future internet

MDPI

Article

A Neural-Symbolic Approach to Extract Trust Patterns in IoT Scenarios

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

[1] Department of Mathematics and Informatics, University of Catania, 95125 Catania, Italy; fabrizio.messina@unict.it
[2] Department DIIES, University "Mediterranea" of Reggio Calabria, 89122 Reggio Calabria, Italy
[3] Department of Psychology, University of Milan-Bicocca, 20126 Milan, Italy; giuseppe.sarne@unimib.it
* Correspondence: domenico.rosaci@unirc.it
† These authors contributed equally to this work.

Abstract: Trust and reputation relationships among objects represent key aspects of smart IoT object communities with social characteristics. In this context, several trustworthiness models have been presented in the literature that could be applied to IoT scenarios; however, most of these approaches use scalar measures to represent different dimensions of trust, which are then integrated into a single global trustworthiness value. Nevertheless, this scalar approach within the IoT context holds a few limitations that emphasize the need for models that can capture complex trust relationships beyond vector-based representations. To overcome these limitations, we already proposed a novel trust model where the trust perceived by one object with respect to another is represented by a directed, weighted graph. In this model, called T-pattern, the vertices represent individual trust dimensions, and the arcs capture the relationships between these dimensions. This model allows the IoT community to represent scenarios where an object may lack direct knowledge of a particular trust dimension, such as reliability, but can infer it from another dimension, like honesty. The proposed model can represent trust structures of the type described, where multiple trust dimensions are interdependent. This work represents a further contribution by presenting the first real implementation of the T-pattern model, where a neural-symbolic approach has been adopted as inference engine. We performed experiments that demonstrate the capability in inferring trust of both the T-pattern and this specific implementation.

Keywords: Internet of Things; reputation; security simulation; T-pattern model

Academic Editors: Wei Yu and Guobin Xu

Received: 3 February 2025
Revised: 27 February 2025
Accepted: 3 March 2025
Published: 6 March 2025

Citation: Messina, F.; Rosaci, D.; Sarnè, G.M.L. A Neural-Symbolic Approach to Extract Trust Patterns in IoT Scenarios. *Future Internet* **2025**, *17*, 116. https://doi.org/10.3390/fi17030116

1. Introduction

Today, the paradigm of programming smart objects is evolving to incorporate social aspects, driven by the necessity for these objects to interact dynamically to perform complex tasks. These tasks may involve exchanging services, making requests, or negotiating contracts [1–3]. The representation of social interactions between software entities has naturally emerged within the field of cooperative smart objects [4,5], where leveraging social characteristics enables collaborative behaviors. Various approaches have been introduced to enhance collaboration among smart objects within their communities [6]. Additionally, in competitive environments such as e-commerce, several models have been developed [7,8] to equip smart objects with social capabilities that improve their ability to achieve specific goals. Across these models, the need to effectively represent trustworthiness among community members remains a consistent requirement. This necessity has led to the

development of trust-based models that address different dimensions of trust relationships. Such models are typically designed to enhance smart object performance, particularly in IoT environments, where objects operate within vast networks and must constantly share information.

When two smart objects interact, the one requesting a service is termed the trustor, while the one providing it is the trustee. A trust relationship between two smart objects often spans multiple dimensions. The term dimension refers to the specific perspective used to evaluate the interaction. Common trust dimensions include attributes such as competence, honesty, security, reliability, and expertise, alongside others that depend on the specific context. These dimensions represent subjective measures that each smart object independently calculates based on its perception of its surroundings. However, in some cases, an additional dimension—reputation—is necessary. This is where the community collectively assigns a trust value to a trustee. Reputation becomes particularly relevant when a smart object o_1 lacks direct knowledge of another smart object o_2 and must instead rely on the community's collective assessment of o_2's trustworthiness.

Various trust and reputation models have been proposed for multi-smart object communities [9,10]. Most of these approaches quantify trust using scalar measures, integrating them into synthetic trustworthiness indicators. However, a key limitation of these models is their reliance on simple scalar values, often structured as trust vectors. For example, the trust of a smart object o_1 toward another object o_2 might be represented by the vector $trust_{o_1,o_2} = [honesty, reliability, expertise, ...]$. These trust vector elements are often interdependent rather than fully independent. For instance, a smart object might infer honesty from reliability, assuming that a reliable partner is also likely to be honest to some degree, factoring in both internal and external conditions. Our study does not introduce new quantitative trust measures but instead presents a model capable of representing scenarios where a smart object lacks direct knowledge of a trust dimension, such as reliability, and infers it from another, like honesty. The proposed model captures the interdependencies among multiple trust dimensions. To address this, we propose a trust and reputation model for social smart object communities, where the trust a smart object o_1 perceives toward another o_2 is not represented as a vector of independent values. Instead, it is modeled as a directed, weighted graph, where nodes represent trust dimensions (termed *trust aspects*), and edges capture their interdependencies. This approach allows smart objects to infer unknown trust dimensions based on available information from other related dimensions.

The proposed model, named T-pattern [11], is designed to capture interdependent trust dimensions. It employs a formalism similar to logical rules. For example, to express that the *honesty* dimension derives from *reliability*, we use the rule $reliability \xrightarrow{z} honesty$. Here, dimensions such as *reliability* and *honesty* are not mere logical literals but real variables with values ranging from 1 to 5, where 5 (and 1) denote maximum (and minimum) trustworthiness, respectively. The parameter z is a real value quantifying the inferred trust variable.This approach allows multiple trust variables to contribute to the derivation of another, avoiding the complexities of traditional logic programming by adjusting z values. Furthermore, we introduce the T-Pattern Network (TPN), a framework that represents trust and reputation values alongside dependencies among trust dimensions across all smart objects.

In prior work, we presented the theoretical foundation of the T-pattern model [11] but did not demonstrate its capacity for trust inference. To fully implement the T-pattern model and network, inference techniques are required to derive logical rules, potentially utilizing neural networks, fuzzy logic, Bayesian methods, or other approaches.

To validate the T-pattern model's ability to infer trust dimensions, we developed its first real implementation. For this purpose, we employed CILIOS (Connectionist Learning

and Inter-Ontology Similarities) [12] as the inference engine. CILIOS is a connectionist learning approach that generates logical rules to model agent behavior using a concept graph. It leverages neural-symbolic networks where input and output nodes correspond to logical variables. By observing actor behavior, CILIOS autonomously derives ontologies, requiring only the definition of relevant concepts and categories by the model designer. Using this CILIOS-based implementation, we conducted experiments to assess the model's effectiveness in enabling smart objects to estimate their partners' trustworthiness.

The obtained results confirm the T-pattern model's ability to accurately represent dependencies between trust dimensions—an anticipated yet unverified outcome before our experiments. Different inference engines could be integrated into the T-pattern model depending on various application requirements, such as scalability, memory efficiency, or processing time. However, addressing these aspects goes beyond the scope of this work and represents a challenge for future research.

1.1. Advantages of T-Patterns in the IoT Context

As discussed, most existing models represent trust and reputation as independent scalar values aggregated into a trust vector. This assumption simplifies real-world interactions, where trust dimensions (e.g., reliability and honesty) can be interdependent. In highly dynamic IoT environments, where agent populations evolve rapidly and newcomers frequently join, information about certain trust dimensions may be initially unavailable, while other dimensions can be inferred from initial interactions. Our research addresses this issue by introducing the following advantages:

- The T-pattern model enables IoT systems to infer unknown trust dimensions, such as reliability, based on available information from related dimensions like honesty.
- It employs an interpretable logical formalism that supports interactions between smart objects at the IoT edge, while the inference process runs in the background at the cloud level.
- T-patterns can help smart objects in an IoT environment to operate in presence of noisy or incomplete information since the smart object can use T-patterns to derive an unknown or uncertain information from another one available or more reliable. Our experiments, although limited to a simulated environment, demonstrated that T-patterns can represent the logical links by the different trust dimensions with an high degree of precision, due to the observation of the whole agent community; thus, an inexpert agent, a newcomer, or an agent that operates in presence of noise and that has not a complete information on a given trust dimension can be advantaged by the possibility to derive it form the T-patterns.
- Our work presents the first real implementation of the T-pattern model, utilizing a neural-symbolic inference engine.
- CILIOS is specifically designed to extract logical rules by directly observing the behavior of a set of agents in a multi-agent system, regardless of the particular network environment and trust conditions, as the logical rules are extracted from a neural network trained at the cloud layer of the multi-agent system and are continuously updated as agents move from one given network environment to another. This feature made CILIOS particularly suitable to be applied to extract T-patterns in an IoT context.

1.2. Plan of the Paper

This paper is organized as follows. In Section 2, we review related work. Section 3 introduces the scenario under study as well as the architecture of our IoT community and in Section 4, we provide an overview of the T-pattern model proposal. In Section 5, we describe the architecture and the neural-symbolic implementation of T-pattern based on

CILIOS and present the experiments carried out and their results that demonstrated the effectiveness of this first version of T-pattern to infer trust. Finally, Section 6 concludes this paper and discusses directions for our ongoing research.

2. Related Works

The field of logic-based approaches for trust and reputation has garnered growing interest, leading to the development of several innovative methods. For instance, in [13], the authors propose formal definitions of various types of trust within a modal logic framework, describing trust as a "mental attitude of an agent" concerning certain properties (epistemic, deontic, and dynamic) attributed to other agents. In [14], the focus shifts to reasoning about quantitative aspects, such as trust levels, through the introduction of $TCTL^G$, a logical language tailored to represent this information. The authors also created a symbolic model-checking algorithm to quantify relationships among agents. Our approach, in contrast, employs the T-pattern model, which, while reminiscent of logical formalism, relies on a unique form of quantifying rule strength rather than logic predicates.

Various studies in the literature leverage machine learning (ML) to address trust issues [15–18]. This alignment fits well with the T-pattern model, as its multi-agent framework inherently supports knowledge learning. Additionally, logic-based neural networks (e.g., *if–then* constructs [12,19]) can be utilized to extract insights effectively from T-pattern models.

Recent research has delved into machine learning and neural network approaches within human–object trust relationships. For instance, Ref. [20] investigated trust assessment in Trust Social Networks (TSNs), considering trust propagation and fusion factors, and proposed the NeuralWalk algorithm for estimating trust factors and predicting relationships. The authors demonstrated that WalkNet, a neural network model for single-hop trust propagation, could inductively predict unknown trust relations using real-world data.

The work presented in [21] addressed trust in the Social Internet of Things (SIoT), where IoT devices interact in a "social manner", making the establishment of trust relationships essential. The authors developed an artificial neural network-based trust framework named "Trust–SIoT", which aims to classify trustworthy objects by identifying complex relationships between inputs and outputs. They highlighted that Trust–SIoT effectively captures various key trust metrics and demonstrated its strong performance through experimental validation within SIoT contexts.

Another noteworthy study discussed in [22] focused on the Internet of Medical Things (IoMT), which aims to enhance the accuracy, reliability, and efficiency of healthcare platforms. The proposed approach, known as NeuroTrust, utilized artificial neural networks to evaluate trust parameters, such as reliability and compatibility, to predict and eliminate malicious nodes that may compromise data integrity. Additionally, a lightweight encryption mechanism is incorporated to strengthen security during data transmission. The experimental results showcased the framework's effectiveness in detecting malicious and compromised nodes, which is crucial for mitigating security threats.

In [17], the authors addressed the issue of compromised or malicious IoT devices. Evaluating trust for these devices is challenging due to the difficulty in measuring various types of trust properties and the associated degrees of belief. To address this, the authors proposed a machine-learning-based trust evaluation method that aggregates network QoS (Quality of Service) properties. A deep learning algorithm was employed to create a behavioral model for each IoT device, quantifying trust as a numerical value by calculating the similarity between observed network behaviors and those predicted by the model.

The study in [23] introduced an innovative intrusion detection method termed the "Taylor-spider monkey optimization-based deep belief network" (Taylor-SMO-based DBN).

This approach incorporates trust factors for intrusion detection, utilizing an optimization algorithm that integrates the Taylor series with the spider monkey optimization (SMO) technique to classify KDD features. The trained deep belief network (DBN) demonstrated superior intrusion detection capabilities compared to other approaches through experimental results.

Trust is also critical in multi-agent systems (MASs), particularly in collaborative agent environments, including IoT applications [24–27]. For example, Ref. [10] discusses a social IoT MAS designed to safeguard trust relationships within the IoT community while minimizing the influence of malicious nodes. Trust evaluation schemes can further enhance federated learning by managing direct trust evidence and recommended trust information, as seen in [28].

Regarding the T-pattern model approach discussed in this paper, it incorporates an "ontology" as part of its logical framework. An ontology is defined as a set of entity descriptions (e.g., classes, relations, and functions) and explicit assumptions represented through a vocabulary that describes reality with a clear and consistent meaning. This framework formalizes knowledge, whether for an agent or a community of agents, using first-order logic where vocabulary items appear as unary (concepts) or binary (relationships) predicates [29]. Notably, two ontologies may use different vocabularies while sharing the same conceptualization.

In agent-based scenarios, ontologies can serve as the foundation of knowledge representation, encompassing key concepts, properties, and relationships, as well as conceptual schemas. This is similar to the framework provided by the JAva DEvelopment Framework (JADE) [30], which models predicates, terms, concepts, actions, and more—akin to the components of agent communication messages. In this context, we adopt ontologies to represent an agent's "viewpoint" regarding interests and behaviors (either its own or those of its owner), typically referred to as a model. This model is adaptable to various frameworks where the agent may operate.

In the past literature, many research work on trust ontologies in multi-agent systems has been proposed. For example, in [31], the authors survey and classify thirteen computational trust models by the trust decision input factors, using such an analysis to create a new comprehensive ontology for trust to facilitate interaction between business systems. Moreover, in [32], it is recognized that an important application area of the Semantic Web is ontology mapping, where different similarities have to be combined into a more reliable and coherent view, which might easily become unreliable if trust is not managed effectively between the different sources. In this paper, the authors propose a solution for managing trust between contradicting beliefs in similarities for ontology mapping based on the fuzzy voting model. In [33], an ontology-based multi-agent virtual enterprise (OMAVE) system is proposed to help SMEs shift from the classical trend of manufacturing part pieces to producing high-value-added, high-tech, innovative products. Furthermore, in [34], the authors introduce an in-depth ontological analysis of the notion of trust, grounded in the Unified Foundational Ontology, and they propose a concrete artifact, namely, the Reference Ontology for Trust, in which the general concept of trust is characterized. In this work, the authors distinguish between two types of trust, namely, social trust and institution-based trust, and they also represent the emergence of risk from trust relations. A systematic review on trust-based negotiation in multi-agent systems (MASs) has been performed in [35], through a bibliometric analysis over the past 25 years of research publications, on three of the most popular scientific databases (Google Scholar, Scopus, and Web of Science). This analysis reveals that this research topic is regaining interest, after some oscillating years, and the impact of its contributions is equivalent to other equally important research variants like ontology and argumentation (in a negotiation scenario). In [36], the authors

aim to handle ontology-based fusion and use multi-agent systems to obtain information fusion from multiple sources/sensors in a secure and integrated manner, with the objective to produce a secure and integrated ontology-based fusion framework by using a multi-agent approach. As for languages and protocols, in [37], the authors proposed a dialogue model, in which multiple agents negotiate the correspondence between two knowledge sets with the support from a Large Language Model (LLM), demonstrating that this approach not only reduces the need for the involvement of a domain expert for ontology alignment but that the results are interpretable despite the use of LLMs.

The Promise Theory [38] has been proposed as a framework for coping with uncertainty in information systems. For example, the use of promises was introduced [39] in a pervasive computing scenario to model the interaction policies between the agents. This approach examines how the autonomic nodes stabilize into a robust functional system in spite of their autonomous decision making and use promises both as a means of modeling a potential specification and as a complementary eye glass for interpreting and understanding emergent behavior. The analysis of promises reveals 'faults' in the policies, which prevent the collaborative functioning of the system as a whole. Successful interactions are the result of a bargaining process. The method of eigenvector centrality is used to locate the most important and vulnerable agents to the functioning of the system. Moreover, in [40,41], the Promise Theory and dimensional analysis was proposed for the Dunbar scaling hierarchy, supported by recent data from group formation in Wikipedia editing. The authors showed how the assumption of a common priority seeds group alignment until the costs associated with attending to the group outweigh the benefits in a detailed balance scenario.

Languages designed to represent the semantics of Web resources, such as OML (Ontology Markup Language) [42], DAML + OIL [43], and SWRL (Semantic Web Rule Language) [44], can also be regarded as ontology models due to their capability to structure semi-structured data. Logic-based approaches have been used extensively in agent systems [45]; for example, Ref. [46] models the state of an agent's environment, while multi-dimensional dynamic logic programming (MDLP) [47] describes the epistemic states of agents.

In our T-pattern model, we employ the approach proposed in CILIOS [12], known as the Information Agent Ontology Model (IAOM). IAOM is designed to represent objects and groups within an agent's environment by assigning them unique "names" within a common vocabulary. Similar to JADE, IAOM's "ontology model" is a class composed of fundamental schemas that all ontologies share, which describe predicates, actions, and concepts pertinent to the agent. In this model, an object in the agent world is identified as an "object", with its associated properties forming an "object-schema". Although object-schemas resemble classes in object-oriented programming (OOP), they are more akin to semi-structured representations such as XML [48]. However, unlike XML, IAOM can model causal relationships using logical formalisms. Additionally, IAOM supports operations on objects and collections through the introduction of (*i*) the "collection" concept, representing a group of objects that may have distinct schemas and be organized into sub-collections, and (*ii*) a set of propositional clauses forming a logic program.

While IAOM includes logical axioms like OML, DAML + OIL, and SWRL, it uniquely models agent actions and distinguishes them from causal implications—defined as logical relationships between events. IAOM can be implemented with a neural-symbolic network, enabling inductive processes that traditional ontology models approach through symbolic, statistical, or connectionist methods. Symbolic methods focus on learning within a symbolic framework, statistical methods utilize probabilistic relational models, and connectionist

approaches, such as neural networks, learn from examples during training using processing units, adaptive connections, and learning processes.

The connectionist approach in CILIOS allows for learning an ontology by observing agent/user behavior and deriving causal implications, incorporating classical and default negation. This capability is particularly valuable for knowledge representation in contexts modeling by T-patterns.

In Table 1, we provide a synthetic comparison between T-pattern model and the other approaches described above to realize trust-based multi-agent systems, considering the main features useful for the IoT context, namely, (a) the interdependence of the different trust dimensions, (b) the possibility of directly extracting the dependencies from the system observation, and (c) the treatment of the uncertainty, where the symbol *Y* (resp. *N*) indicates that the approach considers (resp. does not consider) the correspondent feature.

Table 1. Differences between T-model and the other reviewed approaches.

	Trust Interdependence	Direct Dependencies Extraction	Uncertainty Treatment
T-Model	Y	Y	Y
[33]	N	N	Y
[34]	N	N	Y
[37]	Y	N	N
[36]	Y	N	N
[38]	Y	N	Y

Finally, we want to highlight that the use of T-patterns can be also usefully exploited for enhancing the security of cyber-physical systems. For instance, the necessity of securing modern smart grids requires new networking technology and apposite services designed to cost-effectively secure communications to assets ranging from utility-scale generating units to residential-scale batteries and inverters. This necessity is particularly true for Distributed Energy Resources (DERs) [49], and in this context, the T-pattern framework gives the possibility of introducing an additional security level in the presence of uncertainty in the information (due to, for instance, the introduction of newcomer smart object) that can be faced by deriving unknown trust information from the other one already stored in the system. T-patterns are specifically designed to face these kind of situations.

3. The Scenario

In this Section (for a complete list of symbols used in this paper, please refer to Appendix A), we present the considered environment (E) populated by smart IoT objects and users. In such a context, let o represent a smart IoT object and O denote the set of smart IoT objects living within E. Indeed, in networks with 16,000 smart objects, the average error is approximately 5%. Each object $o \in O$ is associated with a user and acts on their behalf as a prosumer, both producing and consuming data through interactions with other devices in O.

To enable smooth interactions within E, smart IoT objects should establish trust in one another concerning one or more *trust issues*, each representing a distinct aspect of their mutual interactions within E. We define $\Delta = \langle \delta_1, \delta_2, \ldots, \delta_n \rangle$ as a set of n trust issues associated with O in E. Each trust issue δ is characterized by properties such as *Expertise*, *Honesty*, *Security*, and *Reliability*, which hold the following meanings for a generic smart IoT object o:

- *Expertise*. Refers to the level of competence o has in providing knowledgeable opinions within a specific domain. For example, the o's ability to advise its user on corporate stocks.
- *Honesty*. Indicates o's commitment to truthful behavior, free from deception or misleading practices.
- *Security*. Relates to o's handling of confidential data, including safeguarding it from unauthorized access.
- *Reliability*. Measures the consistency and dependability of the services provided by o, with respect to efficiency and effectiveness.

To enable each smart IoT object to quantitatively assess each trust issue in Δ, we introduce the notion of *confidence* ϕ in a trust aspect within E. We define ϕ with respect to a *group of smart objects* in E, denoted as Λ, which, in some cases, may encompass the entire set $O \in E$ of smart IoT objects (i.e., $\Lambda \subseteq O$). Specifically, the confidence that a smart IoT object o_1 assigns to a trust issue $\delta_i \in \Delta$ of another smart IoT object o_2 within the group Λ is represented by $\phi_{o_1,o_2,\Lambda}(\delta_i)$. This confidence can take either (*i*) a real value within $[1,5]$, where 1 (resp., 5) represents the minimum (resp., maximum) trust level of $\phi_{o_1,o_2,\Lambda}(\delta_i)$, or (*ii*) a value of *null* if ϕ has not yet been assessed.

Moreover, within a group $\Lambda \subseteq O$, the confidence perceived by the members of Λ concerning a trust issue δ_i for a smart IoT object o_1 (denoted $\phi_{o_1,o_1,\Lambda}(\delta_i)$) may serve as the *group reputation* of Λ with respect to δ_i. Like individual confidence, this value can range within $[1,5]$ or remain *null* if it has not yet been evaluated. It is worth noting that if Λ coincides with O, then $\phi_{o_1,o_1,\Lambda}(\delta_i)$ will represent the *community reputation* of the all smart IoT objects regarding the trust issue δ_i.

4. A T-Pattern Model Overview

In this section, we provide an overview of the T-pattern model [11].

Before introducing the mathematical formalism, in order to make it clearer and more understandable, we provide a leading example involving two smart IoT objects, denoted as o_1 and o_2 (Figure 1) within an environment E represented by the tuple $\langle O, \Delta, \Phi, \tau \rangle$, where

- $O = \{o_1, o_2\}$ is a set of smart IoT objects;
- $G =$ is a set of groups (in this example, no specific groups are considered, so $G = \{\emptyset\}$);
- $\Delta = \{R, H, S, X\}$ is a set of trust issues, namely reliability (R), honesty (H), security (S), and expertise (X);
- Φ maps the mutual confidence values among the smart IoT objects. In this example, each confidence value is set to null, assuming there is no initial knowledge about the mutual trustworthiness of the other smart IoT objects. (For simplicity, we assume the existence of a single group representing the entire social community E, and thus Λ is omitted).
- τ is a set consisting of two T-patterns (i.e., the links $\langle o_1, o_2, N_{o_1,o_2} \rangle$ and $\langle o_2, o_1, N_{o_2,o_1} \rangle$ in Figure 1, which represent the initial unsymmetrical knowledge of how the two smart IoT objects perceive each other's trustworthiness).

In the proposed example, network N_{o_1,o_2} (on the left in Figure 1) shows that o_1 derives the honesty of o_2 from its reliability using the derivation rule $R \xrightarrow{0.7} H$, with a ratio of 0.7. In other words, o_1 trusts o_2's honesty at 70% of its reliability level, even without prior direct experience of o_2's honesty, provided that o_2's reliability has been verified. Similarly, N_{o_1,o_2} illustrates that o_1 infers the security of o_2 based on its honesty using the derivation rule $H \xrightarrow{0.9} S$, with a ratio of 0.9, again without any previous experience regarding o_2's security.

Finally, in the right part of Figure 1, the network N_{o_2,o_1} reflects the following derivation rules, with the respective ratios indicated.

$$R \xrightarrow{0.7} H; \qquad H \xrightarrow{0.9} S; \qquad H \xrightarrow{0.6} X; \qquad X \xrightarrow{0.8} R \tag{1}$$

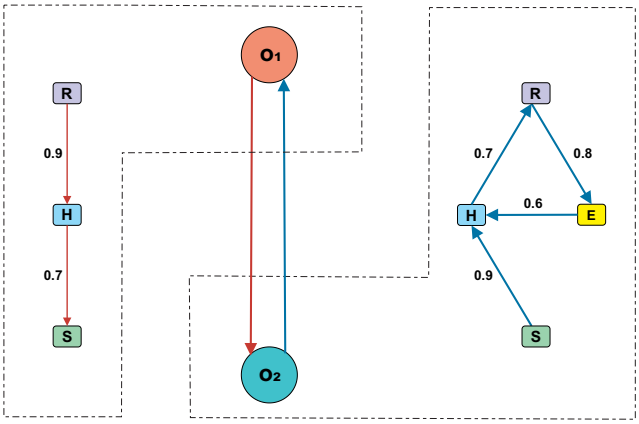

Figure 1. An example of a simple TPN with two smart IoT objects (i.e., o_1 and o_2) considering expertise (X), honesty (H), reliability (R), and security (S) trust issues.

In our framework, a P2P T-pattern is formally defined as a tuple $\tau = \langle N_\Delta, o_1, o_2, \Lambda \rangle$, where N_Δ represents a network that describes peer-to-peer trust relationships for the ordered pair of smart IoT objects o_1 and o_2, in the context of both the group Λ of smart IoT objects and the group Δ of trust issues. In essence, a P2P T-pattern captures how o_1 perceives trust in o_2 within the context of a group Λ for specific trust issues.

Formally, the network $N_\Delta = \langle V, L \rangle$ consists of a set of trust issues $V \subseteq \Delta$ and a set L of links between these trust issues. For any pair of trust issues $\delta_1, \delta_2 \in \Delta$, the link $l \in L$ is represented by the ordered tuple $l = \langle \delta_1, \delta_2, \Lambda, w \rangle$, where Λ is the group context and w is a real-valued weight in the range $[0, 1]$. In the context of a P2P T-pattern, the weight w quantifies the perceived strength of the relationship between trust issues δ_1 and δ_2 as assessed by o_1 in regard to o_2, and is calculated as follows:

$$w = \frac{\phi_{o_1,o_2,\Lambda}(\delta_1)}{\phi_{o_1,o_2,\Lambda}(\delta_2)} \tag{2}$$

which implies that $\phi_{o_1,o_2,\Lambda}(\delta_1) = w \cdot \phi_{o_1,o_2,\Lambda}(\delta_2)$. For example, if the directed link between trust issues δ_1 and δ_2 has an associated weight $w = 0.65$, this indicates that the confidence in δ_2 is 65% of the confidence assigned to δ_1.

We also define a *global* T-pattern as a tuple $\tau = \langle N_\Delta, o_1, o_1, \Lambda \rangle$, representing a single smart IoT object evaluated by the entire group Λ, where $\Lambda \subseteq O$. The weight w, in this case, reflects the collective perception of the relationship between trust issues δ_1 and δ_2 regarding smart object o, and it is computed similarly to Equation (2): $\phi_{o,o,\Lambda}(\delta_1) = w \cdot \phi_{o,o,\Lambda}(\delta_2)$.

To manage a P2P T-pattern for a generic link $l = \langle \delta_1, \delta_2, \Lambda, w \rangle$ within the network N_Δ, we apply the following three rules:

- **Derivation rule (DR)**. It computes $\phi_{o_1,o_2,\Lambda}(\delta_1)$ as $w \cdot \phi_{o_1,o_2,\Lambda}(\delta_2)$. It is denoted as follows:

$$\delta_1 \xrightarrow{w} \delta_2 \tag{3}$$

- **v-Assignment rule (VR)**. It assigns a value $v \in [1, 5] \cup \{null\}$ to the confidence $\gamma_{\delta_1,\delta_2,\Lambda}(o)$. It is denoted as follows:

$$v \rightarrow o \tag{4}$$

- w**-Assignment (ZR)**. It joins a value $c \in [0,1]$ to the value w of a link l. It is denoted as follows:

$$c \rightarrow w \tag{5}$$

These three rules can be automatically applied to both P2P and global T-pattern:

- When, due to Equation (3) or other updates, the confidence $\phi_{o_1,o_2,\Lambda}(\delta)$ (or, respectively, $\phi_{o,o,\Lambda}(\delta)$) for a trust issue δ is updated, then Equation (2) is applied to each link originating from δ represented in N_Δ and is propagated to the other nodes linked to the updated nodes, excluding δ.
- For each link $l = \langle \delta, \delta^*, \Lambda, w \rangle$ directed towards δ^*, Equation (5) is automatically applied, updating w to c, which is calculated as the ratio $\frac{\phi_{o_1,o_2,\Lambda}(\delta)}{\phi_{o_1,o_2,\Lambda}(\delta^*)}$.
- Each time the weight w of a link l is updated by Equation (5), then Equation (2) is automatically applied to the link's destination node and is propagated to other nodes linked to the updated nodes, excluding the origin node of link l.

This procedure can be synthetically described by the pseudo-code shown in Algorithm 1.

Algorithm 1 Trust propagation algorithm

```
for all trust issue δ do
  if φ_{o1,o2,Λ}(δ) OR φ_{o,o,Λ}(δ) is updated then
    for all link l originating from δ ∈ N_Δ do
      Apply and propagate Equation (2)
    end for
    for all link l = ⟨δ, δ*, Λ, w⟩ directed towards δ* do
      Apply Equation (5) automatically
    end for
    for all link l do
      if weight w of l is updated by Equation (5) then
        Apply and propagate Equation (2) automatically
      end if
    end for
  end if
end for
```

In such a way, the T-pattern model can be applied to a large variety of scenarios, from more simple to more complex or adversarial ones (e.g., untrustworthy devices or sudden changes in behavior).

Finally, we define a *T-Pattern Network* (TPN) as a tuple $\langle O, G, \Phi, \Lambda, T \rangle$, where O is the set of smart IoT objects, G represents groups (i.e., $G = \langle \Lambda_1, \Lambda_2, \ldots, \Lambda_n \rangle$), Φ maps each confidence $\phi_{o_1,o_2,\Lambda}$ to a value in $[1,5] \cup null$, Λ is the set of trust issues, and T is the set of T-pattern on Δ, ensuring that no two T-patterns in T are associated with the same triplet (o_1, o_2, Λ). A T-pattern can be considered a link between two smart IoT objects; therefore, a TPN can be viewed as a network where the vertices are smart IoT objects, and the links represent the T-pattern in T. Each link (i.e., T-pattern) has an associated weight defined by (Φ, G_Δ). Based on the above automatic rule activation, the consistency of all T-pattern with the confidence mapping (Γ) is maintained. We highlight that all the trust measures are dimensionless, having values ranging in the interval $[1 \ldots 5]$, where 1 (resp. 5) means minimum (resp. maximum) trust, and for this reason, in the introduced formulas, there is not ever a division by zero.

5. A Neural-Symbolic Implementation of T-Pattern

As discussed in the introductory section, while in [11], we only presented the T-pattern model, in this work, we introduce its first implementation based on the T-Pattern Architec-

ture, equipped with the neural-symbolic inference engine CILIOS [12]. This implementation allowed us to conduct an experimental campaign in a simulated environment populated with smart IoT objects. The goal of such experiments was to verify the effectiveness of the T-pattern model in inferring trust.

Therefore, this section provides a detailed description of the experiments conducted and their corresponding results.

5.1. The T-Pattern Architecture

The *T-Pattern Architecture* (TPA) is a multi-SO architecture designed to manage a T-Pattern Network $NET = \langle O, G, \Delta, \Gamma, P \rangle$, deriving the information needed to update T-patterns by observing the behavior of smart IoT objects and where O is the set of smart objects, G is a set of groups $(\Omega_1, \Omega_2, \cdots, \Omega_n)$, *Delta* is the set of trust issues, Γ is a mapping on a confidence $\gamma_{o_1,o_2,\Lambda}$ which gives a value ranging in $[1,5]$, and P is a set of T-patterns on Δ, such that there are not two T-patterns belonging to P associated with the same ordered triplet (o_1, o_2, Λ).

The TPA is distributed across three logical levels (see Figure 2), where each level follows the automated rules described in Section 4 (see also [11] for additional details):

- A *smart object level*, consisting of *n trust manager smart objects* $tm_1, tm_2, \ldots, tm_n$, where each tm_i is associated with a corresponding smart object $o_i \in O$ of NET and is capable of updating the trust patterns associated with all edges originating from s_i in NET;
- A *group level*, consisting 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 calculating the group reputation $\gamma_{o,o,\Omega_i}(t)$ for each smart object $o \in \Omega_i$ and each trust aspect $k \in K$;
- A *community level*, consisting of a single *community manager CM* of smart objects, which is capable of calculating the community reputation $\gamma_{o,o,S}(t)$ for each smart object $o \in O$ and each trust aspect $\delta \in \Delta$.

The three layers can be managed at three different computational layers: (i) the smart object level is represented by the smart object (IoT), and (ii) the group level can be managed into the Fog [50] and the community level can be mapped into the Cloud.

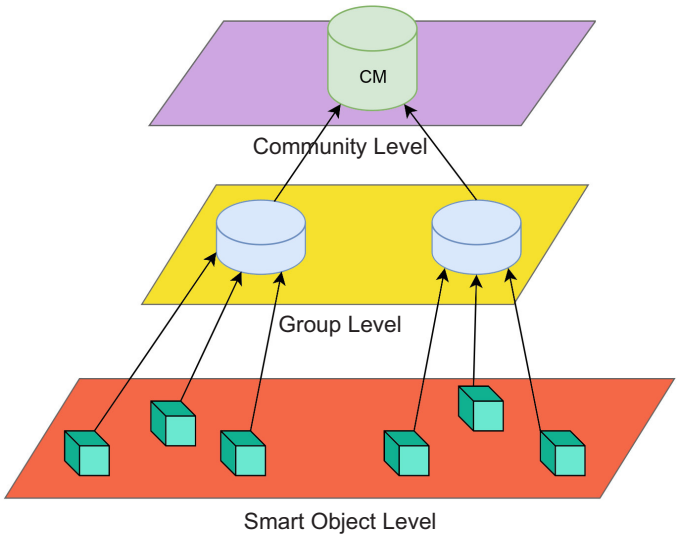

Figure 2. The three-layer TPA architecture.

We highlight that the presence of the three layers gives the possibility of managing each layer with different and independent computational resources as follows: (i) the smart object level is managed by the smart objects, (ii) the group level is managed by

the computational power of the Fog, and, finally, the community level is managed into the Cloud. This way, increasing the number of smart objects or object groups do not significantly impact system performance. In particular, the scalability of the upper service will be easily ensured by the elasticity of Cloud resources.

5.2. The Inference Engine

The T-pattern model requires an inference technique in order to derive suitable logical rules. To this end, we identified in CILIOS [12] a promising inference engine for T-pattern. A brief description of CILIOS is provided below.

CILIOS (Connectionist Inductive Learning and Inter-Ontology Similarities) is a system originally developed to enhance agent collaboration in multi-agent environments (MAS). This system enables the construction of knowledge representations, known as ontologies, that support collaborative decision-making and personalized recommendations.

The architecture of CILIOS combines connectionist learning techniques with symbolic ontology models. In CILIOS, observed behaviors are translated into logical rules using neural-symbolic networks. Each CILIOS agent can dynamically represent and update its own ontology, which describes the relevant concepts and categories for interaction.

By integrating ontologies, CILIOS facilitates efficient cooperation, adapts to changes, and continuously improves interaction capabilities, increasing the system's ability to respond to complex contexts like the Web and IoT networks. Furthermore, the system is designed to be flexible and scalable, supporting a broad range of applications where continuous learning and dynamic adaptation are essential.

In order to understand how CILIOS can be used to extract T-patterns (a complete description can be found in [12]), we briefly describe the underlying idea on which it is conceived. This idea is based on the possibility that symbolic knowledge can be represented by a connectionist system, as a neural network, in order to build an effective learning system. In particular, it is proved that, for each extended logic program P, there exists a feed-forward neural network N with exactly one hidden layer and semi-linear activation functions, which is equivalent to P in the sense that N computes the model of P. In our current application to the IoT described in this paper, a set of T-patterns as, for example, that represented in Equation (2) can be viewed as a logic program P, and it can be derived following the constructive definition provided in [12]. In particular, this construction passes through the construction of a feed-forward neural network that will be trained on the data provided by the smart objects' interactions and that will be then analyzed to be transformed in a logic program representing the T-patterns.

5.3. The Simulation

To evaluate the effectiveness of the T-pattern model in representing trust relationships in a simulated smart IoT environment, we considered the following elements:

- A set of n smart IoT objects O;
- A set of k groups G;
- A set of trust issues $\Delta = \{R, H, S, X\}$, representing reliability (R), honesty (H), security (S), and expertise (X).

5.4. Training Phase

We simulated i mutual interactions between the smart IoT objects in O, applying the CILIOS approach to derive the set τ (containing the extracted T-pattern), as described in Section 4. The set Φ (mapping mutual confidence values among smart IoT objects) was initially randomly generated and refined continuously throughout the training process. We repeated the training phase for different values of $n = 2000, 4000, 8000$, 16,000 and

$k = 5, 10, 15, 20$, resulting in 16 training scenarios, each denoted by $TRAINING_{n,k}$. We used $i = 100{,}000$ (resp. 200,000, 400,000, 800,000) for $n = 2000$ (resp. 4000, 8000, 16,000), assuming that interactions increase linearly with the number of smart IoT objects.

5.5. Test Phase

For each training phase $TRAINING_{n,k}$, we conducted a corresponding test phase $TEST_{n,k}$, simulating i mutual interactions among the smart IoT objects in O. For each pair of smart IoT objects o_h, o_k, only the reliability value was initially known, while honesty, security, and expertise were derived using the T-pattern extracted in the training phase.

As in the training phase, we used $i = 100{,}000$ (resp. 200,000, 400,000, 800,000) for $n = 2000$ (resp. 4000, 8000, 16,000). For each test phase $TEST_{n,k}$ and each pair of smart IoT objects o_p, o_q, we calculated the percentage error $e(o_p, o_q, HON)$ (resp. $e(o_p, o_q, SEC)$, $e(o_p, o_q, EXP)$) by comparing the inferred values for honesty, security, and expertise to the actual values. Finally, we computed the average percentage error $e(HON)$ (resp. $e(SEC)$, $e(EXP)$) across all pairs o_p, o_q, with the results presented in Tables 2–4.

Table 2. Percentage error $e(HON)$ for different values of number of smart IoT objects n and number of groups k.

	n = 2000	**n = 4000**	**n = 8000**	**n = 16,000**
k = 5	0.115	0.092	0.071	0.054
k = 10	0.131	0.112	0.087	0.062
k = 15	0.144	0.121	0.107	0.088
k = 20	0.155	0.134	0.119	0.093

Table 3. Percentage error $e(SEC)$ for different values of number of smart IoT objects n and number of groups k.

	n = 2000	**n = 4000**	**n = 8000**	**n = 16,000**
k = 5	0.111	0.089	0.074	0.051
k = 10	0.129	0.111	0.085	0.065
k = 15	0.140	0.126	0.112	0.086
k = 20	0.151	0.130	0.117	0.096

Table 4. Percentage error $e(EXP)$ for different values of number of smart IoT objects n and number of groups k.

	n = 2000	**n = 4000**	**n = 8000**	**n = 16,000**
k = 5	0.119	0.093	0.075	0.055
k = 10	0.120	0.090	0.084	0.067
k = 15	0.139	0.122	0.110	0.092
k = 20	0.146	0.134	0.122	0.11

The same results are graphically represented in the bar diagrams in Figures 3–5.

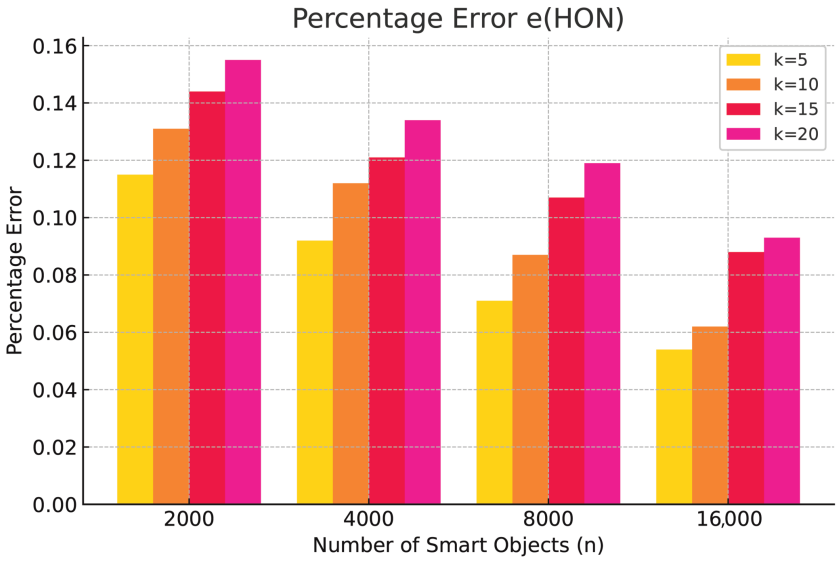

Figure 3. Percentage error $e(HON)$ for different values of number of smart IoT objects n and number of groups k.

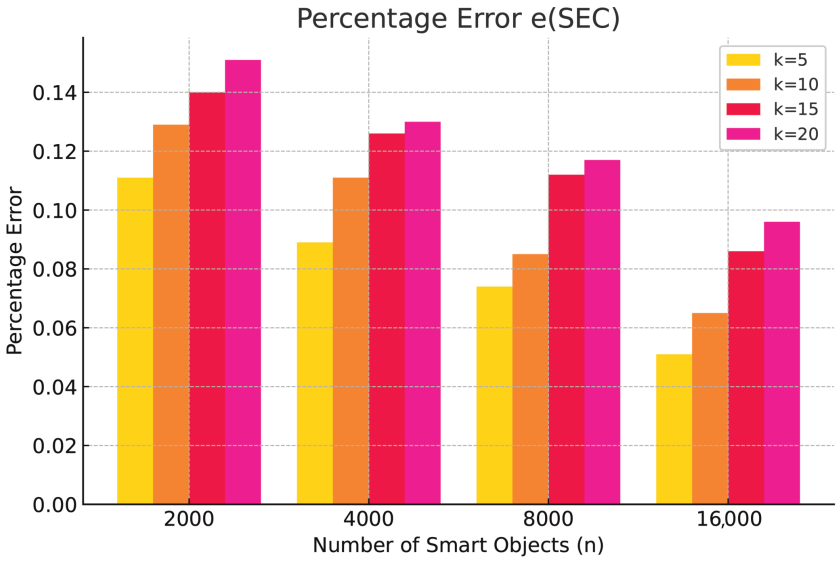

Figure 4. Percentage error $e(SEC)$ for different values of number of smart IoT objects n and number of groups k.

5.6. Discussion of the Results

The experimental results show that the T-pattern model enables smart IoT objects to estimate confidence values in their partners for three trust issues (honesty, security, and expertise) derived from reliability values, with an average error ranging from 5% to 15% with respect to actual values, demonstrating that the model has a sufficient accuracy. Accuracy improves with network size: in networks with 16,000 smart objects, the average error is approximately 5%. In particular, in our simulations, larger networks collected more interactions (the number i), thus providing a more precise inference mechanism during training with respect to small networks. This results do not represent a limitation for small networks because, as we verified in our simulations, even smaller networks, sooner or later, will reach a number of interactions to decrease the measured error.

Figure 5. Percentage error $e(EXP)$ for different values of number of smart IoT objects n and number of groups k.

We also observe that the average error increases with the number of groups, as a higher number of groups complicates the inference process. While the average error with only 5 groups ranges from 5% to 11%, it increases to between 11% and 15% with 20 groups. We highlight that increasing the number of groups imply increasing the number of logical rules that the inference engine has to extract. It has been demonstrated in [12] that the effectiveness of the logical rules in representing the data with which the neural network is trained decreases with the number of logical rules; thus, this result was expected.

As we further discuss in our conclusion, we aim to improve this particular aspect in the near future. We finally observe that the experimental results are consistent across all trust issues derived using the T-pattern model.

5.7. Efficiency and Costs

We highlight that the construction of T-patterns is performed at the cloud level, in background with respect to the ordinary activities of the IoT systems, and thus, it does not significantly impact the execution time of the other tasks involving the smart objects, while T-patterns model consumes a minimal amount of system memory, relating to the storage of a graph of relatively small dimension. This implies that no significant barriers exist for the implementation of this approach to large-scale wireless IoT systems. In order to compute T-patterns using the CILIOS inference engine, we have used, in pur simulation at the cloud level, a 64-bit workstation with a Intel(R) Core(TM) i7-10875H CPU @ 2.30 GHz and 128 GB of RAM. Table 5 shows how computation time changes with varying network sizes.

Table 5. Execution time t (in minutes) of the T-pattern model construction for different values of number of smart IoT objects n and number of groups k.

	n = 2000	**n = 4000**	**n = 8000**	**n = 16,000**
k = 5	12.5	40.9	200.7	744
k = 10	13.6	42.13	212.4	771.12
k = 15	16.2	45.20	221.17	792.28
k = 20	18.8	48.41	233.55	802.24

As a final consideration, we highlight that a different group size impacts the system performance simply based on its impact on the global size n of the multi-agent community. This is derived from several simulations that we performed by fixing k and increasing the size of a given group (that results in an increment of the parameter n). We simply re-obtained the results shown in the tables above.

6. Conclusions

An important emerging aspect in IoT smart object communities is the representation of mutual trust among community members. To address the limitations of current approaches in the literature, which often struggle to capture complex trust relationships in IoT smart object communities, we previously developed a model called the T-pattern. This model maintains trust information through a weighted graph, where nodes represent trust dimensions and edges represent relationships between these dimensions. The strength of the T-pattern model lies in its ability to derive one unknown trust dimension from another by leveraging their interdependence. To validate the effectiveness of T-patterns in inferring trust dimensions, this paper presents experiments conducted with an implementation of this model that uses a neural-symbolic approach as inference engine, which is based on CILIOS. These experiments, designed to estimate trust values for three specific trust aspects—honesty, safety, and competence—based on trustworthiness values, demonstrate the effectiveness and accuracy of this T-pattern implementation in reliably inferring trust.

We note that the calculation of trust measures is the hard part of the presented approach and that the use of neural networks is performed offline. However, we have shown how the training of the model can be executed in the background, while the agents interact in the IoT scenario, using the cloud level of the IoT system without interfering with the ordinary agent processes. The trained model is used when the training is completed and the simulation we have performed show that its exploitation is useful to improve the capability of the system to accurately determine the trust of the different smart objects.

Building on this line of research, after having verified the potentiality of our T-pattern model, our future studies will focus on implementing the proposed model by adopting new inference engines and considering additional scenarios. Moreover, for a more exhaustive analysis, we aim to evaluate our model under different error metrics and assessing its effectiveness with respect to other issues like computational efficiency (e.g., inference time and memory usage), robustness under noise or malicious input, and sensitivity to parameter changes.

Finally, we highlight that our proposal to apply T-patterns to the IoT environment is the first attempt we make at applying our trust model to a complex and distributed Information Systems scenario, and the simulations we have performed should be considered only as a verification of the feasibility of the idea. In our ongoing research, we are planning to apply the T-pattern model to a real IoT environment, directly observing the behavior of smart objects when interacting with each other, constructing the T-patterns based on these observations, and then computing the effectiveness of the model. Moving from simulations to a real environment implies facing novel issues that we did not deal with in this paper, as the management of real-word data noise and the scalability of the system with respect to the dimensions of the smart object population. Secondly, we highlight that the architecture of our system, which is delegated to the cloud-level construction of T-patterns with an inference engine that is used in the background with respect to other ordinary IoT activities, possess the capability to integrate with edge or fog computing frameworks as a system for performing real-time trust estimation. Moreover, it is also possible to study the possibility of combining T-pattern with reinforcement learning integration to create hybrid trust models that might enhance inference accuracy levels.

Author Contributions: All authors collaborated equally on this article. 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. 2024–2026 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: No new data were created or analyzed in this study.

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

Appendix A. Symbol List

For the sake of clarity, the list of adopted symbols is provided below.

Table A1. Symbol list.

Symbol	Meaning
E	the Environment
o	a smart IoT object
O	a set of smart IoT objects
δ	a trust issue
Δ	a set of trust issues
Λ	a group of smart IoT objects
G	a set of groups of smart IoT objects
ϕ	a confidence value
Φ	a set of confidence values
$\phi_{o_1,o_2,\Lambda}(\delta_i)$	the confidence of o_1 for δ_i w.r.t. o_2 within the group Λ
$\phi_{o_1,o_1,\Lambda}(\delta_i)$	the confidence of the group Λ for δ_i w.r.t. o_1
Γ	a mapping on a confidence $\gamma_{o_1,o_2,\Lambda}$ which gives a value ranging in $[1,5]$
τ	a T-pattern $\tau = \langle o_1, o_2, \Lambda, N_\Delta \rangle$ over two smart objects o_1 and o_2, a trust network N_Δ within a group Λ
T	a set of T-pattern
N_Δ	a weighted directed graph representing trust relationships
v	a vertex in N_Δ
V	a set of vertexes in N_Δ
l	a link in N_Δ
L	a set of links in N_Δ
w	a weight on a link $l \in N_\Delta$

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: New York, NY, USA, 2020.
11. Messina, F.; Rosaci, D.; Sarnè, G.M. Applying Trust Patterns to Model Complex Trustworthiness in the Internet of Things. *Electronics* **2024**, *13*, 2107. [CrossRef]
12. Rosaci, D. CILIOS: Connectionist inductive learning and inter-ontology similarities for recommending information agents. *Inf. Syst.* **2007**, *32*, 793–825. [CrossRef]
13. Demolombe, R. Reasoning about trust: A formal logical framework. In *Proceedings of the International Conference on Trust Management*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 291–303.
14. 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.
15. Baier, C.; Katoen, J.P. *Principles of Model Checking*; MIT Press: Cambridge, MA, USA, 2008.
16. Liu, X.; Datta, A.; Lim, E.P. *Computational Trust Models and Machine Learning*; CRC Press: Boca Raton, FL, USA, 2014.
17. Ma, W.; Wang, X.; Hu, M.; Zhou, Q. Machine learning empowered trust evaluation method for IoT devices. *IEEE Access* **2021**, *9*, 65066–65077. [CrossRef]
18. 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]
19. Palmer-Brown, D. Neural Networks for Modal and Virtual Learning. In *Proceedings of the Artificial Intelligence Applications and Innovations III*; Iliadis, L., Vlahavas, I., Bramer, M., Eds.; Springer: Boston, MA, USA, 2009; p. 2.
20. Liu, G.; Li, C.; Yang, Q. Neuralwalk: Trust assessment in online social networks with neural networks. In Proceedings of the IEEE INFOCOM 2019-IEEE Conference on Computer Communications, Paris, France, 29 April–2 May 2019; pp. 1999–2007.
21. Sagar, S.; Mahmood, A.; Wang, K.; Sheng, Q.Z.; Pabani, J.K.; Zhang, W.E. Trust–SIoT: Toward trustworthy object classification in the social internet of things. *IEEE Trans. Netw. Serv. Manag.* **2023**, *20*, 1210–1223. [CrossRef]
22. Awan, K.A.; Din, I.U.; Almogren, A.; Almajed, H.; Mohiuddin, I.; Guizani, M. NeuroTrust—Artificial-neural-network-based intelligent trust management mechanism for large-scale Internet of Medical Things. *IEEE Internet Things J.* **2020**, *8*, 15672–15682. [CrossRef]
23. Bhor, H.N.; Kalla, M. TRUST-based features for detecting the intruders in the Internet of Things network using deep learning. *Comput. Intell.* **2022**, *38*, 438–462. [CrossRef]
24. 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]
25. 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]
26. 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]
27. 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]
28. Guo, J.; Liu, Z.; Tian, S.; Huang, F.; Li, J.; Li, X.; Igorevich, K.K.; Ma, J. 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]
29. Guarino, N. *Formal Ontology in Information Systems: Proceedings of the First International Conference (FOIS'98), June 6–8, Trento, Italy*; IOS Press: Amsterdam, The Netherlands, 1998; Volume 46.
30. Java Agent DEvelopment Framework (JADE). 2024. Available online: http://jade.tilab.com/ (accessed on 10 January 2025).
31. Viljanen, L. Towards an ontology of trust. In *Proceedings of the International Conference on Trust, Privacy and Security in Digital Business*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 175–184.
32. Nagy, M.; Vargas-Vera, M.; Motta, E. Multi-agent Conflict Resolution with Trust for Ontology Mapping. In *Proceedings of the Intelligent Distributed Computing, Systems and Applications: Proceedings of the 2nd International Symposium on Intelligent Distributed Computing–IDC 2008, Catania, Italy, 2008*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 275–280.
33. Sadigh, B.L.; Unver, H.O.; Nikghadam, S.; Dogdu, E.; Ozbayoglu, A.M.; Kilic, S.E. An ontology-based multi-agent virtual enterprise system (OMAVE): Part 1: Domain model l ing and rule management. *Int. J. Comput. Integr. Manuf.* **2017**, *30*, 320–343. [CrossRef]
34. Amaral, G.; Sales, T.P.; Guizzardi, G.; Porello, D. Towards a reference ontology of trust. In Proceedings of the On the Move to Meaningful Internet Systems: OTM 2019 Conferences: Confederated International Conferences: CoopIS, ODBASE, C&TC 2019, Rhodes, Greece, 21–25 October 2019; Proceedings; Springer: Berlin/Heidelberg, Germany, 2019; pp. 3–21.

35. Barbosa, R.; Santos, R.; Novais, P. Trust-based negotiation in multiagent systems: A systematic review. In *Proceedings of the International Conference on Practical Applications of Agents and Multi-Agent Systems*; Springer: Berlin/Heidelberg, Germany, 2023; pp. 133–144.
36. Sobh, T.S. A secure and integrated ontology-based fusion using multi-agent system. *Int. J. Inf. Commun. Technol.* **2024**, *25*, 48–73. [CrossRef]
37. Zhang, S.; Dong, Y.; Zhang, Y.; Payne, T.R.; Zhang, J. Large Language Model Assisted Multi-Agent Dialogue for Ontology Alignment. In Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems, Auckland, New Zealand, 6–10 May 2024; pp. 2594–2596.
38. Burgess, M. *Thinking in Promises: Designing Systems for Cooperation*; O'Reilly Media, Inc.: Newton, MA, USA, 2015.
39. Burgess, M.; Fagernes, S. Autonomic pervasive computing: A smart mall scenario using promise theory. In Proceedings of the 1st IEEE International Workshop on Modelling Autonomic Communications Environments (MACE), Dublin, Ireland, 25–26 October 2006; pp. 133–160.
40. Burgess, M.; Dunbar, R.I. A promise theory perspective on the role of intent in group dynamics. *arXiv* **2024**, arXiv:2402.00598.
41. Burgess, M.; Dunbar, R.I. Group related phenomena in wikipedia edits. *arXiv* **2024**, arXiv:2402.00595.
42. Firesmith, D.; Henderson-Sellers, B.; Graham, I. *OPEN Modeling Language (OML) Reference Manual*; CUP Archive: Cambridge, UK, 1998.
43. McGuinness, D.L.; Fikes, R.; Hendler, J.; Stein, L.A. DAML+ OIL: An ontology language for the Semantic Web. *IEEE Intell. Syst.* **2002**, *17*, 72–80. [CrossRef]
44. Horrocks, I.; Patel-Schneider, P.F.; Boley, H.; Tabet, S.; Grosof, B.; Dean, M. SWRL: A semantic web rule language combining OWL and RuleML. *W3C Memb. Submiss.* **2004**, *21*, 1–31.
45. van der Hoek, W. Logical foundations of agent-based computing. In *ECCAI Advanced Course on Artificial Intelligence*; Springer: Berlin/Heidelberg, Germany, 2001; pp. 50–73.
46. Rao, A.S.; Georgeff, M.P. Decision procedures for BDI logics. *J. Log. Comput.* **1998**, *8*, 293–343. [CrossRef]
47. Alferes, J.J.; Pereira, L.M. Updates plus preferences. In Proceedings of the Logics in Artificial Intelligence: European Workshop, JELIA 2000, Málaga, Spain, 29 September–2 October 2000; Proceedings 7; Springer: Berlin/Heidelberg, Germany, 2000; pp. 345–360.
48. Extensible Markup Language (XML). 2024. Available online: https://www.w3.org/XML/ (accessed on 10 January 2025).
49. Liu, M.; Teng, F.; Zhang, Z.; Ge, P.; Sun, M.; Deng, R.; Cheng, P.; Chen, J. Enhancing cyber-resiliency of der-based smart grid: A survey. *IEEE Trans. Smart Grid* **2024**, *15*, 4998–5030. [CrossRef]
50. Atlam, H.F.; Walters, R.J.; Wills, G.B. Fog computing and the internet of things: A review. *Big Data Cogn. Comput.* **2018**, *2*, 10. [CrossRef]