electronics

MDPI

Article

Applying Trust Patterns to Model Complex Trustworthiness in the Internet of Things

Fabrizio Messina [1,†], Domenico Rosaci [2,*,†] and Giuseppe M. L. Sarnè [3,†]

[1] Department of Mathematics and Informatics (DMI), University of Catania, 95124 Catania, Italy; fabrizio.messina@unict.it
[2] Department of Information Engineering, Infrastructure and Sustainable Energy (DIIES), University "Mediterranea" of Reggio Calabria, 89122 Reggio Calabria, Italy
[3] Department of Psychology, Università Degli Studi of Milano Bicocca, 20126 Milan, Italy; giuseppe.sarne@unimib.it
* Correspondence: domenico.rosaci@unirc.it
† These authors contributed equally to this work.

Abstract: Key aspects of communities of the Internet of Things (IoT) smart objects presenting social aspects are represented by trust and reputation relationships between the objects. Several trustworthiness models have been presented in the literature in the context of multi-smart object community that could be adopted in the IoT scenario; however, most of these approaches represent the different dimensions of trust using scalar measures, then integrating these measures in a global trustworthiness value. In this paper, we discuss the limitation of this approach in the IoT context, highlighting the necessity of modeling complex trust relationships that cannot be captured by a vector-based model, and we propose a new trust model in which the trust perceived by an object with respect to another object is modeled by a directed, weighted graph whose vertices are trust dimensions and whose arcs represent relationships between trust dimensions. By using this new model, we provide the IoT community with the possibility of representing also situations in which an object does not know a trust dimension, e.g., reliability, but it is able to derive it from another one, e.g., honesty. The introduced model can represent any trust structure of the type illustrated above, in which several trust dimensions are mutually dependent.

Keywords: Internet of Things; security; reputation; LoRa; meshtastic; LoWPAN; simulation

Citation: Messina, F.; Rosaci, D.; Sarnè, G.M.L. Applying Trust Patterns to Model Complex Trustworthiness in the Internet of Things. *Electronics* **2024**, *13*, 2107. https://doi.org/10.3390/electronics13112107

Academic Editor: Aryya Gangopadhyay

Received: 6 May 2024
Revised: 24 May 2024
Accepted: 25 May 2024
Published: 29 May 2024

1. Introduction

Nowadays the paradigm of smart object programming is evolving by integrating social aspects, deriving from the need that the objects interact with each other to execute interactive complex tasks, as providing and requesting services, negotiating contracts, etc. [1–3]. The necessity to represent social interactions between software entities naturally emerged in the cooperative smart objects field [4,5], which considers actors using sociality to implement collaborative behaviors. In this context, to enhance collaboration into communities of smart objects, several proposals have been presented [6]. In such a scenario, to deal with egoistic competitors (e.g., see the case of e-commerce), several proposals have been presented [7,8] having the specific purpose of providing the smart objects with specialized social features to better achieve their goals. In all the proposals above, the need emerges to effectively represent trustworthiness between the actors of the community. Such a requirement leads to realize trust-based models which address several aspects of the trust relationships. Trust-based models are generally designed to produce an improvement of the smart object performances, in particular when certain situations arises, as in the IoT, when smart objects are distributed in large networks and have to continuously share information with each other.

When two smart objects interact, the one acting as service requester is denoted as *trustor*, which acts as a service requester, whereas the other is denoted as the *trustee*, and provides e-services.

It is very important to remark that a trust relationship between two smart objects often involves several dimensions, where the term *dimension* means the particular viewpoint adopted to analyze the interaction. Some usual dimensions for representing trust are, for instance, competence, honesty, security, reliability, expertise, etc. These dimensions are subjective measures, whose values are computed by each smart object independently from the other ones. However, often we have the need to consider another dimension of the trust in which the whole smart object community assigns a trust value with respect to a given trustee. This dimension is denoted as the *reputation* of the trustee, and assumes a key role in those situations where a smart object x does not have sufficient knowledge about another smart object b, and then has the need to use b's reputation.

Many trust and reputation models have been proposed in the past in the context of the multi-smart object community [9,10]. Most of these proposals take into account the multi-dimensional nature of trust by using scalar measures and integrating these measures into synthetic indicators of trustworthiness. We here highlight that an evident limitation in this approach is that it considers trust and reputation as simple scalar values, possibly collected into a trust vector. As an example, the trust of a smart object a for another smart object b could be represented by the vector $trust_{a,b} = [honesty, reliability, expertise, \ldots]$. However, often the single elements of the trust vector are note-perceived by the smart object as independent from each other. It could be possible that a trust value, e.g., honesty, is perceived as derivable from another measure, e.g., reliability, since in some cases a smart object could think that a reliable partner should be considered also as honest, with a given degree of probability. To cover this situation, we propose in this paper a new model of trust and reputation for a community of social smart objects, in which the trust perceived by a smart object a with respect to another smart object b is not modeled as a vector of trust values, but instead using a directed, weighted graph whose nodes are trust dimensions (that we will denote as *trust aspects*) and whose edges represent relationships between trust dimensions. This way, we introduce the advantages of also representing situations in which a smart object does not know a trust dimension, e.g., honesty, but it is capable of deriving it from another dimension, e.g., reliability. The model introduced in this paper, called *T-pattern*, is designed to represent any type of situation described above, in which several different trust dimensions are mutually dependent. Our model is based on a formalism that recalls the logical rules. As an example, in order to represent the fact that the dimension *honesty* is derived from the dimension *reliability*, we use the rule $reliability \xrightarrow{z} honesty$. We highlight that, in our formalism, dimensions such as *reliability* and *honesty* are not logical literals, but instead represent real variables assuming values in the interval $[0, 1]$, where 1 (resp. 0) means maximum (resp. minimum) trustworthiness and z is a real value allowing to quantify the value of the derived variable. This way, it is possible to derive the same trust variable simultaneously from many different trust variables, without the difficulty of solving a logic program, but simply by adjusting the z values. Moreover, we also introduce the notion of T-Pattern Network (TPN) as an integrated framework to represent both the trust and reputation values as well as the dependencies between trust dimensions for all the pairs of smart objects. This paper is devoted to only presenting the theoretical formalism of the T-Pattern model and network. In order to practically implement it, it is necessary to apply some inference techniques to derive the logical rules that could be represented, for instance, by neural networks, fuzzy logics, Bayesian approaches, and many other possible solutions. Here, we do not cover this aspect; that is a subject of our ongoing research.

This paper is organized as follows. In Section 2, we deal with some related work. In Section 3, we introduce the scenario we deal with as well as the architecture of our IoT community. In Section 4, we describe in detail our proposal, also giving a significant example of our T-pattern model, whereas Section 5 proposes a distributed architecture,

called TPN, to automatically manage a Trust Pattern Network. Finally, in Section 6, we draw some conclusions and discuss our ongoing research.

2. Related Works

Trust is a basic component of almost social interactions independently from the real or virtual nature of the actors. For such a reason, modeling trust is crucial in a wide number of disciplines and, thanks to its multifaceted nature, several definitions of trust have been provided in the literature based on the specific aspect(s) attended to. Similarly, there exist different approaches to deal with the problem of modeling trust and reputation in multi-agent communities that have been designed in the past. A complete overview of this is behind our aims, but the interested reader may refer to the overwhelming literature on this area, like [11–15], for a deeper analysis.

With respect to our topics, from the viewpoint of logic, in [16], a set of appropriate logic rules about trust are defined in terms of epistemic and dynamic properties for deriving consequences. Consequently, information about trust can be exploited to take appropriate decisions. Another proposal, at a high level of abstraction, can be found in [17], where a logical language, named Trust Computation Tree logic (TCTL), extends the Computation Tree Logic (CTL) [18], a temporal logic for a branching-time model checking able to represent trust. In addition, recently the authors of [19] proposed $TCTL^G$ to quantify the relationships occurring between agents by adopting a logical language able to represent the quantitative aspect of trust and develop a symbolic model checking algorithm. The T-patterns model differs from the logical approaches mentioned above because it adopts a formalism that seems similar to the deductive formalism of logic programs but, unlike them, it does not rely on logical predicates. In fact, it exploits a kind of *quantification* of the strength of rules, which is particularly suitable when trust assessment is to be expressed.

Machine learning techniques are often adopted in complex applications within an increasing number of domains. However, the opaque nature of such techniques can sometimes make it difficult to develop in counterparts a rational confidence and an adequate level of acceptance in the results they provide, notwithstanding their effectiveness [20]. Several approaches in the field of trust evaluation funded on machine learning techniques have been proposed in the literature, and some in-depth studies on such approaches are analyzed in [18,21–23]. In such a scenario, we note that the general multi-agent architecture for automatically learning knowledge encapsulated in T-models is compatible with different learning mechanisms described in the survey referenced above. Therefore, subject to the above considerations, the model proposed by T-patterns is also suitable to conveniently support machine learning techniques.

Among the various artificial intelligence techniques that could potentially be used to develop our model, and in particular to be used in the automatic construction of T models, the connectionist approach seems to be promising. In detail, the use of logic neural networks would enable knowledge extraction from T-patterns via simple logical rules of type *if-then* [24,25], thus also overcoming the opacity aspects that characterize machine learning techniques.

With regard to the applications of trust-based approaches in the field of multi-agent systems (MASs), they have found increasing application over the years and, in particular, in the field of collaborative agents. In the context of trust and collaborative agents, there are several interesting applications of the Internet of Things (IoT) [12,26,27] among which [28] is notable. This study, as part of an MAS, uses a competitive approach in the context of non-cooperative games to optimize the formation of groups of agents that asymptotically maximize social capital with respect to the agents' trustworthiness. The calculated solution to the above problem complies with the Nash equilibrium. Another application was proposed in [10], in which case a social IoT MAS faces attacks aimed at undermining the trust relationships that have been established in the IoT community. To this end, a trust management model has been developed that is resilient to the types of attacks considered by the authors and capable of marginalizing malicious IoT nodes. Although not explicitly

agent-based, the approach proposed in [29] is intrinsically MAS compatible. Here, to address the weaknesses of existing approaches in correctly estimating trustworthiness, taking into account malicious behaviors and dynamic context changes, a general trust model is proposed that can consider competence, willingness, and social relationship in SIoT via two specific functions, Degree of Importance (DoI) and Degree of Contribution (DoC), to calculate competence and helpfulness of actors.

In [30], the authors propose a trust evaluation scheme for federated learning in DTMN, which takes direct trust evidence and recommended trust information into account. In [31], a novel trust cascading-based emergency message dissemination (TCEMD) model is proposed, which incorporates the entity-oriented trust values into data-oriented trust evaluation in an efficient manner.

Other applications of trust-based approaches for MAS embrace a very large number of areas, including, for example, the following: (*i*) in the field of transportation, to increase the safety of autonomous vehicles at intersections [32] or the exchange of reliable information between smart vehicles [33]; (*ii*) in different markets such as, for example, analyzing the influence of trust on market functioning in the market for energy [34] or for perishable goods, where there is no quality indicator and quantities can vary significantly [35]; (*iii*) manage, for various purposes, social networks of agents using a trust and reputation-based overlay layer referring to the relationships between users, the evolution of their reputations over time [36], and the similarity of their interests for increased resilience of the social network to malicious activities and tampering that can ensure reliable transactions [37] and communications [38].

Another important aspect of a trust-based approach is that of privacy protection. In [39], a novel Privacy-Preserving Reputation-Updating scheme for cloud-assisted vehicular networks is proposed, based on the Elliptic Curve Cryptography (ECC) and Paillier algorithms, in which the reputation feedbacks are collected and preprocessed by the honest-but-curious Cloud Service Provider (CSP) in a privacy-preserving manner, and the computation and communication overheads on the TA side can be dramatically reduced. In [40], the authors propose a novel privacy-preserving trust management (PPTM) scheme for emergency message dissemination in SAGIVNs. The proposed scheme can realize precise trust management and strong conditional privacy preservation simultaneously with low communication overhead and can provide strong applicability, strong robustness, and multiple other attractive features.

In summary, some related scientific contributions that for different reasons are close to T-Patterns have been presented here. Compared to them, T-Patterns are able to model trust and reputation, also considering the causal implications existing between the various possible dimensions of trust, in much more complex scenarios than can be modeled through simple trust arrays.

3. The IoT Social Scenario

In this paper, we consider *smart objects* as IoT devices operating on behalf of a human being. We introduce a set S of smart objects—the reader may refer to Table 1 for the list of all the symbols used in this paper—capable of interacting with each other, thus constituting a social community. We also introduce *trust aspects* as follows:

- **Expertise**, which represents the capability of a smart object to give expert opinions in a specific domain. In other words, different know-how related to different domains can be used as trust aspects. For instance, in an e-commerce scenario, we could denote as $e_{financial}$ the expertise of a smart object to give opinions about finance;
- **Honesty** as the capability of the SO to provide a truthful behavior, i.e., how much it is not fraudulent or misleading;
- **Security** as the honesty of an SO in describing how much the smart object confidentially manages private data and does not allow unauthorized access to them;

- **Reliability** represents a measure of the reliability of the services provided by the SO. In other words, reliability represents the degree of reliance that can be placed on the services provided by the smart object, including effectiveness and efficiency.

Now, we denote as $K = \langle k_1, k_2, \ldots, k_n \rangle$ a set of n trust aspects associated with the set of smart objects S, where the elements $k_1, k_2, \ldots, k_n$ represent the names of the trust aspects. Moreover, in order to represent how the smart objects quantitatively evaluate the trust aspects contained in K, we introduce the notion *confidence* in a trust aspect. We characterize such confidence with respect to a *group of smart objects* denoted as Ω which, in a particular case, can coincide with the entire smart object community.

The confidence that a smart object s_1 has in a trust aspect $k^* \in K$ of a smart object s_2 with respect to the group Ω, that we denote as $\gamma_{s_1,s_2,\Omega}(k^*)$, represents the trust that the smart object s_1 assigns to the trust aspect k^* of the smart object $s_2$in the context of the group Ω. The confidence can be either (*i*) a real value belonging to the interval $[0, 1]$, where $\gamma_{s_1,s_2,\Omega}(k^*) = 1$ (resp. $\gamma_{s_1,s_2,\Omega}(k^*) = 0$) means maximum (resp. minimum) trust, or (*ii*) the value *null* if that confidence has not been yet evaluated.

The confidence that a group Ω (which can coincide with the whole social community) has in a trust aspect k^* of a smart object s_1, that we denote as $\gamma_{s_1,s_1,\Omega}(k^*)$, represents the *group reputation* that the group Ω assigns to the trust aspect k^* of the smart object s_1, and it is either a real value belonging to the interval $[0, 1]$, where $\gamma_{s_1,s_1,\Omega}(k^*) = 1$ (resp. $\gamma_{s_1,s_1,\Omega}(k^*) = 0$) means maximum (resp. minimum) reputation or (*ii*) the value *null* if that confidence has not been yet evaluated. If the group Ω includes all the smart objects of the community S, we call $\gamma_{s_1,s_1,S}(k^*)$ *community reputation* of the smart object S with respect to the trust aspect k^*.

Table 1. Symbol list.

Symbol	Meaning
K	a set of trust aspects
k_i	a specific trust aspect
S	a set of smart objects
s_i	a specific smart object
Ω	a group of smart objects
$\gamma_{s_1,s_2,\Omega}(k^*)$	the confidence of s_1 for k^* with respect to s_2 within the group Ω
$\gamma_{s_1,s_1,\Omega}(k^*)$	the confidence of the group Ω for k^* with respect to s_1
N_k	a weighted directed graph representing trust relationships
Γ	a mapping on a confidence $\gamma_{s_1,s_2,\Omega}$ which gives a value belonging in $[0, 1]$
$p = \langle s_1, s_2, \Omega, N_K \rangle$	a T-pattern over two smart objects s_1 and s_2, a trust network N_k within a group Ω

4. The T-Pattern Model

A *T-pattern* is defined as a tuple $p = \langle s_1, s_2, \Omega, N_K \rangle$, where (*i*) N_K is a network (i.e., a weighted directed graph) representing some trust relationships, associated with the ordered pair $\langle s_1, s_2 \rangle$, involving the trust aspects contained in K and (*ii*) Ω is a set of smart objects (a group) in the context of which the trust aspects are evaluated.

Note that we also consider the possibility to have a *global* T-pattern, which is a tuple $\langle s_1, s_1, \Omega, N_K \rangle$; therefore, a global T-pattern is characterized by a single smart object by means of the global evaluations expressed by the whole group Γ (as a special case, by the whole smart object community), whereas in the other cases, where $s_1 \neq s_2$, we have a *peer-to-peer T-Pattern* (*P2P T-pattern* for short), characterizing how a smart object s_1 trusts another smart object s_2.

The N_K network contained in a T-pattern $p = \langle s_1, s_2, \Omega, N_K \rangle$ represents how the smart object s_1 perceives the relationships existing among the trust aspects with respect to the smart object s_2 and in the context of a group Ω. Formally $N_K = \langle Q, E \rangle$ is a network composed of a set of trust aspects $Q \subset K$ and a set of edges E, whereas each edge $e = \langle k_1, k_2, \Omega, z \rangle$ is an ordered tuple such that $k_1, k_2 \in Q$ are two trust aspects, Ω is a group

and z is a real number, belonging to the interval $[0, 1]$. In the case $s_2 \neq s_1$, i.e., in the case of a P2P T-pattern, the value z represents how the smart object s_1 perceives the relationship between the trust aspects k_1 and k_2, with respect to the smart object s_2:

$$z = \frac{\gamma_{s_1,s_2,\Omega}(k_1)}{\gamma_{s_1,s_2,\Omega}(k_2)} \tag{1}$$

Therefore, we can derive $\gamma_{s_1,s_2,\Omega}(k_1)$ as $z \cdot \gamma_{s_1,s_2,\Omega}(k_2)$. For instance, if k_1 is the trust aspect *reliability* and k_2 is the trust aspect *honesty*, the arc $\langle k_1, k_2, 0.7\rangle$ means that the confidence in the honesty of k_2 is evaluated by k_1 as the 70 percent of the confidence assigned of the reliability of k_2.

In the case $s_2 = s_1$, i.e., in the case of a global T-pattern, z represents how the whole group (as a particular case, the whole social community) perceives the relationship between the trust aspects k_1 and k_2, with respect to the smart object s_1. The value of z is also in this case equal to the ratio $\frac{\gamma_{s_1,s_1,\Omega}(k_1)}{\gamma_{s_1,s_1,\Omega}(k_2)}$.

Derivation rule (DR). On the basis of the previous considerations, each arc $e = \langle k_1, k_2, \Omega, z\rangle$ of the N_K network of a P2P T-pattern means that $\gamma_{s_1,s_2,\Omega}(k_1) = z \cdot \gamma_{s_1,s_2,\Omega}(k_2)$. We call such a type of rule a *derivation rule* and we also denote it by the following formalism:

$$k_1 \xrightarrow{z} k_2 \tag{2}$$

Assignment rule (AR). Moreover, we define another type of rule, called *assignment rule*, that assigns to the confidence $\gamma_{k_1,k_2,\Omega}(s)$ a given value $v \in [0, 1] \cup \{null\}$ and we denote it by the following formalism:

$$v \rightarrow s \tag{3}$$

z-assignment (ZR). Finally, we define a third type of rule, called *z-assignment*, that assigns to the value z of an edge e, a value c belonging to the interval $[0, 1]$, denoting it by the following:

$$c \rightarrow z \tag{4}$$

For the global T-Pattern, we have a completely identical formalism to indicate that $\gamma_{s,s,\Omega}(k_1) = z \cdot \gamma_{s,s,\Omega}(k_2)$.

4.1. Automatic Rules Application

The three rules described above can be applied automatically as follows:

- Each time the confidence $\gamma_{s_1,s_2,\Omega}(k)$ (resp. $\gamma_{s,s,\Omega}(k)$ in the case the T-pattern is associated with an auto-edge) of a trust aspect k has been updated (in consequence of the application of the AR rule (3) or by side effect of other updates), automatically the DR rule (2) is applied to each edge outgoing from the node associated with k in the network N_K, and the update is propagated to each other node outgoing from the now updated nodes, except for the node k.
- For all the edges $e = \langle k, k^*, \Omega, z\rangle$ incoming in the node k^*, the ZR rule (4) is automatically applied such that z will be updated by using a value $c = \frac{\gamma_{s_1,s_2,\Omega}(k)}{\gamma_{s_1,s_2,\Omega}(k^*)}$.
- Each time the ZR rule (4) is applied to update the value z of an edge e, it is automatically applied to the DR rule (2) to the final node of the edge e, and such an update is propagated to each other node outgoing from the updated nodes, except for the initial node of the arc e.

4.2. T-Pattern Network

A *T-Pattern Network* (TPN for short) is defined as a tuple $\langle S, G, K, \Gamma, P\rangle$, where S is a set of smart objects, G is a set of groups $(G_1, G_2, \ldots, G_n)$, K is a set of trust aspects, Γ is a mapping that maps each confidence $\gamma_{s_1,s_2,\Omega}$ with a value belonging in $[0, 1] \cup null$, and

P is a set of T-patterns on K, such that there do not exist two T-patterns belonging to P associated with the same ordered triplet (s_1, s_2, Ω).

Since a T-pattern can be viewed as an edge between two smart objects, a TPN can be considered as a network where the nodes are smart objects and the edges are defined by the T-patterns of P, each edge weighted by the pair (Γ, G_K) associated with the corresponding T-pattern. The mechanisms of automatic rule activation described above guarantee the consistence of all the T-patterns with the confidence (Γ) mapping.

4.3. Practical Example

In order to give an overview of the practical application of the formalism introduced in the above section, let us consider Figure 1, where two smart objects are depicted.

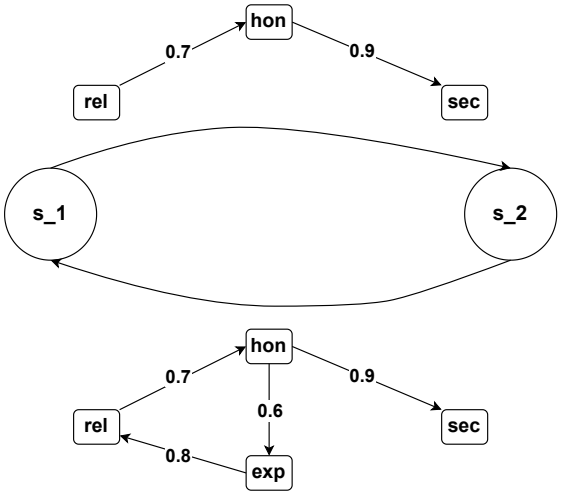

Figure 1. A simple TPN having two smart objects.

The two smart objects, denoted as s_1 and s_2, respectively, live in a multi-smart object community represented by the tuple $\langle S, K, \gamma, G, P \rangle$, where

- $S = \{s_1, s_2\}$ is the set of smart objects;
- $G = \emptyset$; in this example, we do not consider any group;
- $K = \{rel, hon, sec, exp\}$ is a set of trust aspects representing reliability (*rel*), honesty (*hon*), security (*sec*), and expertise (*exp*);
- γ is a mapping representing the confidence values of each smart object with respect to the other smart objects. In this example, we suppose that all the values contained in γ are equal to null; in other words, we depict an initial situation with no knowledge about the mutual trustworthiness of the smart objects. Moreover, we have omitted, for simplicity, indicating the group Ω, since in this case, the unique group is represented by the whole community.
- P is a set containing two T-patterns, graphically depicted by the two arcs $< s_1, s_2, N_{s_1,s_2} >$ and $< s_2, s_1, N_{s_2,s_1} >$, respectively, representing the a priori knowledge we have about how the two smart objects perceive the trustworthiness.

The network N_{s_1,s_2} (upper part of Figure 1) informs us that the smart object s_1 derives the honesty of s_2 from the reliability by using the derivation rule $rel \xrightarrow{0.7} hon$ (represented by the number 0.7 in the upper part of Figure 1). In other words, if s_1 knows the reliability of s_2, it automatically trusts the honesty of s_2 as a percentage of the reliability equal to the 70 percent. Therefore, the smart object s_1 sufficiently believes the honesty of s_2, even in cases where it has not directly experienced such honesty, provided that the reliability of s_2 has been verified.

The network N_{s_1,s_2} (upper part of Figure 1) also informs us that the smart object s_1 derives the security of s_2 from the honesty by using the derivation rule $hon \xrightarrow{0.9} sec$, repre-

sented by the number 0.9 in the upper part of Figure 1. In other words, when s_1 has verified the honesty of s_2, it is also almost sure of its security.

In the initial situation of the confidence values represented by the mapping γ, no information is available and therefore no information can be derived from the derivation rules that can be applied only when some confidence values will be directly derived from the smart objects.

Finally, the meaning of the the network N_{s_2,s_1} (lower part of Figure 1) is analogously represented by the following derivation rules (represented by all the numbers in the lower part of Figure 1):

$$rel \xrightarrow{0.7} hon; \qquad hon \xrightarrow{0.9} sec \tag{5}$$

$$hon \xrightarrow{0.6} exp; \qquad exp \xrightarrow{0.8} rel \tag{6}$$

5. The T-Pattern Architecture (TPA) to Automatically Manage a TPN

In this section, we describe a multi-SO architecture, called *T-Pattern Architecture* (TPA) and depicted in Figure 2, capable of managing a T-Pattern Network $NET = \langle S, G, K, \gamma, P \rangle$, deriving the information needed to update the T-patterns by observing the smart object behavior.

The TPA is distributed on three logical levels, namely:

- A *smart object level*, composed of *n trust manager smart objects* $tm_1, tm_2, \ldots, tm_n$, where each tm_i is associated with the corresponding smart object $s_i \in S$ of NET and is capable of updating the trust patterns associated with all the edges outgoing from s_i in NET. The trust manager tm_i will apply some inferential technique to automatically construct and update the trust pattern;
- A *group level*, composed of *l group manager smart objects* $gm_1, gm_2, \ldots, gm_l$, where each gm_i is associated with the corresponding group $\Omega_i \in G$ of NET and is capable of computing the group reputation $\gamma_{s,s,\Omega_i}(t)$ for all the smart objects $s \in \Omega_i$ and for all the trust aspects $k \in K$, by applying the automated rules described in Section 4.1 considering only the smart objects $s \in \Omega_i$;
- A *community level*, composed from a *community manager smart object* CM, capable of computing the community reputation $\gamma_{s,s,S}(t)$ for all the smart objects $s \in S$ and for all the trust aspects $k \in K$ by applying the automated rules described in Section 4.1 considering all the smart objects $s \in S$.

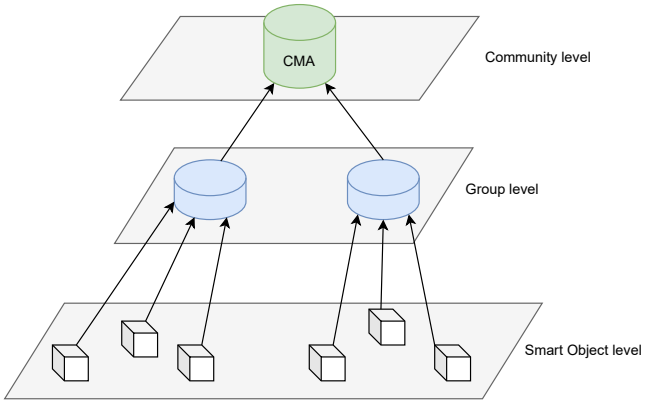

Figure 2. The three-layers of TPA architecture.

Computational Evaluation

As we have higlighted in the previous section, a T-pattern can be viewed as an edge between two smart objects, and then a TPN can be considered as a network where the nodes are smart objects and the edges are defined by the T-patterns of the set P, each edge weighted by the pair (Γ, G_K) associated with the corresponding T-pattern. The construction of a T-Pattern Network then requires a time cost that linearly depends on the cardinality $|P|$ of the set P of the T-patterns. Each T-pattern can have a maximum of $|K|^2$ arcs, where

K is the number of possible trust aspects, therefore the time computational complexity for the updating activity of each trust manager (associated with each smart object) is $O(|P| \cdot |K|^2)$. The distributed architecture of the TPN allows one to efficiently manage the network updating. Indeed, each group manager gm_i exploits the work produced by the trust managers associated with the smart objects of its group Ω_i, and therefore the time computational complexity for the updating activity is simply $O(|\Omega_i|)$. Analogously, the time complexity for the updating activity of the community manager CM is $O|S|$.

6. Conclusions

In this work, we have dealt with the problem of representing mutual trust among a community of smart objects. We have designed a model capable of maintaining trust information by a weighted graph in which nodes are trust dimensions and whose edges represent relationships between trust dimensions. The key point of the proposed model is represented by the scenario in which a smart object does not know a trust dimension. In this case, by the proposed model, it will be possible to derive it from another dimension. We have formalized the notion of T-Pattern Network (TPN), as an integrated framework capable of representing trust and reputation in a community of smart objects. The T-pattern Network also enables the representation of the dependencies between trust dimensions for all the pairs of smart objects.

Finally, we are working on developing suitable mechanisms to build trust patterns, as well as to compute group-level reputation and community reputation starting by the trust patterns. Moreover, it is important to highlight the necessity to deal with privacy protection issues when applying trust patterns in an IoT context. Indeed, trust patterns include aspects revealing information about the use of smart objects that could be perceived as sensible by human users. We are planning to address this issue in our future work.

Author Contributions: Methodology, F.M., D.R. and G.M.L.S.; Investigation, F.M., D.R. and G.M.L.S.; Writing—original draft, F.M., D.R. and G.M.L.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Italian Ministry of University and Research (MUR) Project "T-LADIES" under Grant PRIN 2020TL3X8X and in part by Pia.ce.ri. 2020–2022 funded by the University of Catania and in part by the Project CAL.HUB.RIA funded by the Italian Ministry of Health, Project CUP: F63C22000530001. Local Project CUP: C33C22000540001.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Telang, P.; Singh, M.P.; Yorke-Smith, N. Maintenance of Social Commitments in Multiagent Systems. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtually, 2–9 February 2021; Volume 35, pp. 11369–11377.
2. Jaques, N.; Lazaridou, A.; Hughes, E.; Gulcehre, C.; Ortega, P.; Strouse, D.; Leibo, J.Z.; De Freitas, N. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In Proceedings of the International Conference on Machine Learning. PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 3040–3049.
3. Esmaeili, A.; Mozayani, N.; Motlagh, M.R.J.; Matson, E.T. A socially-based distributed self-organizing algorithm for holonic multi-agent systems: Case study in a task environment. *Cogn. Syst. Res.* **2017**, *43*, 21–44. [CrossRef]
4. Walczak, S. Society of Agents: A framework for multi-agent collaborative problem solving. In *Natural Language Processing: Concepts, Methodologies, Tools, and Applications*; IGI Global: Hershey, PA, USA, 2020; pp. 160–183.
5. Torreño, A.; Onaindia, E.; Komenda, A.; Štolba, M. Cooperative multi-agent planning: A survey. *ACM Comput. Surv. (CSUR)* **2017**, *50*, 1–32. [CrossRef]
6. Khan, W.Z.; Aalsalem, M.Y.; Khan, M.K.; Arshad, Q. When social objects collaborate: Concepts, processing elements, attacks and challenges. *Comput. Electr. Eng.* **2017**, *58*, 397–411. [CrossRef]
7. Jafari, S.; Navidi, H. A game-theoretic approach for modeling competitive diffusion over social networks. *Games* **2018**, *9*, 8. [CrossRef]
8. He, Z.; Han, G.; Cheng, T.; Fan, B.; Dong, J. Evolutionary food quality and location strategies for restaurants in competitive online-to-offline food ordering and delivery markets: An agent-based approach. *Int. J. Prod. Econ.* **2019**, *215*, 61–72. [CrossRef]

9. Kowshalya, A.M.; Valarmathi, M. Trust management for reliable decision making among social objects in the Social Internet of Things. *IET Netw.* **2017**, *6*, 75–80. [CrossRef]
10. Brogan, C.; Smith, J. *Trust Agents: Using the Web to buIld Influence, Improve Reputation, and Earn Trust*; John Wiley & Sons: Hoboken, NJ, USA, 2020.
11. Cho, J.H.; Chan, K.; Adali, S. A survey on trust modeling. *ACM Comput. Surv. (CSUR)* **2015**, *48*, 1–40. [CrossRef]
12. Fortino, G.; Fotia, L.; Messina, F.; Rosaci, D.; Sarné, G.M.L. Trust and reputation in the internet of Things: State-of-the-art and research challenges. *IEEE Access* **2020**, *8*, 60117–60125. [CrossRef]
13. Hoff, K.A.; Bashir, M. Trust in automation: Integrating empirical evidence on factors that influence trust. *Hum. Factors* **2015**, *57*, 407–434. [CrossRef]
14. Jøsang, A.; Ismail, R.; Boyd, C. A survey of trust and reputation systems for online service provision. *Decis. Support Syst.* **2007**, *43*, 618–644. [CrossRef]
15. Wang, J.; Yan, Z.; Wang, H.; Li, T.; Pedrycz, W. A survey on trust models in heterogeneous networks. *IEEE Commun. Surv. Tutor.* **2022**, *24*, 2127–2162. [CrossRef]
16. Demolombe, R. Reasoning about trust: A formal logical framework. In Proceedings of the International Conference on Trust Management, Oxford, UK, 29 March–1 April 2004; pp. 291–303.
17. Drawel, N.; Bentahar, J.; Shakshuki, E. Reasoning about trust and time in a system of agents. *Procedia Comput. Sci.* **2017**, *109*, 632–639. [CrossRef]
18. Baier, C.; Katoen, J.P. *Principles of Model Checking*; MIT Press: Cambridge, MA, USA, 2008.
19. Drawel, N.; Bentahar, J.; Qu, H. Computationally Grounded Quantitative Trust with Time. In Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems, Auckland, New Zealand, 9–13 May 2020; pp. 1837–1839.
20. Burrell, J. How the machine 'thinks': Understanding opacity in machine learning algorithms. *Big Data Soc.* **2016**, *3*, 2053951715622512. [CrossRef]
21. Liu, X.; Datta, A.; Lim, E.P. *Computational Trust Models and Machine Learning*; CRC Press: Boca Raton, FL, USA, 2014.
22. Ma, W.; Wang, X.; Hu, M.; Zhou, Q. Machine learning empowered trust evaluation method for IoT devices. *IEEE Access* **2021**, *9*, 65066–65077. [CrossRef]
23. Wang, J.; Jing, X.; Yan, Z.; Fu, Y.; Pedrycz, W.; Yang, L.T. A survey on trust evaluation based on machine learning. *ACM Comput. Surv. (CSUR)* **2020**, *53*, 1–36. [CrossRef]
24. Palmer-Brown, D. Neural Networks for Modal and Virtual Learning. In Proceedings of the Artificial Intelligence Applications and Innovations III, Thessaloniki, Greece, 23–25 April 2009; p. 2.
25. Rosaci, D. CILIOS: Connectionist inductive learning and inter-ontology similarities for recommending information agents. *Inf. Syst.* **2007**, *32*, 793–825. [CrossRef]
26. Sharma, A.; Pilli, E.S.; Mazumdar, A.P.; Gera, P. Towards trustworthy Internet of Things: A survey on Trust Management applications and schemes. *Comput. Commun.* **2020**, *160*, 475–493. [CrossRef]
27. Hussain, Y.; Zhiqiu, H.; Akbar, M.A.; Alsanad, A.; Alsanad, A.A.A.; Nawaz, A.; Khan, I.A.; Khan, Z.U. Context-aware trust and reputation model for fog-based IoT. *IEEE Access* **2020**, *8*, 31622–31632. [CrossRef]
28. Fortino, G.; Fotia, L.; Messina, F.; Rosaci, D.; Sarné, G.M.L. A meritocratic trust-based group formation in an IoT environment for smart cities. *Future Gener. Comput. Syst.* **2020**, *108*, 34–45. [CrossRef]
29. Wei, L.; Wu, J.; Long, C.; Li, B. On designing context-aware trust model and service delegation for social internet of things. *IEEE Internet Things J.* **2020**, *8*, 4775–4787. [CrossRef]
30. Guo, J.; Liu, Z.; Tian, S.; Huang, F.; Li, J.; Li, X.; Igorevich, K.; TFL-DT, J.M. TFL-DT: A Trust Evaluation Scheme for Federated Learning in Digital Twin for Mobile Networks. *IEEE J. Sel. Areas Commun.* **2023**, *41*, 3548–3560. [CrossRef]
31. Liu, Z.; Weng, J.; Ma, J.; Guo, J.; Feng, B.; Jiang, Z.; Wei, K. TCEMD: A trust cascading-based emergency message dissemination model in VANETs. *IEEE Internet Things J.* **2019**, *7*, 4028–4048. [CrossRef]
32. Chuprov, S.; Viksnin, I.; Kim, I.; Reznikand, L.; Khokhlov, I. Reputation and trust models with data quality metrics for improving autonomous vehicles traffic security and safety. In Proceedings of the 2020 IEEE Systems Security Symposium (SSS), Crystal City, VA, USA, 1 July–1 August 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–8.
33. De Meo, P.; Messina, F.; Postorino, M.N.; Rosaci, D.; Sarné, G.M.L. A reputation framework to share resources into iot-based environments. In Proceedings of the 2017 IEEE 14th International Conference on Networking, Sensing and Control (ICNSC), Calabria, Italy, 16–18 May 2017; pp. 513–518.
34. Andrade, R.; Pinto, T.; Praça, I. Trust model for a multi-agent based simulation of local energy markets. In Proceedings of the International Conference on Practical Applications of Agents and Multi-Agent Systems, L'Aquila, Italy, 16–19 June 2020; pp. 183–194.
35. Van Kooten, O.; Nevejan, C.; Brazier, F.; Oey, M.; Hubers, C. SamenMarkt®, a Proposal for Restoring Trust in the Horticultural Fresh Food Market by Using Multi-Agent System Technology. In *Agricultural Value Chain*; IntechOpen: London, UK, 2018; pp. 19–36.
36. Fortino, G.; Messina, F.; Rosaci, D.; Sarné, G.M.L. Using blockchain in a reputation-based model for grouping agents in the Internet of Things. *IEEE Trans. Eng. Manag.* **2019**, *67*, 1231–1243. [CrossRef]
37. Su, Z.; Liu, L.; Li, M.; Fan, X.; Zhou, Y. Reliable and resilient trust management in distributed service provision networks. *ACM Trans. Web (TWEB)* **2015**, *9*, 1–37. [CrossRef]

38. Das, A.; Islam, M.M. SecuredTrust: A dynamic trust computation model for secured communication in multiagent systems. *IEEE Trans. Dependable Secur. Comput.* **2011**, *9*, 261–274. [CrossRef]
39. Liu, Z.; Wan, L.; Guo, J.; Huang, F.; Feng, X.; Wang, L.; Ma, J. PPRU: A privacy-preserving reputation updating scheme for cloud-assisted vehicular networks. *IEEE Trans. Veh. Technol.* **2023**, 1–16. [CrossRef]
40. Liu, Z.; Weng, J.; Guo, J.; Ma, J.; Huang, F.; Sun, H.; Cheng, Y. PPTM: A privacy-preserving trust management scheme for emergency message dissemination in space–air–ground-integrated vehicular networks. *IEEE Internet Things J.* **2021**, *9*, 5943–5956. [CrossRef]