CTMS-SIOT: A Context-based Trust Management System for the Social Internet of Things

Oumaima Ben Abderrahim
National School of Computer Science
University of Manouba
Tunis, Tunisia
oumaima.benabderrahim@ensi-uma.tn

Mohamed Houcine Elhedhili
National School of Computer Science
University of Manouba
Tunis, Tunisia
med_elhdhili@yahoo.es

Leila Saidane
National School of Computer Science
University of Manouba
Tunis, Tunisia
leila.saidane@ensi.rnu.tn

Abstract—**The social internet of things (SIOT) is a thriving research field that emerged after the integration of social networking concepts in the internet of things. It resulted in the appearance of new and more powerful applications. Indeed, trust management system (TMS) has been considered as an effective security mechanism in the Internet of things. Thus, many research works have been carried to propose trust evaluation and trust prediction methods. In the traditional trust management system, historical behavior data are taken into account to predict the trust value of the network entity, while the context of the network entity is rarely considered. The novelty of our approach can be summed up in three aspects: i) designing a highly scalable trust model, ii) the ability of the system to provide the most trustworthy service provider which takes into account the dynamic aspect of the internet of things, such as context, the capacity object and the social relationship between objects, iii) and the application of decision tree to analyze first the relationship between the different components of the network and the object behavior, and to improve then the decision making.**

Keywords ; IoT, Security, Trust management system, Attacks, Direct observation, Decision tree, Context, Similarity, Trust.

I. Introduction

Internet of Things (IoT) can connect a large number of things through communication networks in various types of applications. In IoT systems, things can be sensors, monitors, smart devices, laptops or even human beings. This new paradigm gave birth to many applications as smart grids, smart cities, smart homes and e-health. These applications aim at improving the human life quality. However, achieving a complete vision of the IoT depends on several factors where the most important one is the security. Indeed, IoT security is a challenging task as known solutions in the Internet might be inadequate for the IoT due to their inherent characteristics, especially mobility and nodes heterogeneity in terms of resources. Besides, resource constrained objects need the support of unconstrained and trustworthy objects to establish securely a network service. Hence, an object must select trustworthy helpers to avoid malicious behaviors. This can be done by constructing a trust management system. Our centralized trust management system contains a local TMS, in each object, and a central TMS on a trust server. Because our model adapts to the dynamics of the network, we introduced the notion of the context. Obviously, the behavior of things varies according to context. Our model is able to select the most trustworthy objects that can provide the requested service for each context even if no history exists about the service provider by using the decision tree tool. We used Jaccard Index to compute the social similarity between the different objects in order to determine the credibility of the objects predicted by the decision tree. Our TMS server contains two modules. The first, called trust module, is responsible for contextual trust computing (T^{cx}) and reputation computing (R). The second module, named learning module and based on decision tree technique, is responsible for behaviours classification and improving the decision making. It can learn from the evaluations and the experiences already existing in the trust table. We demonstrate the effectiveness of the introduced solution CTMS-SIOT against TMS attacks by comparing it with the methods proposed in the literature. The reminder of the paper is organized as follows: In section II, we give a brief overview of the existing IoT trust management systems. In section III, we specify the attack model. Our proposed solution is depicted in section IV. Section V presents the simulation results. Finally, we conclude the paper and we outline future works.

II. Related work

A number of trust management schemes were developed for trust management models to protect the communication in the context of the Internet of things. In [1], Chen et al suggested a trust management system based on fuzzy reputation for the IoT with quality of service trust metrics containing elements as packets forwarding/delivery ratio and energy consumption. However, this system considers that the IoT environment is formed only by sensors, which is far from the reality of the IoT services. In [2], Bao et Al suggested a security architecture for the Internet of Things where most objects are linked to human entities, and are able to establish social relationships as friendship, ownership and community. However, malicious nodes can disturb the network functionality using attacks as self-promoting attacks, bad mouthing and good mouthing attacks.

In [3], authors proposed a distributed trust management protocol for the Internet of things. This protocol is based on encounter and activity rates: That is to say, two nodes coming

978-1-5090-4372-9/17/$31.00 ©2017 IEEE

in touch with each other or involved in a mutual interaction can directly rate each other and exchange trust evaluation of other nodes performing an indirect rate, which resemble to a recommendation. In this protocol, three reference parameters for trust evaluation are used: honesty, cooperativeness and community-interest. Therefore, such a dynamic trust management protocol is able to adaptively adjust the best trust parameter setting in response to dynamically-changing environments in order to maximize the application performances. In fact, the social relationships in IoT systems have attracted the attention of many researchers. SIOT is a similar approach introduced by Niti et al in [4]. It is a subjective model based on social relationships, where each node computes trust on the basis of its own experience and the opinions of common friends. In this system, the transaction weight increases with the importance of the transaction. To evaluate the trust, authors used a feedback system, objects credibility and centrality. In [5], a technique based on the distributed collaborative filtering was developed to select feedback using similarity rating of friend-ship, social contact and community of interest relationships as filter. For scalability, the capacity-limited node only keeps trust information of a subset of nodes. In [6], A Scalable Hybrid Trust & Reputation model for the social internet of Things was proposed, The work assumes that two nodes belonging to the same CoI have specific social interests and strong social ties whereas different communities may have different or controversial views of trust towards the same trustee due to their differing social interests. The goal of the proposed trust management scheme is to make sure that each nodes trust evaluation converges to its community agreement.

In the TMS cited above, the context is not considered in the trust computation, which is not realistic in dynamic networks, we explain in our work how our model can have a trust value dedicated for each context.

In the next section, we will describe our new trust management system called CTMS-SIOT that we proposed for the Social Internet of Things.

III. Proposed trust management system :CTMS-SIOT

Unlike the systems proposed in IoT literature, our trust model proposes an objective mechanism providing dynamic trust value for the same node in different contexts and different services, since the trust computation for objects without considering the context is not realistic in dynamic networks with multi-task nodes. In such networks, each node should have a trust value dedicated for each context. We describe, in this section, the proposed architecture and the trust management system.

A. Architecture

Choosing either a centralized or decentralized approach for trust management depends on several factors, such as the computational complexity. In fact, the nodes in the network are with constrained resources and do not support complex computation and heavy algorithms. Contrariwise, in the decentralized architecture, each node must store the trust information about network node for later use, which results in the reduction of nodes lifetime. Thus, in our work, we choose a centralized architecture as shown in figure 1.

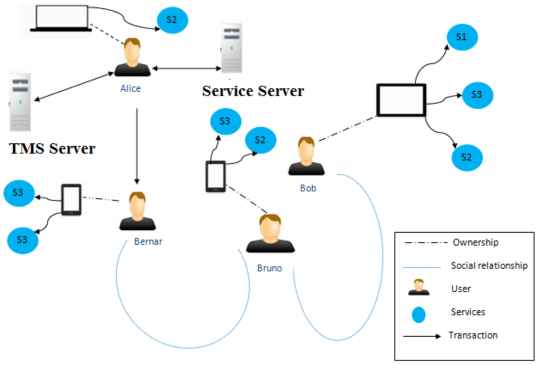

Fig. 1. The architecture of the proposed model

The components of our models are described below:

1) Objects: Our networks are formed by objects with different capacity C. They are used only by their owners. A social relationship, such as friendship, can be established between them. Each object has a local trust system to record the feedback for the limited set of objects.

2) Service server: Is responsible for the service discovery. Indeed, each object that wants to join the SIOT must authenticate itself to the SIOT service server and register the services that can provide, the information about devices (Object profile), contextual information such as $location$, $time$. The service server can propose to the new object to join communities of interest or making a friendship relationship with SIOT users. The service server computes dynamically the similarity between each two objects. We will explain later the computational similarity which will be transferred to the trusted server on request.

3) Trust management server (TMS): Our model has a central trust management system. It receives the feedback from the entities of the network and computes the contextual trust and the reputation. Indeed, our trust management server contains two modules. The first module is designed for the computing of contextual trust ”‘T^{cx}”’ and reputation R; whereas the second module is the learning module used for behaviors classification and prediction. We assume also that this server can be a cloud server and a trust entity. To provide worldwide service, there can be a chain of trust server in order that, if the trust server does not have the trust value for an object, the trust server can inquire about the trust from the next server. To make our model simpler, we use a single server in this work.

B. Trust model Basic Elements

In this section, we define the trust management parameters then trust management steps

We use the following parameters to compute trust:

- Feedback system parameters (F=1, F=0.5, F=2): Parameters are used to evaluate respectively the successful transaction (F=1), uncertain transaction (F=0.5), and failed transaction (F=2) between two object i and j. When malicious behavior is detected, the object will be punished double to discourage malicious behavior and to keep the idea of trust which is easy to lose and hard to get.
- Transaction weight (W_{cx}^{l}): Is used to measure the weight of a transaction l between two objects in the context cx, this weiht is used to prevent an object from behaving well on low weight services to build a good reputation then behaving badly on services with high weight.
- The computation capabilities weight (C): We classify the objects into 3 classes according to their computation capacity.
  - Class 1 (C1) includes low-capacity objects, such as RFIS and sensors.
  - Class 2 (C2) contains objects with an average capacity as the smart-phone.
  - Class 3 (C3) consists of high-capacity objects, such as servers and desktop computers.

  It is assumed that an object can move from one category to another if the resources exceed a predetermined threshold.
- Context weight (W_{cx}): Reflects the importance of the context, as we assume that more interaction in a specific context cx means higher context importance leading to more weight in the trust evaluation.

C. Trust model steps

1) bootstrap: For initial trust value, we assume that each node assigns an initial trust value to its new neighbor based on the relationship links two objects. Authors, in [6], defined objects relationships as owner relationship established if the objects are registered in SIOT by the same user, and collocation relationship (COLR) when the objects are domestic objects or objects of the same workplace. In these two types of relation, it is very unlikely to find a malicious node. For this reason, the highest initial trust values are assigned to these objects. Social relationships (SR) are those established between objects that are encountered occasionally for a common interest, which makes them associated to a small trust values. The relationships are considered as parental relationships (PR) when objects are homogeneous or manufactured in the same period by the same manufacturer, such relations are the most risky, since they are created between objects of the same brand but that never met. Based on the aforementioned reasons, we can define the initial trust value of an objects j as seen by another one i as shown in table 1. These evaluations are stored in the trust manager and used as inputs for the trust management system.

TABLE I
INTIAL TRUST VALUES

Object Relationships	Initial trust values
Ownership relationship (OR)	0.9
Co-location Relationship (COLR)	0.7
Social Relationship (SR)	0.5
Parental Relationship (PR)	0.5

2) Service request: We assume that when Alice needs a service S3, she should send a request by its object A to the service server. After that, the discovery mechanism is triggered and the service server returns the objects that can provide the requested service for example Bob's device, Paul's device. Hence, Alice consults his local trust table to check their trusts. If she does not find histories about them, she sends a trust request query called TRQ=(Id_B, id_P, cx, S) to the trust server to ask about their trusts. ID_B and ID_P are the identifiers of Bob and Paul respectively and their devices identifiers. cx and S designate respectively the context and the evaluated service.

3) service provider selection: Upon receiving a request from Alice, the trust manager starts the entity selection process to return the most trustworthy user among Bob and Paul to the requester. In the best case, service providers have already interacted for the same service and with the same context, Hence, the server return their trust values. Contrariwise, the system may be in a situation where the candidates nodes have not yet been evaluated for the current requested service, or they were in other contexts when the request service has been evaluated. To resolve the problem of this lack of information, we proposed to use an algorithm composed by these steps:

step 1 *Trust prediction based on decision tree*:

In this section, we try to predict the behavior of the service providers for query requesting by deducting the relationship between the different attributes and trust. Thus, we use a decision tree having an easy to follow natural flow. Indeed, the decision tree is a very useful modeling technique since it is easy to be understood, it can provide a simple visual representation of data and it allows extracting the classifications rules. Then, each branch represents a test result and each leaf node specifies a category. In our model, the decision tree can help us to find the relationship between attributes and compute the preference similarity. We adopt the algorithm developed by Quinlan C4.5 [7] to construct our decision tree for each service provider. Hence in the training step, we define first the attributes of our model which are the evaluated node feedback record in TMS as defined in t=(cx, S, C, OR). Where cx is the context in which the transaction is made, S is the evaluated service, C is the capacity of the evaluated object, OR is the object relationship, between the two object. These attributes are given by the trust management server. The classes used, in our model, are the feedback evaluation (T, U, R). They are related respectively to success, failed and uncertain transactions. So, each training sample in the training set is a five-attribute training, denoted as t = (cx′, S′, C′, OR′, Class) and the first dimensions, in T, are depicted as internal nodes of the decision tree; whereas the class is represented as the leaf node of the tree. After the training step, the decision tree is ready for application in the service request process. After the application of decision tree, our system will determine the trustworthy provider nodes called selected nodes.

step 2 *Social similarity computing using Jaccard Similarity Index*:

The second step consists in computing social similarity between the selected nodes and requester node by using the Jaccard Similarity Coefficient which is a statistic tool used for comparing the similarity and diversity of sample sets and is defined as the size of the intersection divided by the size of the union of the sample sets. The sample sets in our model are the friendship-list, the community-list, Object-profile-list. For the normalization, we assume that if two attribute are similar we assign the value 1, otherwise 0.

- Friendship-list similarity (Sim^F) : The friendship similarity is a powerful social relationship (intimacy) for screening recommendations and is computed in equation 1.

$$sim^F(A,B) = J(A,B) = \frac{\mid F_A \cap F_B \mid}{\mid F_A \cup F_B \mid} \qquad (1)$$

  where F_A and F_B are friends lists of A and B

- Community-Interest-list similarity (Sim^{CoI}) : Two users in the same community of interest share similar social interests and are most likely to have common knowledge and standard toward a service provided by the same device. It can be computed in the same way by using equation 2.

$$sim^{CoI}(A,B) = J(A,B) = \frac{\mid COl_A \cap COl_B \mid}{\mid COl_A \cup COl_A \mid} \qquad (2)$$

  Where COl_A and COl_B are the communityIterest-list of users A and B.

- Object-profile similarity (Sim^O) : Like human beings, devices also can have profiles, including devices basic information, such as their manufacturers, owners, and working conditions. Therefore the trust server, the Object-profile similarity between $A(O_A)$ and $B(O_B)$ is computed as follow :

$$sim^O(A,B) = J(A,B) = \frac{\mid O_A \cap O_B \mid}{\mid O_A \cup O_A \mid} \qquad (3)$$

The social similarity between two objects can be weighted by combination of all social similarity metrics considered in this paper; friendship, community of interest, profile devices. To compute the average of the social similarity, we use the following equation:

$$Sim(A,B) = \sum_{k \in \{F,O,CoI\}} Sim^k * SW_k \qquad (4)$$

where $SW_{Sim^F} + SW_{Sim^{CoI}} + SW_{Sim^O} = 1$ and are used to give a weight to each metric according to its importance. The service server send then a similarity query called (SQ) to the trust management server and defined as follow:

$$SQ(A,B) = (id_A, id_B, Sim(A,B))$$

step 3 *selected node's credibility computing*

Inspired by our social life, we assume that the credibility between two objects increases with the increase of the similarity between them. hence after receiving of similarity measures we compute the credibility of selected object B as seen by the requester object A by application of formula 5.

$$Cred_{AB} = \Omega(R_B) + (1 - \Omega(Sim(AB)) \qquad (5)$$

Where R_B is the reputation of the selected node B computed by the server using equation 10, $Sim(AB)$ is the similarity computed between the service provider and requester service and Ω is the weight attributed for each term to keep the value of credibility between [0..1]

4) Transaction and evaluation: We describe in this section, the transaction evaluation, Contextual trust computation, and the reputation computation.

a) Transaction evaluation at node level: After the transaction, the requester node must evaluate the service provider from the selecting node, by the experimentation of its satisfaction, we assume that the feedback=1 if the transaction is successfully completed and feedback=2 for the failed transaction. Indeed when malicious behavior is detected, the object will be punished double to discourage malicious behavior and to keep the idea of trust easy to lost and hard to get. Finally, we assume that feedback=0.5 for uncertain transaction. After transaction, each user must send the evaluation query, called EQ, to the trust server for trust management composted by the id_A of the evaluated object, the identity of the object that was evaluated id_B, feedback F, the context cx, the evaluated service S, the capacity of the evaluated object C_B and object -relationship OR. So the EQ between two objects may be expressed as follows:

$$EQ = (id_A, id_B, cx, C_B, S, OR, F)$$

b) Evaluation on trust server: This part comprises two parts the contextual trust and reputation computing.

- Contextual trust computing "T^{cx}"

The trust server is responsible for computing contextual trust and reputation. So we compute the contextual trust T_j^{cx}

for object j for each context and for each service using the Dirichlet distribution [8], which is the equivalent of the Beta distribution extended from binary events to multiple events. Thus, Dirichlet distribution can address, in a better way, different behaviors in complex environment as IoT. The T_j^{cx} value can be presented by the mathematical expectation of the probability density function using equation6 where α_j^{cx} , β_j^{cx} and λ_j^{cx} are given by equation 7 and are used to compute respectively the successful(S), uncertain(U) and failed(E) transactions respectively, between nodes i and j for a specific context. W_{cx}^l is the weight of the l^{th} transaction (service) and N_{cx} is the number of transactions between nodes i and j for a context cx.

$$T_j^{cx} = \frac{\alpha^{cx} + 1}{\alpha^{cx} + \beta^{cx} + \gamma^{cx} + 3} \tag{6}$$

$$\alpha_j^{cx}, \beta_j^{cx}, \lambda_j^{cx} = F_{ij} * \sum_{l=1}^{l=N_{cx}} *W_{cx}^l \tag{7}$$

This formula is used to compute the contextual trust presenting limits. The time between the transactions is not taken into consideration. Logically the recent feedbacks is more important than the old ones because a malicious object can change its behavior over time. To solve this problem, we introduce, in our work, the forgetting factor to give a weight to each feedback. We define the forgetting factor as shown in equation 8.

$$\phi^l = \lambda^{N_{cx} - l} \tag{8}$$

Where N_{cx} is the total number of transactions for the context cx and l is the current transaction, so we update the value of α and β as follows :

$$\alpha'_{ij}, \beta'_{ij}, \gamma'_{ij} = F * \sum_{l=1}^{l=N_{cx}} *W_{cx}^l * \phi^l \tag{9}$$

- Reputation computation "R"

If node i interacted with node j in m number of contexts cx_1,cx_2,...,cx_m, then we define the reputation of an object j computed by the TMS as the sum of all the contextual trusts realized in different contexts multiplied by the weight of each context as shown in 10

$$R_j = \frac{sum_{k=1}^{k=m} T_j^{cx_k} * W_{cxk}}{NR_{cx}} \tag{10}$$

Where m is the number of contexts in which object j is evaluated and NR_{cx} is number of recommendations received, W_{cxk} is The weight of the context k. We assume that more interaction in a specific context means higher context importance leading to more weight in the trust evaluation. For this reason, we compute W_{cxk} by application of formula 11, where N_{cx} is the number of transactions realized in all the contexts during the period of time q and N_{cxk} is the number of transactions in the context k.

$$W_{cxk} = \frac{N_{cxk}}{N_{cx}} \tag{11}$$

IV. Evaluation

"Up to now, no real implementation of SIOT has been proposed. consequently there is no real dependent data base. We collected data from the various studies presented in the literature to create our database. In fact, authors in [4] assumed that objects with high resources have more capabilities to cheat. For example, if a user wants to know the air conditioning of a room knowing that two objects will provide the service, a sensor and a smart phone. In this case, the risk of malicious behavior increases with the smart phone. We begin by showing the simulation obtained in execution our trust management proposed by Iot devices. We consider a network formed by 100 objects, randomly assigned to 100 users, these users are connected to a social network and can maintain friendly relations. These objects are grouped together in communities of interest, and can request a random service with random time between 1 and 60 seconds. For the analysis, we select one node and we analyze its efficiency to detect the normal and malicious behavior of another nodes,We supposed that each object can provide 3 services with different weights Ws1 = 0.95, Ws2 = 0.03 and Ws3 = 0.1,respectively.

A. Nodes classification based on the forgotten factor

In this section, we assess the effect of the forgotten factor ϕ as defined in equation 8 on the trust value of an honest object and dishonest one, A forgotten factor = 1 hands that all transaction is weighted equated and noting is forgotten, with forgotten factor equal to 0, only the last transaction is remember. Figure 2 and figure 3 show that the value of λ determines the maximum trust value that an object can get.

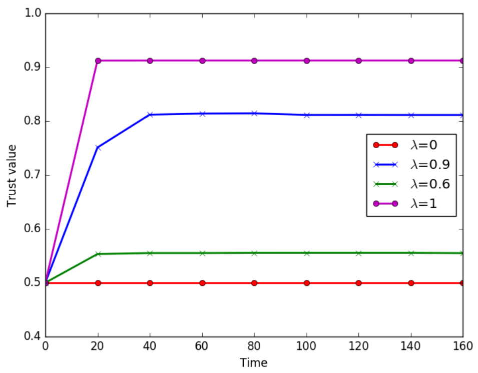

Fig. 2. The effect of the forgotten factor on trust value of normal object

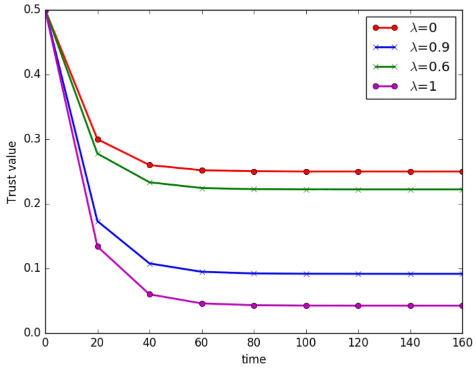

Fig. 3. The effect of the forgotten factor on trust value of malicious object

B. Node behavior detection based on decision tree

We are specifically concerned with trust prediction in SIOT. We applied the machine learning tool WEKA using the C4.5 algorithm to draw the decision tree based on the training set of selected node, capable of providing 3 services; S1, S2 and S3. It is assumed that the selected node has indicated that it is working in the ”‘context = day”’ at the registration time, The training set of selected node are the results of simulations performed with our TMS during 2 hours, The values that can take each attribute are

- Context : We specify two context ”day” and”””night”.
- object-relationship : COlR, SR, PR, OR. as defined in bootstrap phase.
- Object-capacity, we define three types C1, C2, C3, as explained above.
- Service: S1, S2, S3.

Figure 4 shows the result of the decision tree with the object-relationship OR attribute as the root. That is to say, the information of OR seen to be the major contributor, for the similarity degree and the information of context are in the second place. According to the decision tree, the selected node behaves well and provides the requested service when the object relation is Ownership-. When the relationship which brings the two objects are collocation and the context is the day, the object is trustworthy, whereas when the context is night, the risk increases and the behavior becomes risky. Table 2 illustrates the accurateness of the decision tree approach: 91.5% of cases were correctly classified with very small statistical error margins. Experiment results demonstrate that using the decision tree can improve the decision making in the trust managements systems.

TABLE II
SUMMARY OF RESULTS FOR THE DECISION TREE

Correct Classified	91.5 %
Incorrect Classified	8.41 %
Kappa statistic	0.86%
Mean absolute error	0.095%
Root mean squared error	0.232

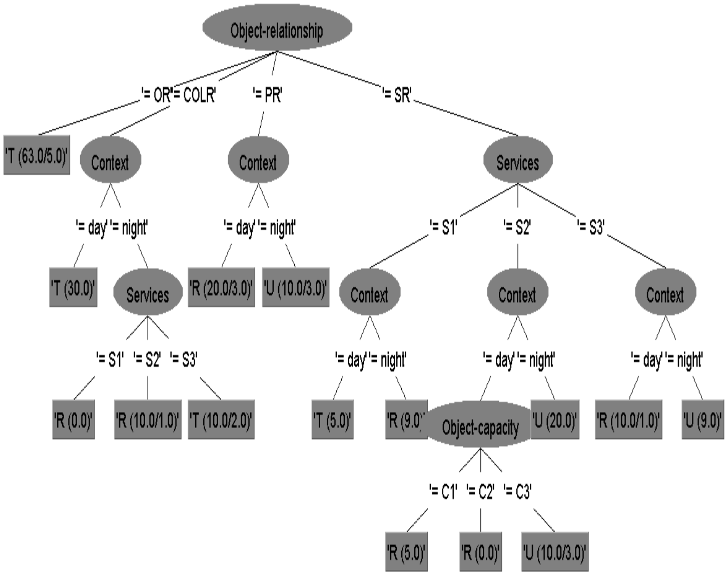

Fig. 4. The decision tree

V. CONCLUSION

In this paper, we have proposed a new centralized trust management system for the social internet of things called CTMS-SIOT. CTMS-SIOT is able to return the trustworthy objects for each context and each service, even if no history on the nodes behaviors exists. Indeed we have introduced a robust algorithm that uses Jaccard Coefficient to compute the Social similarity between objects and decision tree technique to predict nodes behaviors. We notice that the proposed solution has strengthen the decision-making performance. As future work, we plan to implement other machine learning techniques for behaviors predictions.

REFERENCES

[1] D. Chen, G. Chang, D. Sun, J. Li, J. Jia, and X. Wang, “Trm-iot: a trust management model based on fuzzy reputation for internet of things,” *Computer Science and Information Systems*, vol. 8, no. 4, pp. 1207–1228, 2011.

[2] F. Bao and I.-R. Chen, “Dynamic trust management for internet of things applications,” in *Proceedings of the 2012 international workshop on Self-aware internet of things*. ACM, 2012, pp. 1–6.

[3] G. Lize, W. Jingpei, and S. Bin, “Trust management mechanism for internet of things,” *Communications, China*, vol. 11, no. 2, pp. 148–156, 2014.

[4] M. Nitti, R. Girau, L. Atzori, A. Iera, and G. Morabito, “A subjective model for trustworthiness evaluation in the social internet of things,” in *Personal Indoor and Mobile Radio Communications (PIMRC), 2012 IEEE 23rd International Symposium on*. IEEE, 2012, pp. 18–23.

[5] R. Chen, J. Guo, and F. Bao, “Trust management for soa-based iot and its application to service composition,” 2015.

[6] F. Bao, I.-R. Chen, and J. Guo, “Scalable, adaptive and survivable trust management for community of interest based internet of things systems,” in *Autonomous Decentralized Systems (ISADS), 2013 IEEE Eleventh International Symposium on*. IEEE, 2013, pp. 1–7.

[7] J. Quinlan, “C4. 5: Programs for machine learning. c4. 5-programs for machine learning/j. ross quinlan,” 1993.

[8] A. Jøsang and J. Haller, “Dirichlet reputation systems,” in *Availability, Reliability and Security, 2007. ARES 2007. The Second International Conference on*. IEEE, 2007, pp. 112–119.