This paper is a postprint of the paper published online in Computer and Electrical Engineering at DOI https://doi.org/10.1016/j.compeleceng.2021.107568

Trusted Object Framework (TOF): A Clustering Reputation-based Approach Using Edge Computing for Sharing Resources among IoT Smart Objects

Giancarlo Fortino[a], Lidia Fotia[a], Fabrizio Messina[b], Domenico Rosaci[c], Giuseppe M. L. Sarné[d,*]

[a]*Department DIMES, University of Calabria, v. Pietro Bucci, 95126 Arcavacata di Rende (CS)*
[b]*Department DMI, University of Catania, v.le Andrea Doria 6, 95126 Catania (CT)*
[c]*Department DIIES, University "Mediterranea" of Reggio Calabria, Loc. Feo di Vito, 89122 Reggio Cal. (RC)*
[d]*Department of Psychology, University of Milano Bicocca, Pz. dell'Ateneo Nuovo 1, 20126 Milano (MI)*

Abstract

In the Cloud of Things, data are stored and processed in the cloud and the results are sent to IoT Smart Objects (SOs). This architecture generates, among others, a certain overhead in terms of network latencies and generally, increasing costs due to the distances between the cloud and the local IoT networks. Edge computing lies in the middle, encompassing data computing and storage that are performed on the "edge" nearby the SOs networks. Noteworthy, trusting inappropriate counterparts can expose to several potential threats for malicious, fraudulent and/or disliked behaviours, also potentially encouraged by the nature of the relationships. Therefore, in order to mitigate risks given by the selection of unreliable partners, in this paper we propose a clustering reputation-based approach for IoT Edge-based platform. Whenever SOs interact for services, a feedback is sent to an Edge server, which will exploit such feedback to calculate the reputation scores of the SOs. In this way the reputation systems is moved from the cloud to the edge servers. Moreover, in the case a SO moves from its own edge domain to another one, its reputation score will not be lost because it will be still saved on the cloud. To implement our proposal we have designed a distributed *Trusted Object Framework* (TOF) where heterogeneous OSs host and exploit the assistance of associated software agents. To verify efficiency and effectiveness of our approach we carried out some experiments – in a simulated scenario – which confirmed the potential advantages deriving by the adoption of TOF.

Keywords:
Edge Computing; Internet of Things; Trust system; Clustering; Multiagent system

*Giuseppe M. L. Sarné, Department of Psychology, University of Milano Bicocca, Pz. dell'Ateneo Nuovo 1, 20126 Milano (MI), Italy, e-mail: giuseppe.sarne@unimib.it

Email addresses: giancarlo.fortino@unical.it (Giancarlo Fortino), lidia.fotia@dimes.unical.it (Lidia Fotia), messina@dmi.unict.it (Fabrizio Messina), domenico.rosaci@unirc.it (Domenico Rosaci), giuseppe.sarne@unimib.it (Giuseppe M. L. Sarné)

1. Introduction

The Internet of Things (IoT) [2] is gaining an increasing relevance in our daily lives. Its ability to transform the world by realising smart environments is denoted also by adaptive forms of cooperation among smart objects (SOs) and/or users to make available potentially useful and attractive services [3, 4, 5, 6, 1]. Smart Objects (as smart sensors/devices) are able to collect and exchange a lot of data among them through network infrastructures potentially connecting millions of IoT nodes. Moreover, the IoT applications supply accurate network services to their users. When many SOs are connected by IoT techniques, they produce massive data to be processed and further require to provide intelligence to both service providers and users. Formerly, several cloud-based environments have been developed for allowing the access to IoT devices and providing them with communication, computational and storage resources [7], as well as for delegating tasks to other physical and/or virtual components living on the cloud. This solution avoids the consumption of valuable resources from carrying out such activities locally. For example, in the Cloud of Things, data are stored and processed on the cloud and the results are sent to the IoT layer (Smart Objects). Unfortunately, the adoption of this architecture generates overheads as (e.g., network latencies), which represent a very critical aspect for IoT applications that are usually time-sensitive. Indeed, environments such as smart transportation, electricity grid, smart cities and many others, present the "non-negotiable" requirement of quick responsiveness.

In summary, in our vision the cloud computing solution holds some limitations due to the following aspects:

- the computation processes need to be uploaded to the cloud;
- the limited bandwidth and network resources are committed by massive data transmissions;
- the non-negligible network communication latencies.

Moreover, power consumption represents a critical aspect of SOs. As a consequence tasks which are computationally heavy are often moved to devices having more availability and/or computational capabilities then SOs.

In order to solve the problems highlighted above, the edge computing [8, 9, 10] allows computational and communication overhead to be shifted from SOs, having limited power and computational resources, to edge servers provided with significant resources and nearby the SOs. This way, edge computing can relieve the peak in traffic flows, mitigate the bandwidth requirements, reduce the transmission latency during data computing or storage in IoT activities and increase the IoT network lifetime and effectiveness. In such an edge computing IoT scenario, potentially heterogeneous SOs can cooperate with well equipped SOs placed in their proximity, to consume/produce services and/or extract/exchange knowledge.

The idea behind this work is to promote the mutual cooperation among SOs in the scenario above described, making available the resources belonging to the more closed, equipped and performing SOs (usually not those devoted to critical missions), allowing to pay for obtaining a service [11] (for instance, by adopting a micro-payment system [12]).

However, trusting inappropriate counterparts can expose SOs to several potential threats due to malicious, fraudulent and/or disliked behaviours [13]. Risks can significantly increase in

the presence of open and heterogeneous environments and/or when the involved relationships include fee payments or other valuable benefits. We argue that a certain level of confidence and mutual trustworthiness is fundamental for motivating the sharing actors to interact on the basis of a reasonable hope to be engaged in fulfilling interactions. Conversely, a poor level of confidence can compromises the possibility of choosing a reliable partner. To mitigate the risks due to unreliable partners [14], independently on their nature (human, objects or virtual entities [15]), reputation systems [16] can be adopted to create a confidence atmosphere. Reputation systems are capable to provide a measure about the expectation that a trustor has to receive benefits from a trustee by taking into account direct or indirect information about past behaviours or events [17].

In this paper, we propose to introduce a clustering reputation-based approach in an IoT Edge-based platform, such that after two SOs exchanged a service, then the edge server, which they are referring, will receive a feedback sent by each SO about its counterpart. The edge server will exploit such feedback to calculate the reputation scores of the SOs. In other words, for not burdening the SOs, the reputation systems is moved from the cloud, where the most of the past proposals in the literature collocate it to the edge servers. In this way, each edge server calculate a reputation score of its own SOs, being able then of dividing the SOs into clusters generated on the basis of the SOs' reputation. Clustering is then leveraged to realise a competitive environment in which each SO is encouraged to gain a reputation to position itself in the clusters of the most trusted SOs. As we discuss later in this work, the adoption of clustering offers to the SOs a few benefits. For instance, a SO can exploit the reputation of its own cluster to sell resources at the best possible price when it acts as a provider and, vice versa, can purchase them at a good price in case it acts as a consumer. On the other hand, whenever a SO moves from its own edge domain to another one, its reputation will not be lost because it will be still saved on the cloud. This means that on the cloud there will be a periodically updated repository that contains the SOs's reputation score of the overall IoT SO population.

Basing on the premises above, our proposal is aimed to obtain the following advantages:

- the reputation score is calculated on the edge server by relieving the SOs;
- correct SOs behaviours are promoted by means of a competitive mechanism;
- the mobility of SOs is still possible by preserving past reputations, because the cloud allows collaboration among edge servers, as they have the possibility of obtaining the reputation history of SOs.

A further intuition underlying our proposal is of tackling the management of the SOs cooperation by implementing an agent-based framework where each SO hosts a light tamper-proof software agent capable of basic interactions and social behaviours [18, 19]. These software agents, by following a sharing philosophy, are supported by the reputation-system located on its edge server to require/accept of cooperating with a reliable partner placed in their proximity and by taking benefits from their trustworthiness (i.e., from the belonging to a cluster, established based on their reputation value). At the end of the interaction, the interacting agents will issue a feedback to the edge server which will update their reputation scores. It is obvious that on the edge server there exist an agent which autonomously manages the reputation algorithm.

In order to implement the above proposal we designed a distributed *Trusted Object Framework* (TOF) where heterogeneous SOs host and exploit the assistance of associated tamper-prof software agents.TOF is composed of distributed *framework* agents, which provides some basic services to all the object agents. Also, we introduce a clustering reputation-based model in order to evaluate the reputation of an OS as a partner, for assigning each SO to a cluster based on its trustworthiness to allow the best SO partner choice to cooperate for sharing resources. Consider that, each hosted agent on SO monitors resources and communication activities of its host, and disseminates the reputation of its own host by interacting with the other framework component in a safe manner [20]. Instead, each agent hosted on the edge servers manages the reputation of its associated SOs. However, note that in the following issues strictly related to authentication and payment mechanisms are considered as orthogonal with respect to the focus of this proposal, and therefore they are not dealt with in this paper. In order to validate the presented approach, we carried out an experimental campaign by means of a simulated agent scenario, which confirmed the potential advantages deriving by the adoption of TOF.

The rest of the paper is organised as follows. Section 2 gives an overview on the related literature. Section 3 introduces the proposed agent framework, while Section 4 describes the adopted reputation model. The experimental results are presented in Section 5 and in Section 6 some conclusions are drawn, as well as a brief discussion related to potential future works.

2. Related Work

In this section, we first discuss edge computing issues in the IoT context and then we introduce related technologies on a trust and reputation system in IoT. The various aspects related to these topics have been dealt in a large number of scientific contributions and, therefore, their complete contextualization within these backgrounds is beyond our aims. For such a reason, the examined approaches are those that, to the best of our knowledge, come closest to our proposal. However, the interested reader might refer to the numerous existing surveys, among which [21, 22, 23], for a more complete overview.

2.1. Edge Computing in IoT

In the IoT environment, SOs generate more information that can be collected and processed through approaches typical of the big data technology to turn it into something that is useful. There existed the needed of extrapolating helpful information from unstructured big data. For this reason, IoT provides a different vision of big data in data storage, data processing and analytic activities. A proposed solution was that of displacing data into the cloud [24, 25], but the nature of IoT data posed the question of choosing the best technology for performing such activities. Unlike the cloud infrastructure, the edge computing can manage the entire data management processes for its closeness to data sources, i.e., the IoT SOs [26, 27, 28, 29].

More specifically, an edge computing node offers storage, computing and network connectivity to processed at once by the edge computing node nearest to the IoT SOs that generated data. However, the introduction of edge computing requires to solve the problems related to its administration as well as that of allocating resources to IoT SOs. Further issues to be integrated in edge computing are trust, security and privacy that represent severe challenges [30, 31, 32, 33].

In [69], they explore the resource-efficient edge computing issues for intelligent IoT applications. They design a resource-efficient computation offloading algorithm to permit an IoT device to use resources across the local device and the edge cloud in proximity. In particular, to reduce an intelligent IoT device's edge resource occupancy while satisfying the QoS requirement, this algorithm is composed of a delay-aware task graph partition algorithm and the optimal virtual machine selection method. An intrusion-detection system (IDS) is used to mitigate security threats in edge computing. Lin et al. [68] study IDS architecture and resource allocation in edge computing. Their system facilitates multiple resources sharing and heterogeneous resource-demanding allocation. In particular, the authors present a general edge computing IDS architecture and use this as the basis for their model to allocate resources. Also, a single-layer dominant and max–min fair (SDMMF) allocation is used and a multilayer resource allocation scheme is used to cope with the multiple resources fair allocation in multiple layers. In [67], the authors propose a resource allocation model for allocating computing resources in an edge computing platform, named Zenith. It uses a decoupled architecture where the infrastructure management at the Edge Computing Infrastructures (ECIs) is executed independent of the service provisioning and service management performed by the service providers (SPs). Then, they propose an auction-based mechanism for resource contract establishment and a latency-aware scheduling technique that maximises the utility for both ECIPs and SPs.

In [34], the authors summarised the opportunities and challenges of edge computing in IoT and sustained that edge computing solves the technology gaps in IoT. Sun et al. [35] introduced an IoT architecture, named EdgeIoT, to manage data streams from IoT SOs at the mobile edge. Also, Jutila [36] suggested an adaptive edge computing solution to optimise IoT traffic flows and network resources. In [37], the authors proposed a workload allocation scheme in a hierarchical edge network to reduce the response time of task requests. To minimise the service delay for IoT applications, Yousefpour et al. [38] discussed a delay aware policy for the IoT–fog–cloud network. In order to achieve the optimal and stable performance in the IoT-based network, in [39], the authors designed an edge IoT framework to allocate the limited computing resources of fog nodes to IoT users. Yang et al. [40] analysed the joint optimisation of service placement and load dispatching in the mobile cloud systems. Then, they proposed a set of algorithms to reach trade-offs between the latency of users' requests and the cost of service providers.

2.2. Trust and Reputation Systems in IoT

Reputations systems, as well as group formation, play an important role in IoT contexts [41, 42, 43], while cryptographic techniques protect privacy and authentication [44, 45], trust and reputation systems estimate the trustworthiness of potential partners.

First, we introduce the most accurate definitions of reputation and trust in the literature. The reputation [70] of an entity is an expectation of its behaviour based on other entities' observations or the collective information about the entity's past behaviour within a specific context at a given time. Trust [71] is the extent to which one party is willing to depend on something or somebody in a given situation with a feeling of relative security, even though negative consequences are possible.

The reliability achieved from direct experiences is the most important measure for the trust. In addition, any potential partner or service customer might wonder how much the community

(or a subset of it) trusts a certain peer. For this reason, trust is calculated by considering direct agents experiences (reliability) and or opinions of others (reputation). Reliability and reputation are ordinarily merged in a synthetic measure [46, 47, 48, 49].

An important issue in open, competitive and distributed scenarios is to provide a comfortable environment where the involved actors can realise own activities. In this context, it is important to reduce the large number of potential threats due to malicious behaviours [13].

To solve the problem of unreliable partners, in the literature several trust systems have been proposed [14, 15, 50, 51].

For instance, Bao et al. [52] presented a dynamic trust management protocol that exploits the social relationships among IoT objects to develop a trust-based service composition. The trust system introduced in [53] considers the evolution of social relationships over time and adapts itself to the trust swing. Also, in [54] it is highlighted that IoT SOs hold heterogeneous skills and are able to perform complex tasks. Hence, IoT SOs utilize direct experiences and available recommendations to evaluate the trustworthiness of their peers and to match services' demand and offer. In [55], the authors proposed an approach for machine-to-machine applications, named BETaaS, that supplements a trust model based on security, QoS, scalability, availability, and gateways reputation. This model is characterised by evaluating SOs reliability based on the monitoring of their behaviours.

SOs can be arranged in groups/clusters formed by similar peers on the basis of their social interactions and mutual trust evaluations [56]. In forming trust-based groups/clusters in IoT environments, it is very important to consider scalability and countermeasures against different attack typologies, included bad-mouthing attacks. In fact, Alshehri et al. [57] proposed some methods for scalable trust-based IoT clustering joined and for countering bad-mouthing attacks on trust systems. Moreover, their contribution also considers trust computation and trust-based migration of IoT SOs from one cluster to another.

Finally, in [58], the authors studied the convergence among IoT, software agents, and cloud computing [59] to form agent groups/clusters. In particular, they introduced an algorithm to form agent groups/clusters through reliability and reputation gathered by the agents. The experimental results proved that the proposed approach leads to form groups with high values of mutual trust.

3. The IoT Edge-based Scenario

Figure 1 depicts the reference scenario composed by a set of local edge servers, each of them managing some heterogeneous SOs acting into the associated local domain, and working independently of each other. In this context, we assume that SOs share resources for pay. To support this activity, two agents are introduced, respectively the *edge agent* and the *trust agent*, where:

1. An *edge agent*, denoted by e, is associated with the edge domain and supports SOs in their activities;
2. A *trust agent*, a light tamper-proof component denoted by t, manages reputation tasks and monitors hardware resources and communication activities of its host (for detecting malicious activities) and interacts with the agent e associated with its current edge domain.

Note as t works into the interest of both its SO and the framework so that it can not be tampered neither by malicious or by the same SO (i.e., its owner).

In the following, we will refer at two different SOs, named SO_1 and SO_2, by means of their trust agents, denoted as t_1 and t_2, respectively.

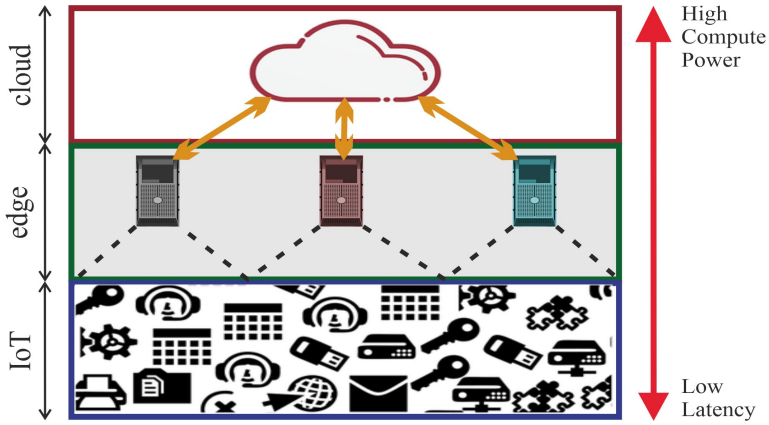

Figure 1: The proposed scenario - level scheme.

The main activities carried out within the TOF framework are:

1. ***Edge agent tasks*** *(e)*:
   • *Affiliation* - When a SO desires to operate within the agent framework, it must carry out an affiliation process with an agent e associated with one of the domains active into the framework. To this aim, once e has verified the "identity" and registered the SO into the framework then t can start to run on the SO. The agent t will be provided by e with *i*) an initial reputation R^0 (set to 0.5 [60, 61]) *ii*) a pair of personal asymmetric cryptographic keys, *iii*) the public key of the edge server, and *iv*) a certificate provided of expiration date and signed by e[1] to witness device identity and reputation[2], also for avoiding significant waste time for latency in interacting with e.
   • *Domain Management* - e provides to associate each SO to a cluster based on its trustworthiness. To this aim, each e *i*) manages the reputation model presented in Section 4, *ii*) updates both the SOs' reputation score and their certificates based on the feedback released by each SO's counterpart and *iii*) sends the updated certificates to both the respective t and to a repository in the cloud (shared with the other edge agents of the framework) to allow SO mobility among the different edge domains forming the TOF framework.
2. ***Trust agent tasks*** *(t)* will perform the following tasks:

[1]Note that each time a SO moves itself from an edge domain to another, then it needs to require a new certificate signed by the edge agent of its new edge domain.

[2]To avoid a partial representation of the reputation into the framework, the adopted solution has been of not adopting the usual second hand approach but of using signed certificates managed by tamper-proof agents, which can be also considered as local stubs of the edge agent.

• *Presentation* - It carries out interactions with t associated with other SOs and sends information about their identities and reputation scores to its associated SO;
• *Interaction* - It manages the feedback and the certificate of its host SO by interacting with the trust agents associated with other SOs.

3. ***SO tasks*** - To share resources, each SO carries out two main tasks named *Service Search* and *Service Provisioning*. More in detail:
• *Service Search* - This task is realised when a SO is looking for a service. In this case, it requires to its t of acquiring the certificates (storing identity and reputation) of SOs operative in the same area, via their hosted trust agents, in order to verify their trustworthiness. Based on the received certificates, it will select those SOs that meet its individual *hazard threshold* [3] (in terms of a cluster, see Section 4) and which ask for a resource r of interest by sending them the descriptor of r, denoted by D_r. Note that if any neighbour object satisfies the trustworthiness criteria, then the search process ends here or it could be repeated after that a time frame will be elapsed.
• *Service Provisioning* - When an SO_1 receives the descriptor D_r of a resource r offered by SO_2, the SO_1 formerly verifies if its hazard threshold is satisfied by SO_2 and only in case of a positive response then the tuple $\langle D_r, C_r \rangle$ will be sent to SO_2, where C_r is the cost that SO_1 is willing to pay for r. If SO_1 and SO_2 agree on C_r, then SO_2 will make the resource r accessible to SO_1. After that r has been consumed, the feedback about their counterparts are sent by both the SOs to e and the trust agents t_1 and t_2 will receive by it the new certificates with their updated reputation, as explained in Section 4. Note as both *i*) possible negotiation strategies and *ii*) the modalities to make accessible the shared resources are behind the aim of this proposal and, therefore, they will not be discussed here.

4. The Agent Clustering

In our framework, each edge agent clusters the SOs belonging to its domain (see Section 1) on the basis of their trustworthiness. To this aim, as clustering criterion the reputation, i.e. *an expectation about future behaviours based on information about observations on past behaviours* [65], is adopted.

In order to highlight better the advantages of the adoption of the clustering, we remark that, from a practical viewpoint, the inclusion of a SO in one or another cluster (on the basis of its reputation score) allows it to obtain some benefits. For instance, a SO can sell resources at the best possible price depending on its own cluster when it acts as a provider and, similarly, it can purchase them at a low price when it acts as a consumer (see Section 5).

4.1. The Reputation Model

The reputation model adopted to estimate the agent trustworthiness, based on the above reputation definition, considers the SO's history, in other words its behaviour in terms of feedback received from its counterpart.

[3] The hazard threshold [62, 63, 64] represents the probability of "failure" that the an entity is willing to accept.

To this purpose, once a resource has been provided by a SO, which acts as a provider, to another SO, which acts as a consumer, then their trust agents will send the respective *feedback*, computed by the SO (e.g., the SO's owner or by an associated personal SO agent), to the edge agent for updating their own reputation scores and certificates. In particular, a feedback should reflect the behaviour of its counterpart also in the light of the own expectations about it. The feedback could differ between provider and consumer SOs because computed based on individual criteria (e.g., cost and quality of the resource, promptness in delivery or payment, etc.).

From a more formal point of view, let be $f^r_{1,2} \in [0,1] \subset \mathbb{R}$ the feedback computed by SO_1 about SO_2 for the resource r, where 0/1 means the minimum/maximum appreciation that SO_1 has about SO_2 for their interaction (i.e., let be $f^r_{2,1}$ the feedback computed by SO_2 about SO_1, with $f^r_{2,1} \neq f^r_{1,2}$). Then the trust agent t_1 (i.e., t_2) will send its feedback, with the cost of r, to the edge agent in a secure way. In turn, the edge agent will provide to update the SOs' reputation scores $R \in [0,1] \subset \mathbb{R}$ based on the feedback sent by the trust agents t_1 and t_2. With respect to the SO_1 (but analogously for SO_2), the new reputation score of SO_1 will be updated as follow:

$$R_1^{new} = \begin{cases} \alpha \cdot R_1^{old} + (1-\alpha) \cdot \Delta^r_{1,2} & \Delta^r_{1,2} > 0 \vee R_1^{old} \geq 0.5 \\ R_1^{old} & \text{otherwise} \end{cases} \tag{1}$$

with

$$\Delta^r_{1,2} = \Big(\beta \cdot \Phi_{2,1} + (1-\beta) \cdot \Psi_{2,1}\Big) \cdot \gamma_2 \cdot f^r_{2,1}$$

and where the parameter $\alpha \in [0,1] \subset \mathbb{R}$ weights the old reputation score R_1^{old} with the new contribution so that higher is the value of α, the lower will be the sensitivity of R and vice versa (see Section 5). The parameters β, γ, Φ and Ψ will be discussed in detail below. Note that against malicious activities, R will be updated only if the logical condition $\Delta^r_{1,2} > 0 \vee R_1^{old} \geq 0.5$ is true. Finally, an updated certificate witnessing the current reputation score (and identity) of SO_1 will be prepared and signed by the edge agent and sent to the trust agent t_1; contextually, the SO's reputation stored onto the cloud repository will be updated.

More in detail, in computing R_1^{new} the edge server takes into account some parameters and, in particular, the reliability of SO_2 in providing honest feedback that we assume here to be represented by its reputation. It is taken into account by means of the parameter γ that will be set to 1 if the SO's reputation is greater or equal than a threshold δ, where $\delta \in [0,1] \subset \mathbb{R}$ (see Section 5, or 0 otherwise. More formally:

$$\gamma = \begin{cases} 1 & R \geq \delta \\ 0 & \textit{otherwise} \end{cases} \tag{2}$$

The parameter Φ refers to the relevance of r that, for the sake of simplicity, we consider represented by its monetary cost C^r. The ratio of this parameter is that the lower the cost, the lower the relevance and, consequently, the effects on the reputation. In such way, it is possible to significantly limiting the benefits of malicious SOs in gaining a reputation for low relevant resources for then loosing it by cheating for high value resources. Moreover, we assume that the

relevance of r will be maximum when its cost will be greater than a suitable framework parameter denoting the maximum cost for r (i.e., C^r_{max}, see below). More formally, Φ is computes as follows:

$$\Phi = \begin{cases} 1 & C^r \geq C^r_{max} \\ \\ \dfrac{C^r}{C^r_{max}} & \textit{otherwise} \end{cases} \tag{3}$$

The parameter Ψ has been introduced against collusive behaviours where two or more SOs mutually and fictitiously increase their feedback with a high frequency. To this aim, Ψ is computed by considering the time-frequency that a feedback was provided to the same SO, as specified in the left part of Eq 4.

$$\Psi = \begin{cases} 1 & f^r < 0.5 \\ \\ \psi(t_2)^{-1} & f^r \geq 0.5 \end{cases} \qquad \psi(t_2) = \begin{cases} 1 & t_1 = t_2 = 0 \\ \psi(t_l) + 1 & t_2 - t_l < T \\ max\Big(1, \psi(t_l) - \Big\lfloor \dfrac{t_2 - t_l}{T} \Big\rfloor\Big) & t_2 - t_l \geq T \end{cases} \tag{4}$$

More specifically, when $f^r < 0.5$, the parameter Ψ is set to 1, otherwise it is computed as the inverse of ψ. The parameter $\psi \in [0, 1] \subset \mathbb{R}$ is initially set to 1 (i.e., the first time that the two SOs mutually interacted) and then it is decreased/increased on the basis of the time elapsed between two consecutive feedback released by the same SOs and the time period T, a framework parameter, see Section 5.

Finally, we observe that this reputation model is resilient against possible effects of malicious communication failures realised, for instance, to avoid of receiving negative feedback [66]. In fact, the presence of the edge server can be always exploited to allow a correct representation of the malicious SO's reputation. However, when a similar malicious behaviour has been detected then the reputation of the guilty SO will be penalised as $R_i^{new} = \eta \cdot R_i^{old}$, where $\eta \in [0, 1[\subset \mathbb{R}$ is a system parameter.

4.2. The Clustering

One of the edge agent's task consists of assigning SOs to clusters on the basis of their reputation scores. The aim of this clustering process is that of realising a competitive environment to promote correct behaviours. In particular, the belonging to a top-level cluster allows to a SO of gaining more/spending less for resources, depending it from the provider/customer role it is playing.

More specifically, the number of clusters depends by the edge domain and, in each edge domain, it can further vary over time in order to be adaptive as more as possible to changes that could occur in that context. For instance, possible criteria for deciding the number of clusters could depend from the number of interaction, the variety of resource offered/purchased or their maximum cost, etc.

Therefore, each edge domain will set the number of clusters and the reputation score thresholds for the access to each cluster in an arbitrary way with the aim of maximising the befits for

honest SOs and, at the same time, maximising the penalty for malicious SOs. To this end, an example will be provided in Section 5.

5. Experiments

The proposed framework has been tested by means of a number of simulations aimed to verify its effectiveness to recognise honest and malicious SOs, to cluster SOs based on their trustworthiness and to verify the economic convenience to adopt such a cluster mechanism for the SOs.

To this aim, for simplicity, we simulated only one TOF edge domain and two populations of SOs living therein. In particular, a first population was formed by 10,000 SOs acting as a consumer and a second one was formed by 500 SOs playing the role of resource providers. The ratio of this choice is represented by the observation that when a SO acts as either a consumer or provider (e.g., prosumers) based on its needs, it must be provided with a specific reputation score for each of the two roles. Therefore, from a practical point of view, the adoption of two separated SO population has not any consequence on the simulation results. Furthermore, a simple profile has been assigned to each SO to provide them with suitable behaviours appropriate to their honest or malicious nature. More in detail, each SO's profile stores information about the: *i*) honest or malicious SO's nature; *ii*) type(s) of malicious behaviours enacted by the SO and the data characterising each of such behaviours. Note how some of these data (e.g., the nature and specific malicious activities carried out by the SO) strictly depend on the simulated scenario and, therefore, were set accordingly to it; differently, other data present in the profile are assigned in a completely or partially random way.

The reputation scores are then exploited to realise a reputation-based clustering able to provide benefits (e.g., economic, easier resources finding, etc.) to honest SOs. In particular, to verify the benefits deriving by the clustering we analysed it also from an economic point of view. Remember that, as previously specified, the buying/selling of a resource will only be possible when the reputation of a counterpart is at least equal to the own hazard threshold. As practical consequence, some SOs will not be considered at all as a counterpart because of its low reputation score. We have also assumed that the final selling/purchasing price of a resource will depend on the cluster to which the SO belongs (see Table 1). In other words, if the two SOs belong to the same cluster then discount and increase are compensated. Otherwise, greater is the difference and greater is the economic advantage/disadvantages for the most/least reliable of them.

More in detail, each experiment consisted of 50 epochs, a number suitable to show stable and significant trends, and for each epoch 2500 SOs purchase resources (i.e. the 25% of the consumer population) by interacting with all the providers. To this end, two IoT scenarios have been simulated, named *A* and *B*, whose behaviours and parameters are explained below and recapped in Table 1. Note that some preliminary simulations lead to the adopted parameter setting.

The scenarios are characterised by the presence of malicious SOs equal to 10% of the entire populations but they take on different behaviours in the two scenarios. In the first scenario *A*, malicious actors always cheat and provide misleading feedback about to their counterparts (e.g., 0 to reliable partners and, vice versa, 1 to unreliable partners), while *ii*) in the scenario *B*

Scenario	Malicious	Malicious Behaviour
A	10%	Cheating and Misleading
B	10%	Alternate cheating
$T = 60\ min$	$R^0 = 0.5$	$C_{max} = [0.10; 5.00]$ \$
	$\alpha = 0.5$	$\beta = 0.5$
Cluster - Reputation range		
1 - $R \in [0.00; 0.45[$		
2 - $R \in [0.45; 0.55[$		
3 - $R \in [0.55; 0.85[$		
4 - $R \in [0.85; 1.00]$		

Table 1: Scenarios simulated with system and clustering parameters

malicious build positive reputations on low cost resources for cheating on high cost resources (this alternate behaviour has a frequency of a high cost resource each four low cost resources for each malicious consumer).

Moreover, an initial reputation score $R^0 = 0.5$ is assigned at all the SOs (see Section 3), and such reputation value was adopted also in the simulations to discriminate honest from cheaters. The time period ruling Eq. 4 was set to 60 minutes, while parameters α and β were both set to 0.5 and, finally the cost of a resource ranged in the domain $[0.01; 1.00]$ \$.

The first analysis we carried out on simulation results was focused on malicious SOs for measuring how many of them are recognised and the average value of their reputation. Figures 2 and 3 show such results for the two scenarios A and B, respectively.

The results for scenario *A* show that *i*) about 10 epochs, i.e. a little number of interactions, are enough for recognising more than the 90% of malicious actors based on their reputation score (i.e., < 0.5) and *ii*) the average reputation of malicious, by starting from the initial value of $R^0 = 0.5$, quickly decreases, as the number of simulated epochs increases, and characterise malicious SOs. The results of the scenario B, shown in Figure 3, are qualitatively quite similar

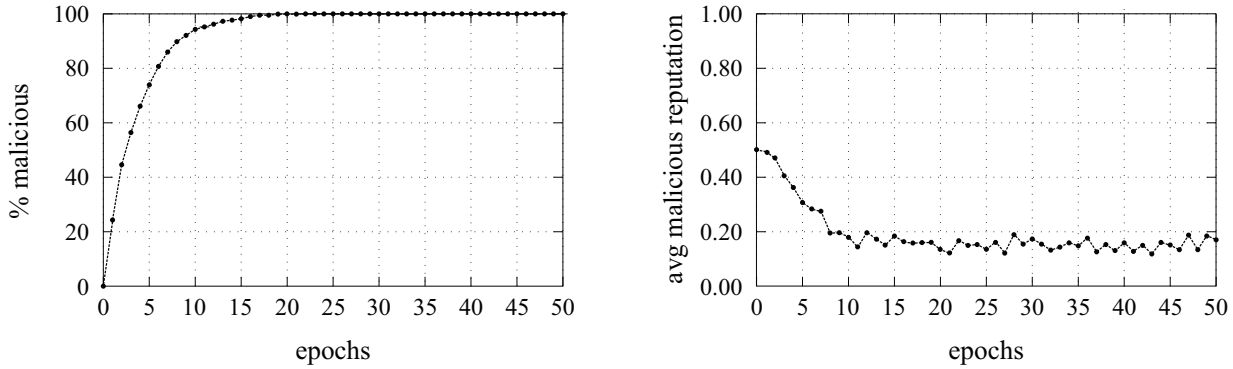

Figure 2: Percentage of malicious identified and the average reputation score of malicious for 50 epochs of simulations. Scenario A

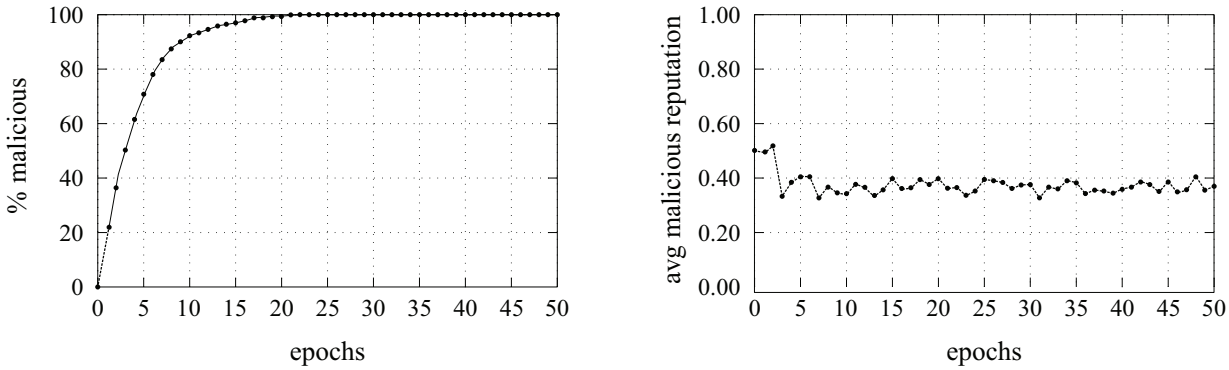

Figure 3: Percentage of malicious identified and the average reputation score of malicious for 50 epochs of simulations. Scenario B

to those obtained for the previous scenario. In particular, the percentage of recognised malicious increases a bit less rapidly, while the average reputation of malicious is always less then R^0 but higher than for the simulation *A* due to the alternate behaviours. It means that the reputation system is resilient also in presence of both dynamic and alternate behaviours.

In order to evaluate the economic advantage induced by the presence of clusters, we present in Figure 4 only the results referred to the scenario *A* given that the differences with scenario *B* are not significant. The percentage of advantage/disadvantage of a provider/client when it has to interact with client/provider, on the basis of their respective belonging cluster, are shown in Table 2. The results obtained by this experiment are shown in Figure 4 and from which we can highlight two aspects. The first aspect is that economic advantages deriving by a greater trustworthiness are evident for both clients and provides and they increases as epochs increase with respect to unreliable SOs. The other aspect can be observed is that the competitive scenario

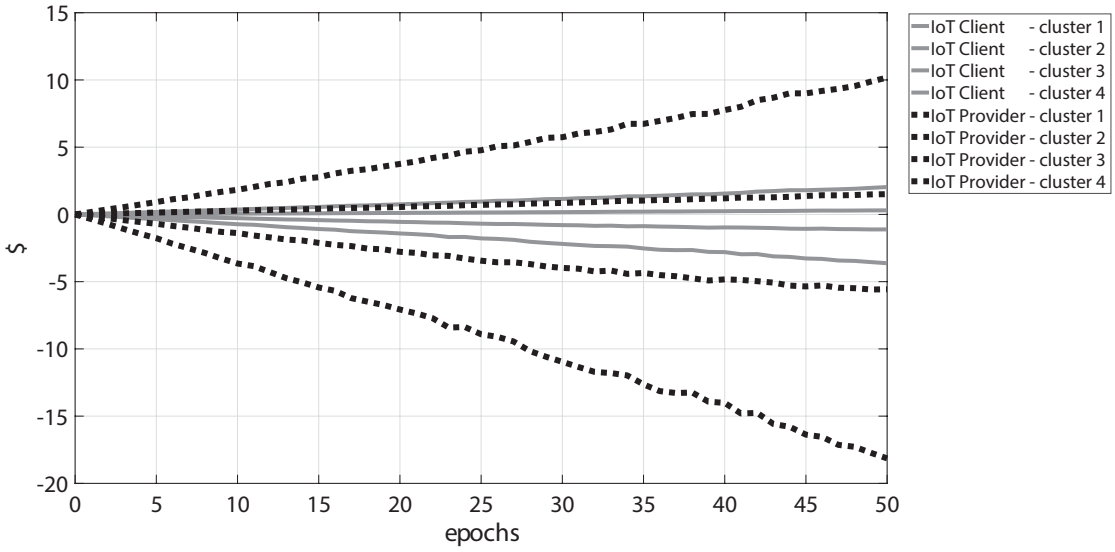

Figure 4: Clustering economic advantage. Scenario A

we designed encourage honest actors to be increasingly honest and marginalizes cheaters. In fact, notwithstanding the rough profiles we adopted, we noted a trend in honest actors to move from clusters 1 and 2 towards clusters denoted by a greater trustworthiness, i.e. 2 and 3 respectively,. Differently, malicious actors do not highlight such a behaviour given their malicious nature. Both the above aspects are positive and desirable.

From these preliminary results we can state that the our framework is effective in quickly identifying all the "malicious" agents (without "false positive") and support the sharing activities (for pay) of TOF.

provider / client	1	2	3	4
1	0%	-5%	-15%	$+25\%$
2	-5%	0%	$+5\%$	$+15\%$
3	-15%	-5%	0%	$+5\%$
4	-25%	-15%	-5%	0%

Table 2: Percentage of advantage/disadvantage of a provider/client vs a client/provider on the basis of their respective belonging cluster

6. Conclusions

In this work we have introduced a reputation-based approach aimed at significantly mitigate the risks, for smart objects, of interacting with unreliable partners. The approach is able to promote correct behaviours in an IoT Edge-based platform. In our approach, whenever two SOs have an interaction for a service, a certain edge server will collect a feedback sent by each SO about its counterpart; from these feedbacks, the edge server will be able to calculate and update the reputation scores of the two SOs. Our approach allows the SO to not be burden, while the cloud is still used only as repository of the SOs' reputation scores. Therefore, in the case a SO moves from its current own domain to another one, our approach operates such that its reputation value will not be lost.

In order to implement the above proposal, we have designed a distributed *Trusted Object Framework* (TOF) where heterogeneous OSs host and exploit the assistance of associated tamper-proof software agents, named trust agents. Moreover, in our framework, each edge server is provided with an agent capable to cluster the SOs belonging to its domain on the basis of their trustworthiness. The affiliation of a SO to one or another cluster allows it to obtain some benefits, since a SO on the basis of the cluster to which it belongs can, for instance, sell resources at a higher cost when acting as a provider and, vice versa, can purchase them at a cheaper price when acting as a consumer.

We have highlighted this advantage by means of some experiments, aimed to verifying the effectiveness of our approach to recognise honest by malicious SOs, to cluster IoT devices based on their trustworthiness and to verify the economic convenience to adopt such a cluster mechanism for the SOs. The results of the experimental campaign show the advantages deriving by a

greater trustworthiness and how they increase in time. Moreover, the experiments also evidence how our approach introduces a competitive scenario in the IoT platform that encourages honest actors and marginalizes malicious agents.

As for the future, in our ongoing research we are studying the possibility to introduce a further level of reputation in the cloud. In particular we are developing an extended model on which the data collected from different edge servers is collected and integrated in the cloud. We are also developing a testbed which is necessary to evaluate the pro and cons of this extended approach.

Acknowledgment

This work has been supported by the project "Pia.ce.ri linea 2 2020-2022", granted by the University of Catania.

References

[1] C. Savaglio, G. Fortino. A Simulation-driven Methodology for IoT Data Mining Based on Edge Computing. ACM Transactions on Internet Technology (TOIT) 21 (2) (2021): 1–22.

[2] K. Ashton, That' internet of things' thing. rfid journal, 22 june 2009 (2009).

[3] G. Kortuem, F. Kawsar, V. Sundramoorthy, D. Fitton, Smart objects as building blocks for the internet of things, IEEE Internet Computing 14 (1) (2010) 44–51.

[4] G. Fortino, W. Russo, C. Savaglio, W. Shen, M. Zhou, Agent-oriented cooperative smart objects: From iot system design to implementation, IEEE Trans. on Systems, Man, and Cybernetics Sys.

[5] M. Bakri, S. Alsharif, H. Alhumyani, E. Ali, R. Mokhtar, R. Saeed, An enhanced cooperative communication scheme for physical uplink shared channel in nb-iot, Wireless Personal Comm. (2021) 1–20.

[6] G. Rehman, A. Ghani, M. Zubair, S. Ghayyure, S. Muhammad, Honesty based democratic scheme to improve community cooperation for internet of things based vehicular delay tolerant networks, Trans. on Emerging Telecomm. Technologies 32 (1) (2021) e4191.

[7] G. Fortino, F. Messina, D. Rosaci, G. M. L. Sarné,, Using trust and local reputation for group formation in the cloud of things, Future Generation Computer Systems 89 (2018) 804–815.

[8] W. Shi, J. Cao, Q. Zhang, Y. Li, L. Xu, Edge computing: Vision and challenges, IEEE IoT Journal 3 (5) (2016) 637–646.

[9] W. Shi, S. Dustdar, The promise of edge computing, Computer 49 (5) (2016) 78–81.

[10] S.-H. Pan, S.-C. Wang, Optimal consensus with dual abnormality mode of cellular iot based on edge computing, Sensors 21 (2) (2021) 671.

[11] P. De Meo, F. Messina, M. N. Postorino, D. Rosaci, G. M. L. Sarné, A reputation framework to share resources into iot-based environments, in: IEEE 14th Int. Conf. on Networking, Sensing and Control, IEEE, 2017, pp. 513–518.

[12] H.-V. Valdes-Benavides, Ruy Alberto, P. Lourdes, Virtual currencies, micropayments and monetary policy: Where are we coming from and where does the industry stand?, in: Handbook on 3D3C Platforms, Springer, 2016, pp. 123–159.

[13] R. Roman, J. Zhou, J. Lopez, On the features and challenges of security and privacy in distributed internet of things, Computer Networks 57 (10) (2013) 2266–2279.

[14] J. Fogel, E. Nehmad, Internet social network communities: Risk taking, trust, and privacy concerns, Computers in Human Behavior 25 (1) (2009) 153–160.

[15] G. Han, J. Jiang, L. Shu, J. Niu, H.-C. Chao, Management and applications of trust in wireless sensor networks: A survey, J. of Computer and System Sciences 80 (3) (2014) 602–617.

[16] M. Levi, A state of trust, Trust and governance 1 (1998) 77–101.

[17] P. Dumouchel, Trust as an action, European J. of Sociology/Archives Européennes de Sociologie 46 (3) (2005) 417–428.

[18] D. Uckelmann, M. Harrison, F. Michahelles, An architectural approach towards the future internet of things, in: Architecting the internet of things, Springer, 2011, pp. 1–24.

[19] G. Fortino, R. Gravina, W. Russo, C. Savaglio, Modeling and simulating internet-of-things systems: A hybrid agent-oriented approach, Computing in Science & Engineering 19 (5) (2017) 68–76.

[20] W. A. Jansen, Countermeasures for mobile agent security, Computer Comm. 23 (17) (2000) 1667–1676.

[21] Z. Yan, P. Zhang, A. V. Vasilakos, A survey on trust management for internet of things, J. of network and computer applications 42 (2014) 120–134.

[22] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin, X. Yang, A survey on the edge computing for the internet of things, IEEE access 6 (2017) 6900–6919.

[23] G. Fortino, L. Fotia, F. Messina, D. Rosaci, G. M. L. Sarné, Trust and reputation in the internet of things: State-of-the-art and research challenges, IEEE Access 8 (2020) 60117–60125.

[24] G. Fortino, L. Fotia, F. Messina, D. Rosaci, G. M. L. Sarné, Forming groups in the cloud of things using trust measures, in: Int. Symp. on Intelligent and Distributed Computing, Springer, 2018, pp. 298–308.

[25] E. Andrade, B. Nogueira, I. de Farias Júnior, D. Araújo, Performance and availability trade-offs in fog–cloud iot environments, J. of Network and Systems Management 29 (1) (2021) 1–27.

[26] X. Wang, X. Wang, Y. Li, Ndn-based iot with edge computing, Future Generation Computer Systems 115 (2021) 397–405.

[27] Y. Zhang, J. Pan, L. Qi, Q. He, Privacy-preserving quality prediction for edge-based iot services, Future Generation Computer Systems 114 (2021) 336–348.

[28] M. Cui, Y. Fei, Y. Liu, A survey on secure deployment of mobile services in edge computing, Security and Comm. Networks 2021.

[29] C. Long, Y. Cao, T. Jiang, Q. Zhang, Edge computing framework for cooperative video processing in multimedia iot systems, IEEE Trans. on Multimedia 20 (5) (2017) 1126–1139.

[30] P. Varshney, Y. Simmhan, Demystifying fog computing: Characterizing architectures, applications and abstractions, in: IEEE 1st Int. Conf. on Fog and Edge Computing, IEEE, 2017, pp. 115–124.

[31] I. Psaras, Decentralised edge-computing and iot through distributed trust, in: Proc. of the 16th Annual Int. Conf. on Mobile Systems, Applications, and Services, 2018, pp. 505–507.

[32] M. Mukherjee, R. Matam, L. Shu, L. Maglaras, M. A. Ferrag, N. Choudhury, V. Kumar, Security and privacy in fog computing: Challenges, IEEE Access 5 (2017) 19293–19304.

[33] N.-N. Dao, N.-T. Dinh, Q.-V. Pham, T. V. Phan, S. Cho, T. Braun, Vulnerabilities in fog/edge computing from architectural perspectives, in: Fog/Edge Computing For Security, Privacy, and Applications, Springer, 2021, pp. 193–212.

[34] M. Chiang, T. Zhang, Fog and iot: An overview of research opportunities, IEEE Internet of Things J. 3 (6) (2016) 854–864.

[35] X. Sun, N. Ansari, Edgeiot: Mobile edge computing for the internet of things, IEEE Communications Magazine 54 (12) (2016) 22–29.

[36] M. Jutila, An adaptive edge router enabling internet of things, IEEE Internet of Things J. 3 (6) (2016) 1061–1069.

[37] Q. Fan, N. Ansari, Workload allocation in hierarchical cloudlet networks, IEEE Communications Letters 22 (4) (2018) 820–823.

[38] A. Yousefpour, G. Ishigaki, J. P. Jue, Fog computing: Towards minimizing delay in the internet of things, in: IEEE int. Conf. on edge computing (EDGE), IEEE, 2017, pp. 17–24.

[39] H. Zhang, Y. Xiao, S. Bu, D. Niyato, F. Yu, Z. Han, Computing resource allocation in three-tier iot fog networks: A joint optimization approach combining stackelberg game and matching, IEEE Internet of Things J. 4 (5) (2017) 1204–1215.

[40] L. Yang, J. Cao, G. Liang, X. Han, Cost aware service placement and load dispatching in mobile cloud systems, IEEE Trans. on Computers 65 (5) (2015) 1440–1452.

[41] G. Fortino, F. Messina, D. Rosaci, G. M. L. Sarné, Using blockchain in a reputation-based model for grouping agents in the internet of things, IEEE Transactions on Engineering Management 67 (4) (2019) 1231–1243.

[42] G. Fortino, F. Messina, D. Rosaci, G. M. L. Sarné, Resiot: An iot social framework resilient to malicious activities, IEEE/CAA Journal of Automatica Sinica 7 (5) (2020) 1263–1278.

[43] X. Fu, Y. Yang, Modeling and analyzing cascading failures for internet of things, Information Sciences 545 (2021) 753–770.

[44] F. Buccafurri, L. Fotia, G. Lax, Allowing non-identifying information disclosure in citizen opinion evaluation, in: Int. Conf. on Electronic Government and the Information Systems Perspective, Springer, 2013, pp. 241–254.

[45] F. Buccafurri, L. Fotia, G. Lax, Privacy-preserving resource evaluation in social networks, in: 10th Annual Int. Conf. on Privacy, Security and Trust, IEEE, 2012, pp. 51–58.

[46] D. Rosaci., G. M. L. Sarné,, S. Garruzzo., Integrating trust measures in multi-agent systems, International Journal of Intelligent Systems 27 (1) (2012) 1–15.

[47] P. De Meo, L. Fotia, F. Messina, D. Rosaci, G. M. L. Sarné, Providing recommendations in social networks by integrating local and global reputation, Information Systems 78 (2018) 58–67.

[48] M. Bahutair, A. Bouguettaya, A. G. Neiat, Multi-perspective trust management framework for crowdsourced iot services, arXiv preprint arXiv:2101.04244.

[49] M. Nambobi, K. Ruth, A. Alli, R. Ssemwogerere, The age of autonomous internet of things devices: Opportunities and challenges of iot, Challenges and Opportunities for the Convergence of IoT, Big Data, and Cloud Computing (2021) 1–16.

[50] B. Rohini, A. R. Nayak, N. Mohankumar, Enhancing security and trust of iot devices–internet of secured things (iost), in: Research in Intelligent and Computing in Engineering, Springer, 2021, pp. 15–24.

[51] V. L. Hallappanavar, M. N. Birje, A reliable trust computing mechanism in fog computing, Int. J. of Cloud Applications and Computing 11 (1) (2021) 1–20.

[52] F. Bao, 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.

[53] R. Chen, F. Bao, J. Guo, Trust-based service management for social internet of things systems, IEEE transactions on dependable and secure computing 13 (6) (2015) 684–696.

[54] Y. B. Saied, A. Olivereau, D. Zeghlache, M. Laurent, Trust management system design for the internet of things: A context-aware and multi-service approach, Computers & Security 39 (2013) 351–365.

[55] C. Vallati, E. Mingozzi, G. Tanganelli, A. Mamelli, D. Sommacampagna, B. Anggorojati, et al., Betaas: A platform for development and execution of machine-to-machine applications in the internet of things, Wireless Personal Comm. 87 (3) (2016) 1071–1091.

[56] E. M. Schooler, D. Zage, J. Sedayao, H. Moustafa, A. Brown, M. Ambrosin, An architectural vision for a data-centric iot: Rethinking things, trust and clouds, in: IEEE 37th Int Conf on Distributed Computing Systems, IEEE, 2017, pp. 1717–1728.

[57] M. D. Alshehri, F. K. Hussain, O. K. Hussain, Clustering-driven intelligent trust management methodology for the internet of things (citm-iot), Mobile networks and applications 23 (3) (2018) 419–431.

[58] G. Fortino, L. Fotia, F. Messina, D. Rosaci, G. M. L. Sarné, Using local trust measures to form agent cot groups 1, Intelligenza Artificiale 14 (1) (2020) 7–18.

[59] M. Chen, Y. Hao, C.-F. Lai, D. Wu, Y. Li, K. Hwang, Opportunistic task scheduling over co-located clouds in mobile environment, IEEE Trans. on Services Computing 11 (3) (2016) 549–561.

[60] G. Zacharia, P. Maes, Trust management through reputation mechanisms, Applied Artificial Intell. 14 (9) (2000) 881–907.

[61] S. Ramchurn, D. Huynh, N. Jennings, Trust in multi-agent systems, Knowledge Engeenering Review 19 (1) (2004) 1–25.

[62] R. Falcone, C. Castelfranchi, Social Trust: a Cognitive Approach., Kluwer, 2001.

[63] F. Perich, J. Undercoffer, L. Kagal, A. Joshi, T. Finin, Y. Yesha, In Reputation We Believe: Query Processing in Mobile Ad-hoc Networks., in: Proc. of the 1st Annual Int. Conf. on Mobile and Ubiquitous Systems: Networking and Services, 2004, 2004, pp. 326–334.

[64] Y.-H. Tan, W. Thoen, An Outline of a Trust Model for Electronic Commerce., Applied Artificial Intelligence 14 (8) (2000) 849–862.

[65] A. Abdul-Rahman, S. Hailes, Supporting trust in virtual communities, in: Proc. of the 33rd Hawaii Int. Conf. on System Sciences, Vol. 6, IEEE Computer Society., 2000, p. 9.

[66] G. Lax, G. M. L. Sarné, CellTrust: a reputation model for C2C commerce, Electronic Commerce Research 8 (4) (2006) 193–216.

[67] Z. Jinlai, P. Balaji, L. Heiko, W. Qingyang, Zenith: Utility-aware resource allocation for edge computing, in: Proc. of the 2017 IEEE international conference on edge computing (EDGE), IEEE Computer Society., 2017, 47–54.

[68] L. Fuhong, Z. Yutong, A. Xingsuo, Y. Ilsun, C. Kim-Kwang Raymond, Fair resource allocation in an intrusion-detection system for edge computing: Ensuring the security of Internet of Things devices, IEEE Consumer Electronics Magazine, 7 (6) (2018) 45–50.

[69] C. Xu, S. Qian, Y. Lei, X. Jie, ThriftyEdge: Resource-efficient edge computing for intelligent IoT applications, IEEE network, 32(1), 2018, 61–65.

[70] A. Farag, Taxonomy of reputation assessment in peer-to-peer systems and analysis of their data retrieval, The Knowledge Engineering Review, (29) 4, 2014, 463–483.

[71] L. McKnight, D. Harrison, N. Chervany, The meanings of trust, (1996) Citeseer.