT system using the underlying block chain, Ethereum [30]. Integrity is the percentage of the message that is error-free when received. The error is calculated by comparing the transmitted message with the received message, which is called the Bit Error Rate (BER) of a transaction. Security is calculated using the Hidden Markov model, which outputs a probabilistic value (0-1). The hidden markov model [31], which is based on an existing model,

helps in identifying intrusions and cyber security breaches in a node. Honesty is the percentage of transactions that were done honestly, calculated using an existing Proof-of-Honesty model [32], which despite being a model used for block chain IoT systems, can be used for any node networks using the proof-of-Honesty method. Privacy is the percentage of private transactions. Privacy breaches can be detected using an existing model which identifies devices that are "listening" to the node and checks if any of those devices are outside of the IoT system [33].

An important aspect of the proposed system is the encapsulation of the learning abilities of Neural Networks based IoT systems with less processing power and complexities. The system will maintain databases for every node containing data of all previous transactions with all its neighbor nodes. The model uses linear regression to predict a numerical value of each of the five trust parameters stated previously, using the database present in each node's storage. The cumulative Trust factor is then calculated as the summation of the five trust parameter values.

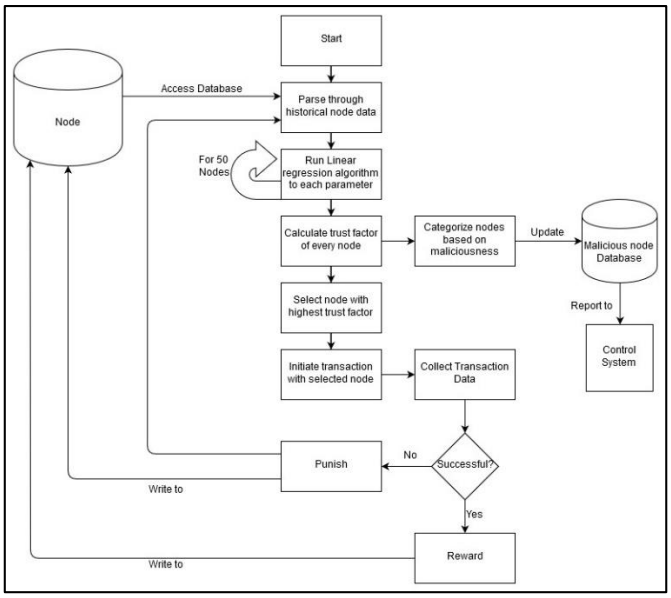

Fig. 1. Flow of processes in the usage of Linear Regression for Node Analysis

Fig. 1 is the flow diagram describing the working of the linear regression algorithm in the trust management system. The linear regression algorithm has two uses in the program. The first is to predict the next trust parameter value of every single node in the system. The second is to analyze the trends of all the trust values of nodes to predict if a node is turning malicious or not. This is done by calculating the slope of the graph of the line created by the linear regression model. There is a threshold value defined in the system. An empirical method can be used for approximating the threshold value, which is tailored to the historical experimental data using permutation tests. Or, the threshold value can be set to 95/100, based on the universally accepted error tolerance value $\alpha = 0.05$. The trust requirements of various industries differ from one another. Therefore, the trust value threshold can

only be set by an expert or an individual who has experience with that particular network of nodes [34]. If the Trust value exceeds the threshold value it is noted as a "working node", else it is noted as a malicious node. If a node's trust value trend is such that its current value is above the threshold value but the linear regression line shows that its trust value is decreasing then the node is classified as a potentially malicious node. The data can then be sent to a control system, which can then assign maintenance or removal services for the malicious nodes and keep the potentially malicious nodes under their supervision.

After every iteration, the trust values are then rewarded or punished based on the transaction that was completed based on the predicted values. The database is then updated with the actual values and then the system can run again.

4. Implementation

The model was simulated in a Python IDLE on a Windows 10, 8 GB RAM, i7-7500U CPU @ 2.70 GHz (4 cores) system. Python has been chosen because of its versatility in the area of artificial intelligence and the fact Python is the programming language of the near future. The databases has been made as Excel sheets for convenience. A 50-node network has been considered for the experiment. Each node has had five parameters and parameter values of the last 20 times each node has been used have been noted. Hence the database has contained about 5000 entries before the first iteration of the experiment giving the program enough data to learn from. The python package "openpyxl" has been used to read and write ".xlsx" files without manually having to update them. In an actual network of nodes, a more optimized ".csv" can be used.

Two databases have been used, one for the trust values of all the nodes and the second for the table that holds the nodes classified into three categories based on their maliciousness. The table has been first created after the 21st Iteration where the trends of trust values of every node have been analyzed. After every iteration, the table has beeen completely updated with the new results of the analysis. The database that contains the trust parameters has been created just to demonstrate the effectiveness of the linear regression algorithm in predicting using a database and it is not an existing database.

4.1. Pseudocode

Step 1. Start

Step 2. Access database and retrieve values of each parameter for all the nodes $N_0 - N_{19}$.

Step 3. Scale all the data to the range (0-20).

Step 4. Add all parameters of each corresponding node and create an array of trust values $T_0 - T_{19}$.

Step 5. Run Linear Regression algorithm over each array $T_0 - T_{19}$ for every node in $N_0 - N_{19}$, create an array of predicted trust values.

Step 6. Sort through the predicted array and choose Node *N* with the highest predicted trust value.

Step 7. Initiate transaction.

Step 8. Collect results of transactions and create an array of actual trust values for each node.

Step 9. If the transaction is successful, the trust parameter is “awarded”.

Step 9.1. Else, the trust parameter is “punished”.

Step 9.2. Return to Step 7 unless the transaction is successful.

Step 10. Analyze trends in trust values.

Step 11. Let Th be the threshold value. And let T_n be the predicted trust value of a node.

Step 12. If T_n >Th: N_n is a working node.

Step 12.1. Else N_n is a malicious node.

Step 13. If linear regression algorithm computes slope of line plotted with T_n over *n* to be less than “0” then N_n is a potentially malicious node.

Step 14. Report to control system.

Step 15. End.

5. Results and discussion

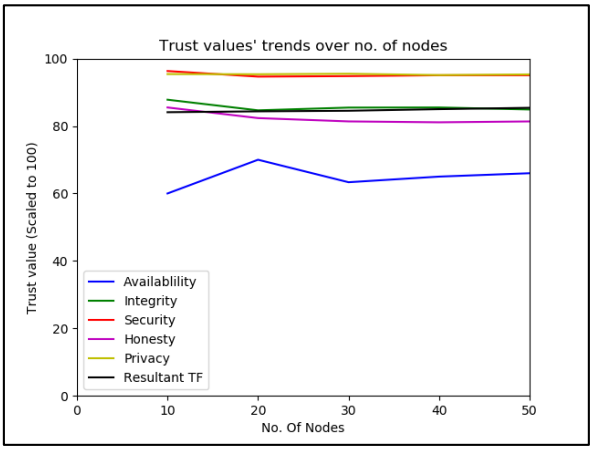

Fig. 2. Trust value trends over the no. of nodes in the system

Overall, our simulation has shown how efficient it is in predicting accurate trust values. A comparison has been done in networks of 10-50 nodes to view the trends in the system using the Linear Regression trust model, which is seen in the graph in Fig. 2. The most significant outcome of the experiment has been its ability to detect malicious nodes almost immediately. The trust value predicted drastically alters after an unsuccessful transaction and the effect are visualized in the graphs below.

In Fig. 3, the plotline clearly shifts significantly just because (in this case) the 21st transaction has been unsuccessful (notice the “0” on the plot corresponding to *x*=21 in the bottom left), therefore it immediately removes the node out of contention of being selected for any further transactions. Furthermore, in the general scenario

tested for all 50 nodes, it has not taken more than two iterations of a certain node's transaction to be unsuccessful to detect a malicious node.

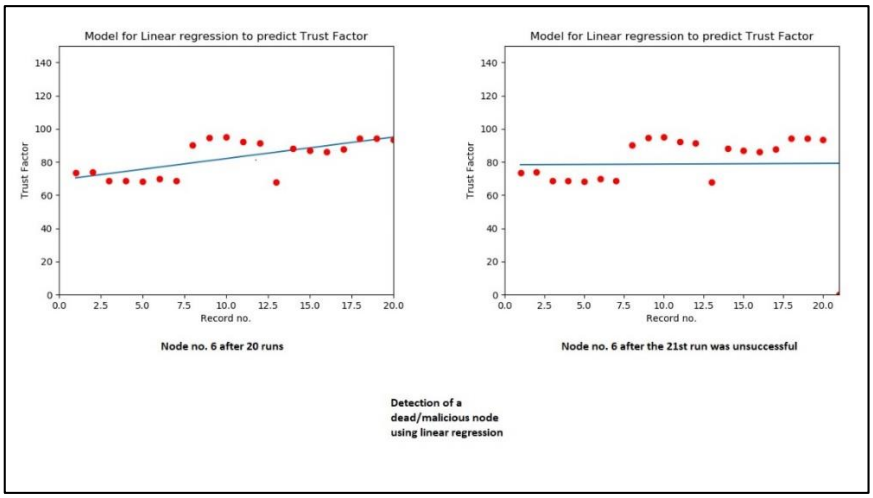

Fig. 3. Comparison of the trust values of a particular node before and after an unsuccessful transaction

5.1. Comparison with other trust models

Traditional trust models use a function – simple or complex to calculate the trust value of a node. But the use of AI immediately gives it an edge over all such traditional models due to its ability to learn and predict. Complex functions can be used on top of the AI model to further strengthen it.

Table 2 is a comparison of our model with the Neural Network models proposed in the literature. The effectiveness of the neural network and the linear regression in a system with seven parameters are compared below. The following comparison is done on the performance of neural networks [27] and linear regression on small datasets [35].

Table 2. Comparison table of NN systems and our LR system

Parameter	Neural networks	Linear Regression
Computational power required	Very high	Low
Storage space required	Extremely high (100× that of LR)	High
Cost of operation/installation	High	Low
Time taken for analysis	High	Very low
Possible extent of optimization	Can be optimized, but will still not be on par with Linear Regression	Can be optimized significantly based on the use case
Effectiveness in limited parameter trust models	Requires a large number of parameters, hence useless	Does not require a large number of parameters, hence ideal
General Effectiveness	Most trust models have linearity, hence neural networks are unnecessary	Linear regression is ideal for a linearly varying dataset

Since scalability and interoperability are integral aspects of IoT systems it is of paramount importance to address them. In this paper, we have provided the framework of an IoT system. What technologies lie within are completely dependent on the application of this framework. The model provides a way for a certain device or node to identify the most trustable node in its vicinity and does not enforce any constraints on how the devices communicate. This framework can be applied to already existing systems that have heterogeneous devices in its network and interoperability will not hinder its performance. As for the scalability, this framework does not have any limitations to its size. In our simulation we have assumed a node to have 49 immediate neighboring nodes which itself is a considerably large number of nodes to have in a particular node's direct vicinity. An IoT system may have a massive network of nodes but any particular node will not have many immediate neighbors, therefore scaling a IoT system by adding more nodes to the network will not affect the efficacy or the accuracy of the model. Even if there were more nodes added as immediate neighbors to a certain node, the increase in resource requirement will not be significant.

6. Conclusion

It can be concluded that the use of linear regression for the analysis of nodes, especially the detection of malicious nodes, is a significant improvement in the field of IoT. Its predictive analysis will help the management take action on malicious nodes before any corruption takes place. It is also established that even though neural networks are a more advanced technique in artificial intelligence it comes at a cost of resources. The use of Linear Regression not only has the same benefits as that of neural networks but is also cost-efficient and optimized for nodes. Since nodes have minimal storage and computational power and the fact that IoT systems have a huge number of such nodes it is necessary to have the time and cost of operation as minimal as possible. Unnecessarily increasing the cost of operation of one node will increase the cost of operation of the whole system by thousands or more. In conclusion, Linear regression for node analysis is a cost-effective method to accurately analyze and predict trust values in a network of nodes and also to predict the malicious properties of a node.

References

1. C h e n, D., G. C h a n g, D. S u n, J. L i, J. J i a, X. W a n g. TRM-IoT: A Trust Management Model Based on Fuzzy Reputation for Internet of Things. – Comput. Sci. Inf. Syst., Vol. **8**, 2011, No 4, pp. 1207-1228.
2. A l m o g r e n, A h m a d, et al. FTM-IoMT: Fuzzy-Based Trust Management for Preventing Sybil Attacks in Internet of Medical Things. – IEEE Internet of Things Journal, 2020.
3. Z h a o, H., X. L i. VectorTrust: Trust Vector Aggregation Scheme for Trust Management in Peer-to-Peer Networks. – The Journal of Supercomputing, Vol. **64**, 2013, No 3, pp. 805-829.
4. B e c k e r, M. Y., P. S e w e l l. Cassandra: Flexible Trust Management, Applied to Electronic Health Records. – In: Proc. of 17th IEEE Computer Security Foundations Workshop, IEEE, 2004.

5. L i, H u i-J i a, et al. Exploring the Trust Management Mechanism in Self-Organizing Complex Network Based on Game Theory. – Physica A: Statistical Mechanics and its Applications, Vol. **542**, 2020, 123514.
6. Y a n, Z., P. Z h a n g, A. V. V a s i l a k o s. A Survey on Trust Management for Internet of Things. – Journal of Network and Computer Applications, Vol. **42**, 2014, pp. 120-134.
7. G u o, J., R. C h e n, J. J. T s a i. A Survey of Trust Computation Models for Service Management in Internet of Things Systems. – Computer Communications, Vol. **97**, 2017, pp. 1-14.
8. C h e n, R., F. B a o, J. G u o. Trust-Based Service Management for Social Internet of Things Systems. – IEEE Transactions on Dependable and Secure Computing, Vol. **13**, 2015, No 6, pp. 684-696.
9. B e r n a b e, J. B., J. L. H. R a m o s, A. F. S. G o m e z. TACIoT: Multidimensional Trust-Aware Access Control System for the Internet of Things. – Soft Computing, Vol. **20**, 2016, No 5, pp. 1763-1779.
10. B e n, Y., A. O l i v e r e a u, D. Z e g h l a c h e, M. L a u r e n t. Trust Management System Design for the Internet of Things : A Context-Aware and Multi-Service Approach. – Comput. Secur., 2013, pp. 1-15.
11. A l s h e h r i, M. D., F. K. H u s s a i n, O. K. H u s s a i n. – Mobile Netw Appl., Vol. **23**, 2018, 419. **https://doi.org/10.1007/s11036-018-1017-z**
12. M a h m u d, M., M. S. K a i s e r, M. M. R a h m a n, et al. Cogn. Comput., Vol. **10**, 2018, 864. **https://doi.org/10.1007/s12559-018-9543-3**
13. M o h a m m a d, M., A. K h a l i d, C. S u b h a s h. RBATMWSN: Recursive Bayesian Approach to Trust Management in Wireless Sensor Networks. 2008, pp. 347-352. DOI: 10.1109/ISSNIP.2007.4496868.
14. C h e n, I. R., F. B a o, J. G u o. Trust-Based Service Management for Social Internet of Things Systems. – IEEE Transactions on Dependable and Secure Computing, 2016.
15. N i t t i, M., R. G i r a u, L. A t z o r i. Trustworthiness Management in the Social Internet of Things. – IEEE Transactions on Knowledge and Data Management, Vol. **26**, 2014, No 5, pp. 1253-1266.
16. C h e n, I. R., et al. Dynamic Trust Management for Delay Tolerant Networks and Its Aapplication to Secure Routing. – IEEE Transactions Parallel and Distributed Systems, Vol. **25**, 2014, No 5, pp. 1200-1210.
17. L i u, Y u-C h a o, et al. A Method for Trust Management in Cloud Computing: Data Coloring by Cloud Watermarking. – International Journal of Automation and Computing, Vol. **8**, 2011, No 3, pp. 280-285.
18. C h e n, I. R., J. G u o, F. B a o. Trust Management for SOA-Based IoT and Its Application to Service Composition. – IEEE Transactions on Service Computing, Vol. **9**, 2016, No 3, pp. 482-495.
19. F r e e m a n, L. C. Centrality on Social Networks. – Social Networks, Vol. **1**, 1979, pp. 215-239.
20. C h e n, Z., R. L i n g, C. M. H u a n g, X. Z h u. A Scheme of Access Service Recommendation for the Social Internet of Things. – International Journal of Communication Systems, Vol. **29**, 2016, No 4, pp. 694-706.
21. N o o r, T. H., et al. Trust Management of Services in Cloud Environments: Obstacles and Solutions. – ACM Computing Surveys (CSUR), Vol. **46**, 2013, No 1, pp. 1-30.
22. Al s h e h r i, M. D., F. K. H u s s a i n. A Fuzzy Security Protocol for Trust Management in the Internet of Things (Fuzzy-IoT). – Computing, Vol. **101**, 2019, pp. 791-818. DOI:10.1007/s00607-018-0685-7.
23. J a v a n m a r d i, S., et al. FRTRUST: A Fuzzy Reputation Based Model for Trust Management in Semantic P2P Grids. – arXiv preprint arXiv:1404.2632, 2014.
24. B e n d i a b, K., S. S h i a e l e s, S. B o u c h e r k h a. A New Dynamic Trust Model for "On Cloud" Federated Identity Management. – In: Proc. of 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS'18), Paris, 2018, pp. 1-5.
25. B e n d i a b, K., N. K o l o k o t r o n i s, S. S h i a e l e s, S. B o u c h e r k h a. WiP: A Novel Blockchain-Based Trust Model for Cloud Identity Management. – In: Proc. of 16th IEEE Intl. Conf. on Dependable, Autonomic and Secure Computing, 16th Int. Conf. on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech'18), Athens, 2018, pp. 724-729.

26. A s i r i, S., A. M i r i. An IoT Trust and Reputation Model Based on Recommender Systems. – In: Proc. of 14th Annual Conference on Privacy, Security and Trust (PST'16), Auckland, 2016, pp. 561-568. DOI: 10.1109/PST.2016.7907017.
27. S i n g h, A. V., V. J u y a l, R. S a g g a r. Trust Based Intelligent Routing Algorithm for Delay Tolerant Network Using Artificial Neural Network. – Wireless Netw., Vol. **23**, 2017, pp. 693-702. DOI:10.1007/s11276-015-1166-y.
28. C h a, I., et al. Trust in M2M Communication. – IEEE Vehicular Technology Magazine, Vol. **4**, 2009, No 3, pp. 69-75.
29. S h a l a, B., et al. Ensuring Trustworthiness for P2P-Based M2M Applications. – Internet Technologies and Applications (ITA), Wrexham, 2017, pp. 58-63.
30. Z h a n g, W., et al. Blockchain-Based Federated Learning for Device Failure Detection in Industrial IoT. – IEEE Internet of Things Journal, 2020.
31. M u h a t i, E., D. B. R a w a t. Hidden Markov Model Enabled Prediction and Visualization of Cyber Agility in IoT Era. – IEEE Internet of Things Journal, 2021.
32. I m r a n, M., et al. PLEDGE: A Proof-of-Honesty Based Consensus Protocol for Blockchain-Based IoT Systems. – In: Proc. of 2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), IEEE, 2020.
33. S i b y, S., R. R. M a i t i, N. O. T i p p e n h a u e r. Iotscanner: Detecting Privacy Threats in Iot Neighborhoods. – In: Proc. of 3rd ACM International Workshop on IoT Privacy, Trust, and Security, 2017.
34. C h e n, Z., L. T i a n, C. L i n. Trust Model of Wireless Sensor Networks and Its Application in Data Fusion. – Sensors, Vol. **17**, 2017. 10.3390/s17040703.
35. M a a d, M. Artificial Neural Networks Advantages and Disadvantages. Bagdad College of Economic Science University, 2018.

Received: 26.04.2021; Second Version: 28.09.2021; Accepted: 12.10.2021