Trust-based Management in IoT Federations

H. Yahyaoui[1], Z. Maamar[2], M. Alkhafajiy[3], and H. Al-Hamadi[1]

[1] *Computer Science Department, Kuwait University, State of Kuwait*
[2] *College of Technological Innovation, Zayed University, U.A.E*
[3] *School of Computer Science, University of Lincoln, UK*

Abstract

This paper presents a trust-based evolutionary game model for managing Internet-of-Things (IoT) federations. The model adopts trust-based payoff to either reward or penalize things based on the behaviors they expose. The model also resorts to monitoring these behaviors to ensure that the share of untrustworthy things in a federation does not hinder the good functioning of trustworthy things in this federation. The trust scores are obtained using direct experience with things and feedback from other things and are integrated into game strategies. These strategies capture the dynamic nature of federations since the population of trustworthy *versus* untrustworthy things changes over time with the aim of retaining the trustworthy ones. To demonstrate the technical doability of the game strategies along with rewarding/penalizing things, a set of experiments were carried out and results were benchmarked as per the existing literature. The results show a better mitigation of attacks such as bad-mouthing and ballot-stuffing on trustworthy things.

Keywords:
Attack, Evolutionary Game, Federation, Internet-of-Things, Trust.

1. Introduction

It is largely accepted that the chip industry has heavily impacted the Information and Communication Technology (ICT) landscape. Today's chips are tiny, powerful, reliable, and affordable allowing to concretize Marker Weiser's vision about the 21^{st} century computer when he states that "*... The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it*" [31]. In line with this vision, the Internet-of-Things (IoT) is also capitalizing on the chip industry advances by enacting the development of cutting-edge systems like skinmarks (printed skin electronics for interaction) and smart eyeglasses (chewing, light exposure, and daily activity monitoring) [3].

6.4 billion connected things were in use in 2016, up 3% from 2015[1], and will reach more than 30.9 billion devices worldwide by 2025[2]. Are all these things trustworthy? And, can all these things be trusted? Things in the IoT world are not always collaborative, cooperative, and predictable [6]. Their silent and transparent "invasion" into our daily lives could become a nightmare, should confidentiality, integrity, and availability be not taken seriously. In fact, things could be dangerous to people's health, safety, and liberty as stated in [22].

Along with the trust concern and to ensure bet-

[1] www.gartner.com/newsroom/id/3165317
[2] techjury.net/blog/internet-of-things-statistics/#gref.

ter use of things, we suggested in a previous work gathering things into federations [14]. Federations' benefits include fostering things' collective over individual behaviors and enforcing cross-thing collaboration despite their silo nature [19, 20]. To sustain our research efforts into federations, this paper addresses the following trust-related concerns: how to enforce trust among things in federations, how to ensure the "longevity" of federations by detecting and rewarding trustworthy behaviors of things, how to mitigate malicious attacks such as bad-mouthing and ballot-stuffing on things while collecting recommendations about things, and how to avoid "chaos" in federations by isolating untrustworthy things? To address these 3 concerns, we resort to the Evolutionary Game Theory (EGT, [18]) constituting a novel way for running federations as games where each thing (i.e., player) will have a strategy that expresses its trust behavior in compliance with the federation's regulations like how to join, when to leave, and how to be rewarded. From a practicability perspective, things' trust behaviors are associated with payoffs that federations use to reward them when they complete the assigned operations as expected. Otherwise, the payoffs are limited and even revised with the option of ejecting things from the federations. Because things are expected to frequently sign-up in and sign-off from federations, as they see fit (using payoffs), this could put the federations at the risk of instability leading to chaos. This risk is mitigated by tracking the trust behavior of each thing in terms of what it did and what it is doing. This helps monitor federation evolution, which is a key characteristic of the EGT.

Our contributions in this work are, but not limited to, as follows:

- Devise an architecture associated with trust-based management of IoT federations.
- Propose a trust-based evolutionary game model (IoT-EG) for managing IoT federations.
- Adopt payoffs to reward and penalize things based on the trustworthy and untrustworthy behaviors they expose.
- Demonstrate the effectiveness of the trust-based evolutionary game model with respect to other related works against malicious attacks such as bad-mouthing and ballot-stuffing.

The rest of this paper is organized as follows. Section 2 motivates the adoption of thing federation. Section 3 presents some fundamental concepts related to the current work. Section 4 discusses existing works. Section 5 proposes an architecture associated with trust-based management of IoT federations. Section 6 details the proposed trust-based evolutionary game model that underpins the management of IoT federations. Experiments and major results related to the proposed game model are analyzed in Section 7. Finally, future research directions and concluding remarks are discussed in Sections 8 and 9, respectively.

2. Motivations

Thing federation constitutes a good way for fostering collaboration between things despite their nature of being confined into silos [14, 20]. For this purpose, many aspects are to be examined like how do things sign-up in federations, how are things monitored when residing in federations, and how do things sign-off and sometimes get expelled from federations. Unfortunately, there are not, to the best of our knowledge, studies addressing these questions. While managing federations could happen from different perspectives like growth, performance, and competitiveness, this paper focuses on trust. Indeed, a federation would like to attract, reward, and retain trustworthy things since, they could make it competitive and reliable. Contrarily, the same federation would like to avoid, penalize, and expel untrustworthy things since for instance, they could drain its resources.

Federation management is dynamic since for instance, trustworthy things can join and then, misbehave for different reasons to the extent that they become untrustworthy. And, untrustworthy things could adopt a different behavior making them trustworthy in the long-run. To mitigate the impact of things' behaviors on federations' stability, we associate federations with regulations that things should abide to. An example of regulations would be the use

of payoffs based on things' trust behaviors. High payoffs reflect things' acceptable behaviors to federations allowing them to reside longer.

During their stay in a federation, things adjust their trust behaviors according to the situations that they run into and with whom they end-up interacting. For instance, they could deal with trustworthy peers allowing them to complete their works on time. Contrarily, they could be delayed when dealing with untrustworthy peers. Changes in trust behaviors could impact a federation's stability over time with a high-turnover of things, i.e., continuous sign in and sign off. This will be addressed as part of the pay-off strategy that needs to be devised.

3. Background

An overview of the concepts of IoT, federation, and EGT is provided hereafter. For additional details, readers are invited to consult the references included in this overview.

3.1. Internet of things

The extensive literature on IoT (e.g., [1], [4], [9], [16], [23], and [28]) does not help suggest a common definition. For instance, in [4], Barnaghi and Sheth present a set of requirements and challenges linked to IoT. Requirements include quality, latency, trust, availability, reliability, and continuity that should impact efficient access and use of IoT data and services. And, challenges result from today's IoT ecosystems that feature billions of dynamic things that make existing search, discovery, and access techniques and solutions inappropriate for IoT data and services. In [1], Abdmeziem et al. discuss IoT characteristics and enabling technologies. The former include distribution, interoperability, scalability, resource scarcity, and security. And, the latter include sensing, communication, and actuating that are mapped onto a 3-layer IoT architecture referred to as perception, network, and application, respectively.

3.2. Federation of things

Despite the abundant literature on the concept of federation, there is not much, to the best of our knowledge, when it comes to federation of things. In [11], Heil et al. define IoT as a context-aware federation of devices. The objective of setting-up this federation is to support users access, connect, and locate arbitrary devices according to their functionalities. Heil et al.'s approach integrates real-world devices into service federations so, that, devices' capabilities are made available for external use. In [17], Mathlouthi and Ben Saoud discuss cloud federation in the context of enabling a flexible composition of System of Systems (SoS). A SoS is about the cooperation of several constituents that are complex, heterogeneous, autonomous, and independently governed, but capable at the same time of working in a cooperative way to achieve common goals. To allow the exposure of thing federation as a service, Celesti et al. discuss IoT-as-a-Service (IoTaaS) in conjunction with the development that cloud computing is going through and that is leading to IoT cloud and cloud federation [5]. Celesti et al. recommend a 3-layer cloud federation reference-architecture that would meet automation and scalability, interoperable resource provisioning, and interoperable security requirements. In [30], Torroglosa-Garcia and Skarmeta-Gomez discuss the interoperability of identity federation systems. These federations unify and simplify user and service management using trust relationships. However, the large number of federations, each focussing on different areas, necessitates their interoperability to ensure a consistent access across all these federations, and, hence, the same digital identity is used. Finally, in [14], Maamar et al. present Thing-Federation-as-a-Service (TFaaS) as a novel way to address the silo constraint impeding the collaboration of things. A federation is a group of things that are put together in order to handle a particular real-life situation like tunnel closure due to car accident. Maamar et al. specialized federation into planned and ad-hoc. The former is put in place ahead of time and, hence, has its thing members already known with respect to a situation's needs and requirements. And, the latter is put in place on-the-fly when none of the planned federations can handle a situation and, hence, needs to have its thing members identified, selected, and finally assigned. To illustrate the tunnel closure, Maamar et al. suggested

a federation of back-up cameras that is automatically activated so, that, live images from the inside of the tunnel are broadcasted to the rescue teams while meeting their non-functional requirements (e.g., upload speed and resolution quality). To handle the closure, 2 cases were identified:

- 1^{st} time closure: an ad-hoc camera federation is set-up (by some engineers) after selecting the necessary cameras with respect to the rescue teams' non-functional requirements. Once the tunnel closure is over, the ad-hoc camera federation becomes a planned camera federation.
- Recurrent tunnel-closure: a planned camera federation, among those that were initiated in the past, is selected with respect to the rescue teams' non-functional requirements. If the selection is unsuccessful, then the case is treated as 1^{st} time tunnel-closure.

The cameras in a federation whether planned or ad-hoc are expected to deliver the best possible quality of service to rescue teams. A camera that is for instance, hacked would follow an untrustworthy strategy by broadcasting low-quality images that could delay the work of the rescue teams and hence, would be penalized. Contrarily, other cameras would follow a trustworthy strategy and hence, would be rewarded by receiving more bandwidth. By adopting different strategies, a game is put in place requiring that trustworthy *versus* untrustworthy cameras should be identified to ensure the stability and longevity of the federations to which they belong. Our proposed game model in Section 6 achieves this identification.

From a competitiveness perspective, a federation could make things sign-off if their performances (e.g., unreliable data and recurrent failure) do not meet the federation's expectations. A thing could also leave a federation, should the business in the federation become less rewarding (e.g., data-sharing rate among the members drops below a threshold).

3.3. Evolutionary game theory

Classical game theory is a framework used to capture the interactions between agents that would like to increase their own payoffs. The agents adopt strategies that expose their behaviors to the environment in which they reside. The strategies that allow agents to maximize their payoffs without gaining more by unilaterally changing their decisions constitute a situation of Nash Equilibrium [21].

Contrarily, EGT emerged as an extension of the classical game theory to capture biological concepts such as generation and evolution. Agents are members of a certain generation and may survive or not the evolution that a generation would go through. An interesting EGT concept is Evolutionary Stable Strategy (ESS) [27]. It characterizes a strategy which cannot be invaded by any other competing strategy if adopted by a large population of competitors. ESS is considered as a refinement to the Nash Equilibrium.

To model evolution, EGT adopts the concept of replicator dynamics [29], which indicates, that for each strategy k, the evolving proportion (*aka* share) of a population's members that follow this strategy. More formally, let P be a population, $\lambda_k(t)$ be the number of players that adopt strategy k in the set of strategies K, and $x(t) = [x_1(t), \ldots, x_k(t), \ldots, x_K(t)]$ be P's state at time t. Let $x_k(t) = \lambda_k(t)/\lambda(t)$ denote P share of players that adopt k at time t, and $\lambda(t) = \sum_{k=1}^{K} \lambda_k(t)$ denote the total number of players in P. Let also u_{kl} be the payoff of a player playing strategy k along with another playing a different strategy l. The dynamics of P's share $x_k(t)$ is defined by $x'_k(t)$, time derivative of $x_k(t)$, and is defined by $x'_k(t) = x_k(t)(\sum_{l=1}^{K} u_{kl}(t)x_l(t) - \bar{u}(t))$, where $\bar{u}(t) = \sum_{k,l=1}^{K} u_{kl}(t)x_k(t)x_l(t)$ is the average payoff of P. Therefore, the players of a strategy increase/decrease their population's shares if their payoffs are higher/lower than the population's average payoff.

We apply the EGT's main concepts to IoT federations management model where population corresponds to federation of things, players correspond to things, and strategy is specialized into either trustworthy T or untrustworthy U. Furthermore, we resort to the concept of dynamics to monitor the behaviors of things in a federation and ensure that the shares of untrustworthy things (i.e., proportion of things that do not comply with a federation's reg-

ulations) are low. Details about the game model are in Section 6.5.

4. Related work

Several trust models and their derivatives like reputation and credibility are reported in the literature. The focus is on those models of type peer-to-peer in the context of IoT by discussing non-game-based and then, game-based.

Among prominent non-game trust and reputation models we discuss here EigenTrust, Peer-Trust, and IoT-Trust. EigenTrust [12] is a reputation model for peer-to-peer systems that aggregates trust recommendations (indirect trust) weighted by the credibility of recommenders to calculate the trust of a peer. EigenTrust computes a global trust value based on the principal eigenvector of normalized local trust values. The trust is used to reduce the number of downloads of malicious files in a network. EigenTrust assumes the existence of trusted peers that can provide good recommendations to address collusion attacks.

PeerTrust [32] is a dynamic peer-to-peer trust model used to quantify and assess the trustworthiness of peers in e-commerce communities. This trustworthiness reflects the degree of trust that other peers in the community have on a given peer based on past experiences. PeerTrust consists of two parts. The first part is a weighted average of the amount of satisfaction that a peer receives for each transaction while the second part adjusts this average by either increasing or decreasing the trust value based on community-specific characteristics and situations. The weight takes into account the credibility of feedback source to counter dishonest feedback, and transaction context to capture the transaction-dependent characteristics.

IoT-Trust [7] is an adaptive and scalable trust management for the composition of applications in service-oriented architecture-based IoT systems. Feedbacks are gathered based on distributed collaborative filtering and direct and indirect trusts are combined using an adaptive filtering technique. In IoT-Trust, the objectives are to mitigate the effect of malicious nodes on interaction quality and to minimize the number of collusion attacks. IoT-Trust considers a simple case in which the direct user satisfaction experience f is a binary value (1/0 for satisfied/unsatisfied). Then, f is considered as an outcome of a Bernoulli trial that is used as a weight for positive observations.

Compared to the afore-mentioned non-game trust models, our future evolutionary game model will provide an analytical study on the conditions that were satisfied using incentives and the role of these incentives in increasing the share of trustworthy things in a federation. Furthermore, our model will promote node stability that is built on top of the proposed trust model to detect peers' unstable trust behaviors.

Recently, a trust binary-game model that captures the interaction between requestors and providers in an IoT environment is presented in [15]. The model includes nodes' behaviors, possible strategies, and payoffs, and is associated with guidelines for designing trust management algorithms. However, the model does not take into account the evolution of untrustworthy and trustworthy things' shares. Contrarily, our evolutionary game model allows federations to monitor the trust behaviors of things, which leads to a more effective detection of untrustworthy things.

Our trust-based IoT federations management framework puts focus on things' short and long-term trust behaviors and assigns incentives to strategies that things adopt throughout their course of actions. This management also relies on an evolutionary game based trust model that provides a formal setting for proving dominant strategies. It takes into account the dynamism of IoT environment, ensures the stability of things' behaviors, and mitigates the effects of malicious attacks such as bad-mouthing and ballot-stuffing. These features altogether are barely touched upon, if not overlooked in most existing trust management models as pinpointed by Sharma et al. [26]. Table 1 compares our trust model to existing ones using five factors detailed below. This comparison's objective is to shed light on the capabilities of our model by integrating different aspects like with whom a thing interacts, how trust evolves over time, and who recommends a thing.

- Direct trust relies on a thing's one-to-one interactions with peers to define its trustworthiness level.
- Indirect trust proceeds differently to direct trust by relying on peers' recommendations about a thing.
- Trust evolution tracks changes of a thing's trustworthiness level over time, which is normal due to the dynamic nature of federations.
- Stability analysis is in line with trust evolution but focuses on the impact of trust changes on a federation's stability over time.
- Provability provides a formal basis for finding the conditions to satisfy prior to reaching equilibrium states.

5. IoT federations management architecture

Fig. 1 is the architecture associated with our Trust-based Management of IoT Federations (TMIF). It is built-upon three layers, thing, federation, and behavior, hosting different modules and repositories.

The thing layer is concerned with defining things' profiles, tracking their ongoing operations, and forming their histories. In this layer, the *profile* module gathers all the necessary details about the operational aspects of a thing such as functionality it offers (e.g., sensing, actuating, and communicating) and quality of performing this functionality. These details are obtained from things' providers as well as thing monitoring. The history of a thing is stored in a *log* that the *history* module updates based on details it receives from the *access* module in the federation layer. History could refer to things' sign-in/out activities in/from federations along with trust-based observations that define things' behaviors. A thing could join a federation at start-up time or re-join others after being expelled, should the federation decide that the thing exposes an unstable behavior due to continued swing from poor to good performance during a period of time. To avoid behavior state explosion, we assume in this work that a thing can join one federation at a time. The thing layer is important for the federation layer since things' profiles and histories are deemed required during sign in. Moreover, the thing layer may serve as a basis for a federation to constitute a list of "friends" that could provide recommendations about unknown things considering to join.

The federation layer is concerned with assembling and dismantling federations along with ensuring the stability of some. In this layer, the *access* module allows things to sign-in/out in/from federations according to these federations' regulations that are defined by the federation's administrator. In addition to the *access* module, the *trust assessment* module collects non-functional details about things, referred to as Quality-of-Thing (QoT) parameters in [24], and recommendations about things so, that, it defines a thing's trust score. QoT parameters reflect how well things did when they took part in federations while recommendations are submitted by things that already interacted with the recommended things. Still in the *trust assessment* module, it periodically feeds the upper layer's *behavior analysis* module with things' trust scores. The *access* module receives entry requests to federations from things located in the lower layer. It decides on either accepting or rejecting each request based on parameters related to a federation such as capacity, which depends on the current number of residing things, stability, which reflects the state of a federation based on the trust behaviors' trends of the majority of its residing things, and history of a thing, which is collected from the lower layer's *history* module. For this purpose, the *access* module coordinates with both the lower layer's *history* module and the upper layer's *stability analysis* module to allow/deny things to join/from joininga federation.

Finally, the behavior layer is concerned with assessing things' trust behaviors and analyzing the stability of federations. On the one hand, this layer's *behavior analysis* module decides on the payoffs that things deserve after continuously collecting trust scores of things from the federation layer's *trust*

Table 1: Comparison between our IoT-EG and some existing trust models

Model	Direct Trust	Indirect Trust	Trust Evolution	Stability Analysis	Provability Provability
EigenTrust [12]	√	√			
PeerTrust [32]	√	√			
IoT-Trust [7]	√	√			
Trust Binary Game [15]	√	√			√
Our IoT-EG	√	√	√	√	√

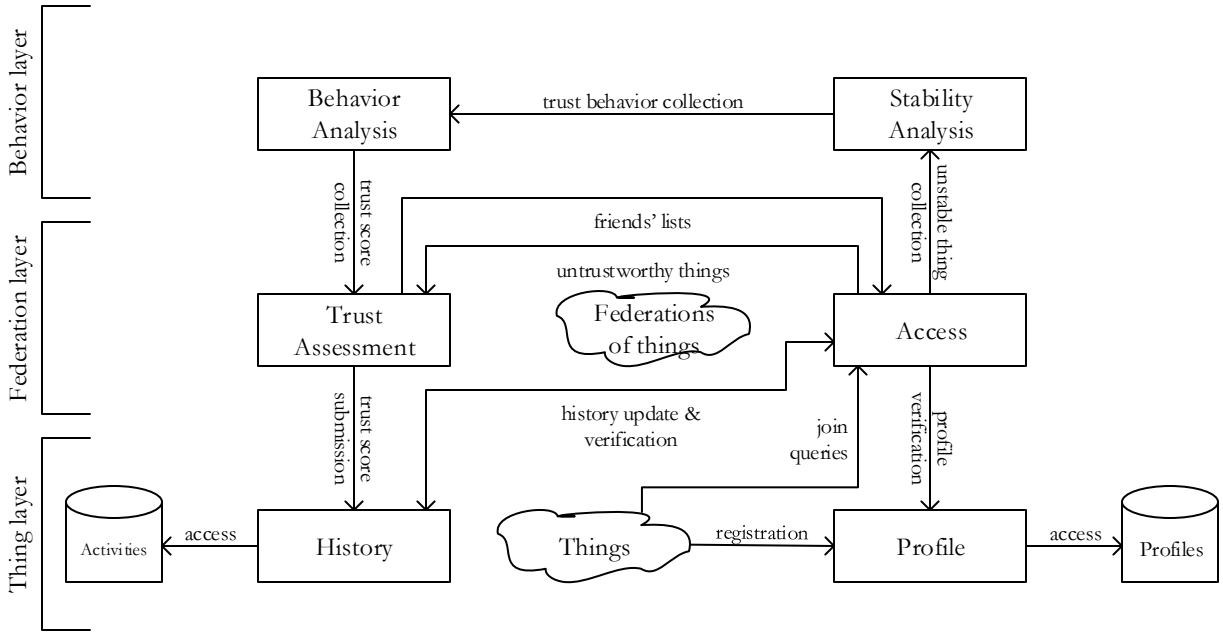


Figure 1: Three-Level representation of the TMIF architecture

assessment module. On the other hand, this layer's *stability analysis* module communicates with the federation layer's *access* module to either allow or forbid federations from accepting new things, should this module reveal that things expose stable/unstable behaviors. Assessing behavior stability is based on analyzing long-term things' behaviors.

6. IoT Trust-based evolutionary game model

Our IoT Evolutionary Game model (IoT-EG), underpinning the functioning of thing federations, mixes short- and long-term trust behavior assessment of things. In this section, the modules involved in this assessment are presented.

6.1. Trust assessment

In Fig. 1, the *trust assessment* module establishes a thing's trust score that has a limited time-span and depends on the quality of collaboration that the thing would have had with peers. Formally, the trust score Tr_i^j of thing i with respect to thing j combines the trust value, the direct-trust value (direct interaction between i and j), and the indirect-trust value (recommendations obtained from the friends of i about j). Equation 1 defines $Tr_i^j(t)$ at time t where $Tr_i^j(t-1)$ is the trust score of thing i with respect to thing j at time $t-1$, $\tilde{QoT}_i^j(t)$ is thing i's weighted average of QoT parameter values after interacting with thing j at time t, $\tilde{R}_i^j(t)$ is the indirect trust as a recommendation value from the friends of thing i about thing j at time t, and λ_1, λ_2 and λ_3 are fine-tuning parameters set by the federation's owner (values in [0,1] such that their sum is 1) that thing i sets to define the weights of direct and indirect trust.

$$Tr_i^j(t) = \lambda_1\ Tr_i^j(t-1) + \lambda_2\ \tilde{QoT}_i^j(t) + \lambda_3\ \tilde{R}_i^j(t) \quad (1)$$

To lighten the notation, the time symbol in defining direct and indirect trust is omitted. Equation 2 defines $\tilde{QoT}_i^j$ where $QoT_i^j(k)$ denotes the value of the k^{th} QoT parameter (among n parameters) for thing i and W_k is the weight of this parameter (value in $[0,1]$).

$$\tilde{QoT}_i^j = \sum_{k=1}^{n} W_k\ QoT_i^j(k) \quad (2)$$

Equation 3 defines the indirect trust where Cr_k^i is the credibility of the k^{th} friend of thing i (value in $[0,1]$) and $R_i^j(k)$ is the rating of thing j by the k^{th} friend among m friends of thing i. Credibility reflects to what extent a thing is accurate in its recommendation and is updated by the *trust assessment* module. It can increase or decrease after assessing how far the recommendation is from the direct trust as per Equation 4. This change in value aims at mitigating the effect of inflated/deflated recommendations when assessing trust scores of things. In this equation, ϵ is a reward/penalty that increases/decreases the credibility depending on the quality of the recommendation.

$$\tilde{R}_i^j = \frac{\sum_{k=1}^{m} Cr_k^i\ R_i^j(k)}{\sum_{k=1}^{m} Cr_k^i} \quad (3)$$

$$Cr_k^i \quad = \quad \begin{cases} Cr_k^i + \epsilon, & \text{if } |R_i^j - QoT_i^j| \leq \epsilon; \\ Cr_k^i - \epsilon, & \text{Otherwise} \end{cases} \quad (4)$$

6.2. Behavior analysis

The *behavior analysis* module performs *short* and *long*-term trust behavior assessment of things in a federation. For short-term trust behavior analysis and after assessing the trust of thing i based on its interactions with thing j, the *behavior analysis* module decides about the strategy that thing i has followed: if Tr_i^j is higher than a federation's trust threshold ξ_{Be} that the federation's administrator sets, then thing i is trustworthy in the interaction it had with thing j. As a result, the *behavior analysis* module announces that thing i has adopted a trustworthy strategy T, otherwise, an untrustworthy strategy U. Establishing a strategy adoption is of paramount importance to determine a thing's payoff as it will be explained later.

The *behavior analysis* module also defines the long-term trust behavior of a thing by collecting its trust

scores. First, it computes the cumulative trust score of thing i in a federation F based on its pairwise trust scores using Equation 5.

$$Tr_i(t) = \frac{\sum_{k \in F} Tr_i^k(t)}{k} \quad (5)$$

As per Equation 6, a trust behavior B_i of thing i is a time series that represents the sequence of trust scores of this thing at different time instances (1 to n).

$$B_i = (Tr_i(1), Tr_i(2), \ldots, Tr_i(n)) \quad (6)$$

As per Equation 7, the behavior of thing i is assessed based on the moving average of its behavior, which denotes the arithmetic mean of the last n collected trust scores.

$$MA_i^t = \frac{\sum_{k=t-n}^{t-1} Tr_i(k)}{n} \quad (7)$$

The *behavior analysis* module would deem the behavior of thing i satisfactory at time t, should MA_i^t be greater than the maximum between the moving average of all things in a federation $\overline{MA_i^t}$ and the federation's trust threshold ξ_{Be}. Should a thing's behavior be not satisfactory, the thing would be expelled from the federation. The main intent of long-term behavior analysis is to "force" a thing to adopt a behavior that is similar to those in the federation and at the same time fulfilling the minimum trust score requirement.

6.3. Stability analysis

In Fig. 1, the *stability* module determines to what extent thing i maintains a consistent behavior compared to other things in a federation F. Equation 8 defines this stability when the standard deviation of its behavior $stdev(B_i)$ is less than a threshold ξ_{ST}.

$$St(i) = stdev(B_i) = \sqrt{\frac{\sum_{k=1}^{n} (Tr_i(k) - \overline{Tr_i(k)})}{n-1}} \quad (8)$$

If the number of things in a federation having unstable behaviors is higher than a stability threshold that is set by a federation's administrator, the federation is deemed unstable and should be dismantled. In conjunction with federation instability, if a thing has an unstable behavior at a certain time instant, it is expelled from the federation.

6.4. Threat Model

In the threat model, things are expected to be malicious by engaging in self-interest operations like overusing resources and delaying the release of services.

While attacks on resources such as Distributed Denial of Service (DDoS) disrupt and/or overwhelm network operations, they can be handled by intrusion detection techniques and do not fall into the scope of this paper. Contrarily, we address trust-related attacks that can disrupt services that a federation makes available to things. Those things that initiate such attacks are identified by assessing their trustworthiness, i.e, calculating the trust score periodically which should help penalize the untrustworthy ones. This is taken care by the trust assessment module (Section 6.1) and payoff strategy (Section 6.5). The main objective is to address attacks like those discussed below:

- Bad-mouthing refers to a recommender (or a group of recommenders) that rates a thing lower than what it deserves for the sake of tarnishing its reputation. Penalizing such a recommender, by reducing its credibility, would warn the recommender against repeating such a behavior and eventually its future recommendations would not be considered, should its credibility become below a threshold ξ_{Cr}.
- Ballot-stuffing refers to a recommender (or a group of recommenders) that inflates a thing's rating for the sake of boosting its reputation and making it more competitive, for example. Penalizing such a recommender by decreasing its credibility would also mitigate the effect of this attack.

6.5. Trust-based EGT model

Federations provide incentives to things so they trustfully collaborate. We endow each federation with an EGT model (IoT-EG) in which incentives depend on how trustworthy a thing is. Such incentives are part of the payoff of each thing and can

increase or decrease depending on the strategy that the thing adopts.

6.5.1. Game strategies

When interacting with thing k, thing i can adopt either a trustworthy strategy T or an untrustworthy strategy U.

The *behavior analysis* module determines the behavior of a thing in a pairwise interaction. Then, the module provides each thing the reward that it deserves based on a payoff matrix. Table 2 outlines a thing's payoff matrix for the proposed evolutionary game where β is a charging incentive parameter for a thing that is following strategy T, γ is the interaction benefit, and σ is interaction cost. It is assumed that β, γ, and σ are strictly positive.

The specification of IoT-EG trust assessment is reported in Algorithm 1. It elaborates the process of computing a thing's trustworthiness in a federation. It is worth to note that the term 'node' will be used interchangeably with 'thing' in the pseudo-code of the algorithm. This algorithm aims to assess the node's trustworthiness during the federation periodically upon an interaction. Therefore, to compute the trust score Tr (e.g., $Tr_i^j(t)$ for $node_i$ toward $node_j$) (algorithm result), the input parameters will includes list of federated things, ξ_{Be}, *Payoff* matrix, QoT, R, Cr, λ, and Pro_t.

Considering a scenario where a set of things are federated to play a game, a thing's trustworthiness will be assessed after each interaction using Equation 1 and following steps 1 and 2 in the algorithm:

- Step 1 implements game interactions: each thing interacts with every single thing in the federation that has a trust score Tr above the federation's trust threshold ξ_{Be} (lines 4-8). It is worth noting that Tr of all things has been boosted in the first iteration since no previous Tr record is computed. The interaction between 2 nodes is evaluated based on a thing's processing time to tasks assigned to each other. Each node passes its assigned tasks to its peers (things) to process them and logs the time it took to process these tasks. The output of this step (i.e., $list_T$ (line 13)) will feed into the next step.
- Step 2 implements trust-score computation: it collects the recommendations from trustworthy nodes, only, (i.e., things) in the federation, i.e., those that have a trust score above the federation trust threshold ξ_{Be} (lines 16-22). Finally, the trustworthiness will be computed using Equation 1 after filtering the recommendation and the output from Step 1 by the weights (i.e., λ) as per lines 25-30. The trust score at line 31 will be the final output of the algorithm.

The payoff is a good means for a federation F to encourage things to be trustworthy. Should that payoff be lower than a minimum value set by the federation's administrator P_F, then the thing would be excluded from the federation.

6.5.2. Game analysis

We discuss hereafter the constraints that should be satisfied to guarantee that a trustworthy strategy prevails over time.

- A strategy T is ESS in 2 cases. The first case is when $2\beta+2\gamma-2\sigma > -\beta-\gamma+\sigma$, i.e., $\beta > \sigma-\gamma$. Any federation, looking to ensure that the strategy T is followed by the majority of its things' members, should be able to provide incentives that should cover the interaction cost based on the following stability constraint: $\beta > \sigma - \gamma$. The second case is when $\beta+\gamma-\sigma = 0$ and $\beta+\gamma-\sigma > -\beta-2\gamma+2\sigma$, i.e., $2\beta+3\gamma-3\sigma > 0$. This means that $\gamma > \sigma$ and that the interaction benefit should be higher than the interaction cost.
- A strategy U risk dominates strategy T iff $2\beta+2\gamma-2\sigma+\beta+\gamma-\sigma < -\beta-\gamma+\sigma-\beta-2\gamma+2\sigma$, i.e., $5\beta+6\gamma-6\sigma < 0$, which means that $\beta < \frac{6}{5}(\sigma-\gamma)$. If a federation provides a charging incentive that is lower than $\frac{6}{5}(\sigma-\gamma)$, the the things may follow strategy U.
- The asymptotically unstable symmetric Nash Equilibrium exists iff 3β + 3γ - $3\sigma \neq 2\beta$ + 3γ - 3σ, i.e, $\beta \neq 0$. The population rate x^* playing risk-dominant strategy U in this case is $\frac{1}{\frac{3\beta+3\gamma-3\sigma}{-2\beta-3\gamma+3\sigma}+1}$, i.e, $x^* = \frac{-2\beta-3\gamma+3\sigma}{\beta}$.

Algorithm 1: IoT-EG trust assessment algorithm

Input: *List of federated things* [$node_i$, $node_j$, .. ,$node_n$];
Result: Computes trust score $Tr_i^j(t)$ for $node_i$ toward $node_j$
Parameters : $federationThreshold$ (ξ_{Be}); $Payoff$; QoT; R; Cr; λ; $ThingsList$ ($list_T$); $ProcessingTime$ (Pro_t); $ThingTask$ $node_{n_t}$

Step 1: *Game interactions* **by**
  $list_T = list[node_{i,..n}] \longleftarrow getFederatedThings$[out $(node_n, QoT)$];
  $list_T = sort(list_T$, by *proximity* ASC);
  for *each* $node_n \in list_T$ **do**
    $node_{n_t} = getTasks(node_n)$;
    if $list_{T^{end}_{node_{n+1}}} \longrightarrow QoT >= \xi_{Be}$ **then**
      $node_n \xrightarrow[node_{n_t}]{interact} list_{T^{end}_{node_{n+1}}}$ (in $node_{n_t}$; out Pro_t) ;
      $node_n$ log (Pro_t) ; ▷ `log` Pro_t `against processed node`
    else
      $list_T = pop(node_{n+1})$; ▷ `remove untrusted node`
    end
  end
  return $list_T$;
End
Step 2: *trustworthiness computation* **by**
  $list_T = sort(list_T$, by QoT DESC);
  for *each* $node_n \in list_T$ **do**
    if $node_n \longrightarrow Tr < \xi_{Be}$ **then**
      $list_T = pop(F_n)$; ▷ `remove untrusted node`
    else
      get R from the trusted node;
      $list_T = list[node_{n_i \longrightarrow n_j)}, R]$; ▷ `new list with R`
    end
  end
  $Tr_i^j(t) = \lambda_1\ Tr_i^j(t-1) + \lambda_2\ \tilde{QoT}_i^j(t) + \lambda_3\ \tilde{R}_i^j(t)$; ▷ `compute trust`
  if $|R_i^j - QoT_i^j| \leq 0.1$ **then**
    $Cr_k^i = min(Cr_k^i + 0.1, 1)$
  else
    $Cr_k^i = max(Cr_k^i - 0.1, 0)$
  end
  return $Tr_i^j(t)$;
End

Table 2: EGT payoff matrix

$$P = \begin{array}{cc} & \begin{array}{cc} T & U \end{array} \\ \begin{array}{c} T \\ U \end{array} & \begin{array}{cc} 2\beta+2\gamma-2\sigma, 2\beta+2\gamma-2\sigma & \beta+\gamma-\sigma, -\beta-\gamma+\sigma \\ -\beta-\gamma+\sigma, \beta+\gamma-\sigma & -\beta-2\gamma+2\sigma, -\beta-2\gamma+2\sigma \end{array} \end{array}$$

6.5.3. Example

Let us consider the case where $\sigma = 20$ and $\gamma = 15$. Table 3 is the payoff matrix.

In this case, if $\beta > \sigma - \gamma$, i.e., $\beta > 5$, the strategy T is ESS. The strategy U risk dominates the strategy T if $\beta < \frac{6}{5}(\sigma - \gamma)$, i.e., $\beta < 6$. So, to lower the risk of having things adopting the strategy U, a federation should provide a charging incentive higher or equal to 6 in this particular setting. Although increasing β would encourage things to follow the strategy T, things may follow the risk dominant strategy U due to problems of strategies selection speed and also depending on the initial distribution, i.e., the initial rate of players playing T *versus* U, as shown in [10, 13, 25]. More precisely, the risk-dominant U has a larger basin of attraction with a rate equal to $\frac{-2\beta+15}{\beta}$ in this special setting.

To mitigate the impact of playing the risk-dominant strategy U by the majority of players, whatever is the initial distribution or the strategies selection speed, a federation should kick-out any thing that has a payoff that is lower than a threshold.

7. Experiments and results

In this section, we discuss the development of the TMIF along with the experiments that were conducted to verify the doability of the IoT-EG model. MATLAB (2020a) running on a Dynabook laptop with Intel Core *i*5-8250U processor and 8 GB of RAM was used.

7.1. Simulation setting

The federation and network elements used during the simulation are presented in Table 4 in terms of network topology, nodes' capabilities, and node interactions/federation.

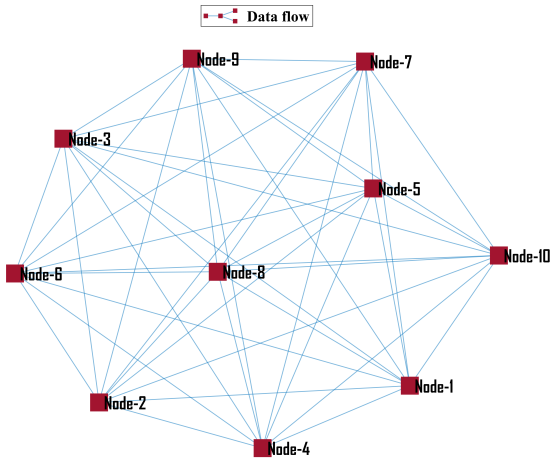


Figure 2: Network graph

- Network topology was modeled as an indirect graph where nodes formed a mesh network as per Fig. 2 (the first 10 nodes in action). The simulation involved 100 nodes joining a federation at the same time. The links between nodes are weighted based on the propagation delay among nodes. Neighbouring nodes have the lowest propagation delay.
- Nodes' capabilities correspond to CPU frequency-based processing powers that during federation will vary from one node to another, 0.3GHz to 1.3GHz [8].
- Node interactions/federation: upon joining a federation, the nodes interact with each other based on the tasks assigned by the federation ($5x10^5$ tasks as specified in Table 4). Each node passes its assigned tasks to its peers to process them and logs the taken time (i.e., processing time) against its processing capability. Regarding the transmission rate between nodes, it is

Table 3: Example of an EGT payoff matrix

$$P = \begin{array}{cc} & \begin{array}{cc} T & U \end{array} \\ \begin{array}{c} T \\ U \end{array} & \begin{array}{cc} 2\beta - 10, 2\beta - 10 & \beta - 5, -\beta + 5 \\ -\beta + 5, \beta - 5 & -\beta + 10, -\beta + 10 \end{array} \end{array}$$

Table 4: Simulation setting

Parameter	Value
Operating system	Win 10
Simulation environment	Matlab 2010a
Number of nodes	100
Node CPU	$[0.3 - 1.3]$GHz
Network topology	mesh
Tasks in each federation	5x10^5
Packet size	$[0.1 - 80]$KB
Federation credibility threshold (ξ_{Cr})	0.65
Federation trust threshold (ξ_{Be})	0.65
Trust parameters $\lambda_{1,2,3}$	λ_1=0.25; λ_2=0.50; λ_3=0.25
Game Payoff β, γ, and σ	β = 6; γ=15; σ=20

high ($\simeq$ 100 Mbps [2]). The logged processing time will be compared to the expected processing time (based on the node's CPU) to determine the trust satisfaction. The bigger the difference, the less trusted the node is, and *vice-versa*. Therefore, after each interaction a node's trust score is reevaluated (using Equation 1) to identify interacted nodes trustworthiness. It is worth noting that the federation reward (i.e., payoff) affects the nodes' behaviors and subsequently their trustworthiness.

7.2. Results and discussion

We first evaluate the performance of the IoT-EG against PeerTrust and IoT-Trust trust assessment algorithms. Fig. 3 demonstrates the performance based on the normalized trust score that is evaluated after each interaction based on the average processing time to all received tasks; considering different packets size (Table 4). However, the number of packets and their sizes are fixed throughout the experiments to ensure consistency across all algorithms.

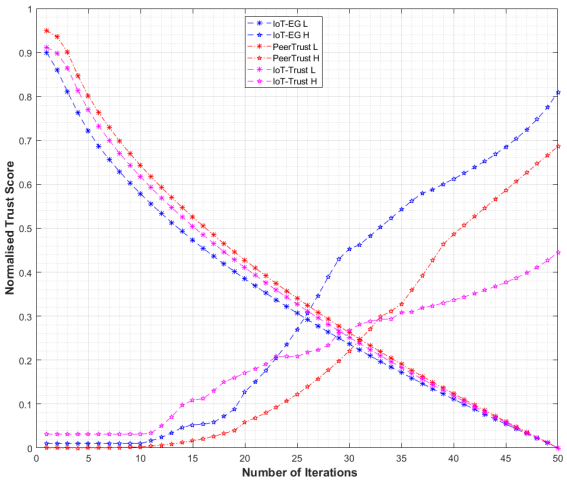
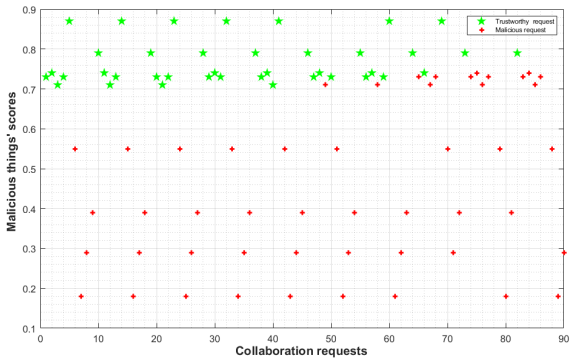


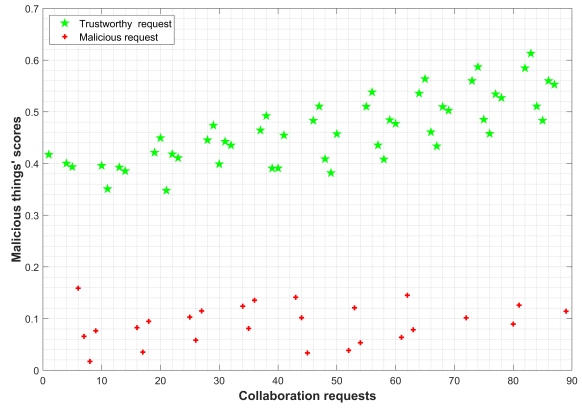
Figure 3: IoT-EG average trust score against PeerTrust and IoT-Trust in terms of high and low payoff

During the experiments, we set the things with different capabilities varying their CPU and processing powers. In Fig. 3 the vertical axis represents the normalized trust score of things while the horizon-

tal axis is the number of iterations in which federations are formed and all algorithms are tested with high and low payoff. The results have generally the same trend. However, it is clear that IoT-EG outperforms the benchmark algorithms. More precisely, IoT-EG penalized more with low payoff compared to other algorithms in identifying untrustworthy nodes. PeerTrust performs better than IoT-Trust in both low and high payoff but still performs less than IoT-EG.



(a) Low payoff



(b) High payoff

Figure 4: Trend of trustworthy requests and malicious request according to the payoff

Fig. 4 shows the results of malicious and trustworthy events based on federation payoff. A malicious event is defined as when a thing takes longer than it should be (considering node's CPU and packets size from Table 4) to process assigned tasks.

In this figure, the number of tasks is set to $1K$ and we had 2 kinds of iterations; the first with high payoff (Fig. 4a) while the second with low payoff (Fig. 4b). The collaboration requests in both figures are grouped according to task outcomes, i.e., whether they are trustworthy (tasks are processed according to node capacity) or malicious (tasks are not processed according to node capacity/delayed) requests. It is clear that with low payoff, the number of malicious events increase as there are not enough incentives for nodes to follow a trustworthy strategy.

Moreover, we have conducted more experiments to monitor the rate of strategies T and U followers over the federation time. Fig. 8 shows the results of the percentages of T and U followers during the game play. It is clear that the payoff, according to low and high incentives, is impacting the percentages of either followers (i.e., T and U) at each timestamp. Hence, the better the payoff, the more T followers and the more successful interactions are within the federation.

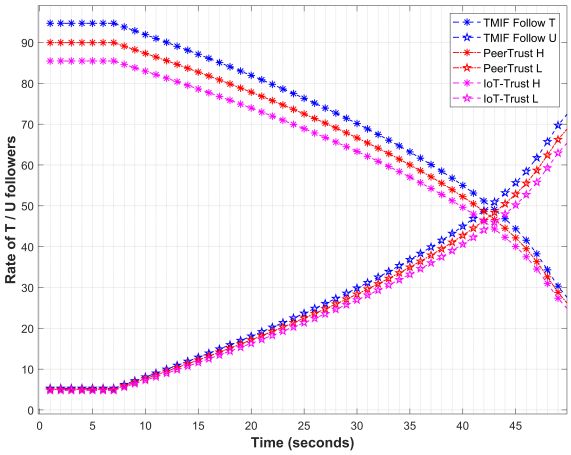


Figure 5: Percentage of node following strategies T and U

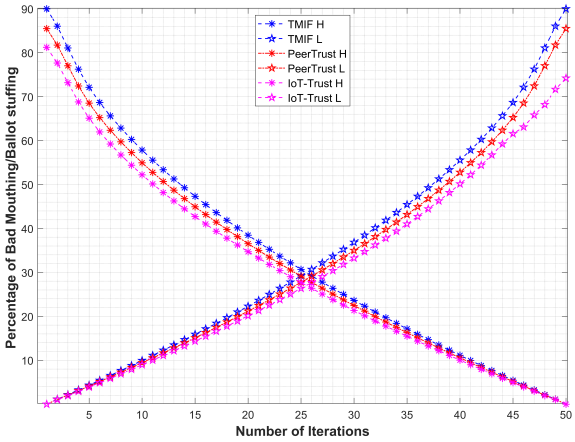


Figure 6: Node's credibility evolution

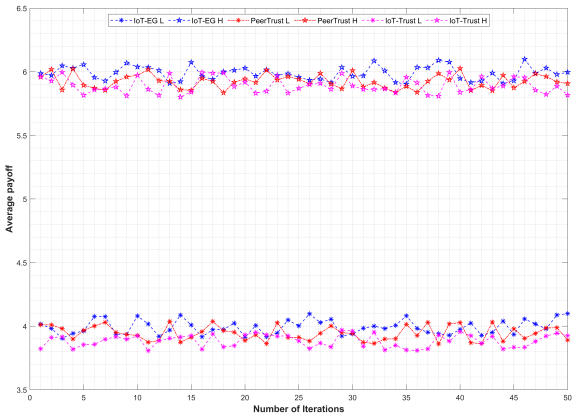


Figure 8: Average payoff in federations

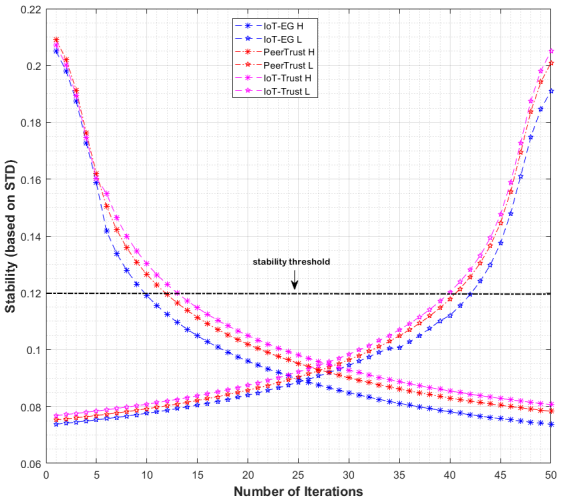


Figure 7: Stable things according to the payoff

The different types of collaboration requests (i.e., untrustworthy and trustworthy requests) will gradually affect the federation coherence and the decision of whether a collaboration request can be accepted or rejected between 2 things. Fig. 6 shows the percentage of bad-mouthing and ballot-stuffing nodes according to federation payoff. We ran this experiment twice: first with an initial percentage of untrustworthy nodes in the federation that is high (i.e., more than half of the nodes in the federation are trustworthy) but in which the incentive is low while in the second run the initial percentage of untrustworthy nodes in the federation is low (i.e., less than half of the nodes in the federation are trustworthy) but the incentive is high. From observing the results, it is clear that the incentive will affect the nodes' decisions regarding which strategy to follow. Therefore, with high incentives, the nodes tend to follow a trustworthy strategy while with low incentives, nodes tend to follow untrustworthy strategy. Fig. 7 expresses this observation for nodes' behaviours in the federation with low and high and payoff. The stability threshold is set to 0.12. Hence, below this threshold the federation is deemed to be unstable and most nodes are likely in it are untrustworthy. The plot shows that IoT-EG strives to keep higher stability (i.e., low standard deviation of trust scores) for high and low rewards compared to PeerTrust and IoT-Trust. Moreover, Fig. 8 shows the average payoff for the nodes in a federation. It

is clear that IoT-EG outperformed PeerTrust and IoT-Trust in both cases (high and low rewards) by having higher average payoff.

8. Future research directions

Our future research directions are multiple covering federation management, trust gamification, strategy selection, to cite just some. Let us start with the first direction where competition among federations to attract and retain things could be based on other factors than incentives. Compatibility and complementarity could allow things to team up when playing complex games. How to find the right peers and ultimately ask for better incentives is a question worth pursuing.

Regarding the second direction that is trust gamification, we would like to enrich the trust-based evolutionary game model with complex and dynamic strategies besides (un)trustworthiness. Indeed, more strategies would be made available to things from which they would select depending on the situations that they will encounter. How to identify and analyze which strategy is being adopted is a question that would fall into trust-gamification. Another question is to make strategies dynamic so they are adopted on-the-fly.

Finally, and in line with the second direction, a research question would concern a probabilistic and not deterministic selection of strategies. This should allow things to weigh in different factors like risk and uncertainty prior to committing to a particular strategy that could either strengthen or undermine their capabilities.

9. Conclusion

This paper presented a trust-based evolutionary game model for managing IoT federations. A federation acted as a platform for hosting things instead of remaining independent offering mechanisms for signing up and signing off, for example. Compared to existing trust models, the role of trust is fostered into the games that things play when they join federations based on factors like trustworthiness. These games capture the dynamic nature of federations that would like to attract, reward, and retain trustworthy things and penalize and expel untrustworthy ones. This exercise of retaining and expelling along with things' signing in and signing off has an impact on the stability of federations that wish to remain active for a longer period of time. The stability in the proposed model can be achieved by analyzing the trust behaviors of things and retaining only those exhibiting stable behaviors.

To demonstrate the technical doability of our trust-based evolutionary game model for managing IoT federations, a set of experiments were conducted by simulating federations according to a specific network topology, things' capabilities, and interactions between things and federations. Results show that high incentives encourage things to follow trustworthy strategies and minimize bad-mouthing and ballot-stuffing attacks on their operations. The model robustness against these attacks has been checked and benchmarked to two similar trust models, IoT-Trust [7] and PeerTrust [32].

References

[1] M. Abdmeziem, D. Tandjaoui, and I. Romdhani. Architecting the Internet of Things: State of the Art. In Anis Koubaa and Elhadi Shakshuki, editors, *Robots and Sensor Clouds*. Springer International Publishing, 55–75, 2016.

[2] M. Al-khafajiy, T. Baker, M. Asim, Z. Guo, R. Ranjan, A. Longo, D. Puthal, and M. Taylor. Comitment: A Fog Computing Trust Management Approach. *Journal of Parallel and Distributed Computing*, 137:1–16, 2020.

[3] O. Amft and K. Van Laerhoven. What Will We Wear After Smartphones? *IEEE Pervasive Computing*, 16(4):80–85, 2017.

[4] P. Barnaghi and A. Sheth. On Searching the Internet of Things: Requirements and Challenges. *IEEE Intelligent Systems*, 31(6):71–75, 2016.

[5] A. Celesti, M. Fazio, M. Giacobbe, A. Puliafito, and M. Villari. Characterizing Cloud

Federation in IoT. In *Proceedings of the 2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA'2016)*, Crans-Montana, Switzerland, 93–98, 2016.

[6] C. Chen and S. Helal. A Device-Centric Approach to a Safer Internet of Things. In *Proceedings of NoME-IoT'2011 Workshop*, Beijing, China, 1–6, 2011.

[7] I. Chen, J. Guoa, and F. Bao. Trust Management for SOA-Based IoT and Its Application to Service Composition. *IEEE Transactions on Services Computing*, 9(3):3444–3449, 2016.

[8] T. Dinh, J. Tang, Q. La, and T. Quek. Offloading in Mobile Edge Computing: Task Allocation and Computational Frequency Scaling. *IEEE Transactions on Communications*, 65(8):3571–3584, 2017.

[9] DZone. The Internet of Things, Application, Protocls, and Best Practices. Technical report, https://dzone.com/guides/iot-applications-protocols-and-best-practices, 2017 (visited in May 2017).

[10] D. Foster and P. Young. Stochastic Evolutionary Game Dynamics. *Theor Popul Biol*, 38:219–232, 1990.

[11] A. Heil, M. Knoll, and T. Weis. The Internet of Things - Context-based Device Federations. In *Proceedings of the 40th Hawaii International Conference on System Sciences (HICSS'2007)*, Hawaii, USA, 1–9, 2007.

[12] S. Kamvar, M. Schlosser, and S. Garica-Molina. The Eigentrust Algorithm for Reputation Management in P2P Networks. In *Proceedings of the 12th International Conference on World Wide Web*, Budapest, Hungary, 640–651, 2003.

[13] M. Kandori, G. Mailath, and R. Rob. Learning, Mutation and Long-Run Equilibria in Games. *Econometrica*, 61:29–56, 1993.

[14] Z. Maamar, K. Boukadi, E. Ugljanin, T. Baker, M. Asim, M. Al-Khafajiy, D. Benslimane, and H. El Alaoui El Abdallaoui. Thing Federation as a Service: Foundations and Demonstration. In *Proceedings of the 8th International Conference on Model and Data Engineering (MEDI'2018)*, Marrakesh, Morocco, 184–197, 2018.

[15] C. Marche and M. Nitti. Binary Trust Game for the Internet of Things. *IoT*, 2(1):50–70, 2021.

[16] B. Mazon-Olivo and A. Pan. Internet of Things: State-of-the-Art, Computing Paradigms and Reference Architectures. *IEEE Latin America Transactions*, 20(1):49–63, 2021.

[17] W. Mathlouthi and N. Ben Saoud. Flexible Composition of System of Systems on Cloud Federation. In *Proceedings of the 2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud'2017)*, Prague, Czech Republic, 358–365, 2017.

[18] J. Maynard-Smith and G. Price. The Logic of Animal Conflict. *Nature*, 246(5427):15–18, 1973.

[19] M. Terrell and Y. Wang and M. Dorow and S. Agrawal and B. Sriraman and Z. Leidall and A. Chandra and J. Weissman. Constellation: An Edge-Based Semantic Runtime System for Internet of Things Applications. *arXiv*, doi = 10.48550/ARXIV.2201.12394, url = https://arxiv.org/abs/2201.12394, 2022.

[20] R. Mihailescu, R. Spalazzese, C. Heyer, and P. Davidsson. A Role-Based Approach for Orchestrating Emergent Configurations in the Internet of Things. In *Proceedings of the Second International Workshop on the Internet of Agents (IoA'2017) held in conjunction with AAMAS'2017*, São Paulo, Brazil, 18–36, 2017.

[21] J. Nash. Non-Cooperative Games. *Annals of Mathematics*, 54:287–295, 1951.

[22] H. Orman. You Let That In? *IEEE Internet Computing*, 21(3):99–102, 2017.

[23] C. Perera, C. Liu, S. Jayawardena, and M. Chen. A Survey on Internet of Things From Industrial Market Perspective. *IEEE Access*, 2:1660–1679, 2014.

[24] A. Qamar, A. Muhammad, Z. Maamar, T. Baker, and S. Saeed. A Quality-of-Things Model for Assessing the Internet-of-Thing's Non-Functional Properties. *Transactions on Emerging Telecommunications Technologies*, 1–13, 2019.

[25] A. Robson and F. Vega-Redondo. Efficient Equilibrium Selection in Evolutionary Games with Random Matching. *J Econ Theory*, 70:65–92, 1996.

[26] A. Sharma, E. Pilli, A. Mazumdar, and P. Gera. Towards trustworthy Internet of Things: A Survey on Trust Management Applications and Schemes. *Computer Communication*, 160:475–493, 2020.

[27] M. Smith. The Theory of Games and the Evolution of Animal Conflict. *Journal of Theoretical Biology*, 47:209–221, 1974.

[28] A. Taivalsaari and T. Mikkonen. A Roadmap to the Programmable World: Software Challenges in the IoT Era. *IEEE Software*, 34(1):72–80, 2017.

[29] P. Taylor and L. Jonker. Evolutionarily Stable Strategies and Game Dynamics. *Mathematical Biosciences*, 40:145–156, 1978.

[30] E. Torroglosa-Garcia and A. Skarmeta-Gomez. Towards Interoperabilty in Identity Federation Systems. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, 8(2):1–6, June 2017.

[31] M. Weiser. The Computer for the 21^{st} Century. *Newsletter ACM SIGMOBILE Mobile Computing and Communications Review*, 3(3):3–11, 1999.

[32] L. Xiong and L. Liu. Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities. *IEEE transactions on Knowledge and Data Engineering*, 16(7):843–857, 2004.