IoT

MDPI

Article

Analysis of Feedback Evaluation for Trust Management Models in the Internet of Things †

Claudio Marche [1], Luigi Serreli [1] and Michele Nitti [1,2,*]

[1] DIEE, University of Cagliari, 09123 Cagliari, Italy; claudio.marche@unica.it (C.M.); l.serreli2@studenti.unica.it (L.S.)

[2] Research Unit of Cagliari, National Telecommunication Inter University Consortium, G.P.Usberti 181A, 43124 Parma, Italy

* Correspondence: michele.nitti@unica.it

† This paper is an extended version of our paper published: Nitti, M.; Girau, R.; Atzori, L.; Pilloni, V. Trustworthiness management in the IoT: The importance of the feedback. In Proceedings of the 2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN), Paris, France, 7–9 March 2017; doi:10.1109/ICIN.2017.7899434.

Abstract: The Internet of Things (IoT) is transforming the world into an ecosystem of objects that communicate with each other to enrich our lives. The devices' collaboration allows the creation of complex applications, where each object can provide one or more services needed for global benefit. The information moves to nodes in a peer-to-peer network, in which the concept of trustworthiness is essential. Trust and Reputation Models (TRMs) are developed with the goal of guaranteeing that actions taken by entities in a system reflect their trustworthiness values and to prevent these values from being manipulated by malicious entities. The cornerstone of any TRM is the ability to generate a coherent evaluation of the information received. Indeed, the feedback generated by the consumers of the services has a vital role as the source of any trust model. In this paper, we focus on the generation of the feedback and propose different metrics to evaluate it. Moreover, we illustrate a new collusive attack that influences the evaluation of the received services. Simulations with a real IoT dataset show the importance of feedback generation and the impact of the new proposed attack.

Keywords: Internet of Things; trust models; feedback evaluation; collusive attacks

check for updates

Citation: Marche, C.; Serreli, L.; Nitti, M. Analysis of Feedback Evaluation for Trust Management Models in the Internet of Things. *IoT* **2021**, *2*, 498–509. https://doi.org/10.3390/iot2030025

Academic Editor: Hyun-Ho Choi

Received: 8 July 2021
Accepted: 6 August 2021
Published: 11 August 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The Internet of Things (IoT) aims to embody into the internet a large number of heterogeneous and pervasive objects that, through standard communication protocols and unique addressing schemes, provide services to the final users [1]. The number and types of applications that can be realized using these technologies is ever increasing with the most powerful ones requiring things to collaborate by exchanging information and services, e.g., cars that exchange information about the traffic status and the sprinkler that obtains triggering events from distributed smoke detectors.

Such future IoT applications will likely be developed, making use of a service-oriented architecture, where each device can play the role of a service provider, a service requester, or both. IoT is moving towards a model where things look for other things to provide composite services for the benefit of human beings (object-object interaction) [2]. With such an interaction model, it is essential to understand how the information provided by each object can be processed automatically by any other peer in the system. This cannot clearly disregard the level of trustworthiness of the object providing information and services, which should take into account its profile and history. Trust and Reputation Models (TRMs) have the goal to guarantee that actions taken by entities in a system reflect their trustworthiness values, and to prevent these values from being manipulated by malicious entities.

The cornerstone to any TRMs is their ability to generate a coherent evaluation of the information received. From the analysis of the past works, an element that is frequently

missing and only sometimes superficially treated is the generation of feedback. Indeed, when an object receives the requested service, it needs to evaluate whether the service is consistent with its query and so rates it [3]. The feedback has the fundamental role of being the source of every trust model, which processes the ones received in the past to drive the computation of the trust level. Until now, the proposed algorithms have assumed that a node is able to perfectly rate the received service and only focus on the trust computation techniques.

Feedback generation is strictly tied to how well the information received matches the request. To this end, the requester has to compute a reference value, which is usually not available since an object does not have any knowledge regarding the ground-truth value of the requested information. Our paper works in this direction with the goal to estimate a reference value and to use it in order to assign feedback to the providers, and thus provides the following contributions:

- First, we define the problem of feedback evaluation in the IoT. We also propose different metrics to evaluate a reference value and how this value should be used to rate services.
- Second, we propose a new collusive attack on trust that aims to influence the evaluation of the reference value and thus to confuse the IoT network. Moreover, we test the resiliency of existing models against it and show how the proposed TRMs are not able to efficiently identify such an attack.
- Third, by using a dataset of real IoT objects, we conduct extensive experiments to analyze the impact of the proposed metrics and the eventual errors in the feedback assignment. Furthermore, we compare different TRMs in order to study the impact of the service and feedback evaluation.

The rest of the paper is organized as follows: Section 2 presents a brief survey on trust management models and, in particular, their approach to the problem of feedback evaluation. In Section 3, we define the problem, introduce the used notations and illustrate the proposed service and feedback evaluation model. Section 4 presents the model performance under different conditions, while Section 5 draws final remarks.

2. State of the Art

Trustworthiness is recognized as a critical factor for the Internet of Things, and the different algorithms used to implement it play an essential role [4]. Generally, trust among devices allows to share opinions between them and can be used by a requester to decide whether to require information from the provider [5]. In recent years, many researchers have examined the problem of trust, so the literature is now quite abundant. This section provides a brief overview regarding the background of trustworthiness management in the IoT. In particular, we focus on the use of feedback and its importance for trust algorithms, taking into consideration the main characteristics discussed in the following, and we do not intend to cover all of the published papers.

Dimension: The feedback is expressed by a value that may be continuous or discrete. The feedback values are generated on the basis of the considered model, and they are used to compose the trust values.

Sharing: The feedback could be available for all the nodes in the network or only for the releasing node. Generally, it is shared in order to improve a recommendation mechanism. Otherwise, it is not shared and is used only by the agent responsible for the trust values.

Metrics: The feedback consists of the evaluation of the interaction, and it is based on defined metrics. Usually, the models make use of QoS metrics, such as memory consumption or energy cost. However, often the authors do not depict how the feedback is generated and assume that an agent is able to perfectly evaluate a service without providing any explanation.

Source: The feedback could be released by different sources—an IoT node, an agent, or an end-user. For the decentralized mechanism, generally, the source might be the

requester node. Otherwise, usually in centralized models, an agent is responsible for feedback generation.

Among the analyzed works considering the concept of trust in the IoT, in [6] the authors propose a centralized trust management architecture based on IoT clusters for countering bad-mouthing attacks. The architecture allows IoT devices and applications to contact each other, making use of trust manager nodes responsible for the communication and data transporting. The trust manager generates the feedback and evaluates the interactions, thanks to QoS parameters, such as memory efficiency. In addition, it takes care of all the trust values and all clusters' management.

Another model against bad-mouthing attacks is described in [7]. The authors illustrate a decentralized trust management model, using direct and indirect observations. Each requester node evaluates the received service according to their resource-consuming capabilities and other QoS parameters (each service has different energy requirements, memory, and processing in a node). The produced feedback consists of a continuous value, where high values require more processing capacity, while differently low values do not require many resources. Nodes share feedback with each other, and thus, it is used to calculate trust values.

Among the approaches, those developed to oppose other types of attacks are described [8,9]. In the first work, the authors propose a hybrid trust protocol to allow users to query services toward IoT providers. Each user can evaluate a provider in a trusted environment, where the trust is composed of recommendations from neighbors and direct experience. The feedback reports are shared to a cloud server that takes care of all the trust values. Unfortunately, the authors do not provide any information about the feedback evaluation. In the second one, the authors take into account machine-learning principles to classify the trust features and combine them to produce a final trust value in a decentralized architecture. Each requester node selects a provider based on a machine-learning algorithm and considers feedback values in $\{0, 1\}$, where 0 represents a scarce service and 1 a good one.

Furthermore, relevant approaches developed for attacks in a specific scenario are presented in [10,11]. In the first paper, the authors illustrate a trusted approach for vehicle-to-vehicle communications, combining vehicle certificates and trust management. Vehicles communicate with each other in a peer-to-peer distribution model and can take decisions based on the reputation of all nodes. Each object evaluates by itself the feedback and shares it with all the other nodes in the network. Additionally, in [11], the authors depict a decentralized technique to improve performance in a vehicular network. An intelligent protocol is developed in order to achieve a good decision with inaccurate and erroneous information and adapt to a dynamic communication environment. Each node evaluates the feedback with QoS metrics, such as the data traffic or the link quality, and expresses it with a continuous variable. Any feedback is shared.

Moreover, two recent works, refs [12,13] illustrate a trust model for a general scenario. The authors in [12] propose a trust management framework for collaborative IoT applications. The trust of each node is based on past interactions, recommendations and QoS parameters. Moreover, the requester node evaluates the feedback and depicts it in a continuous range based on the specific scenario. The sharing through the neighbour nodes allows improving collaboration and resistance against collusive attacks. Furthermore, in [14], the authors develop a trust architecture that integrates software-defined networks to improve organization and reputation management. A reputation node in charge takes care of the nodes' trust, according to their operations. The node evaluates the interaction measuring node operations and considers the feedback into three layers: normal, fault and malicious.

In recent years, many researchers have tried to improve the reliability of trust models in terms of the ability to detect malicious behaviors, e.g., by making use of machine learning [15], or employing Markov matrices [16]. In the first work, the authors propose a decentralized trust management model for social IoT. With the benefit of social networks,

such as improved searching mechanisms and resources discovery, the approach attempts to detect the most trust attacks in the state of the art. Each requester node evaluates the interaction by itself in a continuous range $[0, 1]$ and shares the feedback with its neighbor and pre-trusted nodes. Any information about the metrics used for the evaluation is illustrated. In the second one, discrete feedback is used to evaluate the nodes' interactions in a scenario with various types of attacks. Each node evaluates the feedback and shares it with its neighbor nodes; this value and the recommendations are employed to model the individual trust value.

Table 1 shows a classification of the trust models based on the use of feedback values and takes into consideration the characteristics mentioned above.

Table 1. Comparison of trust management models: main aspects of the feedback.

Reference	Dimension	Sharing	Metrics	Source
Alshehri et al. [6]	n.a.	Trust Manager	QoS-based	Trust Manager
Mendoza et al. [7]	Continuous	Neighbor nodes	QoS-based	Requester node
Chen et al. [8]	Continuous	Cloud server	n.a.	Owner user
Jayasinghe et al. [9]	Discrete	Not shared	n.a.	Requester node
García et al. [10]	n.a.	All nodes	n.a.	Requester node
Wu et al. [11]	Continuous	Not shared	QoS-based	Requester Node
Adewuyi et al. [12]	Continuous	Neighbor nodes	n.a.	Requester node
Chen et al. [14]	Discrete	Reputation nodes	QoS-based	Reputation node
Marche al. [15]	Continuous	Neighbor/Pre-trusted nodes	n.a.	Requester node
Wang et al. [16]	Discrete	Neighbor/Pre-trusted nodes	n.a.	Requester node

3. Feedback Evaluation

3.1. Reference Scenario

The focus of this paper is to propose a feedback evaluation mechanism that is able to rate the service received by the requester.

In our model, the set of nodes in the network is represented by $\mathcal{N} = \{n_1, ..., n_i, ...n_I\}$ with cardinality I, where n_i is the generic node. Every node in our network can provide one or more services, so $\mathcal{S}_j$ is the set of services that can be provided by n_j. The reference scenario is then represented by an application installed in a node n_i, or in the connected cloud space, requesting a particular service S_h: a service discovery component in the network is able to return to n_i a list of potential providers $\mathcal{P}_h = \{n_j \in \mathcal{N} : S_h \in \mathcal{S}_j\}$. At this point, TRMs usually assume that the requester is able to perfectly evaluate the service, and then it has to select only one of the providers in $\mathcal{P}_h$ based on their level of trust. However, the requester does not know the ground-truth value v^h of the service and has, then, no means to evaluate whether the received service is good or not and its level of accuracy. In order to assess a value for the service to be used by the application, the requester has to contact more than one provider in $\mathcal{P}_h$: from every provider n_j, the requester receives a value v_j^h for the service S_h and then has to aggregate all the received values into a reference value v^{h*}. Moreover, to compute the trust level of every provider, a TRM has to rely on the previous interactions among the nodes in the form of feedback, which represents how a requester is satisfied with the received service. After every transaction, the requester n_i has to assign a feedback to all the providers to evaluate the service, based on the provided values v_j^h and on their "distance" from the computed reference value v^{h*}. Each feedback can be expressed using values in the continuous range $[0, 1]$, where 1 is used when the requester is fully satisfied by the service and 0 otherwise.

Figure 1 provides a simple example of a generic graph $\mathcal{N} = \{n_1, ..., n_1 1\}$, with each node capable of providing one or more services, as highlighted in the grey clouds; n_1 is the

node that is requesting the service S_7, as highlighted in the white cloud; $\mathcal{P}_h = \{n_5, n_6, n_9\}$ is the set of nodes that can provide the requested service. For each of the providers in $\mathcal{P}_h$, the requester n_1 receives a value for the service S_7 (red lines in the Figure), then aggregates them to compute the reference value v^{7*}. Based on this value, n_1 can rate the providers individually and assign to each of them feedback (dotted green lines in the Figure) that can be used to update their trustworthiness levels.

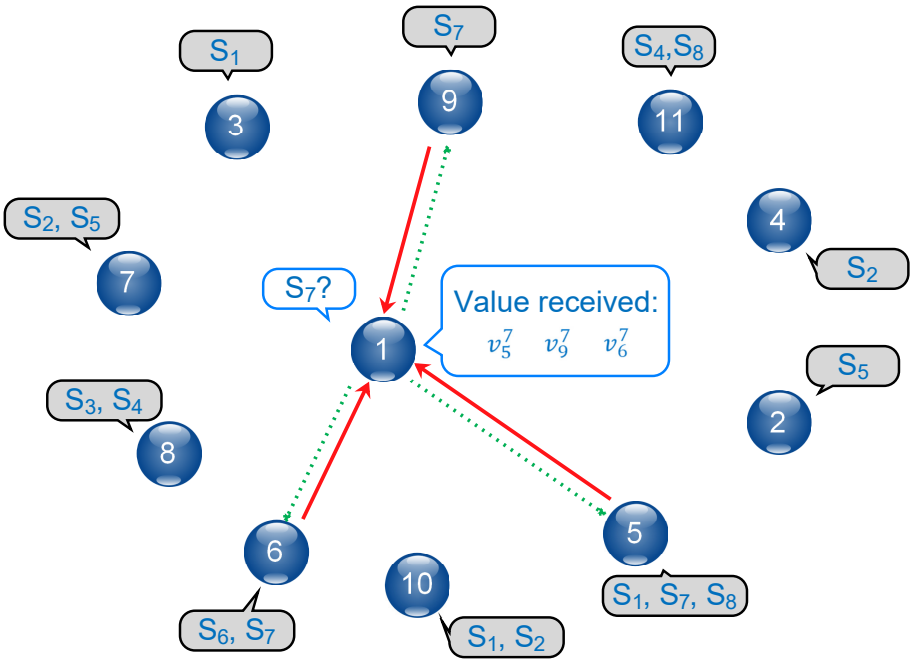

Figure 1. Reference scenario.

The goal of the feedback evaluation algorithm is two-fold: compute the reference value v^{h*} that will be used by the application and estimate the reliability of the providers. This step is fundamental to help the requester to assess a solid value for the requested service and to reward/penalize the providers so as to avoid any malicious node in future transactions.

3.2. Service and Feedback Evaluation Model

According to the presented scenario, we propose a service and feedback evaluation model, where each node, during a transaction, calculates the reference value v^{h*} to be passed to the application. This value is then used as a reference to compute the feedback that will be assigned to the different providers. In this way, every TRM can have information regarding the past interactions available to be used for the trust computation value.

Whenever a node n_i has the need to retrieve a service, it has to select a subset of providers from the list of potential providers $\mathcal{P}_h$. To this end, the requester immediately discards any provider with a trust value lower than a given threshold TH, so that the reliable providers are included in the set $\mathcal{T}_h = \{n_j \in \mathcal{P}_h : T_{ij} \geq TH\}$, where T_{ij} is the trust of node n_i toward node n_j. The value of the threshold has to be decided based on the TRM implemented in the system since different models have different dynamics to label a node as malicious. From this set, the requester contacts the most trustworthy providers of M in order to actually require a service's value.

When the requester receives all the service's values from the providers, it has to implement some mechanism in order to infer the reference value v^{h*} to be passed to the application. Several strategies can be used:

- Mean of all the values obtained by the M providers;
- Sum of all the values obtained by the M providers weighted by their trustworthiness;
- Median of all the values obtained by the M providers.

Since every application requires a certain accuracy, it is possible to assign feedback to the different providers. For simplicity, we consider the ground-truth value to be 0, and we always normalize the application accuracy so that values in the interval $[-1, 1]$ are acceptable by the application (blue area in Figure 2). The reference value v^{h*}, calculated with one of the metrics proposed above, can deviate from the ground-truth value; in this case, if v^{h*} is still within the application accuracy, i.e., $-1 \leq v^{h*} \leq 1$, the transaction is labeled as successful. Otherwise, it means the malicious attack was able to confuse the network and the transaction was unsuccessful.

The requester is unable to assess the outcome of the transaction, so despite its result, it has to assign feedback to each of the providers. The maximum feedback is assigned to those providers that sent exactly the reference value v^{h*}, while the other providers receive lower feedback based on how much the provided value is distant from v^{h*}, as shown by the orange areas in Figure 2. Nodes that have provided values with a distance equal to the accuracy of the application, i.e., $v^{h*} - 1$ and $v^{h*} + 1$, which represent the points of greatest uncertainty, are assigned feedback equal to 0.5. For intermediate values, the feedback follows a linear behavior, but other approaches are feasible and could also depend on the application at hand.

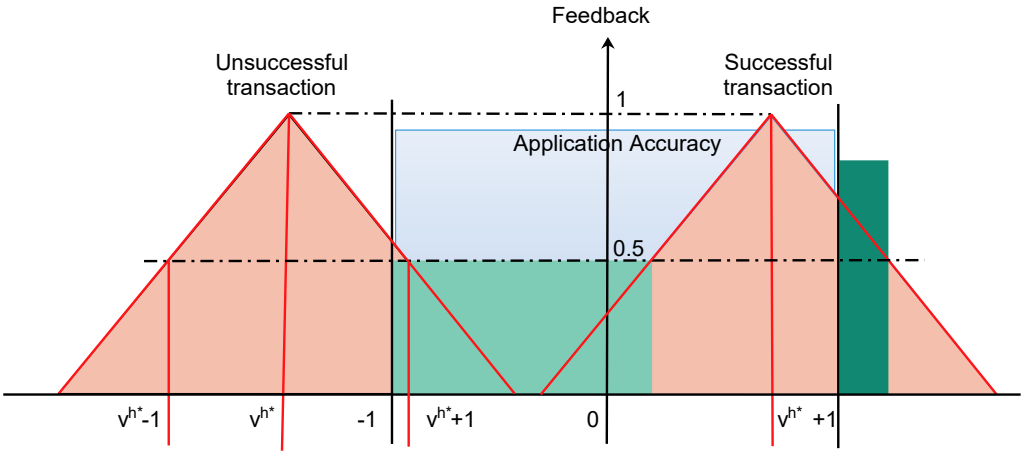

Figure 2. Feedback evaluation.

Due to the difference between the pseudo and the ground-truth value, feedback assignment leads to error in the evaluation of the providers: benevolent nodes that provided values in the light green area are given feedback lower than 0.5 and, in some cases, even 0, while malicious nodes that provided values unacceptable by the application, i.e., outside of the blue area of application accuracy, are given positive feedback, as it is the case of the dark green area in Figure 2.

The introduction of a feedback evaluation system leads to a new malicious behavior that would not be possible if the requester is able to perfectly rate the received service. Generally, malicious behavior is a strategic behavior corresponding to an opportunistic participant who cheats whenever it is advantageous for it to do so. The goal of a node performing maliciously is usually to provide low quality or false services in order to save its own resources; at the same time, it aims to maintain a high value of trust toward the rest of the network so that other nodes will be agreeable in providing their services when requested. A group of nodes (collusive attack) can work together to provide the same malicious value so as to influence the reference value v^{h*}, let the requester believe that it is the correct value to be passed to the application, and assign to them a positive feedback. To the best of our knowledge, this is the first time this attack is presented; therefore, TRMs were never tested against it and we do not know their ability to detect and react to such an attack.

4. Experiments and Results

4.1. Simulation Setup

In order to test our trustworthiness model, we need a large dataset of a SIoT scenario. To this end, we make use of the dataset made available by [17]; it consists of a network of 16,216 devices owned by 4000 users and by the municipality of Santander (Spain), who created their own relations over 11 days. Moreover, the authors share a set of real services and applications offered and requested by the nodes, which are useful to emulate interactions among nodes. We decide to consider only a connected sub-network of around 800 nodes to increase the probability of two nodes interacting with each other. Furthermore, a model of interaction among the nodes is also needed to understand which devices are more likely to interact; trust models are usually tested considering random interactions among nodes without taking into account the behavior of objects that generate queries of services when interacting with the other peers. To this end, we have adopted the query generation model presented in [17], so that at the start of each transaction, the simulator can choose the requester and select all the possible providers. Among all the providers, those up to M are contacted to require the service. The impact of this value is investigated in the first set of simulations. However, all the providers with a trustworthiness value lower than $TH = 0.2$ are automatically discarded and are not considered providers.

Two main behaviors are implemented in the network: one is cooperative and benevolent so that a node always provides good services acceptable by the application, i.e., within the interval $[-1; 1]$. However, in order to simulate different devices' accuracy, the values provided by the nodes follow a uniform or a normal distribution with different values of the variance σ^2, ranging from 0.20 to 0.40. The other behavior is a malicious one, where a node tries to disrupt the network by providing scarce services, i.e., services with values outside of the application accuracy. In particular, malicious nodes provide services with errors up to 300% of the application accuracy. We consider that 25% of the nodes in the network are malicious, implementing a different kind of attack [18].

Table 2 shows all the configuration parameters for the proposed simulations. Every set of simulations is repeated 10 times, and the values shown in the Figures are the average of the obtained values in each run.

Table 2. Simulation setup parameters.

Parameter	Value
Number of nodes	791
Percentage of malicious nodes	25%
Trustworthiness lower threshold (TH)	0.2
Percentage of errors provided by malicious nodes	300%
Variance of service generation distribution for benevolent nodes (σ^2)	$\{0.2, 0.3, 0.4\}$

4.2. Results

This section aims to evaluate the performance of the service and feedback evaluation. The first set of simulations aims to analyze the transaction success rate, i.e., the ratio between the number of successful transactions and the total number of transactions: a transaction is considered successful if the reference value is within the application accuracy. Figure 3 shows the performance of the Marche et al. trust algorithm [15] when the requester can perfectly evaluate the service received and when it implements one of the three possible strategies to infer the reference value and for different values of the numbers of providers M. The mentioned model is designed for a social IoT scenario and makes use of a subjective approach, where every node has its own vision of the network and relies on the recommendations from its friends to speed up the evaluation of trust. In these simulations, malicious nodes implement a strategic behavior corresponding to an

opportunistic participant who acts maliciously with everyone. This is the most basic attack: a node always provides bad services and recommendations, regardless of the requester.

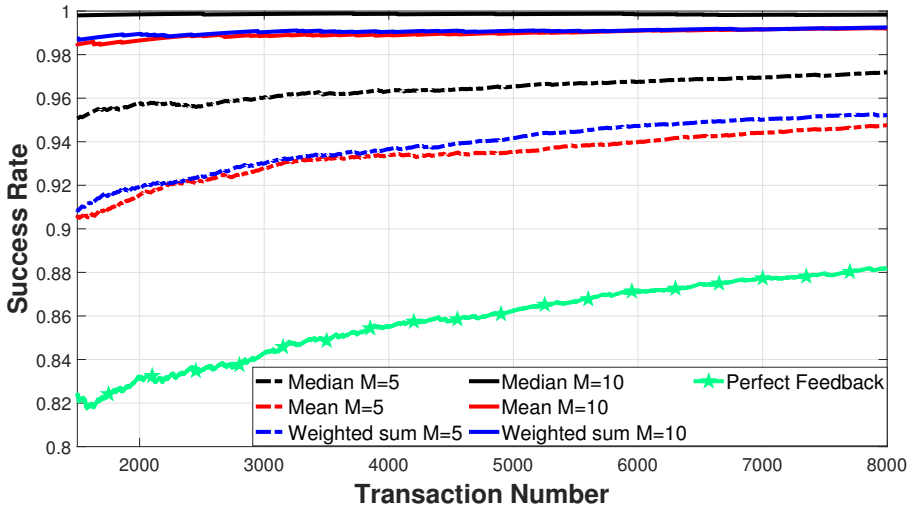

Figure 3. Transaction success rate for all strategies.

Surprisingly, the trust model that makes use of a perfect feedback evaluation has the lowest success rate out of all the versions: this is related to the number of providers M contacted by the requester. A requester with perfect feedback evaluation only interacts with one provider and thus they will need many transactions to accumulate enough experience to accurately evaluate the providers. In the approach proposed in this paper, multiple providers are required to infer the reference value and, even if each of them receives not perfect feedback, the overall learning process of the trust algorithm speeds up. Moreover, the presence of a higher number of providers enables increasing the success rate since it is difficult for few malicious nodes to change the reference value enough for the transaction to be considered malicious. The approach based on the median, dark lines in Figure is the most reliable since there must be $M/2 + 1$ malicious nodes in order for the transaction to be labeled as unsuccessful; using the mean is not ideal since a single malicious node providing a value very distant from the ground-truth value could significantly change the reference value. Finally, the weighted sum is able to obtain good results since it smooths out this effect by weighting every received value for the trustworthiness of the node that provided it.

We evaluated the system performance by analyzing the mean of malicious nodes per transaction, the mean and standard deviation of the reference value, the reference value errors, and the number of errors in the feedback assignment. This last parameter considers both when a malicious node gains a good score, i.e., feedback higher than 0.5, and when a benevolent node receives feedback lower than 0.5. Table 3 shows these results when considering 25% of malicious nodes but with two different types of attacks: the first six simulations implement the basic attack with malicious nodes always providing bad services, while the last six simulations show what happens with the collusive attack described in Section 3.2. Coherently with Figure 3, when implementing only the basic attack, the median is the approach with the lowest percentage of malicious transactions, which means that it is able to efficiently isolate malicious nodes. This can also be inferred by the accuracy of the reference value and by the number of times it is outside of the bounds accepted by the application. Moreover, it confirms how selecting a higher number of providers leads to better results. Differently, if malicious nodes implement the collusive behavior, all the strategies show worse performance: if we consider the error percentage in calculating the reference value, i.e., the number of times the reference value is outside of the application accuracy, we can notice how this percentage is an order of magnitude higher w.r.t. the basic attack; in particular, the median approach with $M = 5$ goes from

3.33% errors to 16.48%, thereby showing the threat of the collusive behavior. Overall, the median approach involves a higher number of malicious nodes per transaction and it has more difficulty in correctly evaluating malicious nodes, as can be inferred by the error percentage in providing feedback, which is higher w.r.t. to the other two approaches. However, the median approach still shows the lowest percentage of errors when computing the reference value.

Table 3. Comparison between simple and collusive attacks.

	Malicious Nodes per Transaction [%]	Mean v^{h*}	SD (v^{h*})	Reference Value Errors [%]	Feedback Errors Malicious vs. Benevolent [%]
Median, M = 10	9.9	0.0025	0.27	0.16	1.57–13.97
Mean, M = 10	10.1	0.0041	0.44	0.81	2.11–15.05
Weighted sum, M = 10	10.1	0.0028	0.43	0.76	2.02–14.85
Median, M = 5	12.4	0.0346	0.79	3.33	3.44–18.81
Mean, M = 5	12.2	0.0305	0.81	5.52	3.08–18.36
Weighted sum, M = 5	12.4	0.0161	0.81	5.15	3.44–20.07
Median, M = 10, Collusive	11.3	0.0050	0.98	5.15	4.60–16.60
Mean, M = 10, Collusive	10.9	0.0035	0.83	7.73	3.31–19.74
Weighted sum, M = 10, Collusive	11.0	0.0228	0.84	8.01	3.49–19.85
Median, M = 5, Collusive	15.4	0.0060	1.41	14.68	9.26–26.66
Mean, M = 5, Collusive	14.6	0.0284	1.36	15.87	7.21–26.86
Weighted sum, M = 5, Collusive	14.6	0.0226	1.38	16.43	7.56–26.69

In order to better understand this behavior, we tested the performance of the trust algorithm at varying the percentage of the malicious nodes. Figure 4 refers to a scenario where all malicious nodes implement the collusive behavior: on the left (Figure 4a), there is the case with $M = 5$ providers, while on the right (Figure 4b) we consider $M = 10$ providers. Two approaches are evaluated: the median (solid lines) and the weighted sum (dotted lines). The figure confirms the table above: with more providers, the trust algorithm performs slightly better but at the expense of more traffic exchanged among the requester and providers. However, when comparing these results with the original trust algorithm, which implemented a perfect evaluation of the service received (xxx line in Figure), we can notice how the success rate greatly decreases: for 50% of malicious nodes, the algorithm proposed by Marche et al. shows a success rate higher than 80%, while, with the collusive attack, it is not able to even reach 60% of successful transactions with a decrease of over 20%.

Finally, the last set of simulations aims to understand how different trust algorithms react with 20% of nodes implementing the collusive attack and considering the median as the strategy to infer the reference value. Other than the algorithm of Marche et al., we have also implemented three other algorithms, namely those of Chen et al. [19], Adewuyi et al. [12] and Mendoza et al. [7].

The model proposed by Chen et al. is similar to the one described by Marche et al.: they are both designed for a social IoT environment, where each node computes the trust value of the rest of the network by itself, so that each device has subjective values of trust toward the rest of the network. Those of Adewuyi et al. and Mendoza et al. are non-social algorithms, based on a distributed approach, which rely on recommendations from neighbor nodes and QoS parameters: trust is computed making use of a weighted sum among these parameters, but while Adewuyi et al. focus on historical interactions, making use of a time window of past interactions, Mendoza et al. concentrate only on the last transaction and the type of service. Differences in the performance of the models can depend on the structure of the network considered and on the types of service/information

requested. To this end, we did not consider our ad-hoc network, but we have adopted the IoT dataset opportunistically re-scaled to a size comparable to their experiments, as described in the previous subsection. Moreover, we have considered the same requests for all the four models, so we are confident that the obtained results are consistent with those obtained by the authors.

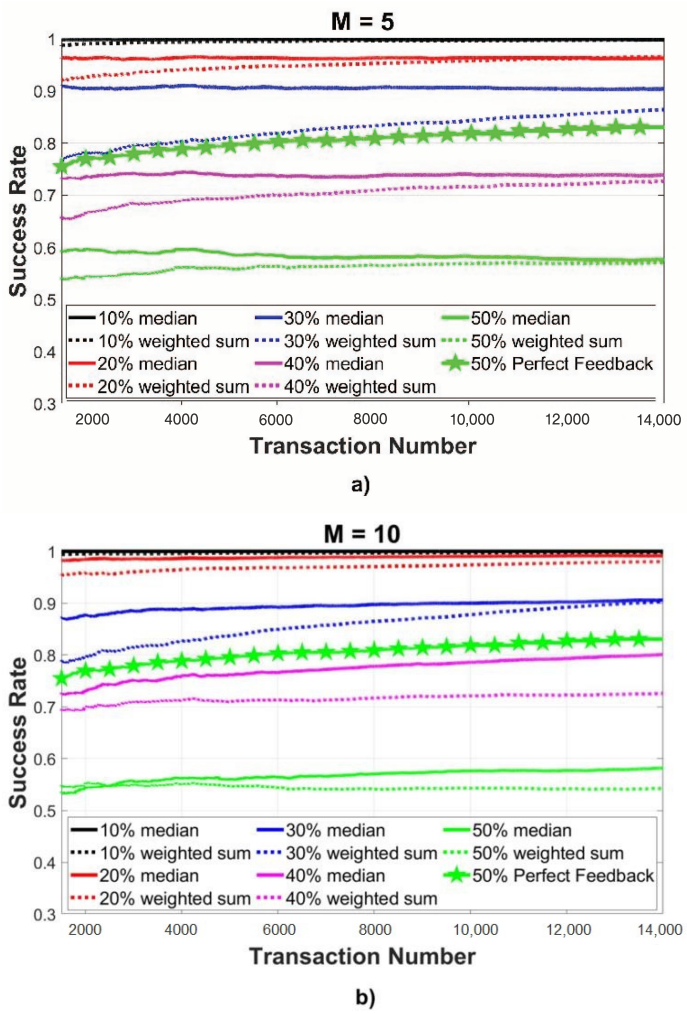

Figure 4. Transaction success rate at increasing values of % of malicious nodes, with $M = 5$ (**a**) and $M = 10$ (**b**) providers.

Figure 5 shows the comparison of the four trust models, considering a number of providers $M = 10$. We can notice that all the algorithms have a high success rate of over 90%, but Chen et al. and Mendoza et al. show an SR decreasing over time. This is due to the collusive attack that is able to confuse the network. The two mentioned algorithms have a short dynamic, i.e., that trust values of both benevolent and malicious nodes, and are concentrated around 0.5, so they are more prone to errors. Whenever there is a transaction where the collusive malicious attack is successful, the malicious nodes are able to obtain the highest value of feedback, since the reference value would correspond with the malicious value provided by all the nodes. So, even if really slowly, these nodes would impact more and more transactions, thus making the success rate decrease over time.

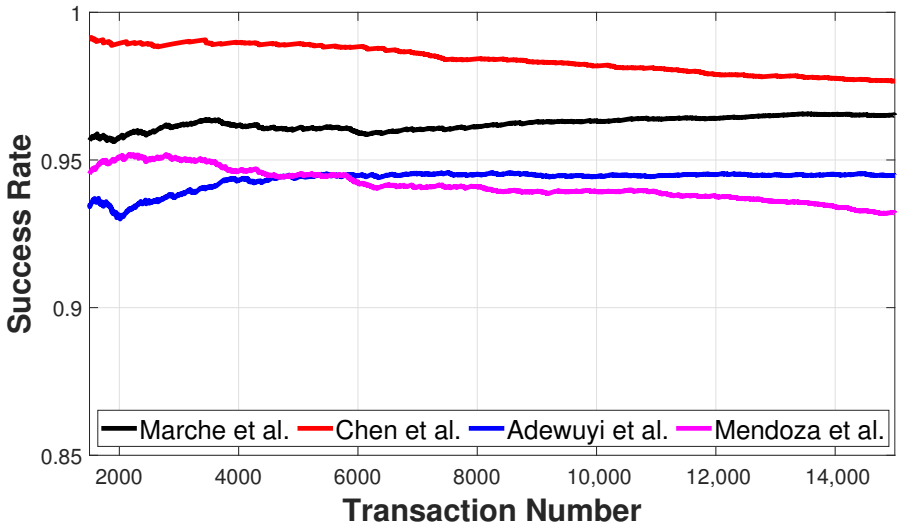

Figure 5. Transaction success rate for different trust management models.

5. Conclusions

In this paper, we have defined the problem of feedback evaluation in the IoT and proposed different metrics to evaluate a reference value that should be used to rate the service received by the providers. The introduction of a mechanism to assign feedback to the services providers leads to error in the evaluation of the providers, due to the difference between the pseudo and the ground-truth value. This means that, even without any malicious nodes, benevolent devices could be assigned with negative feedback. Furthermore, we have observed that, regardless of the ability to provide perfect feedback, choosing more providers enables trust management systems to converge faster. Among the different approaches proposed, the median one is the most reliable since there must be $M/2+1$ malicious nodes in order for the transaction to be labeled as unsuccessful. However, by choosing more providers, a new malicious attack can be proposed: it is a collusive attack on trust, where a group of nodes works together to provide the same malicious value. The attack aims to influence the evaluation of the reference value and, thus, to confuse the IoT network. The proposed service and feedback evaluation methods were applied to well-known trust algorithms; the experiments have shown their importance and the challenges of generating consistent feedback for the trust evaluation in the exchange of services in IoT.

We plan to extend our methods by considering a more realistic scenario. In particular, we want to differentiate the accuracy of each application and consider that each node has different accuracies based on the service provided. This way, also benevolent nodes with low accuracy can provide values outside the range accepted by the application and, therefore, receive low ratings. We then expect that the goal of TRMs will shift: they will not only have to prevent malicious entities from hampering the services, but they will have to select the best provider based on the application at hand.

Author Contributions: Software, C.M. and L.S.; Writing—original draft, and Writing—review & editing, C.M., L.S. and M.N.; validation and visualization, C.M. and M.N.; formal analysis, and supervision, M.N. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported partially by Italian Ministry of University and Research (MIUR), within the PON R&I 2014–2020 framework (Project AIM (Attrazione e Mobilità Internazionale)) and has partially received funding from the European Union's Horizon 2020 research and innovation program under Grant Agreement No. 957228.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available in http://social-iot.org/index.php?p=downloads at [10.1016/j.comnet.2020.107248].

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

References

1. Lin, J.; Yu, W.; Zhang, N.; Yang, X.; Zhang, H.; Zhao, W. A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications. *IEEE Internet Things J.* **2017**, *4*, 1125–1142. [CrossRef]
2. Al-Fuqaha, A.; Guizani, M.; Mohammadi, M.; Aledhari, M.; Ayyash, M. Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 2347–2376. [CrossRef]
3. Nitti, M.; Girau, R.; Atzori, L.; Pilloni, V. Trustworthiness management in the IoT: The importance of the feedback. In Proceedings of the 2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN), Paris, France, 7–9 March 2017; pp. 325–327.
4. 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]
5. Pourghebleh, B.; Wakil, K.; Navimipour, N.J. A comprehensive study on the trust management techniques in the internet of things. *IEEE Internet Things J.* **2019**, *6*, 9326–9337. [CrossRef]
6. Alshehri, M.D.; Hussain, F.K.; Hussain, O.K. Clustering-driven intelligent trust management methodology for the internet of things (CITM-IoT). *Mob. Netw. Appl.* **2018**, *23*, 419–431. [CrossRef]
7. Mendoza, C.V.L.; Kleinschmidt, J.H. A distributed trust management mechanism for the Internet of things using a multi-service approach. *Wirel. Pers. Commun.* **2018**, *103*, 2501–2513. [CrossRef]
8. Chen, R.; Guo, J.; Wang, D.C.; Tsai, J.J.; Al-Hamadi, H.; You, I. Trust-based service management for mobile cloud IoT systems. *IEEE Trans. Netw. Serv. Manag.* **2018**, *16*, 246–263. [CrossRef]
9. Jayasinghe, U.; Lee, G.M.; Um, T.W.; Shi, Q. Machine learning based trust computational model for IoT services. *IEEE Trans. Sustain. Comput.* **2018**, *4*, 39–52. [CrossRef]
10. García-Magariño, I.; Sendra, S.; Lacuesta, R.; Lloret, J. Security in vehicles with IoT by prioritization rules, vehicle certificates, and trust management. *IEEE Internet Things J.* **2018**, *6*, 5927–5934. [CrossRef]
11. Wu, C.; Liu, Z.; Zhang, D.; Yoshinaga, T.; Ji, Y. Spatial intelligence toward trustworthy vehicular IoT. *IEEE Commun. Mag.* **2018**, *56*, 22–27. [CrossRef]
12. Adewuyi, A.A.; Cheng, H.; Shi, Q.; Cao, J.; MacDermott, Á.; Wang, X. CTRUST: A dynamic trust model for collaborative applications in the Internet of Things. *IEEE Internet Things J.* **2019**, *6*, 5432–5445. [CrossRef]
13. Xu, X.; Liu, X.; Xu, Z.; Dai, F.; Zhang, X.; Qi, L. Trust-oriented IoT service placement for smart cities in edge computing. *IEEE Internet Things J.* **2019**, *7*, 4084–4091. [CrossRef]
14. Chen, J.; Tian, Z.; Cui, X.; Yin, L.; Wang, X. Trust architecture and reputation evaluation for internet of things. *J. Ambient Intell. Humaniz. Comput.* **2019**, *10*, 3099–3107. [CrossRef]
15. Marche, C.; Nitti, M. Trust-related Attacks and their Detection: A Trust Management Model for the Social IoT. *IEEE Trans. Netw. Serv. Manag.* **2020**. [CrossRef]
16. Wang, B.; Li, M.; Jin, X.; Guo, C. A reliable IoT edge computing trust management mechanism for smart cities. *IEEE Access* **2020**, *8*, 46373–46399. [CrossRef]
17. Marche, C.; Atzori, L.; Pilloni, V.; Nitti, M. How to exploit the Social Internet of Things: Query Generation Model and Device Profiles' Dataset. *Comput. Netw.* **2020**, *174*, 107248. [CrossRef]
18. Altaf, A.; Abbas, H.; Iqbal, F.; Derhab, A. Trust models of internet of smart things: A survey, open issues, and future directions. *J. Netw. Comput. Appl.* **2019**, *137*, 93–111. [CrossRef]
19. Chen, Z.; Ling, R.; Huang, C.M.; Zhu, X. A scheme of access service recommendation for the Social Internet of Things. *Int. J. Commun. Syst.* **2016**, *29*, 694–706. [CrossRef]