COMMUNICATIONS IN
SCIENCE AND TECHNOLOGY
Homepage: cst.kipmi.or.id

QS-Trust: An IoT ecosystem security model incorporating quality of service and social factors for trust assessment

Warsun Najib*, Selo Sulistyo, Widyawan

Department of Electrical and Information Engineering, Universitas Gadjah Mada, Yogyakarta 55281, Indonesia

Article history:
Received: 1 April 2024 / Received in revised form: 20 June 2024 / Accepted: 23 June 2024

Abstract

In the rapidly growing and increasingly complex Internet of Things (IoT) ecosystem, securing communication and data exchanges between devices is a major concern. To address this, we proposed QS-Trust, a trust-based security model considering both Quality of Service (QoS) and social parameters. QS-Trust uses a trust value to determine the trust level between devices and employs a QoS-aware trust-based algorithm to improve the security of data transmissions. Additionally, the model incorporates intelligence parameters such as computing power, memory capacity, device behavior and context information to enhance the accuracy of trust evaluation. Our simulation results demonstrated that QS-Trust effectively improved the security of the IoT ecosystem while maintaining the high level of QoS. The execution time of QS-Trust was in the range of 21 to 128 milliseconds, which is efficient for real-time IoT applications. QS-Trust offers a promising solution for securing the IoT ecosystem. The QS-Trust model effectively addresses the challenges of maintaining accurate and up-to-date trust levels in dynamic IoT environments through its decentralized approach, multi-factor evaluations, and adaptive algorithms. By continuously monitoring device performance and interactions and dynamically adjusting trust scores, QS-Trust ensures that the IoT network remains secure and reliable.

Keywords: Internet of things; IoT security; QoS security model; trust based IoT security

1. Introduction

Standard security protocols designed for information technology cannot be straightforwardly utilized for Internet of Things (IoT) systems. This divergence stems from the fact that IoT entities exhibit dynamic behaviors distinct from conventional computer networks. Elements within IoT systems possess attributes marked by dynamism, involving frequent associations and disassociations, alongside constraints on resources such as battery power, processing capabilities, and memory [1]. Furthermore, the connectivity of communication networks within IoT ecosystems often exhibits volatility due to the intermittent availability of communication links. The distinct characteristics of IoT and its associated security challenges, as a consequence, diverge from those encountered in typical computer networks. A notable vulnerability faced by IoT pertains to privacy breaches [2]. To mitigate this, it becomes imperative to impose restrictions on data and user access, ensuring that interactions among IoT entities remain confined to trusted counterparts.

Trust-based security is a type of security model based upon the principles of trust and reputation. In a trust-based security system, entities are given a certain level of trust or reputation based on their capability, previous actions and interactions with other entities in the system [3]. In a trust-based security model, trust is established through a variety of means, such as identity verification [4,5], relationship verification [6], and reputation verification [7,8].

There are several shortcomings identified in current trust and security solutions. Conventional trust evaluation schemes are dependent upon fixed, predetermined thresholds, rules, and static models, which might be inappropriate for dynamic and heterogeneous IoT environments [9,10]. In dynamic IoT environment where devices frequently join and leave the network [2,11,12], maintaining accurate and up-to-dated trust level can be challenging. Trust models must quickly adapt to the changes to remain effective. Our QS-Trust model solves this problem by maintaining the trust value of each IoT devices based on interaction history between devices.

Traditional models often rely on binary or coarse-grained trust metrics, which can be insufficient for accurate trust evaluation [13]. This approach can be insufficient for accurate trust evaluation in IoT environments. Typically, these models

* Corresponding author. Telp.: +62-274-552305
Email: warsun@ugm.ac.id

https://doi.org/10.21924/cst.9.1.2024.1419

classify interactions simplistically, either as trustworthy or as untrustworthy without considering the nuanced behavior and varying degrees of reliability exhibited by devices and users. As IoT ecosystems grow and diversify, the need for more granular and detailed trust metrics becomes evident. The detailed Quality of Service (QoS) metrics and sophisticated algorithms, like those used in our QS-Trust, provide a more comprehensive and accurate evaluation of trust by accounting for a range of factors such as computing power, memory capacity, and social interactions, which are critical for ensuring robust and secure IoT operations.

Based on the architecture, trust models can be divided into centralized [14,15,16] and decentralized architecture [5], [17,18]. Centralized architecture has several disadvantages such as single point of failure, scalability issue, and massive storage requirement. Due to the dynamic behaviors of sensor nodes and their resources limitation, establishing reliable end-to-end communication channels especially with external nodes, could be either unachievable or prohibitively costly. Therefore, in our trust model, we utilized a decentralized approach by defining IoT community and IoT ecosystem.

A security framework founded on trust forms a strategy for establishing a secure Internet of Things (IoT) setting. It can alternatively be perceived as a technique for gauging the trustworthiness of an entity within an IoT framework. The levels of assurance are derived from computations, both immediate and drawn from past encounters [19,20,21]. The outcome of these trust evaluations serves as an endorsement for entities deemed reliable.

The primary objective of this research is to propose a trust model as a solution for the current research gap as mentioned previously by incorporating both Quality of Service (QoS) and social dimensions as parameters in trust value computation. The proposed method demonstrated superior speed compared to prior research efforts. Additionally, it was proven valuable for coordinator selection within an IoT community, offering enhanced granularity in election scoring. The experiment also demonstrated the algorithm's efficiency in processing time across various IoT devices.

2. Materials and Methods

In this study, we proposed a model representing a collective assembly of IoT devices, forming what we term as an "IoT Community". The "IoT Ecosystem" concept was introduced to describe a setting in which IoT devices function as a part of an IoT system. Fig. 1 shows a common representation of an IoT ecosystem. This ecosystem consists of three interconnected IoT communities that work together to provide a specific service, forming the IoT landscape.

The main purpose of the security model is to provide a secure environment that guarantees the protection of operations, interactions, and data transfers within IoT systems, defending them against a broad range of potential threats. This model is designed to enhance protection against the breaches of data privacy as well.

2.1. IoT Community

Within the proposed model, a grouping of IoT entities represented as nodes constitutes an entity referred to as an "IoT community." An IoT community, denoted as "N," is defined in the following manner:

$$N = \{d_1, d_2, d_3, \ldots, d_M\} \tag{1}$$

In this context, the symbol d_i signifies the identity of a generic IoT entity, while M represents the total number of entities within a community. Within this framework, we defined the network using a graph denoted as G = {N, E}, where $E \subseteq \{N \times N\}$ represents the set of edges. Each edge within this set signifies a relationship between an IoT node and other nodes within the network.

We defined S_j as a set of services that can be provided by d_j object. The consumer of the service is represented by d_i, which requests a particular service S_h. We assumed that the service discovery module in the IoT system receives the request of this service from d_i and returns a set of nodes $Zh = \{d_j \in N : S_h \in S_j\}$ to it that provide the service S_h.

An IoT community has an ability to connect and interact with other communities, thereby forming an IoT ecosystem. In this study, the phrase "IoT ecosystem" was used to depict an environment where devices in an IoT system or application operated. IoT Ecosystem E is defined as follows (Equation 2).

$$E = \{N_1, N_2, N_3, \ldots, N_k\} \tag{2}$$

Fig. 1 illustrates an Internet of Things (IoT) ecosystem, which is made up of three separate communities. An IoT ecosystem refers to a group of IoT communities that collaborate and interact to accomplish a specific objective and deliver certain services.

Each IoT community consists of a set of devices or objects that communicate with each other within a specific region or system. An IoT coordinator is a specific device or object within an IoT community chosen to play the role of managing interactions with other communities. This coordinator is typically selected from nodes that possess superior computing resources, such as processing power, memory, and battery life, and that have the highest trust level. A node or object, in this context, refers to a tangible device, which could be a smart device, sensor, or actuator.

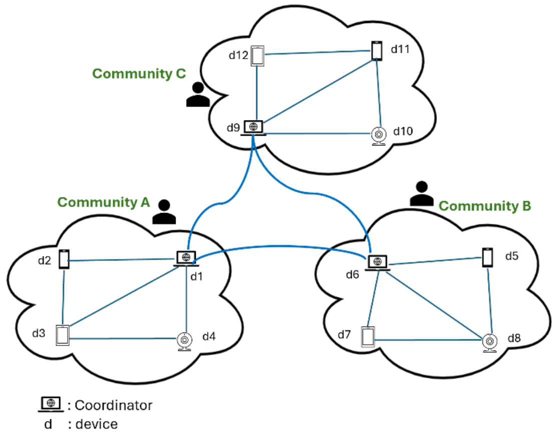

Fig. 1. IoT Ecosystem model

2.2. Trust parameter

Trust parameters are metric that help to ascertain the trustworthiness of a device for inclusion in the trust model. The trust calculation model utilizes two types of metrics: those based on the quality of service (QoS) and those rooted in social aspect of the IoT system.

[21] has defined a trust composition model, which requires entities to determine the relevant trust properties for calculating trust. This model encompasses both Quality of Service (QoS) trust model and Social trust model.

- *QoS model*: It refers to the degree of confidence that one entity in the IoT has in another entity's ability to properly perform its functions. This model evaluates trust by considering a number of factors such as competence, cooperativeness, reliability, and the capability to complete tasks [22].
- *Social model*: It depicts the social connections between the owners of IoT entities, and this mapping is used to determine the trustworthiness of an IoT entity. To calculate trust, these models take into account trust properties such as intimacy, honesty, privacy, centrality, and connectivity [23,24].

QoS provided by an IoT device greatly determines the security aspect. A device with a high level of QoS (e.g. computing capability, bandwidth and memory capacity) can respond quickly when a security threat occurs and can meet the algorithm's computing needs quickly. Also, IoT devices can be viewed as entities that have social relationships. For instance, when a device has a good reputation based on previous interaction history, it will have a good reputation; therefore, it obtains a high trust value.

2.2.1. Centrality

Entities (or nodes) within IoT systems frequently have numerous connections to other entities. They also partake in interactions, either to request or to offer a service. The centrality of an entity offers insights about the community, given that the entity has numerous relationships or is involved in many transactions. Such entity is considered to have a central position within the community [20,25]. As shown in Table 1, this parameter is classified and ranged accordingly. The centrality is scored to a range from zero to one and computed based on the number of entities connected to this particular entity.

Table 1. Trust value based on the number of connections.

Number of connections	Value
$1 < C \leq 20$	0.0 - 0.3
$20 < C \leq 50$	0.3 – 0.5
$50 < C \leq 100$	0.5 – 1.0
$C > 100$	1.0

Centrality offers insights into the social network by indicating that a node with numerous connections or transactions is likely to occupy a central position within the network. As noted in [26], it is linked to group efficiency in problem-solving, leadership perception, and the personal satisfaction of participants. The trust value is normalized into a range from 0.0 to 1.0. By normalizing the degree centrality and mapping it to specified trust value ranges, we can ascertain the trust value of a node in an IoT network. This approach ensures that more central nodes, which are likely to be more influential and involved in more transactions, are assigned higher trust values, reflecting their importance and reliability within the network.

2.2.2. Intelligence

One metric under the QoS classification is device intelligence. The computational abilities of an IoT device are largely dependent on the resources available to the device. Trust values derived from computational capabilities has a static value. This makes sense as IoT objects within an application are typically installed based on specific specifications, which include their power and resources (such as RAM, CPU, and battery). Table 2 presents the classification of IoT devices according to their intelligence factor.

Table 2. Trust score based on computing power.

Device Class	Device Type	Score
Class-1	Desktop, laptop	1.0
Class-2	Smartphone	0.8
Class-3	Smartwatch, smart camera, set top box	0.6
Class-4	Sensor	0.4
Class-5	RFID	0.2

Desktops and laptops generally have the highest computing power and memory capacity among IoT devices. They can handle any complex tasks and large volumes of data efficiently, making them highly reliable and trustworthy. Smartphones, while slightly less powerful than desktops and laptops, still offer significant computing capabilities and memory. They are versatile and widely used, making them relatively trustworthy. Devices in class-3 category have moderate computing power and memory. They are specialized for specific functions and generally perform well within their intended scope, but may not be as robust as higher-class devices. Sensors typically have limited computing power and memory for being designed to collect and transmit data rather than process it extensively. Their trust value reflects their role as data collectors rather than processors. RFID devices have minimal computing power and memory. They are primarily used for identification and basic data transmission. Their trust value is the lowest due to their limited capabilities.

Table 3. Trust score based on device's memory size.

Memory Size	Score
0 < memory < 2 GB	0.2
2 GB ≤ memory < 4 GB	0.4
4 GB ≤ memory < 8 GB	0.8
8 GB ≤ memory < 16 GB	0.9
Memory > 16 GB	1.0

Table 3 describes scoring for IoT objects based on the memory capacity owned by the objects. The memory capacity of IoT devices is vital, especially when the object acts as a coordinator on its IoT community. The larger the memory size, the greater the ability of the object to become a community coordinator. The coordinator election process will use this parameter as input metrics.

Memory capacity is also a crucial factor in determining the trustworthiness of IoT devices. Devices with higher memory capacities can handle more data and perform more complex operations, thus enhancing their reliability and trustworthiness. Devices with memory capacities up to 2 GB are generally limited in their data processing capabilities. They are often used for basic functions such as simple sensors or RFID tags. Their trust value is lower because they cannot handle any complex tasks and have limited capacity for secure data handling. Devices with more than 16 GB of memory have the highest processing power and capacity for secure data handling. They are the most reliable and trusted within the network.

3. Results and Discussion

This chapter discusses the results of this research including the coordinator election process, trust assessment, and the simulation results of the proposed trust-based security model.

3.1. Coordinator election of IoT community

In an IoT community, a coordinator is elected to manage the network and ensure smooth communication between connected devices. The selection of a coordinator is based on several parameters, including processor capabilities, memory capacity, and the number of connected objects. The processor and memory determine the coordinator's ability to handle the data processing and storage needs of the network. The more powerful the processor and larger the memory, the better the coordinator to handle the demands of the network.

The quantity of interconnected objects holds significant importance in the selection process of the coordinator. A coordinator with a larger number of connected objects is better equipped to handle the communication needs of the network. Overall, the coordinator with the best combination of processor capabilities, memory capacity, and the number of connected objects is typically elected to manage the network.

Each IoT community has a coordinator elected from IoT objects of the community's members. Unlike [25] and [27] which only considered centrality, our coordinator election algorithm was conducted based upon three parameters: device class, memory size, and number of connections.

IoT community coordinator election uses the following algorithm:

Step 1: Initialisation.

a. Initializing a list of devices in the IoT community that are eligible to be coordinators.

b. Setting object properties including object-Id, device class, memory size, and number of connected objects.

Step 2: Calculation of the election score.

a. Calculating a score for each device in the list based on its processor capabilities, memory capacity, and the number of connected objects.

b. The score for each device is calculated using a formula as in Equation (3):

$$S = \alpha P + \beta M + \delta C \tag{3}$$

S is the election Score, and P is the processor capability, evaluated based on Table 2. M is memory capacity, assessed based on Table 3. C represents the number of connected objects. α, β, δ refers to the weighting of each factor. Weighting factor α, β, δ can be determined based on how important each parameter is to the context of IoT application (system). To keep the value stays between 0 and 1, weighting factor $\alpha + \beta + \delta = 1$.

c. Sorting the list of devices in descending order based on their score.

Step 3: Post-processing

a. The device with the highest score is elected as the coordinator for the IoT community.

b. If multiple devices have the same highest score, a tiebreaker such as the device that has been a coordinator for the least amount of time can be used to decide the winner.

c. Once a coordinator is elected, the devices in the IoT community are notified and the coordinator begins to manage the network.

d. Re-calculating the election score when a new object with a higher device class joins the community.

In this coordinator election experiment, we used eight IoT objects member of community 'A'. Table 4 presents the object properties such as object ID, object class, RAM size and number of connected objects.

Table 4. Election score for community coordinator

ID	Class of IoT Object	RAM (GB)	No of Conn	Score based only on computing power	Score based on both computing power and centrality
A1	Class-1	8	5	0.7	0.775
A2	Class-2	2	2	0.6	0.630
A3	Class-3	1	3	0.4	0,445
A4	Class-3	1	2	0.4	0.430
A5	Class-1	8	1	0.7	0.715
A6	Class-3	2	3	0.5	0.545
A7	Class-2	4	4	0.6	0.660
A8	Class-2	4	1	0.6	0.630

Table 4 shows the results of the coordinator election score for the community coordinator of IoT community A. It shows the comparison of the scores between the two methods. The first method was calculated based only on computing power (processor and memory) and the second method was calculated based on computing power and centrality (number of connected objects). The second method could provide benefits, by utilizing the centrality of the community coordinator selection process, which can be carried out with more detailed score gradations.

The election method also has some advantages in balancing coordination. A node with high computing power but low centrality might not be as effective in managing and coordinating the network as a node with balanced attributes. The combined method ensures that nodes selected as coordinators have both sufficient resources and strategic connectivity. This approach has advantages compared to other method proposed in [25] (centrality based leader election) and [27] which only considered the highest device identity.

3.2. Trust assessment

This section discusses both QoS trust assessment and social trust, including theirs parameters, formula, and simulation results.

3.2.1. QoS trust assessment

The evaluation of QoS trust was based on two key elements: intelligence (I) and centrality (C). The calculation employed a formula that took into account weighting factors alpha and beta to provide opportunity to give different portions of contribution as seen in Equation 4. The capability factor encompassed both the processing capabilities and memory capacity of the device, while the number of connections between an IoT object and other objects in the network determined the centrality factor.

$$T_i^{qos} = \alpha C_i + \beta I_i \quad (4)$$

$$C_i = \log\left(1 + \sum_{j=1}^{k}(n_i, n_j)\right) \text{ if connection} \le 100$$

$$C_i = 1 \qquad \text{if connection} > 100 \quad (5)$$

T_i represents the QoS trust value, C_i is trust value based on centrality, while I_i represents the trust value of an IoT object d_i based on its intelligence (computing power and memory capacity. The intelligence factor was calculated using Eq. 6. The sigma (n_i,n_j) factor indicated how many number of connections between IoT objects. The logarithmic function was chosen so that the granularity of the centrality score could cover a large range of values. For nodes that had connections with more than 100 other nodes, it was assumed to have a maximum score equal to 1. It was understood that nodes with high connections (more than 100 links) had almost the same capabilities in terms of centrality in their role as nodes in the IoT community.

The intelligence factor I encompassed both the device's processing capabilities (p) and memory capacity (m), as seen in Equation 6.

$$I_i = (p_i + m_i) / 2 \quad (6)$$

where I_i is the intelligence factor of device d_i, while p_i represents processor specification (as calculated based on Table 2), and m_i represents the memory capacity (calculated based on Table 3) of the IoT object.

In the following experiment, we used an IoT object with properties {(object-ID: "A1"), (memory capacity: 8 GB), and (device class: "Class-1")}.

Example calculation. Consider a smartphone with 4 GB of memory and 30 connections:

- Computing power (p_i) score. A smartphone is categorized as Class-2 device. The computing power score $p_i = 0.8$
- Memory capacity (m_i) score. With 4 GB of memory, memory capacity score $m_i = 0.8$
- The smartphone has 30 connections in the range of (21 < 30 < 50). Assuming a normalized centrality of 0.31 within the range of 0.3 to 0.5.
- Overall, trust score assumes the same weighting factors α = 0.5 and β=0.5. Trust score calculated using Equation 4. = (0.5×0.8) + (0.5×0.31) = 0.4+0.155 = 0.555.

Fig. 2 to Fig. 7 shows the effects of variation of weighting factors α and β (from Equation 4) to the trust value of QoS Trust. The orange line shows the trust value based only on centrality (number of connections). The blue line shows the value of QoS trust, calculated using Equation 4 based on both two parameters: intelligence and centrality.

From the trust assessment, it can be seen that in the IoT networks, trust evaluation can be enhanced by incorporating the intelligence factor, which combines computing power, memory capacity, and centrality. This comprehensive approach provides a more accurate measure of a device's capability and reliability.

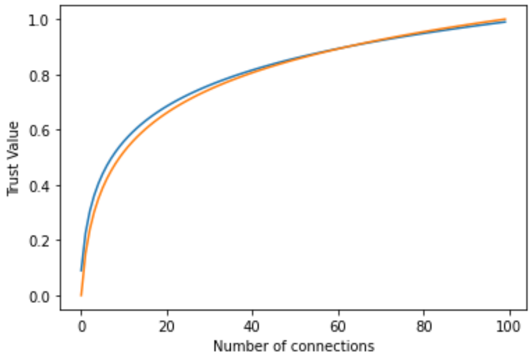

Fig. 2. Trust value as a function of centrality, α = 0.1; β = 0.9

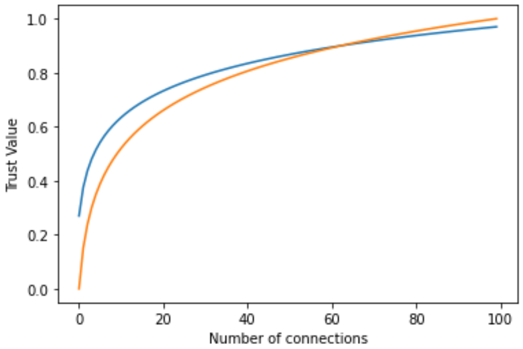

Fig. 3. Trust value as a function of centrality, α = 0.3; β = 0.7

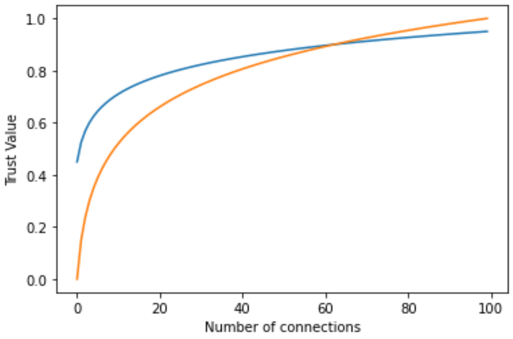

Fig. 4. Trust value as a function of centrality, α = 0.5; β = 0.5

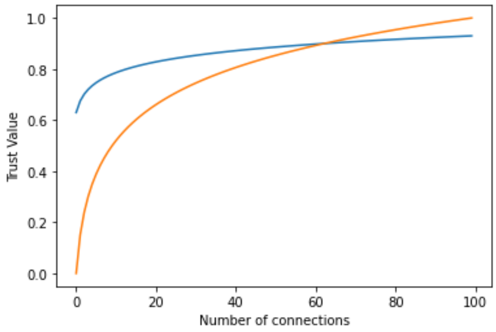

Fig. 5. Trust value as a function of centrality, α = 0.7; β = 0.3

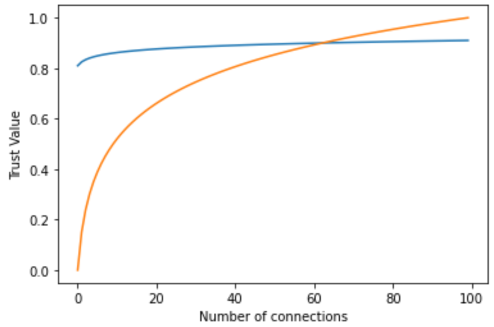

Fig. 6. Trust value as a function of centrality, α = 0.9; β = 0.1

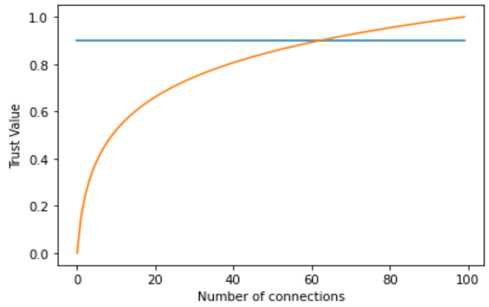

Fig. 7. Trust value as a function of centrality, α = 1.0; β = 0.0

The intelligence factor provides a balanced measure of the device's overall capability by considering both computing power and memory capacity. This metric is crucial in IoT environments where devices need to be assessed not just on single parameters but on their combined ability to perform tasks efficiently and securely.

The overall trust score combines the device's computational capabilities, memory capacity, and its role in the network to provide a comprehensive measure of trustworthiness. Devices with higher scores are more reliable and secure, suitable for critical tasks within the IoT ecosystem.

Integrating computing power, memory capacity, and centrality into a single trust evaluation framework allows for a nuanced and robust assessment of IoT devices. This method helps in making informed decisions regarding device trustworthiness, enhancing the security and efficiency of IoT networks.

3.2.2. Social trust assessment

The trust value between IoT objects can be assessed with a social approach in the sense that the trust level of an object can change dependent upon the interaction feedback between objects. Successful interaction will increase trust between the two interacting objects. On the other hand, an interaction that fails will reduce trust between the two objects. The observation of the history of interactions between objects can be done in two ways: direct and indirect, as seen in the illustration in Fig. 8.

In direct observation, after an object interacts to other object, the result is recorded either success or fail. Successful interaction will increase the trust value of related object with 0.01. Unsuccessful transactions, in contrast, will result in a reduction in the trust value of by minus 0.01.

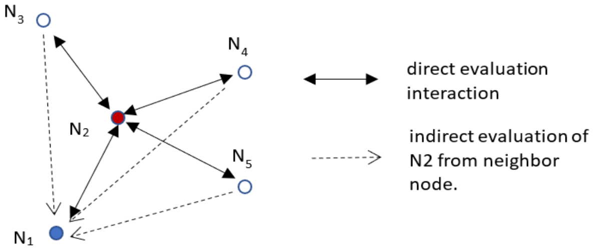
N3
N4
N2
N5
N1
direct evaluation interaction
indirect evaluation of N2 from neighbor node.

Fig. 8. Direct and indirect evaluation based on object interaction.

Indirect observation is calculated based on the transaction history between objects. When the p_i object calculates the trust value of the p_j object, p_i will see the transaction history of other objects against p_j. The transaction history of this p_j object can be asked to the community coordinator who stores the transaction history of each community member object.

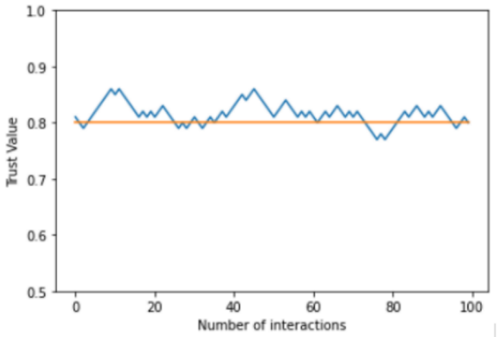

Fig. 9. Fluctuation of the trust value (blue line) as a function of interaction history on an object.

Fig. 9 shows the effect of object interaction to the trust value. In this experiment, the initial trust was set to 0.8 (orange line). In the experiment, it was simulated that an object interacted with another object 100 times with the outcome of each interaction being successful or unsuccessful, determined randomly. In these 100 interactions, the trust value fluctuated in the range of 0.76 to 0.87. The trust value of the object (blue line) would decrease or increase depending on the result of each interaction (succeed or fail) with other IoT objects.

The method is straightforward and easy to implement. The trust value changes are directly tied to the outcomes of interactions. This simplicity offer advantages such as real-time responsiveness, compared to other complex algorithm, which requires complex processing such as in [12,2]. In addition, since each interaction only slightly affects the trust value, occasional outliers (unusual success or failure) have a limited impact on the overall trust level. This makes the system robust against sporadic anomalies.

3.3. Processing time evaluation

This section delves into the outcomes of applying the trust computation algorithm on a variety of IoT devices. The emphasis of the examination is on determining the time it takes for the proposed trust algorithm to run on different IoT devices. This is crucial due to the wide range of IoT devices, each with its own unique computational abilities.

This evaluation is important since IoT devices typically feature limited computational resources, including lower processing power, memory, and battery life. Evaluating the execution time helps to ensure that the algorithms can run efficiently without depleting these limited resources, thereby extending device longevity and maintaining functionality.

The simulation involved 10 nodes within a single IoT community. The weighting factors α, and β were all assigned the same value of 0.5. This implied that in this experiment, we assumed an equal contribution from both QoS trust and social trust. The experiment was conducted on four different IoT devices including desktop PC, laptop, and raspberry as described in Table 5. We selected devices with varying specifications to evaluate the performance of the model across diverse IoT devices. Each measurement was performed ten times, after which we computed the mean value and deviation of the processing time.

Table 5. Processing time on different devices.

Device Type	Device Type	Mean (millisecond)	Std Dev (millisecond)
Desktop PC-1	Desktop PC, Intel Core Duo 2,93 GHz, 4 GB RAM, Windows 10	33.10	3.94
Laptop PC-2	Laptop PC, Intel I7, 8 GB RAM, Windows 10	30.13	2.38
Tablet PC-3	Tablet PC, Intel Core I7-8550U, 1,8 GHz, 16 GB RAM, Windows 11	21.42	3.29
Raspi-4B	Raspi with ARMv7 processor, 8 GB RAM, Raspbian OS	128.66	9.91

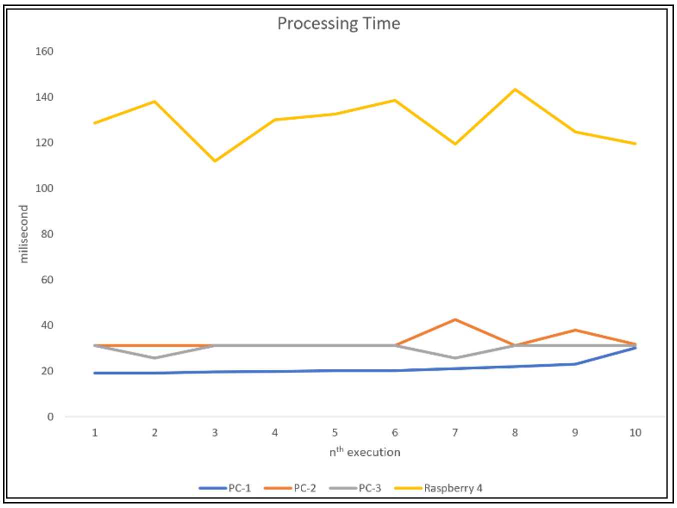

Fig. 10. Processing time of proposed model in several type of devices

Table 5 showcases the algorithm's execution time across various devices. From the results, it is evident that PC-3 was found as the most efficient device in terms of processing time for the proposed trust algorithm, clocking the fastest average time of 21.42 milliseconds. This aligns with the fact that Tablet PC-3 boasts the highest hardware profile. Despite having higher RAM, PC-2's execution time is nearly identical to that of PC-1 with times of 30.1 and 33.1 milliseconds respectively. The Raspberry device recorded the longest execution time, clocking in at 128.6 milliseconds. This was understandable, given that the Raspberry device's computing capabilities are more limited compared to a desktop PC. The execution times observed in the experiment is visualized in Fig. 10.

This result underscored the importance of considering device capabilities in IoT applications, especially for trust-based security algorithms. Future work should explore optimization techniques to improve algorithm efficiency on lower-end devices like the Raspberry Pi.

3.4. Key features of QS-trust model

The proposed QS-Trust model provides trust evaluation framework for IoT devices, as discussed, combines both QoS and social parameter involving computing power, memory capacity, centrality, and interaction history to provide a robust and comprehensive trust score assessment. This approach ensures that decisions about device reliability and security are based on multiple parameters, offering a more nuanced view than traditional binary or coarse-grained models.

3.4.1. Decentralized trust management

IoT Ecosystem and Community Structure: The QS-Trust model divides the larger IoT ecosystem into smaller, manageable IoT communities. Each community handles local trust evaluations, reducing the complexity and improving the scalability of trust management.

Localized Trust Calculations: By evaluating trust within smaller communities, the model can quickly adapt to changes, as each community can autonomously update trust levels based on local interactions.

3.4.2. Multi-factor trust evaluation

Comprehensive Trust Metrics: QS-Trust incorporates multiple factors such as computing power, memory capacity, centrality, and interaction history, which provide a detailed assessment of a device's capabilities and its role in the network.

Dynamic Adjustment: Trust scores are continually updated based on real-time performance and interactions, ensuring that the trust evaluation remains accurate as devices join or leave. This approach is also used in [12,21].

3.4.3. Centrality and context awareness

Dynamic Centrality Adjustment: The centrality score, which reflects the device's importance and connectivity within the network, is dynamically adjusted. This allows the network to quickly recognize and adapt to changes in the device's role. This method is also utilized in [25].

Context-Specific Trust Evaluations: Trust assessments can be tailored to the specific context of each community, allowing for more precise and relevant evaluations that can quickly adjust to the dynamic nature of the network.

4. Conclusion

This study introduces QS-Trust, a security model for IoT based on trust, aimed at establishing a secure IoT ecosystem. The model considers both the quality of service (QoS) and social parameters to determine the trust value of an object. The QoS parameters used in trust assessment are device resources, namely memory capacity and processing capability. Meanwhile, the social parameters involved in this research were friendship relations and interaction history between IoT objects. The QS-Trust model effectively addressed the challenges of maintaining accurate and up-to-date trust levels in dynamic IoT environments through its decentralized approach, multi-factor evaluations, and adaptive algorithms. By continuously monitoring device performance and interactions, and dynamically adjusting trust scores, QS-Trust ensured that the network remained secure, reliable, and capable of adapting to frequent changes.

The execution time of QS-Trust was in the range of 21 to 128 milliseconds, which is efficient for real-time IoT system. QS-Trust offers a promising solution for securing the IoT ecosystem. Future research will focus on developing a framework for implementing the proposed model, mitigating security disturbances related to trust, and increasing data privacy of IoT objects.

References

1. R. Roman, P. Najera, and J. Lopez, *Securing the Internet of Things*, Computer (Long. Beach. Calif) 44 (2011) 51–58.
2. C. Marche and M. Nitti, *Trust-Related Attacks and Their Detection: A Trust Management Model for the Social IoT*, IEEE Trans. Netw. Serv. Manag. 18 (2021) 3297–3308.
3. S. Joshi and D. K. Mishra, *A roadmap towards trust management & privacy preservation in mobile ad hoc networks*, Proc. 2016 Int. Conf. ICT Business, Ind. Gov. ICTBIG 2016, 2017.
4. S. F. Auliya, L. E. Nugroho, and N. A. Setiawan, *A review on smartphone usage data for user identification and user profiling*, Commun. Sci. Technol. 6 (2021) 25–34.
5. B. Tang, H. Kang, J. Fan, Q. Li, and R. Sandhu, *IoT passport: A blockchain-based trust framework for collaborative internet-of-things*, Proc. ACM Symp. Access Control Model. Technol. SACMAT, (2019) 83–92.
6. D. M. Menon and N. Radhika, *A Trust-Based Framework and Deep Learning-Based Attack Detection for Smart Grid Home Area Network,* Int. J. Intell. Eng. Syst. 15 (2022) 106–116.
7. J. Chen, Z. Tian, X. Cui, L. Yin, and X. Wang, *Trust architecture and reputation evaluation for internet of things*, J. Ambient Intell. Humaniz. Comput. 0 (2018) 1–9.
8. N. B. Truong, T. W. Um, and G. M. Lee, *A Reputation and Knowledge Based Trust Service Platform for Trustworthy Social Internet of Things,* Innov. Clouds, Internet Networks (2016).
9. T. Khan *et al.*, *An efficient trust-based decision-making approach for WSNs: Machine learning oriented approach*, Comput. Commun. 209 (2023) 217–229.
10. S. Javanmardi, M. Shojafar, S. Shariatmadari, and S. S. Ahrabi, *FR trust: A fuzzy reputation-based model for trust management in semantic P2P grids,* Int. J. Grid Util. Comput. 6 (2015) 57–66.
11. I. Uddin, M. Guizani, B. S. Kim, S. Hassan, and M. K. Khan, *Trust management techniques for the internet of things: A survey,* IEEE Access, 7 (2019) 29763–29787.
12. X. Wu, *A robust and adaptive trust management system for guaranteeing the availability in the internet of things environments*, KSII Trans. Internet Inf. Syst. 12 (2018) 2396–2413.
13. V. Mohammadi, A. M. Rahmani, A. M. Darwesh, and A. Sahafi, *Trust-based recommendation systems in Internet of Things: a systematic literature review,* Human-centric Comput. Inf. Sci., 9 (2019) 21.
14. Y. B Saied, A. Olivereau, D. Zeghlache, and M. Laurent, *Trust management system design for the Internet of Things: A context-aware and multi-service approach*, Comput. Secur. 39 (2013) 351–365.
15. Y. B. Saied, A. Olivereau, D. Zeghlache, and M. Laurent, *Lightweight collaborative key establishment scheme for the Internet of Things*, Comput. Networks 64 (2014) 273–295.
16. Y. Chae, L. C. DiPippo, and Y. L. Sun, *Trust Management for Defending On-Off Attacks*, IEEE Trans. Parallel Distrib. Syst. 26 (2015) 1178–1191.
17. G. D. Putra, V. Dedeoglu, S. S. Kanhere, and R. Jurdak, *Trust Management in Decentralized IoT Access Control System*, IEEE Int. Conf. Blockchain Cryptocurrency, ICBC 2020 (2020).
18. V. Suryani, S. Sulistyo, and W. Widyawan, *ConTrust: A trust model to enhance the privacy in internet of things*, Int. J. Intell. Eng. Syst. 10 (2017) 30–37.
19. M. Masmoudi, W. Abdelghani, and I. Amous, *Deep Learning for Trust-Related Attacks Detection in Social Internet of Things*, in Lecture Notes on Data Engineering and Communications Technologies, Cham: Springer International Publishing (41) 2020.
20. M. Nitti, R. Girau, L. Atzori, A. Iera, and G. Morabito, *A subjective model for trustworthiness evaluation in the social Internet of Things,* IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications - (PIMRC), Sydney: IEEE (2012) 18–23.
21. C. Boudagdigue, A. Benslimane, A. Kobbane, and J. Liu, *Trust Management in Industrial Internet of Things,* IEEE Trans. Inf. Forensics Secur. 15 (2020) 3667–3682.
22. J. Guo, I. R. Chen, and J. J. P. Tsai, *A survey of trust computation models for service management in internet of things systems*, Comput. Commun. 97 (2017) 1–14.
23. N. B. Truong, H. Lee, B. Askwith, and G. M. Lee, *Toward a trust evaluation mechanism in the social internet of things*, Sensors (Switzerland) 17 (2017) 1–24.
24. L. Atzori, A. Iera, and G. Morabito, *SIoT: Giving a social structure to the internet of things*, IEEE Commun. Lett., 15 (2011) 1193–1195.
25. A. Favier, L. Arantes, J. Lejeune, and P. Sens, *Centrality-Based Eventual Leader Election in Dynamic Networks*, IEEE 20th Int. Symp. Netw. Comput. Appl. NCA (2021).
26. M. Nitti, R. Girau, and L. Atzori, *Trustworthiness Management in the Social Internet of Things*, IEEE Trans. Knowl. Data Eng. 26 (2014) 1253–1266.
27. M. Numan, F. Subhan, W. Z. Khan, B. Assiri, and N. Armi, *Well-organized bully leader election algorithm for distributed system,* Proc. - 2018 Int. Conf. Radar, Antenna, Microwave, Electron. Telecommun. ICRAMET (2018) 5–10.