© <2021>. This post-print version is made available under the CC-BY-NC-ND 4.0 license http://creativecommons.org/licenses/by-nc-nd/4.0/Þ published paper is available at DOI:https://doi.org/10.1016/j.simpat.2020.102261

A Blockchain-based Group Formation Strategy for Optimizing the Social Reputation Capital of an IoT Scenario

Giancarlo Fortino*, Lidia Fotia§, Fabrizio Messina¶, Domenico Rosaci‖, Giuseppe M. L. Sarné §

*Depart. DIMES, Univ. of Calabria, Via P. Bucci, 87036 Rende (CS), Italy, giancarlo.fortino@unical.it

§Depart. DICEAM, Univ. Mediterranea, Loc. Feo di Vito, 89122 Reggio Cal. (RC), Italy, {lidia.fotia, sarne}@unirc.it

¶Depart. DMI, Univ. of Catania, Viale Andrea Doria 6, 95126 Catania (CT), Italy, messina@dmi.unict.it

‖Depart. DIIES, Univ. Mediterranea, Loc. Feo di Vito, 89122 Reggio Cal. (RC), Italy, domenico.rosaci@unirc.it

Abstract

The "Internet of Things" (IoT) provide humans and smart objects with attractive services, based on the advanced features of the IoT devices, like high sensing, real-time acting and reasoning.

In our previous research we have highlighted that these features can be improved by promoting cooperation between smart objects, and we introduced the association between Multi-Agent Systems and IoT devices. In that context, we focused on the issue of accurately choosing the best partners for cooperation, in a scenario composed by several federations. We proposed a reputation model and we have shown that the model leads to detect agents having unreliable or misleading behaviors and that the model itself can be profitably used to form groups of agents that mutually cooperate for improving the effectiveness of their tasks. In this further contribution, we focus on the important issue of the group formation, by arguing that in practical IoT situations it is necessary to improve the group formation strategy to provide it with greater adaptability. To this end we introduce – in a particular

IoT context described in this work – a two-phase group formation algorithm to support the reputation model. Experimental results prove that the adoption of the group formation algorithm, along with the proposed reputation model provides a few benefits to the whole IoT ecosystem.

Index Terms

Blockchain, IoT, Reputation Capital, Group Formation, Multi-agent System.

I. Introduction

The main actors in the "Internet of Things" (IoT) [2] are the IoT "smart" *objects* (i.e., things), that are physical (or virtual) entities with embedded computational, sensing and communication abilities [3], where each object is identifiable and traceable in space and time. In order to reach their goals, these smart objects need to be provided with human-like capabilities. In our previous research [1], we have highlighted that the features above can be improved by introducing complex forms of collaboration between smart objects. This consideration leaded us to propose the association of the Multi-Agent Systems technology with the IoT devices. The convergence of IoT and MAS allows ubiquitous and heterogeneous devices to exploit the needed services in a scalable and pervasive way, thanks to the agent's social attitude to interact and collaborate. In this scenario, the possibility of constructing a wide network composed by heterogeneous smart objects becomes very interesting, since this type of objects are capable of living, acting as prosumers and moving among different, federated environments.

In [1] we already focused on the issue of accurately choosing the best partners for cooperation, in a scenario where IoT devices migrate among different federated environments, where the IoT objects will be unreferenced in terms of its trustworthiness. To this aim, in [1] we proposed a reputation model for characterizing the social reputation of each agent. We have proved that the proposed model leads to detect agents having unreliable or misleading behaviors, designing a strategy by which these agents pay higher costs for obtaining services in the IoT environment, with respect to more reliable and honest agents. However, the cooperation among objects, also small and low-cost, implies their capability of having social interactions with other devices. Then, as preliminary result, we have proved that the designed model can also be profitably used for forming groups of agents that cooperate for improving the effectiveness of their tasks.

In this work, we further delve into the significant aspect of group formation. In particular, we

argue that in practical situations the group formation strategy should be adaptive with respect to the characteristics of the global IoT network [4], [5]. This leads us to introduce, with respect to our past researches, a group formation algorithm which is particularly suitable to also extend the reputation model already introduced in [1] to take into account the perspective above.

With respect to [1], in this work we extend the idea of modeling the reputation of each agent by using a sort of "personal capital" represented by the sum of the feedback received by the agents during the interactions with the other agents. Furthermore, we also propose to employ blockchain technology to maintain information about agents trustworthiness (i.e, the personal capital). Blockchain technology can ensure trust and data integrity to anonymous entities through decentralized, distributed P2P network [6], and the use of cryptographic validation techniques, representing the safe replacement of third parties or centralized authorities to certify such a reputation. Then we extend the original idea of using blockchain to support a trust and reputation model, that we have presented in [7].

While in that paper we only focused on the reputation model, and presented a first version of an algorithm to form groups of agents, in this new paper we particularly deal with the group formation strategy. This new strategy is based on a two-phases algorithm that, differently from the approach presented in [7] fixing the number of the groups and the threshold of reputation for joining with a group, gives instead the possibility to compute the aforementioned parameters based on the characteristics of the IoT objects. Moreover, a new, large experimental campaign has been here implemented to validate the advantage introduced by our approach.

Then, as the core of our contribution, we introduce a new algorithm to support the group formation based on the trustworthiness stored in the blockchain. In particular, this new technical contribution consists of proposing an approach based on a competitive IoT scenario, able to enhance the whole social capital of the community, by implementing a strategy conceived to prevent possible collusive and misleading activities to incorrectly increasing the personal capital of the agents. Moreover, the new strategy of group formation we have introduced implies that an agent desiring to join with a group having a high average reputation, needs to first obtain a high personal capital of reputation.

As further contribution with respect to the previous work, we will present the results of a large experimental campaign aimed at verifying the good performance of the group formation procedure in terms of group composition; in particular we will show that the group formation

procedure is capable to increase the overall reputation capital of the IoT community.

The paper is organized as follows. In Section II, we survey the related work of the recent literature, while in Section III the competitive IoT scenario is presented. Section IV illustrates the reputation-blockchain mechanism. Section VI presents our group formation algorithm and Section VII describes and discusses the results of our experimental campaign and in Section VIII the conclusions are carried out.

II. Related Work

Distributed Systems (DS) are subject to a greater number of threats for malicious and/or disliked behaviors [8] than Centralized ones. This problem becomes critical in the case of open and competitive DSs, but cryptographic techniques [9], trust and reputation systems [10], [11], can reduce risks and support user (i.e., agents) activities.

In particular, cryptographic techniques guarantee protection from external attacks by assuring privacy and counterparts authentication [12]. Conversely, trust and reputation systems allow to esteem the trustworthiness of possible partners, to restrict the possibility of interacting with unreliable users [13], considering the direct experiences (reliability) and/or view point of others (reputation) [14], [15], [16], [17].

A. Trust and reputation systems

Trust criteria are adopted both in real and virtual contexts [18], [19]. Many existing studies propose different points of view [20], [21], [22], and a lot of analysis, architectures and models. The accuracy in esteeming trust depends on three principal factors: quality and number of the information sources [23], the use of a centralized or distributed context [24] and the type of aggregation used for the trust (i.e., local or global way) [25].

As shown in [26], a reputation system must content the following properties: *i*) Entities must have a strong background of past experience so that they can predict future behavior and prevent whitewashing strategies; *ii*) New interactions must be based on past experiences; *iii*) Entities must grant and spread their feedback in the scenario (that is more complicated into DSs).

Reputation scores in DSs can be propagated by means of: *i*) a distributed and synchronized repositories; *ii*) data collected from feedbacks of participants having mutual interactions and/or the opinion of others [27].

In real and virtual communities, trust and reputation systems are helpful also in the formation of groups. In particular, they are used to recommend to a group (member) of a community which are the best members (groups) for the affiliation with (i.e., group recommendation) [28]. In this way, the formed groups result more stable over time with respect to other strategies [29]. For example, Aikebaie et al. [30] propose a trust-based procedure to form groups in a Peer-to-Peer (P2P) system through the agent trustworthiness. The advantages of this model are witnessed by the simulation campaigns. In [31], the authors demonstrated that a formation mechanism for long-term coalitions based on trust agent relationships brings benefits to the system and the agents.

Furthermore, the IoT world can benefit from the adoption of trust and reputation criteria for improving the device performance [32], [33]. Bao et al. [34] proposed a system in which the IoT devices mutually trust their counterparts and propagate their trust value in the form of recommendations with a *word of mouth* mechanism. In [35], a trust system that is adaptable to the evolution of social relationships over time is proposed. The convergence among IoT, software agents and cloud computing to form groups of agents (each one associated with an IoT device and living on the cloud) by means of an algorithm that combines reliability and reputation collected by the agents is described in [36]. Finally, Alshehri et al. [37] introduced a scalable trust management algorithm to form trust-based clusters/groups of IoT devices and allowing trust-based inter-cluster migration of IoT nodes. Recall that the IoT nodes can gain or lose trust values when collaborate with other nodes of their clusters.

B. Blockchain

A blockchain is a chain of data blocks chronologically ordered and replicated on more ledgers that, by means of a distributed consensus protocol, autonomously maintain their local copies "synchronized" making the blockchain difficult to be controlled, tampered or deleted. After that each block has been validated by the consensus it becomes permanent, immutable and accessible [38]. More in detail, each block consists of a header (i.e., identifier, timestamp, number of stored transactions, size of the block and the hash of the previous block in the chain) and the data stored.

The use of the blockchain technology to support smart-contracts (i.e., "a computerized transaction protocol that executes the terms of a contract" [39]), which has been realized for the

first time by the Ethereum [40] blockchain, has allowed new applications of blockchain. To this purpose, many features are made available to develop smart contracts, for example, Ethereum [40] makes available a Touring-complete language programming to develop code for smart-contracts. Nowadays, a great number of blockchain platforms allow to implement smart contracts, for example Hyperledger [41], Ripple [42], Stellar [43] and Tendermint [44].

A very critical aspect of blockchains is represented by the computational complexity placed on the ledger by the exploited consensus protocol. For this reason, different consensus protocols have been proposed/implemented [45], which mainly differ in computational complexity which, in turn, is tightly linked to the robustness against manipulations. These factors need to be considered when applying blockchain to the IoT contexts. For example, the *Proof of Work* (PoW) initially adopted by the Bitcoin currency [46] demands to resolve a computationally expansive hashing puzzle for making valid and adding a new block. Hence, the application of the PoW in an IoT context often requires the adoption of other technologies as, for instance, the cloud computing. Also the long latency, low scalability and poor environmental friendly of the PoW given impulse to the development of other consensus mechanisms [47] by relaxing some security requirements. There are several alternatives to the PoW, such as the *Proof of Stake* (PoS) that is based on the concept that there is something at stake; different its variants have been presented [48], [49], [50].

In the IoT area, Christidis et al. [51] discussed a proposal relying on blockchain and smart-contract technologies, to validate transactions by facilitating and supporting the autonomous workflows and services sharing occurring among IoT devices with benefits in terms of payments, trading, shipping and supply-chain management. In a distributed context, also including IoT devices, the auhors [52] designed Trustchain, which allows blockchain-based trusted transactions in an scalability, openness and Sybil-resistance environment by introducing a consensus protocol alternative to the PoW.

Recently, in the literature, different types of blockchains have been proposed that allow to propagate trust and reputation scores without the use of trusted and powerful third-party. We mention for example *Islands of Trust*, proposed in [53], that consents to spread trust across different IoT domains by using two blockchains: one is a private credit-based blockchain based on the reputation and the other is used for payments. Respect to the system in [53], our proposal has three important differences: *i*) the adoption of a reputation system based on the reputation

capital score; *ii*) we use a single blockchain; *iii*) the use of smart contracts;

III. The underlying IoT scenario

We suppose that our scenario contains a number N of heterogeneous IoT objects (see Figure 1), each of them supported by a software agent. In such a scenario, the agents can interact in the execution of a task on the basis of smart contracts.

Let A denote the set of software agents, and GN the *Global Network* consisting of several federated *Local Networks* (say LN). A particular agent, called *Local Network Administrator* (LNA_m), is delegated to administrate each local network by providing some basic administrative service to the agents of its network. Moreover, we assume that in each local network agents can form groups taking into account their reputation capital verified by the blockchain and, in turn, each group can affiliate agents exclusively belonging to its local network. We highlight that agents are free both of requesting to join with or leave a group active on their current local network and of shifting from a local network to another one and, in this case, they will have to leave the groups they belonged to in the local network of origin. Furthermore, we also assume that each group of a local network m is administrated by the respective LNA_m that, to maximize the effectiveness of the group itself, can also contact other agents to join with or delete from its administrated group those agents having an insufficient reputation capital.

For convenience, the set of agents A in GN and their relationships are represented by means of a graph $G = \langle N, L \rangle$, where N and L represent its sets of nodes and oriented links, respectively. In particular, each node of N is associated with a unique agent and each link of L represents a relationship occurring between two agents (see intermediate layer of Figure 1).

IV. The Reputation Model

Here we will introduce the notion of Reputation Capital. In this approach, the reliability of a consumer is guaranteed by the blockchain (in this case, it must pay the service to another agent), while the reputation capital of a provider witnesses its ability to provide quality service.

The Reputation Capital (RC) is represented by a numerical value – a real positive number – computed on the basis of the past interactions among agents on the basis of the following requirements:

- the more recent the activities, the better the weight of the feedback;

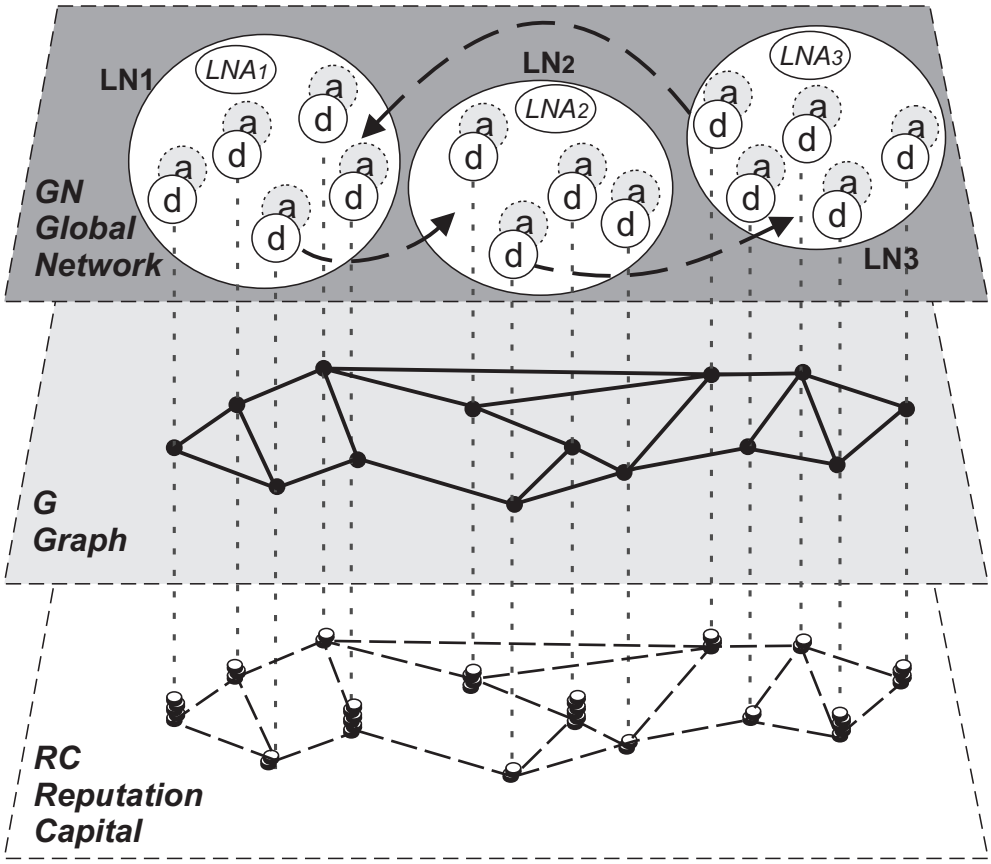

Fig. 1. The proposed IoT scenario

- countermeasures should be taken against collusive behaviors among agents (i.e., behaviors aiming at mutually and incorrectly increasing their reputation capitals);
- alternate behaviors should be hindered, by assigning a feedback value which indicates a "bad" behavior;
- the behavior of "habitual" complainers should be limited by measuring their "credibility", as reasonably proposed in [54].

In particular, high (resp., small) values of RC indicate a good (resp., bad) reputation.

An initial value of reputation capital is introduced as a countermeasure over whitewashing strategies of dishonest agents that aim at coming back in the system to take a fresh favorable reputation [55]. Moreover, the value should be chosen in order not to penalize a newcomer [56].

In particular, when the provisioning of a service is required to an agent a_i (the provider) by another agent a_j (the consumer), this latter provides a feedback $\phi_{ji} \in [0, 1] \subset \mathbb{R}$ representing the

satisfaction of a_j for the service received from a_i. For convenience, we introduce a threshold equal to 0.5 to discern "bad" and "good" feedbacks. A value $\phi < 0.5$ will represent a "bad" feedback. This choice is based on the assumption that the evaluations given by the agents follow a linear scale. Anyway the threshold can be selected in many different ways, depending on the evaluation model behind the rational of the agents.

In order to calculate RC, we do not take into account all the interactions, but only those classified as *qualified interactions*: to this end, whenever a provider agent a_i releases a service s to a consumer agent a_j, the latter assigns a feedback $\phi_{ji} \in [0, 1] \subset \mathtt{R}$ to express its own satisfaction. If the interaction is "qualified" then ϕ_{ji} is used to update the RC_i of a_i (the same considerations are made if the agent a_j is a provider). Therefore, the *Relevance* R of an interaction is calculated as c_s/C, if $c_s < C$, otherwise it is set to 1. The parameter c_s is the cost of a service s and the term C represents the price threshold at which the relevance is computed at its maximum value 1. The previous definition of relevance allow us to give the further definition of "qualified interaction": an interaction is qualified if i) $\phi < 0.5$ or ii) $R \geq \phi \geq 0.5$. The ratio behind this last definition is to discern interactions having values $\phi \geq 0.5$ and, at the same time, a value of relevance $R < 0.5$.

Based on the last h qualified interactions of the provider agent a_i occurred with different h agents (a measure aimed to contrast collusive behaviors), its RC_i is updated in the following way:

$$RC_i = \sum_{n=1}^{h} \omega_n \cdot \delta_n \cdot R_{n,i} \cdot \phi_{n,i} \tag{1}$$

where *i*) ω gives more importance to the most recent qualified interactions and *ii*) δ mitigates the impact of agents that release a low feedback (e.g., $\phi < 0.5$) in order to obtain personal benefits (for an agent, it is set as the complement to 1 of the ratio between the number of low feedbacks (NF) with respect to the total number of its interactions (NT)).

V. Using Smart Contracts for Managing the Reputation Capital

It is well known that security for distributed IoT can be supported by the blockchain technology, as discussed in Section II. In particular, the blockchain [46] ensures trust and data integrity to unknown and anonymous entities (permissionless), through decentralized, distributed, open, and

unchangeable ledger saving data across a peer-to-peer (P2P) network by using cryptographic technologies to identify source and sink of the data, support transactions and management of complex digital assets, smart contracts and data content. Thus, the blockchain represents the safe replacement of third parties or centralized authorities to certify information about trustworthiness [57], [58]. In an IoT context, the choice of the most suitable blockchain platform tightly depends on several characteristics like, for instance, device characteristics (including communication and power capabilities), the operative context, the permisioned or permissionless nature of the blockchain and so on. To this purpose, there are several protocols potentially suitable (in a native or not way) for the IoT world including, for instance, Ethereum [40] (for permissionless blockchain but with some limitations), Hyperledger Fabric [59] (for private, permisisoned blockchains) and IOTA [60] (supporting both the modalities).

In the following, we provide a short description of the role of the blockchain technology – as part of our proposal – to "certify" the trustworthiness of the reputation associated with the agents.

Into a LN, the request for a service is equivalent to run a smart contract on the blockchain platform in order to check and implement all the contractual obligations. Even though our proposal is independent from a specific (smart-contract) blockchain, in this setting we propose to exploit the Ethereum platform for convenience: (*i*) the presence of documented API; (*ii*) the availability of an own cryptocurrency (i.e., Ether) to pay in GN and (*iii*) the availability of several simulation tools to create a personal local Ethereum blockchain for testing and development. However, we remark as our proposal is independent from a specific (smart-contract) blockchain and other blockchain platforms could be used in place of Ethereum without it affects our simulations results (see Section VII).

The smart contract encloses the necessary steps to update both RC of the provider and the following information about:

- the identifier of the agent who is associated the reputation capital and the identifier of its LN;
- the number of interactions of the agent (i) successfully completed, (ii) aborted, (iii) with a low feedback (i.e., $\phi < 0.5$), (iv) with no feedback;
- the list of the q transactions concurring to form the current RC, where each rows is a tuple composed by:

- the identifier of the counterpart agent and its LN;
- the price and the date of the transaction;
- ϕ assigned to the interaction.

Each LNA administrates a private list of the agents currently living in the LN (sorted according to its identifiers), and saving the actual (e.g., updated) RC, a timestamp and the reference on the blockchain that link to where the interesting information is stored. Observe that the list is accessible only by the LNA and the blockchain managers. Also, when an agent starts an interaction, it receives a certificate signed by a private key that contains all the information mentioned above.

VI. The Group Formation Procedure

Now, we will illustrate the procedure designed to be executed by each LNA to form groups of objects on the basis of their RC values. In this context, the role of a group is that of offering the possibility to implement a form of collaboration among the smart objects, whenever a service s is free only if the consumer agent belongs to the same group of its provider.

The approach for group formation is composed by two temporal phases, as described in sections VI-A and VI-B. In fact, the design of this procedure represents a new strategy that, differently from the approach presented in [7], fixing the number of the groups and the threshold of reputation for joining with a group, gives instead the possibility to compute the aforementioned parameters based on the characteristics of the IoT objects.

A. Phase 1: Determining number of groups and RC Thresholds

In this phase, we determine the number ng of the groups and the thresholds $n_1, n_2, .., n_{ng}$ that result as optimal from the viewpoints of the social capital of the whole community.

To this end, we consider that each of the N objects of the IoT community has its reputation capital RC, and thus the problem to solve can be viewed as that of partitioned a set of N objects into ng groups in the best way from the viewpoint of the homogeneity of each group (i.e., the objects of each groups should be the most similar as possible in terms of reputation capital). In our approach, we propose to solve this problem by applying the well-known K-means clustering algorithm, that has the goal of minimizing the total variance intra-cluster [61].

Therefore, the input of this phase is the vector composed by the N values of reputation capital of the IoT objects, and the output will be the number of determined clusters, that we will use as the value ng in the next phase, and the minimum reputation capital of each cluster, that we will use as threshold value in the next phase.

We highlight that actually, the number ng of clusters in the K-means algorithm is an input parameter. Indeed, for an integer $k \geq 1$, the so-called k-means clustering method consists in partitioning a random sample $X_1, .., X_n$ of real values in k groups by minimizing the empirical distortion $W(k) = \frac{1}{n} \sum_{i=1}^{n} min_{j=1}^{k} \|X_i - c_j\|^2$.

Therefore, a good choice of k is essential for building meaningful clusters. Several procedures for choosing k can be found in [62], while [63] compares the performances of the main approaches. The proposed methods existing in the literature can be divided in two main types, global or local. Global procedures consist in performing clustering for different values of k and then retaining the value minimizing or maximizing some function of k. In local procedures, it must be decided at each step whether a cluster should be partitioned (or two groups merged into a single one). In our proposal, we have decided to adopt the classical approach of Calinski and Harabasz [64], that proposes to choose the value of k which maximizes an index based on the quotient $\frac{B(k)/(k-1)}{W(k)/(n-k)}$, where n is the number of values to be clustered, $W(k)$ is the empirical distortion whereas $B(k) = \sum_{j=1}^{k} \|c_j - \bar{c}\|$, where $\bar{c}$ is the mean of the values, is the between sum of squares. The value of k determined as above is that we will use as ng parameter in the next phase. Moreover, for each cluster l, $l = 1, ., ng$, we compute the minimum value n_l of the reputation capitals RC_j belonging to the cluster g_l, and we use the set of all these n_l as threshold parameters (Γ_l) for the next phase. In addition, the approach of Calinski and Harabasz [64] establishes that the sum of all k is equal to n, where n is the number of agents. For this reason, an agent cannot be part of more than one group. In addition, the agent aims to join the group with the highest reputation and which offers him the opportunity to join.

B. Phase 2: Populating the groups

The parameters computed in the first phase represents only a starting point of the designed strategy. Indeed, the reputation capital of the agents will change in time, therefore the procedure aimed at determining parameters ng. Since such a repetition will be actually infeasible in a practical situation, we choose to maintain the parameters computed in the phase 1 (ng and n_i),

and to periodically execute the second phase of the algorithm, which dynamically moves an object from a group to another based on the varying value of each individual reputation capital.

Now, we explain the procedure executed by each LNA, which is shown in Algorithm 1.

The affiliation of each agent will be checked periodically by the administrator in order to maximize its RC. Therefore, an agent undergoes the following procedures:

(*i*) it is displaced to another group that better matches with it, in terms of reputation capital (RC), or (*ii*) it is eliminated from all group because its RC value is not enough to join the groups in the local network.

More in detail:

1) A denotes the agents currently inserted in LN;
2) G denotes the groups in LN;
3) g_l denotes the l-th group in G (i.e., $G = \{g_l\}$).

The Agent administrator LNA operates on two datasets named DA and DG. The two dataset store the following information:

1) the reputation capital RC_i and the timestamp (τ_i) of the last measurement for the each agent a_i living in the local network LN;
2) the data of each group g_l sorted by its value Γ_l (i.e., the RC threshold), the IDs of the agents present in every g_l, the timestamp (ρ_l) of the last "group analysis" (see Algorithm 1)

The procedure shown in the Algorithm 1 is performed periodically. It is also triggered whenever an agent asks to be part of a group. In particular, the LNA checks on the blockchain the RC of all the agents in the LN and updates the dataset DA with the RC values (Lines $1-5$). In particular, Ψ represents the time threeshold after which the value of RC can be considered "old". Then, for each group g_l of the LN, LNA verifies whether its members a_i still satisfies the affiliation requirements ($RC_i > \Gamma_l$), otherwise the procedure *Assign( )* is executed (Lines $6-14$) to find another suitable group for a_i, as explained later in this section. Finally, for each agent asking to join a group g_k, the LNA tests if its RC is greater or equal to the threshold Γ_k; in this case the agent can join the group g_k, (lines $17-18$) otherwise the request is rejected.

In Algorithm 2, the function *Assign( )* receives the following inputs: a_i, the number of groups ng and the datasets DA and DG. First of all, *Assign ( )* verifies if RC_i is lower than the lower threshold Γ_1; in this case a_i is removed from any group in G as long as RC_i will have an inadequate value to be part of a new group. Otherwise, a_i is assigned to any group g_k having a

threshold Γ_k lower than RC_i. To this end, we remark to the reader that the scanning order of G is not specified. If G is visited in ascending order ($k = 1, 2, \dots$), the agent a_i would be assigned always to the first group g_1. Viceversa (descending order) the selected group will be the first one having a suitable threshold $\Gamma_k < RC_i$. In our implementation the groups are selected at random (and each group already selected is excluded from the set for the subsequent extraction) in order to not fill the groups from the first one or the last one. Such strategy allowed us to provide to the groups (those having a suitable threshold $\Gamma_k < RC_i$), the same chance to host the agent a_i.

Algorithm 1 The procedure performed by every LNA of the IoT ecosystem.

Input: DA, DG, Ψ, ng;

for all $a_i \in LN$ **do**
  if $(t - \tau_i) \geq \Psi$ **then**
    ask to the blockchain to update RC of a_i and DA
  end if
end for
for all $g_l \in G$ **do**
  if $((t - \rho_l) \geq \Psi)$ **then**
    for all $a_i \in g_l$ **do**
      if $(RC_i < \Gamma_l)$ **then**
        Assign (a_i, ng, DA, DG)
      end if
    end for
  end if
end for
for all $a_i \in LN$ requesting to be a member of a group g_k **do**
  if $RC_i \geq \Gamma_k$ **then**
    put data of a_i into the dataset DG for the group g_k
  else
    reject the request of a_i and communicate the decision to a_i
  end if
end for

Algorithm 2 *Assign* (a_i, ng, DA, DG).

```
if (RC_i < Γ_1) then
    remove a_i from the dataset DA
    remove a_i from each group in DG
else
    for all g_k ∈ G do
        if RC_i ≥ Γ_k then
            put data of a_i into the dataset DG for the group g_k
            return
        end if
    end for
end if
```

VII. Experiments

An experimental campaign has been carried out to evaluate the performance of our approach also with respect to the two phases of the group formation algorithm. In particular, the experiments have been performed by adopting different values for both the horizon parameter (h) and the number of active malicious devices living in the IoT environment. In such a way, we have investigated on: *i*) the effectiveness in identifying malicious actors by implementing several types of attacks at the same time, as described in Section IV; *ii*) the dynamics of group affiliations occurred in presence of a competitive scenario (see Section VI); *iii*) the costs sustained for services by the IoT members.

Simulations have been carried out on an IoT simulator based on the ACOSO methodology and platform, which can simulate IoT networks and IoT devices in a scalable way and can support also the construction of real IoT systems [5], [65]. This choice has provided greater versatility in developing and executing experiments and elaborating the results (also) in real time. To simulate smart-contracts, although we remark as our proposal is independent from a specific (smart-contract) blockchain, we referred to the well known Ethereum blockchain platform [40] (see Section V). To this aim, a local Ethereum environment has been exploited to simulate smart contracts. More specifically, smart contracts have been written in Solidity [66], compiled on the Ethereum Virtual Machine and public JAVA scripting libraries for Ethereum and the open source

Truffle Suite [67] have been used. However, as a verification, some test smart contracts have been executed on the Ethereum blockchain confirming the simulation results.

In the following of the section, both the parameter setting and the experimental results will be presented and analyzed.

A. *Parameter Setting*

In our experiments we considered a single federated $LN \in GN$ where each IoT device belongs to the LN, and it is supported by a personal software agent. For each of these agents we simulated a long sequence of interactions with the other members. In performing the experiments, we varied in LN the percentage of cheaters device/agents. In particular, during the simulations cheaters carried out, in an endless manner, different sequences of malicious activities (i.e., collusive, alternate, complainer and noising). More in detail, simulations have been carried out by adopting for the main parameters the following setting:

- The number of IoT devices/agents in LN has been set to 10^3 units.
- For each interaction two agents have been randomly chosen, the former played the role of consumer of the service while the other one played the role of service provider.
- Simulations consisted of 10^3 epochs where each epoch, in turn, is formed from 10^3 interactions so that, in average, each agent plays the role of consumer and provider one time for epoch[1]. Note as less than 10 epochs are necessary to have "stable" results.
- The initial RC score has been set to 1.0 as the result of a preliminary trial and error procedure we carried out. We verified that such a value is effective in discouraging whitewashing strategies without penalizing the honest ones too much.
- The parameters c_s and C are respectively the cost of a service s (randomly set between $0.1\$ \div 1.5\$$) and the maximum cost (set to $1\$$) after which the relevance (R) of the service is assumed as "saturated" (i.e., $R = 1$). Note that the setting of cost parameters depends closely on the reference scenario.
- The *horizon* parameter values $h = 4$, 6, 8 and 10 have been adopted. In particular, $h = 4$ and $h = 10$ can be considered as two bounds; in fact, for horizons lower than 4 the system

[1]Note that after four epochs (i.e., $h = 4$) only the 0.17% of the overall agent population has not carry out any interaction neither as provider nor as consumer; as the epochs increase this percentage decreases until the 0.01% at the 6th epochs and zero from the 7th epoch onward.

performances are not acceptable, while they become quite stable with horizons equal or higher than $h = 10$.

- The percentage of cheaters varied from 5% to 20% (with a step of the 5%) of the LN population.
- *Malicious behaviors* (e.g., alternative, collusive, complainer or noising) are carried out by cheating agents. To make their discovery more difficult for our reputation model, regardless of the type of attack they perform, each of these agents adopt a different strategy. In particular, with respect to the number of time that a cheater has been selected, an attack may occur randomly, or with a frequency (i.e., "malicious:correct") of $1:1$, $1:h/2$ and $1:h$. In detail, the malicious behaviors carried out are:
  - *alternate* - this behavior consists of gaining reputation in low value interactions for then spending it by cheating in presence of interaction having a high value; this type of behavior has been simulated by adopting the rates above specified;
  - *collusive* - this activity happens when two or more devices performs mutual interactions aimed to increase their RCs;
  - *complainer* - a malicious behavior aimed to damage own counterparts by systemically releasing low feedback; also this activity has been simulated with the same rates previously adopted;
  - *noising* - it is a voluntarily interruption of an interaction; it is performed with a rate of 1 each 100 interactions. The presence of the blockchain contributes to hinder the possibility to carry out noising interactions in presence of unreliable consumer devices so that any RC penalization will be given in this instance to the involved actors.

The execution of the first phase of the group formation algorithm allowed us, as the best choice, to set the number of groups of LN to 3 and to adopt as thresholds to be affiliated to them the RC values of 2.5, 4.5 and 6.0, respectively.

B. Results

This section describes the results of the performed simulation campaign. Such results have proved the advantages given in a competitive scenarios by the use of *i*) the reputation capital, *ii*) the proposed group formation strategy and *iii*) a blockchain.

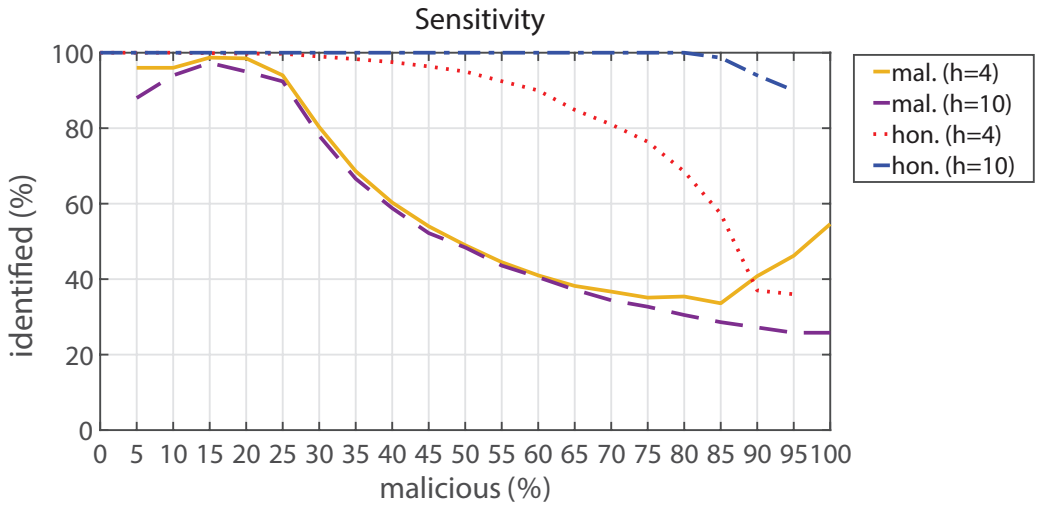

Fig. 2. Sensitivity for $h = 4$ and $h = 10$ n recognizing the nature of the device (i.e., malicious or honest) as the population composition varies.

1) Malicious identification: Firstly, our approach highlighted a high accuracy in recognizing cheaters (only based on the values of their RC scores) as both the horizon and the number of malicious increase. The sensitivity of our model is represented in Figure 2. Results show that the reputation capital, for the considered horizons, works well until the percentage of malicious is less than 25%, while as soon as the percentage of malicious agents is higher than 25%, the performance decreases. This percentage of malicious may appears as a kind of limitation of the presented approach; however, note that in a community, having about 25% of malicious members is unrealistic.

The results related to the capability of recognizing honest agents are quite heterogeneous. Indeed, for $h = 4$ the performance decreases when malicious are more than 50%, while for $h = 10$ this happen only when the malicious are more than 95% of the overall LN population. This led us to define as a bound on the reputation capital, that its capability to recognize malicious agents decreases when they are $\geq 25\%$ of all the LN members and this almost independently from the considered horizon. Therefore, this experiment highlighted that the adoption of the reputation capital provides high accuracy in recognizing the nature of agents and a high resilience to the presence high percentages of cheaters.

2) Group affiliation: To test the effectiveness of the group formation algorithm (see Section VI) another series of experiments have been carried out and the obtained results are depicted in the Figures 3, 4, 5, 6, corresponding to different horizons $h = 4, 6, 8, 10$, and different percentage of malicious, i.e. 5%, 10%, 15%, 20% and 25%. In particular, as above specified by means of the first phase of the group formation algorithm, the number of groups has been set to 3, and their admittance RC thresholds for the agent affiliation were determinate in 2.5, 4.5 and 6.0 (see Section VI).

The experimental results confirmed our expectancy by showing has the group formation algorithm allows a correct dynamic about the distributions of the devices among the groups and permit us to highlight the following considerations:

- the RC of honest agents increases significantly more quickly than malicious ones that, however, never assume high values of RC;
- the malicious devices cannot be affiliated with any group being their RC score lower of the lowest group admittance threshold and, in average, this result becomes stable in few epochs (i.e., about $5 \div 10$ epochs);
- a low horizon implies that honest agents can reach no so high values of RC and, consequently, the number of agents affiliated with the best groups having a higher admittance threshold is generally limited;
- a high horizon leads honest agents to reach high values of RC so that the most part of them is able to belong to the the best groups.

Finally, we observe that, as soon as malicious agents increase in percentage, high horizons become desirable and with an horizon $h = 8$, or greater, all the malicious agents are correctly recognized and do not belong to any group (e.g., it is easy to note that the number of agents not belonging to any groups corresponds to the number of malicious agents).

3) Costs: The results of the third experiments, represented in Figures 7 and 8, highlighted that there are not economic advantages for malicious devices that pay from 1.5 to 4.2 times always more than honest devices. This results are referred to the 25-th epoch.

C. Discussion

The results of the experiments we carried out to test our approach allows us to argue that it is resilient against malicious attacks. Almost all the simulated attacks have been detected in few

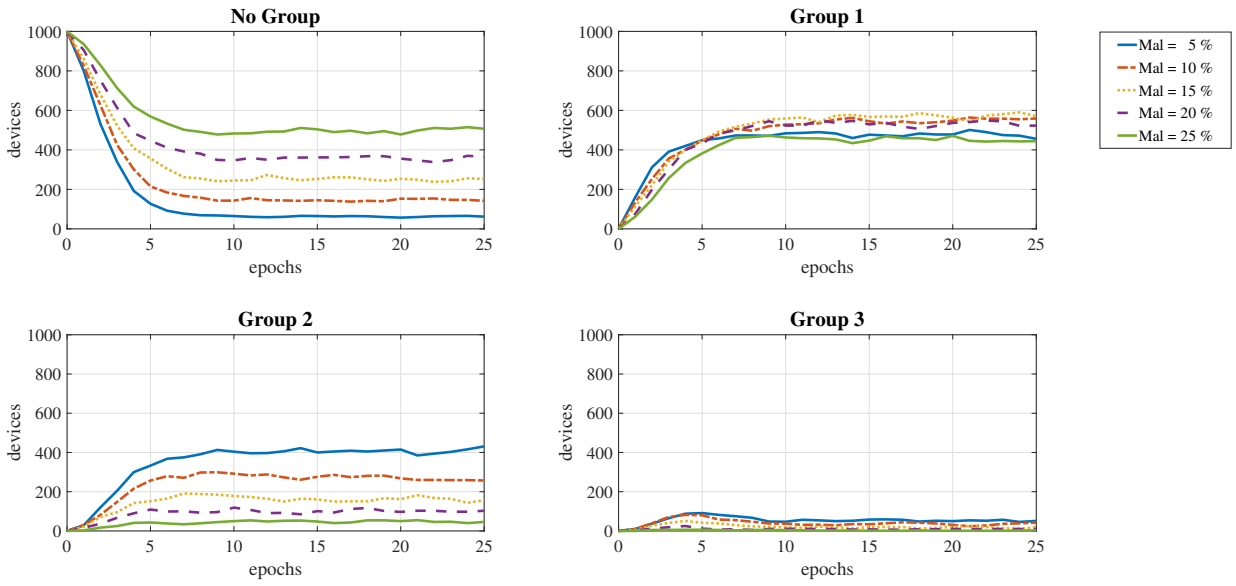

Fig. 3. Group affiliation for $h = 4$ and a presence of the 5%, 10%, 15%, 20% and 25% of malicious devices with respect the overall agent population.

epochs (usually in less than 5) when the percentage of malicious agents is lower than $\sim 25\%$ of all the agents, which is a high enough threshold. Furthermore, the new group formation algorithm works correctly and malicious agents always pay more than honest ones for services. We highlight as such results strongly depend on the adopted horizon, the percentage of cheaters and the number of groups active in LN. However, the benefits in terms of performance and resilience, given by the adoption of the reputation capital, the group formation algorithm and the blockchain, are evident. In particular, the opportunity of dynamically setting the number of groups and their RC admittance thresholds provides high adaptivity with respect to any changes that may occur in the LN population and/or the percentage of malicious agents.

VIII. Conclusions

In this work, we deal with Federations of IoT networks comprises myriad of heterogeneous, smart IoT devices. In our context, devices shift among local networks, and cooperate to attain own targets with their peers.

The level of "satisfaction" of the singol device must be fairly high at the end of interactions, therefore it is important to select reliable collaborators. This is complex problem when device

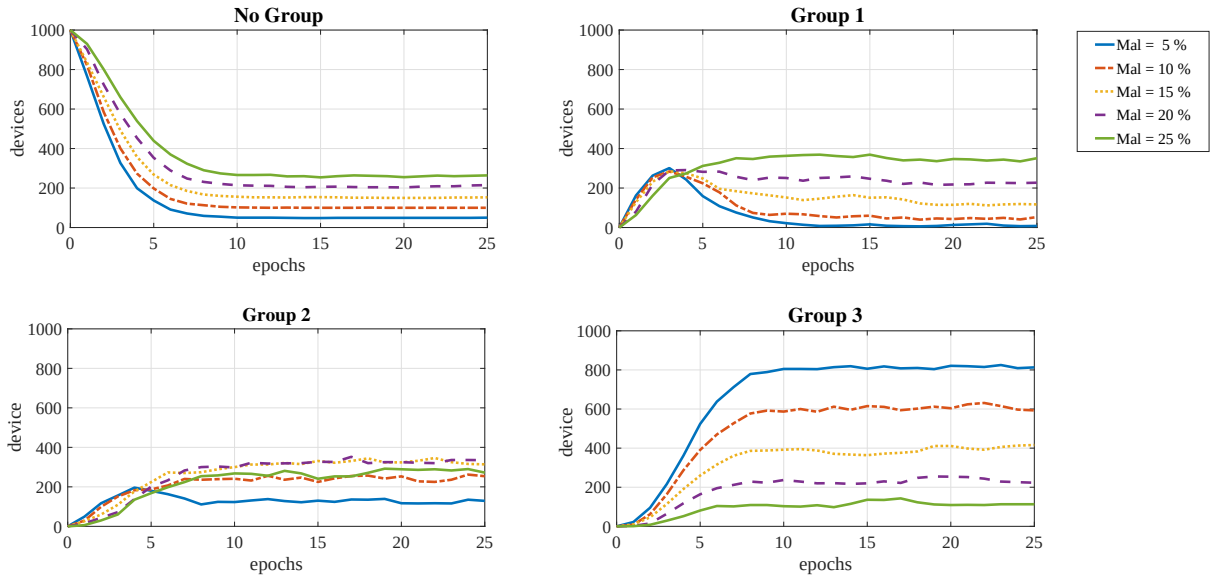

Fig. 4. Group affiliation for $h = 6$ and a presence of the 5%, 10%, 15%, 20% and 25% of malicious devices with respect the overall agent population.

interactions embroil critical and/or complex activities (like, the release of resources). In this case, the reputation is a measure that can help in that choice, if it is appropriately spread. For this reason, we present an approach where a software agent is associated with an IoT device in order to use its social attitudes to collaborate to perform groups. To ensure reliable partnerships, we define the reputation capital, a score that depends on the devices' feedback. In our scenario, we utilize the blockchain technology to propagate the reputation, without the use of centralized component. Also, based on the reputation capital scores, every device can expect satisfactory interactions and economic advantages if it associates with groups of reliable agents.

In detail, we introduce *i*) a appropriate reputation capital model that develops the countermeasures against collusive and malicious behaviors and *ii*) a distributed group formation algorithm that subdivides the agents in groups considering their reputation capital score. Finally, we carry out an experimental campaign to check efficiency and effectiveness of the presented solution. The results highlights how the synergy obtained by the combined adoption of the reputation capital model, the group formation algorithm along with the blockchain provides benefits to the agents operating in the IoT environment. In the future, we will perform an experimental

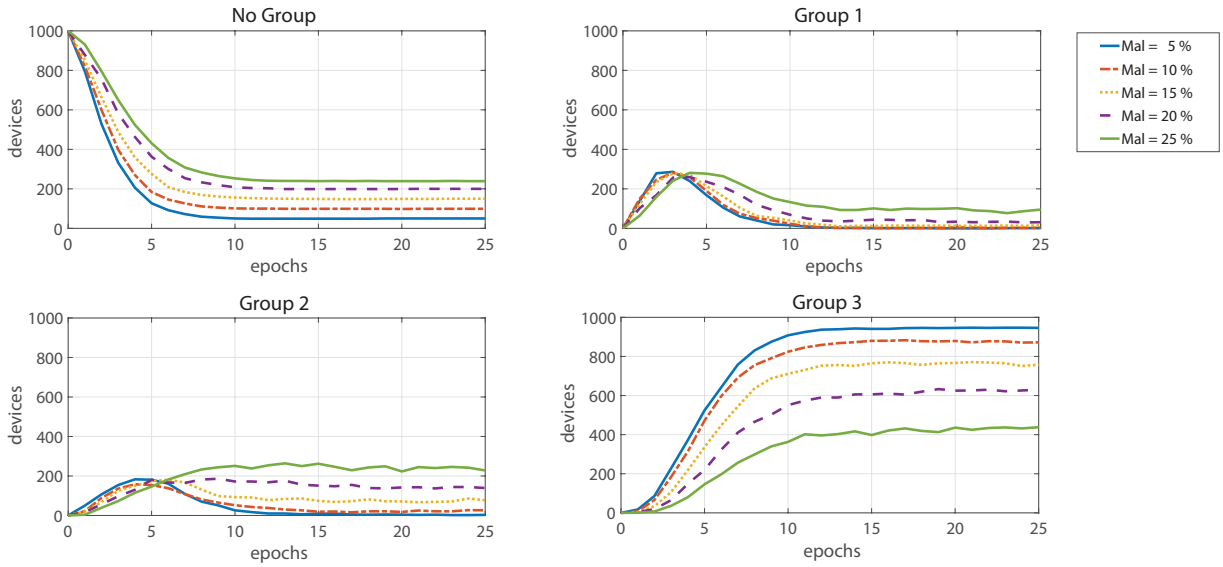

Fig. 5. Group affiliation for $h = 8$ and a presence of the 5%, 10%, 15%, 20% and 25% of malicious devices with respect the overall agent population.

campaign in a real IoT scenario to best prove the benefits inserted by our model.

ACKNOWLEDGEMENT

This work has been partially supported by the University of Catania, Piano per la Ricerca 2016-2018 - Linea Di Intervento 1 "CHANCE" II Edizione - and PIA.CE.RI. 2020-2022 (University of Catania) and by the Network and Complex Systems (**NeCS**) Laboratory at the University Mediterranea of Reggio Calabria, Department Of Civil, Energy and Materials Engineering (DICEAM).

REFERENCES

[1] G. Fortino, F. Messina, D. Rosaci, and G. M. L. Sarné, "Using blockchain for reputation-based cooperation in federated iot domains," in *Intelligent Distributed Computing XIII*, I. Kotenko, C. Badica, V. Desnitsky, D. El Baz, and M. Ivanovic, Eds. Springer International Publishing, 2020, pp. 3–12.

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

[3] G. Fortino, A. Guerrieri, W. Russo, and C. Savaglio, "Middlewares for smart objects and smart environments: overview and comparison," in *Internet of Things Based on Smart Objects*. Springer, 2014, pp. 1–27.

[4] G. Kortuem, F. Kawsar, V. Sundramoorthy, and D. Fitton, "Smart objects as building blocks for the internet of things," *IEEE Internet Computing*, vol. 14, no. 1, pp. 44–51, 2010.

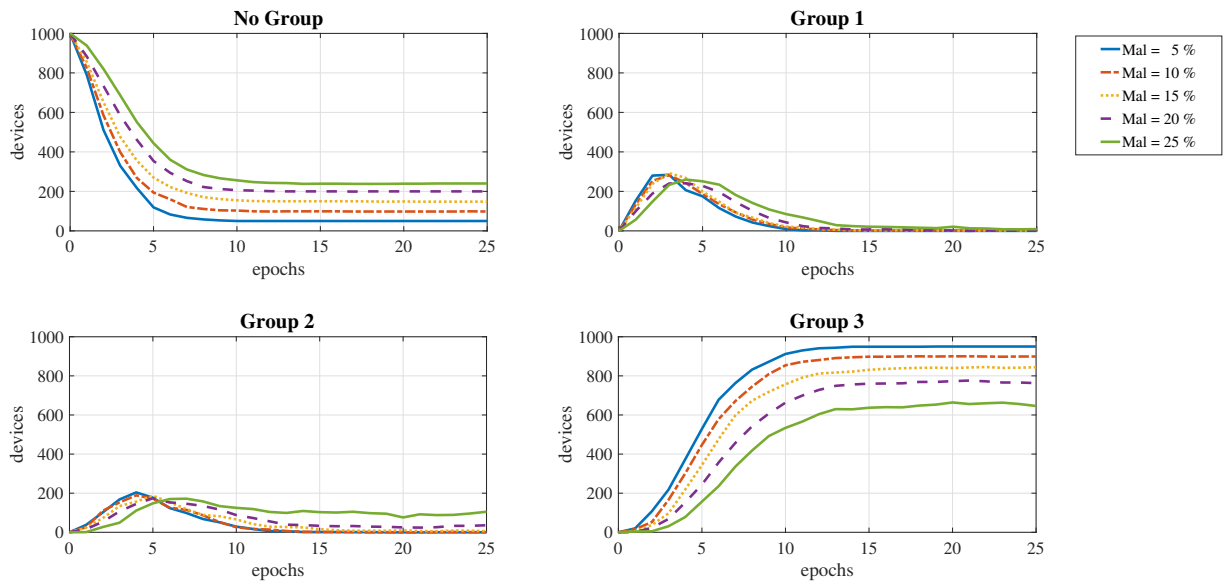

Fig. 6. Group affiliation for $h = 10$ and a presence of the 5%, 10%, 15%, 20% and 25% of malicious devices with respect the overall agent population.

[5] G. Fortino, W. Russo, C. Savaglio, W. Shen, and M. Zhou, "Agent-oriented cooperative smart objects: From iot system design to implementation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems - in press (2018). DOI:10.1109TSMC.2017.2780618*, 2018.

[6] G. F. Coulouris, J. Dollimore, and T. Kindberg, *Distributed systems: concepts and design*. pearson education, 2005.

[7] 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 Transactions on Engineering Management*, vol. july, pp. 1–13, 2019.

[8] R. Roman, J. Zhou, and J. Lopez, "On the features and challenges of security and privacy in distributed internet of things," *Computer Networks*, vol. 57, no. 10, pp. 2266–2279, 2013.

[9] M. G. V. Kumar and U. Ragupathy, "A survey on current key issues and status in cryptography," in *Wireless Communications, Signal Processing and Networking (WiSPNET), Int. Conf. on*. IEEE, 2016, pp. 205–210.

[10] J. Sabater-Mir and M. Paolucci, "On representation and aggregation of social evaluations in computational trust and reputation models," *Int. J. of Approximate Reasoning*, vol. 46, no. 3, pp. 458–483, 2007.

[11] Y. Han, "A survey of trust and reputation management systems in wireless communications," *Proc. of IEEE*, vol. 98, no. 10, pp. 291–298, 2009.

[12] L. Zhu, K. Gai, and M. Li, "Security and privacy issues in internet of things," in *Blockchain Technology in Internet of Things*. Springer, 2019, pp. 29–40.

[13] J. Fogel and E. Nehmad, "Internet social network communities: Risk taking, trust, and privacy concerns," *Computers in Human Behavior*, vol. 25, no. 1, pp. 153–160, 2009.

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

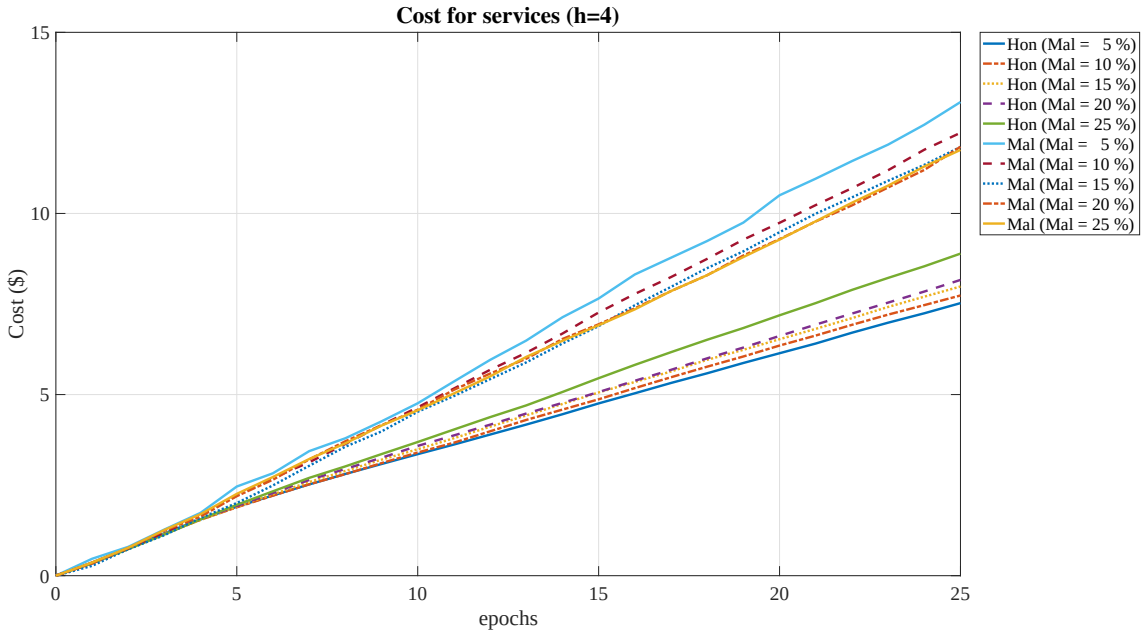

Fig. 7. Cost for services (in $) paid from honest ($h$) and malicious ($m$) devices for $h = 4$ and different malicious percentages (mal)

[15] T. Huynh, N. Jennings, and N. Shadbolt, "An integrated trust and reputation model for open multi-agent system," *Autonmous Agent and Multi Agent Systems*, vol. 13, 2006.

[16] D. Rosaci, G. M. L. Sarnè, and S. Garruzzo, "Integrating trust measures in multiagent systems," *International Journal of Intelligent Systems*, vol. 27, no. 1, pp. 1–15, 2012.

[17] L. Xiong and L. Liu, "Supporting reputation-based trust for peer-to-peer electronic communities," *IEEE Transaction on Knowledge and Data Engineering*, vol. 16, no. 7, pp. 843–857, 2004.

[18] J. Heidemann, M. Klier, and F. Probst, "Online social networks: A survey of a global phenomenon," *Computer Networks*, vol. 56, no. 18, pp. 3866–3878, 2012.

[19] J. Zhan and X. Fang, "Social computing: the state of the art," *Int. J. of Social Computing and Cyber-Physical Systems*, vol. 1, no. 1, pp. 1–12, 2011.

[20] A. Comi, L. Fotia, F. Messina, D. Rosaci, and G. M. L. Sarné, "A partnership-based approach to improve qos on federated computing infrastructures," *Information Sciences*, vol. 367, pp. 246–258, 2016.

[21] T. French, N. Bessis, F. Xhafa, and C. Maple, "Towards a corporate governance trust agent scoring model for collaborative virtual organisations," *Int. J. Grid and Utility Computing*, vol. 2, no. 2, pp. 98–108, 2011.

[22] S. Gächter, B. Herrmann, and C. Thöni, "Trust, voluntary cooperation, and socio-economic background: survey and experimental evidence," *J. Economic Behavior & Organization*, vol. 55, no. 4, pp. 505–531, 2004.

[23] T. Huynh, N. Jennings, and N. Shadbolt, "An integrated trust and reputation model for open multi-agent systems," *Autonomous Agents and Multi-Agent Systems*, vol. 13, no. 2, pp. 119–154, 2006.

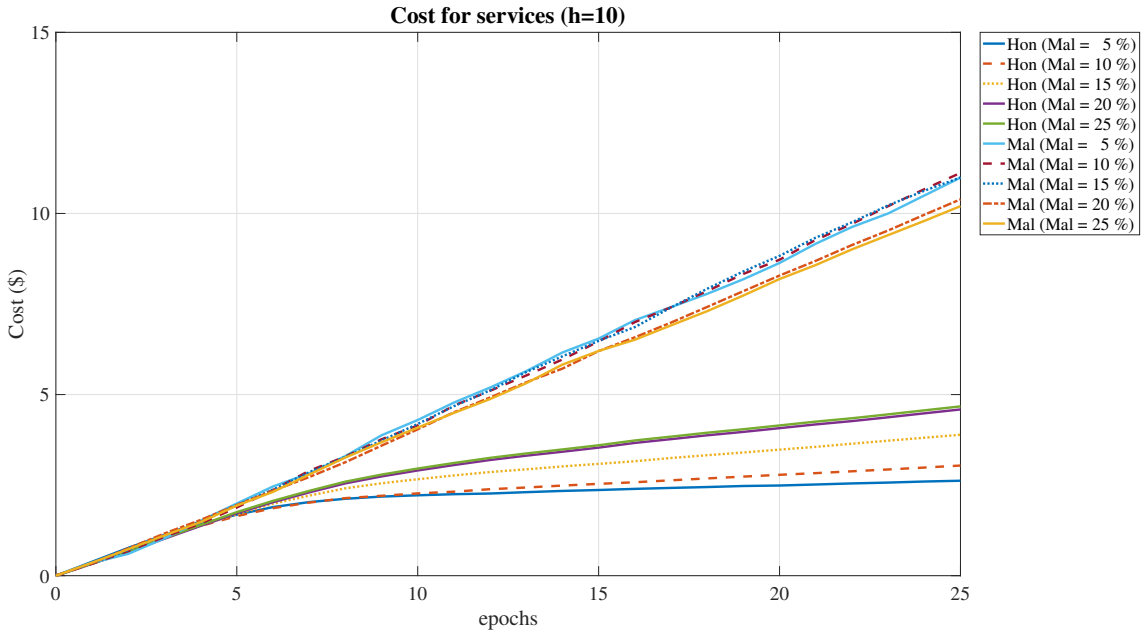

Fig. 8. Cost for services (in $) paid from honest ($h$) and malicious ($m$) devices for $h = 10$ and different malicious percentages (mal)

[24] S. A. Ghasempouri and B. T. Ladani, "Modeling trust and reputation systems in hostile environments," *Future Generation Computer Systems*, vol. 99, pp. 571–592, 2019.

[25] Y. Kim and H. Song, "Strategies for predicting local trust based on trust propagation in social networks," *Knowledge-Based Systems*, vol. 24, no. 8, pp. 1360–1371, 2011.

[26] P. Resnick, R. Zeckhauser, F. E., and K. Kuwabara, "Reputation systems," *Communication of ACM*, vol. 43, no. 12, pp. 45–48, 2000.

[27] M. Momani and S. Challa, "Survey of trust models in different network domains," *arXiv preprint arXiv:1010.0168*, 2010.

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

[29] P. De Meo, E. Ferrara, D. Rosaci, and G. M. L. Sarnè, "Trust and compactness in social network groups," *ACM Transactions on Cybernetics*, vol. 45, no. 2, pp. 205–2016, 2015.

[30] A. Aikebaier, T. Enokido, and M. Takizawa, "Trustworthy group making algorithm in distributed systems," *Human-centric Computing and Information Sciences*, vol. 1, no. 1, p. 6, 2011.

[31] S. Breban and J. Vassileva, "Using inter-agent trust relationships for efficient coalition formation," in *Conference of the Canadian Society for Computational Studies of Intelligence*. Springer, 2002, pp. 221–236.

[32] 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*, p. 102409, 2019.

[33] 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.

[34] F. Bao and I.-R. Chen, "Dynamic trust management for internet of things applications," in *Proc. of the 2012 int. work. on Self-aware internet of things*. ACM, 2012, pp. 1–6.

[35] R. Chen, F. Bao, and J. Guo, "Trust-based service management for social internet of things systems," *IEEE transactions on dependable and secure computing*, vol. 13, no. 6, pp. 684–696, 2016.

[36] 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.

[37] M. D. Alshehri, F. K. Hussain, and O. K. Hussain, "Clustering-driven intelligent trust management methodology for the internet of things (citm-iot)," *Mobile Networks and Applications*, pp. 1–13, 2018.

[38] M. Pilkington, "11 blockchain technology: principles and applications," *Research handbook on digital transformations*, p. 225, 2016.

[39] N. Szabo, "Smart contracts," *Unpublished manuscript*, 1994.

[40] https://www.ethereum.org, 2020.

[41] https://www.hyperledger.org, 2020.

[42] https://ripple.com/, 2020.

[43] https://www.stellar.org, 2020.

[44] https://tendermint.com, 2020.

[45] N. Chalaemwongwan and W. Kurutach, "State of the art and challenges facing consensus protocols on blockchain," in *Information Networking (ICOIN), 2018 International Conference on*. IEEE, 2018, pp. 957–962.

[46] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[47] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *Big Data, 2017 IEEE Int. Congress on*. IEEE, 2017, pp. 557–564.

[48] P. Vasin, "Blackcoin?s proof-of-stake protocol v2," *URL: https://blackcoin. co/blackcoin-pos-protocolv2-whitepaper. pdf*, 2014.

[49] D. Larimer, "Delegated proof-of-stake," *Bitshare whitepaper*, 2014.

[50] V. Buterin and V. Griffith, "Casper the friendly finality gadget," *arXiv preprint arXiv:1710.09437*, 2017.

[51] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *Ieee Access*, vol. 4, pp. 2292–2303, 2016.

[52] P. Otte, M. de Vos, and J. Pouwelse, "Trustchain: A sybil-resistant scalable blockchain," *Future Generation Computer Systems*, 2017.

[53] R. Di Pietro, X. Salleras, M. Signorini, and E. Waisbard, "A blockchain-based trust system for the internet of things," in *Proc. 23nd ACM Symp. Access Control Models and Technologies*. ACM, 2018, pp. 77–83.

[54] R. Falcone and C. Castelfranchi, "The Socio-cognitive Dynamics of Trust: Does Trust Create Trust?." in *Proc. of the Workshop on Deception, Fraud, and Trust in Agent Societies Held During the Autonomous Agents Conf.* London, UK.: Springer-Verlag, 2001, pp. 55–72.

[55] G. Zacharia and P. Maes, "Trust management through reputation mechanisms," *Applied Artificial Intell.*, vol. 14, no. 9, pp. 881–907, 2000.

[56] S. Ramchurn, D. Huynh, and N. Jennings, "Trust in multi-agent systems," *Knowledge Engeenering Review*, vol. 19, no. 1, pp. 1–25, 2004.

[57] M. A. Khan and K. Salah, "Iot security: Review, blockchain solutions, and open challenges," *Future Generation Computer Systems*, vol. 82, pp. 395–411, 2018.

[58] P. J. Taylor, T. Dargahi, A. Dehghantanha, R. M. Parizi, and K.-K. R. Choo, "A systematic literature review of blockchain cyber security," *Digital Communications and Networks*, 2019.

[59] https://www.hyperledger.org/use/fabric, 2020.

[60] https://www.iota.org/, 2020.

[61] L. Rokach, "A survey of clustering algorithms," in *Data mining and knowledge discovery handbook*. Springer, 2009, pp. 269–298.

[62] G. W. Milligan and M. C. Cooper, "An examination of procedures for determining the number of clusters in a data set," *Psychometrika*, vol. 50, no. 2, pp. 159–179, 1985.

[63] A. Gordon, "A survey of constrained classification," *Computational Statistics & Data Analysis*, vol. 21, no. 1, pp. 17–29, 1996.

[64] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics-theory and Methods*, vol. 3, no. 1, pp. 1–27, 1974.

[65] G. Fortino, R. Gravina, W. Russo, and C. Savaglio, "Modeling and simulating internet-of-things systems: A hybrid agent-oriented approach," *Computing in Science & Engineering*, vol. 19, no. 5, pp. 68–76, 2017.

[66] C. Dannen, *Introducing Ethereum and solidity*. Springer, 2017, vol. 1.

[67] https://www.trufflesuite.com, 2020.

Giancarlo Fortino is full Professor of Computer Engineering at the Dept. of Informatics, Modeling, Electronics, and Systems of the University of Calabria (Unical), Italy. He received a Ph.D. in Computer Engineering from Unical, in 1995 and 2000, respectively. He is also adjunct professor at Wuhan University of Technology (Wuhan, China) and senior research fellow at the Italian National Research Council ICAR Institute. His research interests include agent-based computing, wireless (body) sensor networks, and Internet of Things. He is author of over 350 papers in intl journals, conferences and books. 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 MsC Degree in Telecommunication Engineering from the University Mediterranea of Reggio Calabria in 2010 and a PhD in Information Engineering from the University Mediterranea of Reggio Calabria in 2014. Currently she is Post-doc fellow at the University Mediterranea of Reggio Calabria. Her research interests include social network and social internetworking analysis, privacy, security, trust and reputation, intelligent agents.

Fabrizio Messina received his Ph.D. in Computer Science from the Department of Mathematics and Informatics at the University of Catania, Italy in 2009. He is currently serving as assistant professor in the same department. His research interest includes Distributed systems, Complex Networks, Simulation systems, Trust. 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 assistant professor of Computer Science at the Department of Civil, Energy, Environment and Materials Engineering at the University Mediterranea of Reggio Calabria, 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 of Big Data and Cognitive Computing (MDPI). Contact him at sarne@unirc.it.