A Social Edge-Based IoT Framework Using Reputation-Based Clustering for Enhancing Competitiveness

A Social Edge-based IoT Framework Using Reputation-based Clustering for Enhancing Competitiveness

Giancarlo Fortino, *Fellow Member, IEEE*, Lidia Fotia, Fabrizio Messina, Domenico Rosaci, Giuseppe M. L. Sarnè, *Senior Member, IEEE*

This is a post-print version of the paper published in IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, VOL. 10, NO. 4, AUGUST 2023, DOI: 10.1109/TCSS.2022.3208376

***Abstract*—The introduction of the IoT technology and its pervasive penetration in our daily life implies that IoT smart objects can often participate in social events, and in this case their paradigm of interaction is commonly denoted as Social IoT (SIoT). In order to make reliable transactions into a SIoT scenario, in this paper we introduce a multi-agent SIoT architecture, which integrates a reputation system based on a clustering of the SOs. In our framework, when a SO looks for a resource and detects a reliable partner having that resource, then the two SOs can interact to make a transaction, and at the end of the transaction each of them provides a feedback about the partner to the local reputation system of the edge server, in order to update the SOs' reputation scores. Moreover, each edge server sends these reputation scores to the cloud, which updates the associated reputation values of the SOs, deriving from experiences coming from all the edge domains. This architecture provides a given object, moving from an edge domain to another one, with the possibility to have an updated value of its reputation, represented by the value stored in the cloud. We have validated our approach by a campaign of simulations, whose results seem particularly promising.**

***Index Terms*—Clustering, Competitiveness, Edge Computing, Internet of Things, Reputation System.**

I. Introduction

In current Internet of Things (IoT) applications, a large number of smart objects are mutually linked and exchange information to provide several, often sophisticated, e-services [1], [2]. These objects are commonly exploited to detect physical events (e.g. motion, temperature, etc.) from the environment that they live in, and the values of the measures are then transmitted to a closer access points to be analyzed and to support intelligent decisions [3]. Moreover, with the highly pervasive penetration of IoT, these smart objects can also be involved in human-to-machine or machine-to-machine interlocutors [4] (i.e., in social events). In these latter cases they are generally denoted as Social Objects. The paradigm of interaction involving social objects is commonly known as Social IoT (SIoT) [5], [6]. In words, an SIoT can be considered as a particular IoT environment in which the smart objects are autonomously enabled to create relationships with other objects, thus realizing an ecosystem which allows objects to live within a social structure based on relationships. Based on the structure, the support of numerous applications, and networking services, the SIoT is preferable over the traditional IoT [7]. However, some aspects like, for instance, the trustworthiness of users and network navigability [8]–[10], are important challenges that generate users' fears of data disclosure and privacy violations [11], [12]. Thus, it is crucial in such a context to provide reliable data analyses by using trust-based properties

Furthermore, the particular case of the SIoT scenario generates a massive traffic of data, due to the frequent interactions between the SOs, and we have to consider that, in order to guarantee effectiveness and efficiency of the applications [13], [14], a crucial issue is to provide adequate computational and storage resources to the SOSs and intelligently manage communication between them. In the past, a large number of cloud-based proposals have been presented to provide the necessary resources [15] in general IoT contexts; however, the most of these proposals introduce important limitations in terms of computing performances, making unfeasible the development of applications in the particular case of SIoT scenario. In fact, cloud based proposals often generate computational overhead, due to the necessity to upload processes in the cloud, and this affects the power consumption of the SOs.

A different option with respect to cloud solutions is represented by Edge Computing [16], [17], which moves both SOs computational and data communication costs from the SOs to some edge servers, that are in the neighborhood of the SOs and are provided with suitable storage and computational resources. Introducing Edge Computing architectures, SOs can fruitfully ~~benefit of a~~ decrement ~~of the~~ transmission latency and ~~a reduction of the~~ power consumption.

Our paper is positioned in the above context, having the purpose of improving mutual cooperation of SOs in a SIoT environment [18]. In particular, we highlight that, when a SO is looking for resources in its SIoT community, it should choose the most suitable partners which to interact with. This necessity arises from the fact that in a SIoT situation, we can have a competition between SOs in exchanging resources, and therefore an SO can be exposed at the risk of potential malicious activities performed by fraudulent agents trying

G. Fortino is with the Department DIMES, Univ. of Calabria, Via P. Bucci, 87036 Rende (CS), Italy, (e-mail: giancarlo.fortino@unical.it).

L. Fotia is with the Department DIEM, Univ. of Salerno, Via Giovanni Paolo II, 132 - 84084 Fisciano (SA), Italy, (e-mail: lfotia@unisa.it).

F. Messina is with the Department of Mathematics and Computer Science, University of Catania, Italy (e-mail: messina@dmi.unict.it).

D. Rosaci is with the Department DIIES , University of Reggio Calabria Via Graziella, Loc. Feo di Vito, 89122, Reggio Calabria, Italy (e-mail: domenico.rosaci@unirc.it).

G.M.L. Sarné is with the Department of Psychology, University of Milano Bicocca, P.za dell'Ateneo Nuovo 1, 20126 Milano (MI), Italy, (e-mail: giuseppe.sarne@unimib.it).

to gain undue benefits [19]. Such an issue becomes mostly significant in presence of mobile SOs, moving from an edge environment to another one, since in this case the possibility of interacting with an unknown level of trustworthiness is very high. This small degree of confidence in the trustworthiness of the SOs significantly reduces the possibilities to interact with reliable partners for concluding satisfactory transactions [20]–[22]. In this situation, in order to mitigate these risks and to realize an environment where it is possible to be informed about the reliability of each SO, the choice of adopting a reputation system appears unavoidable. We highlight that a reputation system generally provides a global measure of the trustworthiness that the community has in a given trustee, computed exploiting the feedback about the trustee's activities provided by the SOs which interacted with it. This global measure, called *reputation* of the trustee, can be viewed as the expectations that a trustor has to satisfactorily interact with that trustee [23]–[25].

A. Our research proposal

In order to address the research issues above discussed, in this paper we present an agent-based SIoT framework, which integrates a reputation system based on a clustering of the SOs. In our framework, each SO living in a given edged domain (i.e., a community of SOs hosted in a given edge server) is associated with a software agent able of having social-based interactions with other agents associated with the other objects of the domain [26], [27]. These "social activities" of the SOs are performed by the support of an apposite reputation system local to the domain associated with the edge server. In our framework, when a SO looks for a resource and detects a reliable partner having that resource, then the two SOs can interact to make a transaction, and at the end of the transaction each of them provides a feedback about the partner to the local reputation system of the edge server, in order to update the SOs' reputation scores [28]–[30]. Moreover, each edge server periodically sends these reputation scores to the cloud server, which updates the associated reputation values of the SOs, deriving from experiences coming from all the edge domains. This architecture provides a given object, moving among edge domains, with the possibility to have an updated value of its reputation, represented by that stored on the central server.

Furthermore, in our framework, each edge server performs a clustering of the SOs into different groups, based on the SO reputation scores. This clustering activity aims at increasing the SOs' competitiveness in the edge domain (see Section V), motivating each SO to obtain the highest reputation scores which will provide it with the possibility of joining with the best clusters (i.e., those clusters) containing SOs that have the best probability to provide satisfactory transactions.

B. Advantages and limitations of the proposal

We have validated our approach by a campaign of simulations, whose results seem particularly promising. In particular, we have verified the capability of the framework to identify honest and malicious SOs and the quality of the clusters, and the performed simulations show that the framework is resilient also in presence of concomitant malicious behaviors. A campaign of experiments that we have performed on simulated data shows the advantages introduced by our solution, which is capable of detecting malicious SOs, by clustering them based on of their reputation. In particular, an original feature of our strategy is that it quickly identifies malicious actors, whose number continuously decreases in time, although a limitation of our approach is that the presence of alternate behaviors implies that the percentage of recognized malicious actors decreases somewhat less rapidly. Moreover, another original contribution of our proposal with respect of the state of the art is that we have observed that the honest actors moved towards the best groups, while the malicious moved towards the worst ~~ones~~ groups. It is important to highlight that a price to pay for obtaining the advantages shown above is represented by the possible communication overhead due to the necessity of continuously updating the cloud repository from the edge entities.

The paper has the following organization. In Section II we introduces our IoT agent framework, while in Section III the Cluster Reputation Model is presented. An overview of some related work is in Section IV and in Section V we describe some experimental results. Finally, in Section VI we discuss our final conclusions.

II. The Social IoT Edge-based Framework

The architecture of the proposed social edge-based framework (denoted by **W**) is depicted in Figure 1 (a list of symbols is provided in the Appendix). It represents the reference SIoT world we consider and includes:

i) a population of heterogeneous SOs which can move across the different domains forming the framework. Each SO is associated with a ~~*Trust agent*~~ t (see below) in order to take advantage of their intrinsic social attitude;
ii) a set of edge-based IoT domains, where each edge domain is managed by an associated agent to support SOs affiliation;
iii) a cloud (**C**) is associated with an agent (denoted by **c**) supporting the set of edge domains active in the edge-based framework.

In the following, we assume that each SO will be affiliated with one of the edge domains active in our edge-based framework. Moreover, in each edge domain will be active more clusters (groups) and all the SOs affiliated with that edge domain will be also associated with one of its groups based on their own reputation score (see Section III)

The activities that SOs will perform in each edge domain consist in exchanging (i.e. offering/purchasing) resources with other SOs (belonging to the same edge domain) based on their group membership, so that the better the group, the greater the probability of having satisfactory interactions.

For this purpose, different and coordinates activities need to be carried out by the framework. More precisely:

- SO activities Each SO performs the following tasks:

1) *Activation.* This task is carried out the first time that a SO is activated in our edge-based framework. To this aim, let SO be an IoT device (i.e. its associated trust agent **t**)

and let **E** and **e** be an edge domain active in **W** and its associated agent, respectively. To begin this process, an SO affiliation request is required to be sent to **e**, after which the SO's agent **t** will become active and the domain agent **e** will provide it with:

i) a reputation score, set to 0.5 for default, it will be exploited also to initially affiliate the SO with a group active on that edge domain (see Section III);
ii) a personal pair of asymmetric cryptographic keys;
iii) the public key of **e**;
iv) a light certificate, signed by **e**, witnessing the SO's identity, group membership and a timestamp on the certificate expiration date[1].

Then **e** will inform **c** about the SO's affiliation with its edge domain and will send the SO's certificate to it.

2) *Resource Search.* The SOs interested in a resource r perform this task by means of simple message exchange. More in detail, the trust agent **t**, hosted from its associated SO, that is interested in r (i.e. consumer) sends to the other agents, joined with the same its edge domain, a message consisting of:

i) the consumer group membership;
ii) a descriptor of the searched resource.

All the SOs able to satisfy this request (i.e. providers) can replay with a signed message consisting of:

i) the provider group membership;
ii) the descriptor of the offered resource, including its price.

3) *Resource Provisioning.* When for a consumer's request one or more providers' messages have been received by the consumer, then this task is performed. More in detail, let SO_i and SO_j be the consumer and the provider SOs and let $\mathbf{t}_i$ and $\mathbf{t}_j$ be their respective trust agents. When the SO_j resource offer is chosen by SO_i, then $\mathbf{t}_j$ and $\mathbf{t}_i$ will act as follows:

i) each of the two trust agents has to sign (with their private keys) the own certificate, mutually provide to exchange it and then check that of the own counterpart;
ii) carry out a possible negotiation stage;
iii) perform the resource delivery and payment steps.

Note that negotiation, delivery and payment issues are assumed to be orthogonal tasks in the light of the main focus of our proposal.

4) *Trust Management.* After that the previous task is ended, each SO sends a message addressed to **e** and consisting of *i*) a feedback, released to represent its satisfaction degree about the exchanged resource and the counterpart behavior and *ii*) the counterpart's signed certificate. This message will be signed with the sender agent's private key.
When ***e*** receives the agent's message, then it:

i) updates the reputation scores of the involved SOs (see Section III);
ii) issues the new SOs' certificates with also the updated group membership (see Section III-B);
iii) sends the new certificates to the two involved SOs and to the ledger managed by its cloud agent **c**, in order to maintain updated also these data on the cloud.

5) *Domain Change.* To change the edge domain, an SO (i.e. its trust agent) sends its current certificate (signed with its private key) to the server of the new edge domain that will issue a new signed certificate for both the SO and **c**. After this, the SO (i.e. its agent) can leave its current edge domain and migrates to the new edge domain.

[1]For sake of simplicity, we assumed that each certificate has a daily validity and after it is elapsed, a new certificate should be issued by **e**.

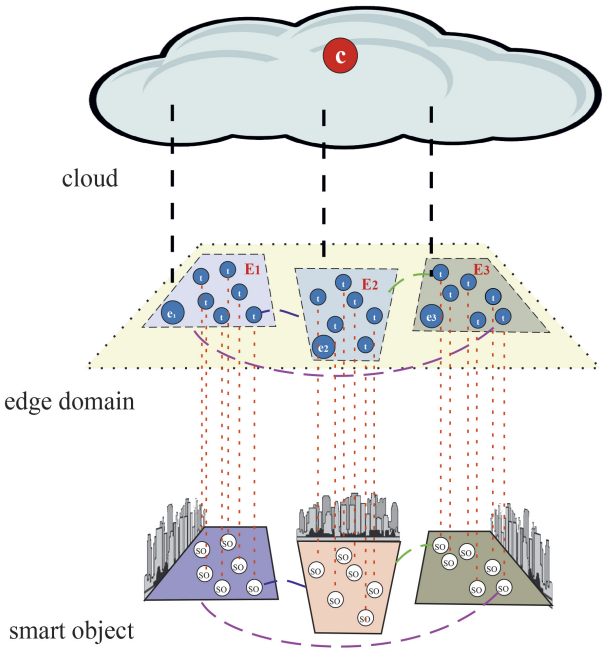

Fig. 1. The architecture of the SIoT framework **W**.

- Edge activities. The edge server (i.e. its edge agent) managing an active IoT edge domain in **W**, besides the *Activation*, *Trust Management*, and *Domain Change* activities previously described, locally manages and updates a ledger of all the SOs currently affiliated with its edge domain, also by exploiting information stored on the cloud, and reissues expired certificates.

- Cloud activities. The cloud (i.e. its cloud agent) associated with our edge-based framework, supports SOs and edge servers in *Domain Change* activities by managing a ledger of all SOs updated certificates with their current reputation scores, group affiliation and edge domain affiliation.

III. The Cluster-Reputation Model

In this Section, the Cluster-Reputation Model (CRM), designed for the SIoT scenario presented above, is described

in detail with respect to its reputation and SO clustering components.

A. The Reputation Model

In particular, in our SIoT scenario a reputation score is associated with each SO. It represents synthetically the history of the SO behavior with respect to the honesty and quality of its interactions in W, as witnessed by the counterparts by means of the feedback they released to reflect their satisfaction degree about the performed interactions.

In order to present the proposed reputation model, let SO_i and SO_j be two smart objects, respectively associated with the trust agents $\mathbf{t}_i$ and $\mathbf{t}_j$, which live on an edge domain $\mathbf{E} \in \mathbf{W}$.

As described in the previous section, once SO_i and SO_j have interacted for a resource r, then each SO releases a feedback ranging in $[0,1] \subset \mathbb{R}$ to the edge agent $\mathbf{e}$, where 0 (i.e. 1) means the minimum (i.e. maximum) appreciation about the role (of consumer or provider of r) played by the own counterpart. More in detail, for a given resource r, let $F_{i,j}$ be the feedback computed by SO_i about SO_j and let $F_{j,i}$ be the feedback computed by SO_j about SO_i. Then the trust agents $\mathbf{t}_i$ and $\mathbf{t}_j$ will send the respective $F_{i,j}$ and $F_{j,i}$ to the edge agent $\mathbf{e}$ in order to updating the reputation scores (i.e. $R_i,\ R_j \in [0,1] \subset \mathbb{R}$) of SO_i and SO_j.

The computation of the reputation score of SO_i depends on the computation of parameter ϑ (named *Interaction*), which takes into account the feedback $F_{j,i}$ given by SO_j (i.e. the trustor) about SO_i (i.e. the trustee), the economic relevance (δ) of the resource r, the frequency (ϕ) of the interactions occurred between two SOs and a parameter (ξ) related to the trustor reputation. More formally, for an interaction occurred between SO_i and SO_j for a given resource r, then $\vartheta^r_{i1,j}$ is computed as:

$$\vartheta_{i,j} = f\Big(\delta_{i,j}, \phi_{i,j}\Big) \cdot \xi_j \cdot F_{j,i} \tag{1}$$

where:

- The parameter $\delta \in [0,1] \subset \mathbb{R}$ considers the relevance of the resource r on the basis of its monetary price p with respect to a price threshold P_{max}, that is a system parameter. More in detail:
  i) when the resource has no cost, then the parameter δ is set to 0;
  ii) when the cost of the resource is greater than a given threshold, then the relevance of r will be assumed as maximum (i.e. 1);
  iii) otherwise, in all the remaining cases, δ will be equal to the ratio between p and P_{max}.

  In other words the lower the cost of the resource, the lower the impact of that resource on the reputation. This mechanism is used to avoid malicious behaviors aimed at gaining reputation for unimportant resources, to be spent for important resources having a high value.
  More formally, the parameter δ is computed as:

$$\delta = \begin{cases} 0 & \textit{if} \quad p = 0 \\ \dfrac{c}{C_{Max}} & \textit{if} \quad 0 < p \leq P_{max} \\ 1 & \textit{if} \quad p > P_{max} \end{cases} \tag{2}$$

- The parameter $\phi \in [0,1] \subset \mathbb{R}$ has been designed to limit collusive activities carried out by two or more SOs for increasing their reputations by mutually providing positive feedback about each other with high frequency. More in detail, the value of ϕ will be set to 1 in presence of negative feedback (i.e. $F < 0.5$); conversely, it will be computed as the inverse of the parameter μ (see below). More formally, the parameter ϕ for a given resource r is computed as:

$$\phi = \begin{cases} 1 & F < 0.5 \\ \mu^{-1} & F \geq 0.5 \end{cases} \tag{3}$$

  where the parameter $\mu \in [0,1] \subset \mathbb{R}$ (preliminary set to 1) is ruled by the closeness in time with which the involved SOs (i.e. agents) mutually provide a feedback about each other. In particular, given two consecutive feedback exchanged by the same pair of SOs and let Δt be the time elapsed between them, then the new value of the parameter μ is computed as:

$$\mu = \begin{cases} max\Big(1, \mu^{old}\Big) & \Delta t < T \\ max\Big(1, \mu^{old} - \dfrac{\Delta t}{T}\Big) & \Delta t \geq T \end{cases} \tag{4}$$

  In other words, when two SOs mutually interacted for the first time, the parameter μ is initially set to 1 and decreased or increased according to the time elapsed between two consecutive feedback with respect to the system time threshold T.
- The function f depends on the parameters δ and ϕ, and returns a value in the domain $[0,1] \subset \mathbb{R}$. The ratio of this function is to limit potential collusive and alternate activities between two SOs by penalizing all iterations characterized by a low or null relevance of the resource and a high value of the frequency parameter. Therefore, when the values of δ and ϕ are both minimum (i.e. 0) the risk of malicious activities is maximum, while when one or both the values of the parameters δ and ϕ are maximum (i.e. 1), or close to the maximum, the risk is minimum.
  To determine the function $f(\delta, \phi)$, some preliminary tests have been carried out. As a result, we found that the desired behavior for $f(\delta, \phi)$ can be obtained by the following family of non linear functions:

$$f(\delta, \phi) = \begin{cases} \sqrt{\delta^z + \phi^z} & \textit{if} \quad f(\delta, \phi) \leq 1 \\ 1 & \textit{otherwise} \end{cases} \tag{5}$$

where the parameter z defines in the detail the behavior of $f(\delta, \phi)$. Tests have also allowed to set suitably the value of z. More specifically, the best compromise in terms of performance was obtained for $z = 1.35$. However, note that even varying the value of z of about $\pm 10\%$, then the variations in performance are not very significant. In Figure 2 is depicted the function f when the parameters δ and ϕ vary and z is set to 1.35.

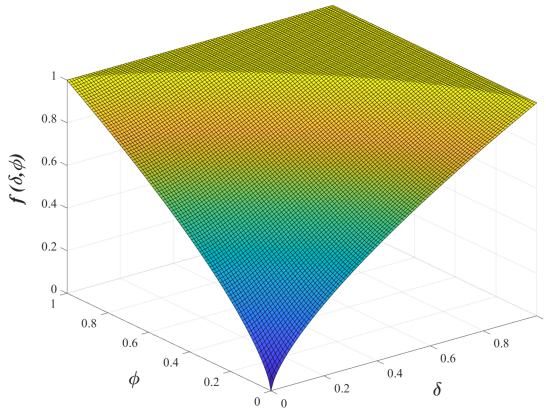

Fig. 2. The function $f(\delta, \phi) \in [0, 1] \subset \mathbb{R}$, depending on the parameters δ and ϕ, computed for $z = 1.35$.

- The value of the parameter ξ, named *trustor relevance*, is computed on the basis of the trustor reputation by means of a step function ruled by the system threshold $\sigma \in [0, 1] \subset \mathbb{R}$ (a parameter of the reputation model). More specifically, the parameter ξ (named *trustor relevance threshold*) is set to 1 if $R \geq \sigma$, otherwise it is set to 0. More formally, the parameter ξ is computed as:

$$\xi = \begin{cases} 1 & \textit{if} \quad R \geq \sigma \\ 0 & \textit{otherwise} \end{cases} \tag{6}$$

- The feedback F is released by the trustor about the trustee in the domain $[0, 1] \subset \mathbb{R}$ on the basis of its satisfaction degree with respect to both the SO's honesty and the quality of the occurred interaction which, in turn, also depends on the resource r.

Once that parameter $\vartheta_{i,j}$, for the resource r, has been computed the the reputation of SO_i (and similarly for SO_j) is updated. More in detail, the reputation of SO_i will not always updated but only when the condition $\vartheta > 0 \vee R^{new} \geq 0.5$ will be satisfied, otherwise it will not be modified. This is a condition to further marginalize untrustworthy actors and potential malicious activities. More formally, the new value of the reputation of SO_i (i.e. ${R_i}^{new}$) will be computed by the edge agent **e** as:

$${R_i}^{new} = \begin{cases} \alpha \cdot {R_i}^{new} + (1 - \alpha) \cdot \vartheta_{i,j} \\ \qquad\qquad \theta_{i,j} > 0 \vee {R_i}^{new} \geq 0.5 \\ {R_i}^{new} \qquad\qquad \textit{otherwise} \end{cases} \tag{7}$$

where $\alpha \in [0, 1] \subset \mathbb{R}$ is a parameter weighting the current reputation score ${R_i}^{new}$ with respect to the contribution $\vartheta^r_{i,j}$. In other words, the parameter α influences the reputation system behavior and the higher is the value of α, the lower is the reputation score sensitivity towards new interactions.

When the reputation updating process ends, then the edge agent **e** verifies if, on the basis of the updated reputation, the SO's group membership remains unchanged or it must be updated. Finally, a new certificate with the updated both SO's reputation score and group membership is issued, signed by **e** and sent to both the agents **t** and **c**.

B. Smart Object Clustering

The SO clustering is the second element of the CRM and leverages on the previously calculated reputation scores

More in detail, to further promote correct SO behaviors in **W**, in each affiliated edge domain the SOs are arranged in groups based on their reputation scores. In this way, a competitive environment capable to identify malicious SOs as quickly as possible (by assuming that the better the group an SO belongs to, the greater the probability of having satisfactory interactions with it) it is built. In addition, basing interactions on an SO's group membership, as an alternative to using its reputation score, allows us to moderate the effects of the small fluctuations to which reputation is normally subject. Therefore, this choice allows for less artifact and more stable system behavior over short time intervals.

To classify SOs into groups, in each edge domain the associated edge agent implements a K-Means Clustering algorithm [31] daily. Part of the motivation behind our choice was the low computational complexity and flexibility of this algorithm. Two essential aspects in the proposed SIoT scenario. Indeed, this clustering algorithm has the advantage of being quite fast, since it is required to only calculate the distances between the points and the centers of each group and, therefore, it has a linear complexity $O(n)$. Furthermore, each edge agent must choose arbitrarily the number of active groups in its domain, a number that could vary over time. This is not a disadvantage from our perspective, because it easily allows us to modify the edge domain *behavior* also to consider changes occurring in the SO population and optimize reputation-related processes. However, other clustering algorithms can be adopted as an alternative to the one chosen here.

We remark that each time the group membership of an SO changes, then the edge agent must issue and sign a new certificate for both the SO's and the cloud agents to witness the new group affiliation.

IV. Related Work

In this section, some contributions that, in our opinion, come closer to our proposal are presented. In particular, we will deal with trust, recommendation and group formation issues in SIoT scenarios.

A. Trust and recommendation in social IoT

The concept of trust in (social) IoT has been widely analyzed in the research literature.

Authors of [32] analyze the trust management problem in IoT as a crucial "tool" for relevant and trustworthy services, data fusion and mining, while the authors of [33] proposed a trust solution using blockchain for collaborative IoT contexts. While the former presents a complete analysis on trust management in IoT, by investigating the trust properties and propose goals of IoT trust management, the latter presents a three levels solution for trust management and IoT collaborations.

In a recent work [34] the problem of establishing trust in remote IoT devices is addressed in a remote way through attestation. A state-of-the-art on the attestation techniques from an IoT device viewpoint is provided by the authors, proving that each of them assume a specific role in IoT trust establishment.

An interesting, very recent work written by Marche at Al. [35] deals with possible attacks and malfunctions in the IoT with particular reference to the "social" aspect of IoT, by proposing a trust management model to deal with various type of trust attacks in IoT. A few simulations results showed how malicious actors could be marginalized in the network. On the other hand, the shortcomings is represented increasing the transactions number required from this model to converge.

The main relation between trust and Social IoT has been provided in [36], where the authors analyzed limitations of the existing models of trust for SIoT, and they proposed a comprehensive model of trust specifically designed for the social IoT, by analyzing fundamental roles as trustor and trustee They clarified the trust concepts in the SIoT in several different issues (i.e., trustor and trustee mutuality, inferential trust transfer, and trustworthiness in a dynamic context), and conducted several simulations to analyze the social IoT performance with the introduction of their trust model.

We also compared two different research works about trust for SIot, which reach the important goal of adaptability by addressing different concerns. The former, by Chen at Al. [37] proposes and analyzes the adaptive trust management concept for SIoT systems with particular emphasis on the dynamic nature of social relationship among IoT peers. They also propose an adaptive trust management protocol. The other [38] deals with the important requirement of sharing data for recommendation services in IoT contexts. They identified Social IoT (SIoT) as platform to share such data by building an object's profile exploiting IoT applications data. This leads to a better adaptation of IoT services to users' requirements and, as a consequence, an improved user experience.

The interested reader might also refer to [20], [21], [39]–[41] for further information on the matter.

B. Social IoT and group

Social IoT (SIoT) enables cooperation between devices to support IoT applications in different areas. In this way it is possible to connect humans, devices, and resources by adopting human like social relationships.

In [42], the authors analyze the problem of increasing network traffic and propose the Social Correlation Group (SCG). It selects those nodes similar for associated information by applying a form of social correlation; the SGC is created, aligned and updated. Also, the various SCGs are constructed to minimize the area involved in discovery step and reduce the costs for their creation and updating the nodes' social correlation.

Concerning the Multiaccess Edge Computing (MEC)-enabled SIoT, Yang et al. [43] introduce a network embedding-based solution for scalable network navigation via leveraging the social similarity of SIoT. They characterize the social relationship from the contents point of view since MEC enables the edge with cache storage. In particular, they leverage the hierarchical structure, along with the social relationship among nodes to extract the hierarchical social group from SIoT. In each social group, they designate a node that locates at the upper layer to be the group leader since it has higher computation and storage capacity than its group members. The social groups can be overlapped because the leader of a group can also be a member of another group. A node will choose to join the group of which the leader has the highest similarity with this node.

The SIoT involves a very large number of services available on the network, so the selection of reliable service providers is an important issue. In [44], the authors propose the Trustworthy Group-based Service Management (TGSM). They select those service providers that best fitting the actors' needs considering trustworthiness measures and performance appraisal. In the trustworthiness assessment a penalty-based approach is adopted to tackle selfish object behaviors by adopting the HITS algorithm which, in turn, is based on a global dynamic approach. The authors also suggest a new architecture for SIoT network using a service-based grouping approach.

In [45], the authors present a scheme for secure content sharing (SCS) to achieve a balance between security and quality of experience (QoE) taking into account social trust. To obtain the social trust, they utilize a random walk strategy that is based on the User-Content-Social Group graph modeling users' preferences. Also, they introduce a hierarchical model of gaming to split the problem of optimization into two main sub-problems, namely user coupling and channel selection. Specifically, the first one sub-problem consists of a matching game using a peer effect and the other one consists of a safe channel selection game using the oriented hypergraph as the game space. Finally, they conceive an uncoupled-user concurrent learning algorithm (UUCL) to achieve an optimal pure Nash equilibrium. In [46] a competitive SIoT framework of federated domains is described and where mobile IoT devices compete, on the basis of their reputation, for belonging to best clusters.

This overview has described some significant contributions of our interest in SIoT computing. However, none of them present the same features of our framework employing the edge- and cloud computing, agent technology, clustering algorithm and a reputation system with the aim to enhance the competitiveness among IoT devices and quickly marginalize

malicious IoT devices. The interested reader might also refer to [28], [47] for further information on the matter.

V. Experiments

The proposed CRM model has been simulated in a SIoT framework environment where, based on the reputation scores and the clustering process, its capability to both identify honest and malicious SOs and promote competitiveness has been verified.

In particular, we simulated a single edge domain, a population of 10000 SOs acting as consumers and a population of 500 playing the role of resource providers. Furthermore, each agent was associated with a SO as previously described in this paper.

Moreover, to create a competitive context, we assumed that when a SO purchases a resource offered by a seller belonging to a class better than that of the SO buyer, this later has to pay an overprice. Vice versa, when the SO seller belongs to a class worst than that of the SO buyer then its offered resource will have to be discounted.

Each experiment was conducted for 25 epochs. In turn, for each epoch 2500 *search for resource* processes have been simulated, i.e. the 25% of the SO population having the role of consumer interacted with the SO seller population.

An initial reputation score of 0.5 has been assigned to all the SOs, independently from their role (see Section II). The cost p^r of a resource r randomly varied in the range $[1, 1.5]$ \$, while P_{Max} was set to 1 \$. The system time threshold T has been set to 1 epoch, the other system threshold σ was set to 0.5 and, finally, the parameter α was set to 0.5. In addition, the number of cluster in our simulated edge domain was set to 4.

The effectiveness of our framework in identifying honest and malicious SOs was tested by simulating two different scenarios, identified as *S1* and *S2* respectively. Both the scenarios *S1* and *S2* were characterized by the presence of the 10% of the SO populations of consumers and providers set as malicious at different degrees.

We conceived these two scenarios to test our framework in critical conditions where more and concomitant attacks were carried out by malicious SOs in order to damage of honest ones. More in detail, for *S1*, malicious actors deceived, colluded and provided misleading[2] feedback about their counterparts. Moreover, in the scenario *S2* malicious SOs also performed alternate behaviors for gaining positive reputations when interact for low cost resources, while deceive for high cost resources.

In the simulations *S1* and *S2*, as a measure of effectiveness we have considered the percentage of malicious recognized based on the value of their reputation referred to their overall number (remember that when the reputation score is lower than 0.5 an SO is considered as malicious). The results of these two simulations with respect to the SO consumer population are shown in Figure 3 and 4 for scenario *S1* and scenario *S2*, respectively. Consider that the results of the experiments for the SO seller population in the two scenarios are practically overlapped.

More in detail, the experimental results for the scenario *S1* (Figure 3) show that after 10 epochs more than the 90% of malicious actors have been identified and their number continues to decrease, as the number of simulated epochs increases. Note as at the 17-th epoch the number of malicious SOs unrecognized has became neglecting. The results of the scenario B (Figure 4) are substantially similar to the scenario *S1*, showing that the percentage of recognized malicious actors decreases somewhat less rapidly than for the previous simulation and it is due to the presence also of alternate behaviors. Analyzing the results of these first experiments, it is evident the good performance of the proposed reputation system that allows a rapid identification of malicious in the two considered scenarios

We have also monitored the dynamic of the cluster process. After the first epoch, since at startup all the SOs have the same initial reputation of 0.5, the SOs begin to be allocated among the various clusters based on the value of their reputation scores. We tested this mechanism by setting the number of clusters to 4. Clusters have been identified with numbers varying from 1 to 4, where 1 is referred to the group of SOs less reliable, while 4 is associated with the group including those SOs considered as the most reliable. In Figure 5 is shown how in the scenario *S1* the SO consumer population of clusters varies as epochs increase. Note also as in the scenarios *S1* and *S2*, the consumer and seller SO populations have behaviors almost overlapped with respect to the cluster dynamic.

In Figure 5 it is interesting to observe as the epochs increase, the honest actors moved towards the best group (i.e. cluster 4), while the malicious moved towards the worst one (i.e. cluster 1). In particular, the central groups (i.e. clusters 2 and 3) initially increased in numerosity for then decreasing as the SOs moved toward the best group. Moreover, since the assignment of SOs to groups is based on the reputation scores, it is easy to

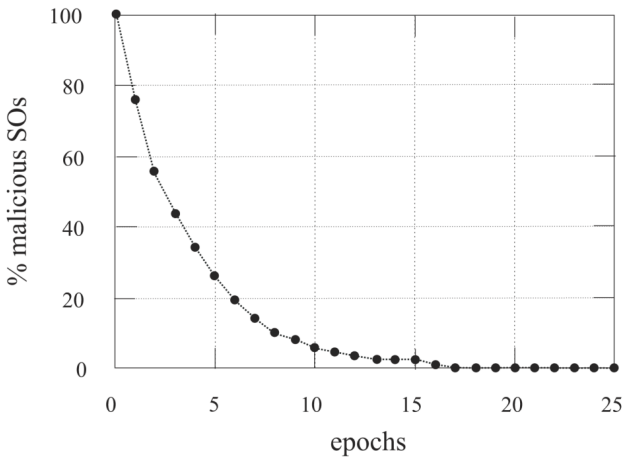

Fig. 3. Scenario *S1*: Percentage of active malicious not recognized.

[2]In other words, malicious SO will assign 0 to partners who have been shown to be trustworthy in behavior and resource quality and, conversely, 1 to unreliable partners.

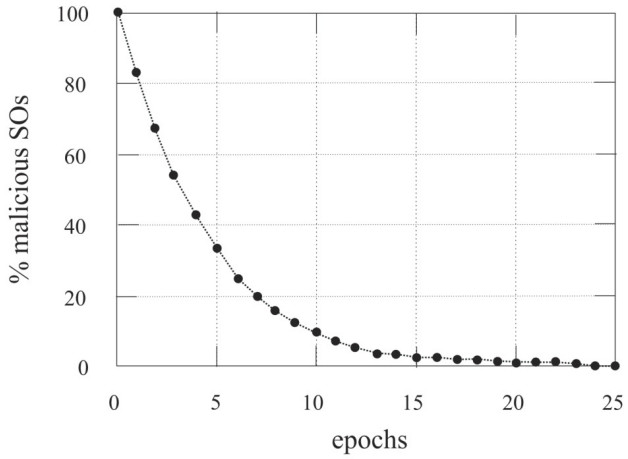

Fig. 4. Scenario *S2*: Percentage of active malicious not recognized.

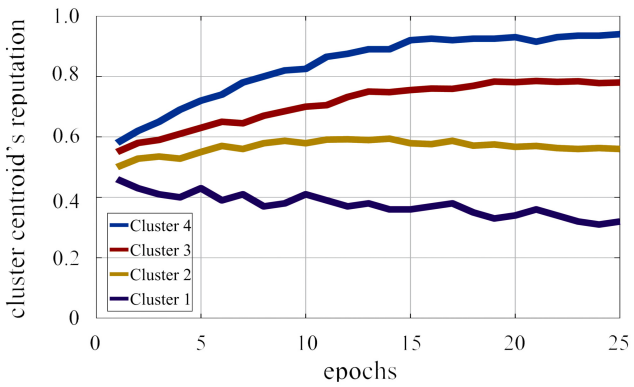

Fig. 6. Cluster centroids.

verify as the numerosity of the cluster 1 follows the process of identification of the malicious SOs, as it is shown in Figure 5.

To complete this analysis on the dynamics of the cluster process, in Figure 6 is shown how the values of the centroids (representing a reputation score) of the clusters vary as the epochs change.

Finally, to confirm the ability of our framework to quickly marginalize the malicious actors, in Figure 7 the number of transactions carried out by malicious as the epochs increase is depicted. From this further confirmation, it is evident that the effects derived from the presence of malicious actors are limited in time (i.e., epochs).

According to the experimental findings presented and discussed here, it can be stated that our *i*) SIoT framework is resilient with respect to the considered malicious behaviors also in presence of a significant percentage of malicious SOs and, moreover, *ii*) the adoption a of a reputation-based clustering strategy contributes to introduce competitiveness by promoting correct behaviors and contributing to marginalize dishonest actors which,as a consequence, have also to pay more than honest ones.

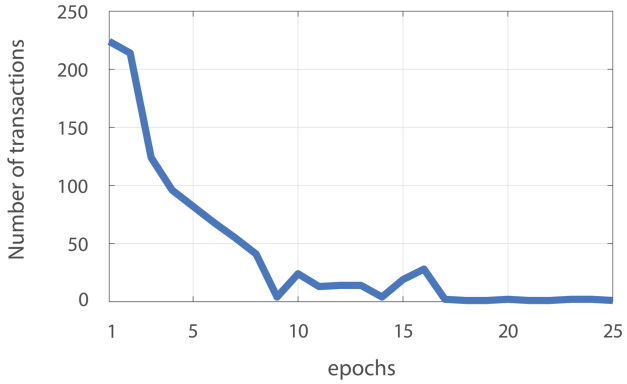

Fig. 7. Number of transactions performed by malicious SO consumers.

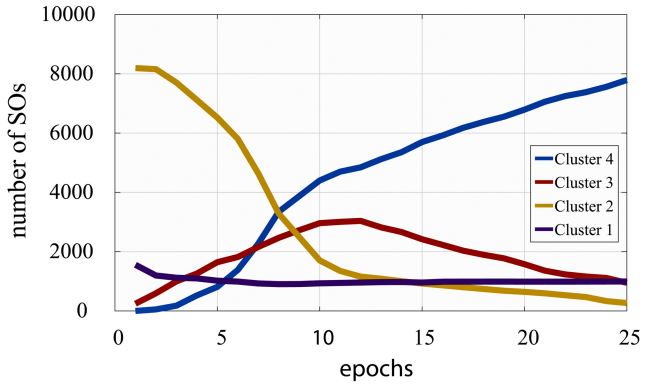

Fig. 5. Clusters composition.

VI. Conclusions

In this paper, we face a key issue of SIoT environments, i.e. that of of avoiding SOs interactions with unreliable partners in an Edge-based context. More in particular, in our framework, when two SOs interact with each other, the edge server associated with that domain updates the reputation values of the two SOs, based on the feedback provided by each SO about its partner. Furthermore, we exploit a cloud server to store the SOs' reputation values. This way, when an object moves from a local edge domain to another one, the value of its reputation will be always available on the cloud.

To realize our solution, we have associated each SO with a software agent, and we have designed a clustering algorithm capable of partitioning different SOs in different clusters, based on their reputation values, in order to increase the SOs' competitiveness in the framework.

A campaign of experiments that we have performed on simulated data shows the advantages introduced by our solution, which is capable of detecting malicious SOs, by clustering them based on of their reputation. In particular, we have

highlighted that malicious actors are quickly identified and their number continuously decreases in time, although the presence of alternate behaviors implies that the percentage of recognized malicious actors decreases somewhat less rapidly. Moreover, we have observed that the honest actors moved towards the best groups, while the malicious moved towards the worst ones. Finally, we have shown that SIoT framework is resilient with respect to the presence of malicious behaviors and the adoption a of our reputation-based clustering approach introduces competitiveness in the community, by promoting correct behaviors and marginalizing dishonest agents.

Our future work on this research line is currently focused on introducing a new, third reputation level in the cloud, with the purpose of clearly separating the local (edge-context) reputation level from a global reputation level taking into account both the whole life of each SO and some interesting agent-based technologies focused on some promising role-based model and methodology for social IoTs like E-CARGO and RBC [48], [49].

ACKNOWLEDGMENT

This work was partially supported by the MUR projects "T-LADIES" (PRIN 2020TL3X8X) and by Pia.ce.ri. 2020-2022 granted by the University of Catania.

APPENDIX

TABLE I
TABLE OF SYMBOLS

Symbol	Meaning
W	Edge-based Framework
E	Edge
e	Edge agent
C	Cloud
c	Cloud agent
SO	Smart Object
t	Smart Object agent
F	Feedback
R	Reputation
ϑ	Interaction parameter
δ	Resource relevance parameter
p	Resource price
P_{max}	Maximum resource price threshold
ϕ	Collusive parameter
μ	Feedback closeness parameter
ξ	Trustor relevance parameter
σ	Trustor relevance threshold

REFERENCES

[1] I. C. Ng and S. Y. Wakenshaw, "The internet-of-things: Review and research directions," *Int. Journal of Research in Marketing*, vol. 34, no. 1, pp. 3–21, 2017.

[2] S. Pattar, R. Buyya, K. R. Venugopal, S. Iyengar, and L. Patnaik, "Searching for the iot resources: fundamentals, requirements, comprehensive review, and future directions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2101–2132, 2018.

[3] M. Zhang, H. Zhao, R. Zheng, Q. Wu, and W. Wei, "Cognitive internet of things: concepts and application example," *Int. Journal of Computer Science Issues (IJCSI)*, vol. 9, no. 6, p. 151, 2012.

[4] M. S. Raza, T. Zongsheng, and M. Muslam, "A review of human-to-machine and machine-to-machine approaches for internet of things," in *2020 Int. Conf. on Communications, Signal Processing, and Applications*. IEEE, 2021, pp. 1–5.

[5] M. Lippi, M. Mamei, S. Mariani, and F. Zambonelli, "An argumentation-based perspective over the social iot," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2537–2547, 2017.

[6] B. Afzal, M. Umair, G. A. Shah, and E. Ahmed, "Enabling iot platforms for social iot applications: Vision, feature mapping, and challenges," *Future Generation Computer Systems*, vol. 92, pp. 718–731, 2019.

[7] M. Roopa, S. Pattar, R. Buyya, K. R. Venugopal, S. Iyengar, and L. Patnaik, "Social internet of things (siot): Foundations, thrust areas, systematic review and future directions," *Computer Communications*, vol. 139, pp. 32–57, 2019.

[8] M. Nitti, L. Atzori, and I. P. Cvijikj, "Network navigability in the social internet of things," in *IEEE world forum on IoT*. IEEE, 2014, pp. 405–410.

[9] S. Suhail, C. Hong, M. Lodhi, F. Zafar, A. Khan, and F. Bashir, "Data trustworthiness in iot," in *2018 Int. Conf. on Information Networking*. IEEE, 2018, pp. 414–419.

[10] F. M. R. Junior and C. A. Kamienski, "A survey on trustworthiness for the internet of things," *IEEE Access*, vol. 9, pp. 42 493–42 514, 2021.

[11] A. Ukil, S. Bandyopadhyay, and A. Pal, "Iot-privacy: To be private or not to be private," in *2014 IEEE Conf. on Computer Communications Work.* IEEE, 2014, pp. 123–124.

[12] M. Seliem, K. Elgazzar, and K. Khalil, "Towards privacy preserving iot environments: a survey," *Wireless Communications and Mobile Computing*, vol. 2018, 2018.

[13] H. Rehman, M. Asif, and M. Ahmad, "Future applications and research challenges of iot," in *Int. Conf. on information and communication technologies*. IEEE, 2017, pp. 68–74.

[14] L. Wei, J. Wu, C. Long, and B. Li, "On designing context-aware trust model and service delegation for social internet of things," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4775–4787, 2020.

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

[16] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin, and X. Yang, "A survey on the edge computing for the internet of things," *IEEE access*, vol. 6, pp. 6900–6919, 2017.

[17] W. Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, and A. Ahmed, "Edge computing: A survey," *Future Generation Computer Systems*, vol. 97, pp. 219–235, 2019.

[18] N. Truong, T.-W. Um, and G. Lee, "A reputation and knowledge based trust service platform for trustworthy social internet of things," *Innovations in clouds, internet and networks*, pp. 104–111, 2016.

[19] A. K. Sikder, G. Petracca, H. Aksu, T. Jaeger, and A. S. Uluagac, "A survey on sensor-based threats to internet-of-things (iot) devices and applications," *arXiv preprint arXiv:1802.02041*, 2018.

[20] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in internet of things: The road ahead," *Computer networks*, vol. 76, pp. 146–164, 2015.

[21] A. I. A. Ahmed, S. H. Ab Hamid, A. Gani, M. K. Khan *et al.*, "Trust and reputation for internet of things: Fundamentals, taxonomy, and open research challenges," *Journal of Network and Computer Applications*, vol. 145, p. 102409, 2019.

[22] J. Zhang, Z. Bhuiyan, X. Yang, A. Singh, D. F. Hsu, and E. Luo, "Trustworthy target tracking with collaborative deep reinforcement learning in edgeai-aided iot," *IEEE Trans. on Industrial Informatics*, vol. 18, no. 2, pp. 1301–1309, 2021.

[23] P. Dumouchel, "Trust as an action," *European J. of Sociology/Archives Européennes de Sociologie*, vol. 46, no. 3, pp. 417–428, 2005.

[24] G. Fortino, L. Fotia, F. Messina, D. Rosaci, and G. M. L. Sarné, "Trust and reputation in the internet of things: State-of-the-art and research challenges," *IEEE Access*, vol. 8, pp. 60 117–60 125, 2020.

[25] Y. Xu, M. Bhuiyan, T. Wang, X. Zhou, and A. Singh, "C-fdrl: Context-aware privacy-preserving offloading through federated deep reinforcement learning in cloud-enabled iot," *IEEE Trans. on Industrial Informatics*, 2022.

[26] F. Cicirelli, A. Guerrieri, G. Spezzano, A. Vinci, O. Briante, A. Iera, and G. Ruggeri, "Edge computing and social internet of things for large-scale smart environments development," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2557–2571, 2017.

[27] Z. Maamar, N. Faci, S. Kallel, M. Sellami, and E. Ugljanin, "Software agents meet internet of things," *Internet Technology Letters*, vol. 1, no. 3, p. e17, 2018.

[28] A. Comi, L. Fotia, F. Messina, D. Rosaci, and G. M. L. Sarné, "Group-trust: Finding trust-based group structures in social communities," in *Int. Symp. on Intelligent Distributed Computing*. Springer, 2016, pp. 143–152.

[29] J. Guo, R. Chen, and J. J. Tsai, "A survey of trust computation models for service management in internet of things systems," *Computer Communications*, vol. 97, pp. 1–14, 2017.
[30] M. Nitti, R. Girau, L. Atzori, and V. Pilloni, "Trustworthiness management in the iot: The importance of the feedback," in *Conf. on Innovations in Clouds, Internet and Networks*. IEEE, 2017, pp. 325–327.
[31] A. Likas, N. Vlassis, and J. J. Verbeek, "The global k-means clustering algorithm," *Pattern recognition*, vol. 36, no. 2, pp. 451–461, 2003.
[32] Z. Yan, P. Zhang, and A. V. Vasilakos, "A survey on trust management for internet of things," *Journal of network and computer applications*, vol. 42, pp. 120–134, 2014.
[33] B. Tang, H. Kang, J. Fan, Q. Li, and R. Sandhu, "Iot passport: A blockchain-based trust framework for collaborative internet-of-things," in *24th ACM symp. on access control models and technologies*, 2019, pp. 83–92.
[34] T. Abera, N. Asokan, L. Davi, F. Koushanfar, A. Paverd, A. Sadeghi, and G. Tsudik, "Things, trouble, trust: on building trust in iot systems," in *53rd ACM/EDAC/IEEE Design Automation Conf.* IEEE, 2016, pp. 1–6.
[35] C. Marche and M. Nitti, "Trust-related attacks and their detection: A trust management model for the social iot," *IEEE Trans. on Network and Service Management*, vol. 18, no. 3, pp. 3297–3308, 2020.
[36] Z. Lin and L. Dong, "Clarifying trust in social internet of things," *IEEE Trans. on Knowledge and Data Engine.*, vol. 30, no. 2, pp. 234–248, 2017.
[37] R. Chen, F. Bao, and J. Guo, "Trust-based service management for social internet of things systems," *IEEE Trans. on dependable and secure computing*, vol. 13, no. 6, pp. 684–696, 2015.
[38] Y. Saleem, N. Crespi, M. Rehmani, R. Copeland, D. Hussein, and E. Bertin, "Exploitation of social iot for recommendation services," in *IEEE 3rd World Forum on Internet of Things*. IEEE, 2016, pp. 359–364.
[39] P. De Meo, F. Messina, M. N. Postorino, D. Rosaci, and 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.
[40] I. U. Din, M. Guizani, B.-S. Kim, S. Hassan, and M. K. Khan, "Trust management techniques for the internet of things: A survey," *IEEE Access*, vol. 7, pp. 29 763–29 787, 2018.
[41] B. Pourghebleh, K. Wakil, and N. J. Navimipour, "A comprehensive study on the trust management techniques in the internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9326–9337, 2019.
[42] D.-H. Kang, H.-S. Choi, and W.-S. Rhee, "Social correlation group generation mechanism in social iot environment," in *2016 Eighth Int. Conf. on Ubiquitous and Future Networks*. IEEE, 2016, pp. 514–519.
[43] W. Yang, Y. Qin, and R. Li, "A network embedding-based approach for scalable network navigability in content-centric social iot," *IEEE Internet of Things Journal*, 2022.
[44] B. Farahbakhsh, A. Fanian, and M. Manshaei, "Tgsm: Towards trustworthy group-based service management for social iot," *Internet of Things*, vol. 13, p. 100312, 2021.
[45] B. Wang, Y. Sun, T. Q. Duong, L. D. Nguyen, and N. Zhao, "Security enhanced content sharing in social iot: A directed hypergraph-based learning scheme," *IEEE Trans. on Vehicular Technology*, vol. 69, no. 4, pp. 4412–4425, 2020.
[46] G. Fortino, F. Messina, D. Rosaci, and G. M. L. Sarné, "Using blockchain in a reputation-based model for grouping agents in the internet of things," *IEEE Trans. on Engineering Management*, vol. 67, no. 4, pp. 1231–1243, 2019.
[47] A. Comi, L. Fotia, F. Messina, D. Rosaci, and G. M. L. Sarnè, "A qos-aware, trust-based aggregation model for grid federations," in *OTM Confederated Int. Conf." On the Move to Meaningful Internet Systems"*. Springer, 2014, pp. 277–294.
[48] H. Zhu, *E-CARGO and Role-based Collaboration: Modeling and Solving Problems in the Complex World*. John Wiley & Sons, 2021.
[49] ——, "Computational social simulation with e-cargo: Comparison between collectivism and individualism," *IEEE Transactions on Computational Social Systems*, vol. 7, no. 6, pp. 1345–1357, 2020.

Giancarlo Fortino Giancarlo Fortino (IEEE Fellow '22) is Full Professor of Computer Engineering at the Dept of Informatics, Modeling, Electronics, and Systems of the University of Calabria (Unical), Italy. He is Highly Cited Researcher 2002-2021 in Computer Science. His research interests include wearable computing systems, Internet of Things, and Cyber-security. He is author of 550+ papers in int'l journals, conferences and books. He is (founding) series editor of the IEEE Press Book Series on Human-Machine Systems and of the Springer Internet of Things series, and is AE of premier IEEE Transactions. He is cofounder and CEO of SenSysCal S.r.l., a Unical spinoff focused on innovative IoT systems. Fortino is currently member of the IEEE SMCS BoG and chair of the IEEE SMCS Italian Chapter.

Lidia fotia received her PhD in Information Engineering from the University Mediterranea of Reggio Calabria in 2014. She is currently serving as assistant professor with tenure track at the department DIEM at the University of Salerno. Her research interests include social network and social internetworking analysis, privacy, security, trust and reputation, intelligent agents. Contact her at lfotia@unisa.it

Fabrizio Messina Fabrizio Messina received his Ph.D. in Computer Science from the Department of Department of Computer Science and Mathematics at the University of Catania, Italy in 2009. He is currently serving as assistant professor with tenure track in the same department. His research interest includes Distributed systems, Simulation systems, Trust and reputation. Contact him at fabrizio.messina@unict.it.

Domenico Rosaci is Associated Professor of Computer Science at the Department of Information, Infrastructures and Sustainable Energy Engineering at the University Mediterranea of Reggio Calabria, Italy. In 1999, he took the PhD in Electronic Engineering. His research interests include distributed artificial intelligence, multi-agent systems, trust and reputation in social communities. He is a member of a number of conference PCs and he is Associate Editor of Journal of Universal Computer Science (Springer). Contact him at domenico.rosaci@unirc.it.

Giuseppe M. L. Sarné is Associated Professor of Computer Science at the Department of Psychology at the University of Milan Bicocca, Italy. His main research interests include distributed artificial intelligence, multi-agent systems, trust and reputation systems. He is a member of a number of conference PCs and he is Associate Editor of E-Commerce Research and Applications (Elsevier) and member of the Editorial Board of Big Data and Cognitive Computing (MDPI). He is IEEE senior member and member of the IEEE technical committee on Hyper-intelligence. Contact him at giuseppe.sarne@unimib.it.