



UNICA

UNIVERSITÀ
DEGLI STUDI
DI CAGLIARI



Università di Cagliari

UNICA IRIS Institutional Research Information System

This is the Author's *accepted* manuscript version of the following contribution:

C. Marche and M. Nitti, "An Investigation of Trust Management Systems in Digital Twin Architectures," *2025 IEEE Globecom Workshops (GC Wkshps)*, Taipei, Taiwan, 2025, pp. 544-549.

© 2025 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

The publisher's version is available at:

<http://dx.doi.org/10.1109/GCWkshps68340.2025.11591210>

When citing, please refer to the published version.

An Investigation of Trust Management Systems in Digital Twin Architectures

Claudio Marche

*Department of Electrical and Electronic Engineering
University of Cagliari
Cagliari, Italy
claudio.marche@unica.it*

Michele Nitti

*Department of Electrical and Electronic Engineering
University of Cagliari
Cagliari, Italy
michele.nitti@unica.it*

Abstract—As the Internet of Things (IoT) continues to expand, Digital Twin (DT) architectures are adopted to represent physical devices with virtual entities, enabling scalable and decentralized systems. In this context, Trust Management Systems (TMSs) have been introduced to assess the trustworthiness of entities, supporting decisions in service exchange interactions. However, most existing TMSs assume ideal conditions, overlooking issues such as service evaluation inaccuracies and unintentional provider faults. This paper investigates the limitations of current TMSs in DT architectures by analyzing their structure and evaluating their performance under scenarios with errors. A classification of state-of-the-art trust models is first presented, followed by a general mathematical formulation that depicts the main inputs of trust computation. Then, several representative TMSs are simulated and evaluated under scenarios characterized by malicious behaviors and errors, both in service provisioning and evaluation. The results highlight important deficiencies in current designs, particularly in distinguishing between faulty but benevolent nodes and malicious entities.

Index Terms—Internet of Things, Digital Twins, Trustworthiness Management, DT Architectures.

I. INTRODUCTION

The Internet of Things (IoT) is in continuous expansion, connecting billions of heterogeneous devices across industrial, urban, and domestic environments [1]. These devices are expected to interact autonomously, share information, and provide services to meet diverse application requirements. As the number of IoT systems increases, traditional architectures based on centralized control face scalability challenges [2]. To address these issues, the concept of Digital Twins (DTs) has emerged as a paradigm for designing IoT architectures [3].

A Digital Twin is a virtual representation of a physical object, capturing its current state, capabilities, and behavior [4]. DTs abstract the complexity of physical devices and expose their functionalities in the cloud or at the edge, improving service discovery. In modern IoT architectures, DTs act as autonomous entities capable of providing services, requesting functionalities from others, and coordinating to manage complex tasks. This decentralized control enhances scalability while reducing the reliance on centralized systems [5]. As DTs frequently interact with previously unknown digital entities, evaluating the reliability of potential collaborators becomes an important challenge [6].

To address this issue, Trust Management Systems (TMSs) have been introduced as a means of supporting decisions in networks [7]. TMSs are responsible for evaluating the behavior of entities and the quality of the exchanged services, assigning trust scores that influence future interactions [8]. Supporting traditional mechanisms like Service Level Agreements (SLAs) and Key Performance Indicators (KPIs) [9], which are effective in centralized scenarios, TMSs offer a complementary and additional solution.

Despite the extensive body of work on trust management, existing TMSs often assume ideal conditions, overlooking aspects such as the possibility of erroneous service evaluations or unintentional errors in service provision. Many TMSs are designed to identify and isolate malicious nodes but lack the capability to distinguish between malicious behavior and cooperative nodes affected by errors [10]. This limitation can lead to the exclusion of benevolent nodes, reducing network cooperation and system performance.

In this paper, an investigation into the structure of existing TMSs is conducted, with a particular focus on their behavior in real scenarios with errors. The manuscript provides a review of the most representative TMSs from the literature, highlighting their inputs. Then, it defines a generic mathematical representation to formalize the trust computation and its evolution over time. Finally, it simulates several state-of-the-art TMSs under various conditions to identify their strengths and deficiencies. The main contributions of this work are summarized as follows: (i) classification of the most relevant TMSs in the literature, identifying common inputs in trust evaluation; (ii) formulation of a general trust model, isolating its components to enable comparative analysis; (iii) performance evaluation of selected TMSs in both ideal and error scenarios, demonstrating how current designs often fail to handle issues such as service evaluation inaccuracies and provider faults.

The remainder of the paper is organized as follows. Section II introduces the concept of trust in DT architectures and provides a classification of existing TMSs based on their inputs. Section III describes the reference scenario and Section IV formalizes a generic trust model, while Section V reports the results of an evaluation of representative TMSs under various conditions. Finally, Section VI concludes the paper.

II. THE IMPORTANCE OF TRUST

As IoT systems become increasingly populated by a large number of connected devices, ensuring cooperation among them is a critical task [11]. These devices must be capable of communicating to meet the continuous service demands generated by user applications, whether by providing real-time data or sharing previously collected information [12]. To optimize service discovery in such environments, DTs have been introduced as virtual counterparts of physical objects. Acting as autonomous entities, DTs abstract the devices' complexity and enable interaction between devices by exposing their functionalities directly in the cloud [13]. In DT architectures, each virtual entity can operate both as a provider and a consumer of services, engaging with other DTs in the network to satisfy application requirements [14].

To analyze how trust is managed within DT architectures, this section offers a brief classification of existing models based on the information they use to evaluate trust. In general, the trust inputs of the most well-known TMSs fall into three main categories: node characteristics, historical behavior, and recommendations from other nodes.

The first category refers to models that assess trust using the static characteristics of providers. These models generally incorporate objective attributes such as Quality of Service (QoS) indicators or social properties. For instance, the model proposed in [15] adopts an evaluation based on parameters such as network throughput, service availability, and execution time. The model also includes a decay function to weight outdated data. Similarly, in the model described in [16], trust is derived from both direct interactions and social relationships. The framework quantifies trust not only through interaction history but also via the network's social graph, emphasizing factors like friendship and community membership. The authors also propose a central weighting system to prioritize evaluations from more influential nodes in the network. Moreover, the proposal in [17] expands on these ideas by incorporating energy parameters and the trust level of the recommending node into the evaluation. This model considers the physical limitations of IoT devices, such as energy consumption and available resources, alongside their placement in a social network. It adjusts trust weights based on the node's social features and energy metrics. Finally, authors in [18] focus on computational capabilities and responsiveness. Their approach uses entropy and the number of service failures to evaluate the reliability of a provider in real time. The system introduces penalty factors for unresponsive nodes, ensuring that nodes with low processing capability are not overloaded.

The second group includes models that build trust based on historical behavior observed during past interactions. These systems track feedback over time to update trust scores. In [19], trust dynamics are captured through a learning mechanism that adjusts trust based on trends in service quality. This model incorporates time windows and weighted averages to detect behavioral patterns. Moreover, authors in [20] propose

TABLE I
CLASSIFICATION OF TMSs.

Ref.	Nodes Characteristics	Historical Interactions	Recommendations
[15]	QoS	✓	✓
[16]	Social	✓	✓
[17]	QoS and Social	✓	✓
[18]	QoS	✓	✓
[19]	QoS and Social	✓	✓
[20]	×	✓	✓
[21]	×	✓	×
[22]	×	✓	✓
[23]	×	✓	✓
[24]	×	✓	✓

a distributed trust mechanism where trust is evaluated through an objective function aggregating past experiences. Their approach focuses on analyzing long-term patterns rather than single interactions. The architecture ensures that evaluations are propagated while avoiding centralized authorities. Another relevant contribution is illustrated in [21], who integrate blockchain to preserve interaction feedback. Their model allows for immutable trust history, reducing the risk of tampered evaluations. The blockchain also enables transparent access to feedback, enhancing transparency in the trust evaluation process.

The third category includes models that calculate trust based on recommendations from other nodes. Among them, in [22] authors propose a trust model that collects external opinions while filtering them based on the similarity of service. Their system adjusts trust weights by assessing the overlap between requested and previously provided services. A different strategy is presented in [23], where a reputation management system based on blockchain ensures resistance to manipulation. This model leverages smart contracts to manage and verify recommendations, supporting transparency. Additionally, it incorporates credibility scores for recommenders, which are updated based on their past accuracy. Finally, a recent proposal is described in [24], designed for secure data sharing in vehicle environments. The model integrates both direct and indirect trust components, where direct trust is computed based on multiple behavioral metrics, while indirect trust, on the other hand, is derived from recommendations provided by neighboring nodes. An important feature is represented by the mechanism for identifying false recommendations. This is achieved through similarity analysis, allowing the system to mitigate the influence of malicious recommenders.

To sum it up, Table I provides a description of the analyzed TMSs, based on the type of input data considered in trust computation, i.e., node characteristics, historical interactions, and recommendations. To the best of our knowledge, although the analyzed TMSs introduce contributions and improvements in identifying malicious behavior, they still exhibit several limitations. In particular, they tend to focus exclusively on malicious nodes, without considering that even cooperative

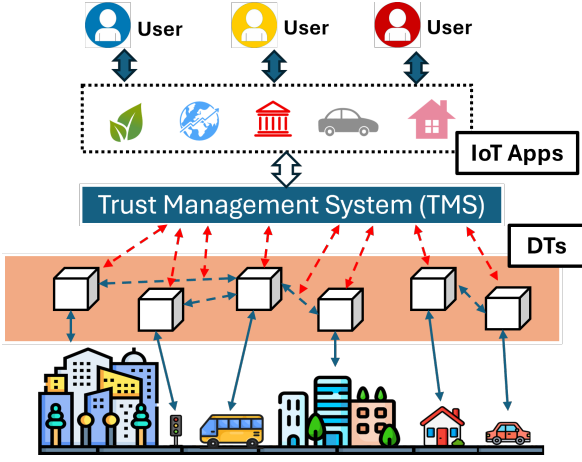


Fig. 1. Reference scenario.

nodes may exhibit faulty behavior. The possibility that a service provider may unintentionally deliver scarce services, due to malfunctions, is often overlooked. Similarly, many models assume that the requester is always able to perfectly evaluate the service received. An efficient TMS should account for these types of errors to avoid benevolent nodes classified as malicious. This is especially important in environments where the number of available providers varies significantly, since the selection strategy may change depending on the redundancy of the providers. In general, current TMSs do not consider this aspect in their design. Furthermore, models that rely on recommendations typically adopt them as a way to accelerate trust propagation. While this mechanism is beneficial for disseminating trust scores, it also introduces a major vulnerability: the same speed can disseminate erroneous evaluations.

In light of these observations, this paper investigates the structure of the most well-known TMSs in the literature, providing a mathematical formulation that captures their components. Building on this formulation, a set of simulations is carried out to assess TMSs under both ideal and error scenarios.

III. SCENARIO

As illustrated in Figure 1, the considered scenario enables users to deploy and run applications in the cloud or in the edge, and rely on their or external DTs to discover available services. DTs, acting as autonomous entities, not only provide data and functionalities to applications but may also interact with other DTs in the network to retrieve additional services required to satisfy applications [25]. However, these interactions introduce risks due to potential malfunctions and errors, or malicious behaviors originating from either the DTs or the associated physical objects.

To mitigate such risks, TMSs play an important role in DT architectures. A TMS estimates the trust of both individual DTs and their interactions, considering that DTs may

operate in a distributed environment. It assists applications in selecting reliable service providers and enables DTs to assess the reliability of requesters in interactions. Trust evaluation must consider not only direct experiences but also multi-hop interactions, where erroneous or opportunistic behaviors can disrupt reliability. As a result, designing robust TMSs requires addressing both intentional malicious behaviors and unintended errors, to support cooperation among digital entities.

IV. STRUCTURE OF A TMS

According to the analysis of the literature, the structure of a generic TMS can be decomposed into three main components. Each component contributes to the computation of the trust score based on different inputs: the intrinsic characteristics of the node, the recommendations received from other nodes, and the history of previous interactions. Building on the analysis of numerous state-of-the-art approaches, especially the most relevant and recent ones discussed in Section II, this section proposes a unified model that captures the essential structure of TMSs. This generic model enables a representation of the main trust components and their dynamic evolution over time, allowing the analysis of trust behaviour as it evolves with the sequence of interactions among entities.

The trust value $T_{rp}(k)$ assigned by a DT requester r to a DT provider p at the k -th interaction is computed as a function of the three elements, as illustrated as follows:

$$T_{rp}(k) = g(f_e(E_p(k)), f_o(O_p(k)), f_i(I_{rp}(k))) \quad (1)$$

where each function $f_x(\cdot)$ describes a different aspect of the trust evaluation. The trust is calculated through a specific aggregation function g , which maps the input parameters into a trust score.

The first contribution, $f_e(E_p(k))$, considers the static characteristics of the provider entity p . These characteristics include both quantitative QoS metrics, such as availability and computation capabilities, and social attributes, such as community belonging and centrality. Formally, this term can be expressed as:

$$f_e(E_p(k)) = \phi_e(Q_p(k), S_p(k)) \quad (2)$$

where $Q_p(k)$ and $S_p(k)$ represent the QoS and social parameters at interaction k , respectively. An aggregation function, represented with ϕ_e , is used to combine these inputs into a unified evaluation of the node's profile.

The second contribution $f_o(O_p(k))$, incorporates reputation information collected from other nodes that previously interacted with the provider. Each node that provides a recommendation acts as a recommender; all the recommendations are usually weighted based on the trust level that the requester assigns to the recommender. This component is formally defined as:

$$f_o(O_p(k)) = \phi_o(\{(T_{jp}(k), w_j)\}_{j \in \mathcal{R}}) \quad (3)$$

where $\mathcal{R}$ is the set of recommenders and w_j is a weighting factor used to give more importance to a recommendation with respect to others. The function ϕ_o aggregates the weighted recommendations into a single score representing the reputation of the provider.

Finally, the third contribution, $f_i(I_{rp}(k))$, models the evolution of trust based on the history of interactions between the requester and the provider. Each past interaction k is associated with a feedback value $v_{rp}(k)$, expressing the evaluation that the requester assigned to the provided service. This component is defined as:

$$f_i(I_{rp}(k)) = \phi_i \left(v_{rp}(k)_{k=1}^L \right) \quad (4)$$

where L represents the number of past interactions considered in the computation. This may correspond to the total number of interactions between the two nodes, or to a fixed size window of the most recent ones. The function ϕ_i can incorporate averages or exponential decay, to weigh past feedback appropriately.

In general, due to the heterogeneity of TMSs, the general model defined by Equation 1 can be difficult to analyze or simulate directly. For this reason, and to enable a more general study of the trust algorithms, the proposed analysis adopts a formulation that focuses on how trust evolves incrementally over time:

$$T_{rp}(k) = T_{rp}(k-1) + \Delta T_{rp}(k) \quad (5)$$

where $T_{rp}(k-1)$ represents the trust computed in the previous interaction, and $\Delta T_{rp}(k)$ denotes the variation resulting from the most recent interaction. This variation depends on the evaluation $v_{rp}(k)$ in response to the behaviour of the provider during the interaction.

$$\Delta T_{rp}(k) = \begin{cases} \Delta T_{rp}^+(k) & \text{if } v_{rp}(k) \geq T_{rp}(k-1) \\ -\Delta T_{rp}^-(k) & \text{if } v_{rp}(k) < T_{rp}(k-1) \end{cases} \quad (6)$$

If the evaluation is greater than or equal to the evaluation of the provider, i.e. its trust value $T_{rp}(k-1)$ until its previous interaction, the actual interaction is considered benevolent, and the trust is incremented by a positive variation $\Delta T_{rp}^+(k)$. Conversely, if the evaluation falls below the trust value, the interaction is labelled as malicious, and the trust is decreased by a negative variation $\Delta T_{rp}^-(k)$.

This representation allows to analyze and compare trust models independently of the specific mechanisms used to compute the trust. In this sense, the following section leverages this formulation to examine various TMSs proposed in the literature, highlighting both their limitations and the directions that should be taken to enhance their effectiveness.

V. EXPERIMENTAL EVALUATION

A. Simulation Setup

To assess the TMSs under different network conditions, a simulation is carried out involving 100 nodes. A total of

50,000 interactions are generated, ensuring that each node can act multiple times both as a requester and as a provider. At each step, one node is randomly chosen to initiate a request, while the potential providers are restricted to 10% of the remaining nodes. For those models that rely on indirect trust, 5% of randomly selected recommenders are also included in the trust evaluation process. As a reference point for comparison, a basic strategy, labelled as *Benchmark* in all figures and the final table, is implemented. This approach relies only on the last direct interaction, without considering past feedback or recommendations. Trust is treated in binary terms: if the most recent service is satisfactory, the provider is trusted (value 1); otherwise, it is marked as untrustworthy (value 0) and excluded from future selections. The benchmark is compared against five representative TMSs, selected for their relevance and recognition in the literature. Each of these models incorporates a different combination of trust inputs, including node characteristics, behavioral history, and recommendations. Specifically, the selected models are those proposed in Adewuyi et al. [15], Nitti et al. [16], Chen et al. [17], Marche et al. [19], and Mendoza et al. [20], all previously introduced in Section II. All the models and the simulation environment have been implemented using MATLAB.

B. Evaluation of the TMSs

This section presents a set of simulations to evaluate the performance of the selected TMSs under different network conditions. The goal is to identify and analyze the main deficiencies that emerge when these models are tested outside of ideal scenarios. To illustrate the overall impact of errors on the performance of the different TMSs, the models are tested under varying error probabilities, without distinguishing between evaluation or provision faults. Specifically, an error corresponds either to a wrong evaluation by the requester, incorrectly classifying a good service as malicious, or to a fault in the provider's behavior, resulting in a scarce service that is perceived as malicious.

The first simulation, shown in Figure 2, compares the behavior of the TMSs in a network where 20% of the nodes behave maliciously. The performance is compared through a success rate analysis, which measures the ratio of successful interactions to the total number of them. Two distinct conditions are examined: the upper graph represents the ideal case, typically adopted in the literature, where no errors are introduced; the lower graph introduces a more realistic one, where random errors affect the system. Specifically, a 10% error probability is applied both to the requesters' evaluation of received services and to the providers' service delivery. As observed in the ideal case (upper graph), the best performance is achieved by the benchmark, which is the simplest among those tested. This outcome is expected, as in the absence of errors, the best performance lies in the speed with which the TMSs discard malicious nodes: a TMS that eliminates malicious nodes as quickly as possible performs best in such a scenario. This

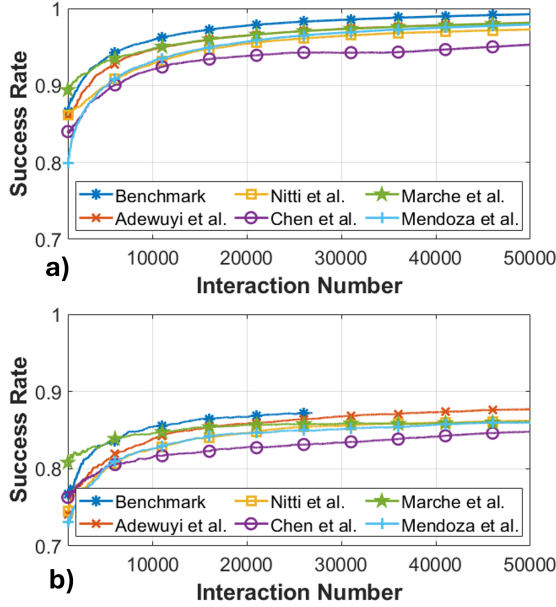


Fig. 2. Success rate of TMSs in the presence of 20% malicious nodes. a) Ideal scenario without errors. b) realistic scenario with 10% of probability of errors in both service evaluation and provision.

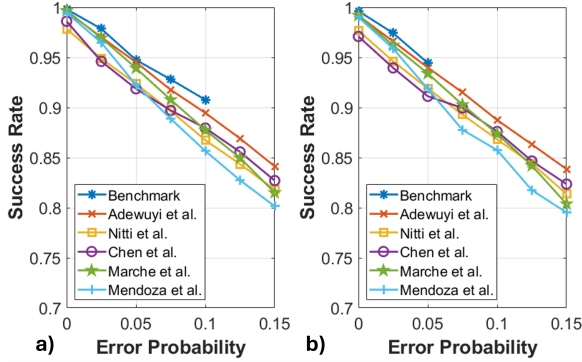


Fig. 3. Impact of increasing error probability on TMSs in a network with 20% malicious nodes. Left: 50% of nodes are available as potential providers. Right: 25% providers availability.

result could suggest that, under perfect conditions, there is no practical need for more sophisticated TMSs. However, the scenario changes significantly when errors are introduced (lower graph). In this setting, errors lead to the classification of all the nodes, benevolent or not, as malicious, preventing the benchmark from finding available providers and causing the system to halt. In contrast, complex TMSs, although not designed to handle errors, continue to function and maintain acceptable performance levels. These models are able to discriminate between malicious behavior and occasional faults, thus preserving cooperation within the network by giving nodes affected by errors a chance to recover trust over time.

The second set of simulations, illustrated in Figure 3, analyzes the TMSs under varying error probabilities, with

TABLE II
AVERAGE TRUST VARIATIONS, SUCCESS SELECTION RATE, AND PERCENTAGE OF DISCARDED BENEVOLENT NODES UNDER TWO ERROR PROBABILITIES.

Ref.	Err.	$\Delta T_{rp}^+(k)$	$\Delta T_{rp}^-(k)$	Success Selection Rate	Benevolent Discarded
Bench.	$\forall$ Err.	0	1	-	100%
[15]	0.05	0.0011	0.0214	98.93%	6.58%
	0.15	0.0034	0.0192	98.89%	27.64%
[16]	0.05	0.0015	0.0260	97.93%	7.05%
	0.15	0.0042	0.0233	97.91%	12.56%
[17]	0.05	0.0034	0.0612	97.94%	7.08%
	0.15	0.0062	0.0349	97.88%	29.64%
[19]	0.05	0.0489	0.2264	86.11%	35.18%
	0.15	0.1478	0.3696	83.61%	48.19%
[20]	0.05	0.0895	1.2598	88.24%	39.69%
	0.15	0.2735	1.1716	85.48%	41.70%

the percentage of malicious nodes fixed at 20%. The figure presents two separate simulations: in the left graph, the potential provider set corresponds to 50% of the total nodes, while in the right graph it is reduced to 25%. These values differ from the 10% used in the previous experiment. As shown in both graphs, the benchmark deteriorates in performance as the error probability increases. When only a small percentage of nodes are available for selection, even low error rates cause the model to misclassify all nodes as malicious. The more complex TMSs exhibit varied behaviors: some models perform better under low error conditions, while others demonstrate better performance as error rates increase with respect to other TMSs. Moreover, the graphs illustrate the impact of the number of providers. Reducing the percentage of providers means fewer alternatives are available, and the ability of a TMS to differentiate between errors and malicious behavior becomes critical for some TMS with respect to others. The motivations under these different behaviours are better explained in the next simulation.

The final set of simulations is summarized in Table II. This experiment leverages the formulation introduced in Section IV to examine the behavior of each TMS based on the average of trust variations, both positive and negative. Two performance metrics are evaluated: the percentage of benevolent nodes that are discarded, and the success selection rate, defined as the percentage of interactions in which the TMS correctly selected a benevolent node over a malicious one. Each model is tested under two error probabilities: 0.05 and 0.15. The results show that while most TMSs maintain high success selection rates, some of them discard a significant number of benevolent nodes. This issue becomes more apparent in models where the negative trust variation is particularly large. In such cases, even a modest error rate causes repeated trust penalties for cooperative nodes, and the small positive variation is not sufficient to offset the degradation. As a result, these nodes are excluded from future interactions despite not exhibiting malicious intent. This behavior is not strictly linked to the ratio between positive and negative variations, but rather to the combined effect of

high penalization and the frequency of errors, which together amplify the risk of wrong classifications. Additionally, models characterized by high negative variation tend to perform worse not only in terms of benevolent node exclusion but also in success selection rate, which in some cases falls below 90%. In general, the table depicts how all TMSs fail to distinguish between faulty but benevolent nodes and malicious ones. Even with a modest error probability of 0.05, the percentage of discarded benevolent nodes exceeds 5% for all TMSs, and increases at higher errors.

These findings underscore a limitation in the current design of TMs: the lack of mechanisms to handle errors. Without accounting for possible errors, both in requesting and in providing services, the performance of TMSs can degrade substantially.

VI. CONCLUSIONS

In recent years, DT architectures have been increasingly adopted in Internet of Things (IoT) scenarios to improve scalability and service exchange. However, ensuring cooperation among a large number of digital entities requires effective mechanisms for trust assessment. In this context, TMSs represent an important component to evaluate the reliability of nodes and to guide decisions in service selection.

This paper analyzes several well-known TMSs, with particular attention to their performance in realistic scenarios that include both malicious behavior and errors. A general mathematical formulation is proposed to abstract the trust computation and highlight its components. Based on this formulation, simulations are conducted to compare five state-of-the-art models under varying network conditions.

The results show that although existing TMSs perform well under ideal assumptions, they are often unable to distinguish between malicious nodes and benevolent nodes affected by errors. This leads to the exclusion of reliable nodes and a consequent reduction in the overall performance.

ACKNOWLEDGMENT

This work has supported by the European Union—NextGenerationEU (D.M. 737/2021 - Interdisciplinary research initiatives on transversal topics for PNR) - Project Tile “SUSTAIN - flexible Sensors for secUre and truSTed crowdsensing environmentAI applicatioNs”, – CUP F25F21002720001.

REFERENCES

- [1] O. Aouedi, T.-H. Vu, A. Sacco, D. C. Nguyen, K. Piamrat, G. Marchetto, and Q.-V. Pham, “A survey on intelligent internet of things: Applications, security, privacy, and future directions,” *IEEE communications surveys & tutorials*, 2024.
- [2] T. Bu, Z. Huang, K. Zhang, Y. Wang, H. Song, J. Zhou, Z. Ren, and S. Liu, “Task scheduling in the internet of things: challenges, solutions, and future trends,” *Cluster Computing*, vol. 27, no. 1, pp. 1017–1046, 2024.
- [3] Y. Kalyani and R. Collier, “The role of multi-agents in digital twin implementation: Short survey,” *ACM Computing Surveys*, vol. 57, no. 3, pp. 1–15, 2024.
- [4] A. Hakiri, A. Gokhale, S. B. Yahia, and N. Mellouli, “A comprehensive survey on digital twin for future networks and emerging internet of things industry,” *Computer Networks*, p. 110350, 2024.
- [5] Y. Wang, Z. Su, S. Guo, M. Dai, T. H. Luan, and Y. Liu, “A survey on digital twins: Architecture, enabling technologies, security and privacy, and future prospects,” *IEEE Internet of Things Journal*, vol. 10, no. 17, pp. 14965–14987, 2023.
- [6] S. A. Alhandi, H. Kamaludin, and N. A. M. Alduais, “Trust evaluation model in iot environment: a comprehensive survey,” *Ieee Access*, vol. 11, pp. 11165–11182, 2023.
- [7] K. Iranshahi, J. Brun, T. Arnold, T. Sergi, and U. C. Müller, “Digital twins: Recent advances and future directions in engineering fields,” *Intelligent Systems with Applications*, vol. 26, p. 200516, 2025.
- [8] H. Wu, P. Ji, H. Ma, and L. Xing, “A comprehensive review of digital twin from the perspective of total process: Data, models, networks and applications,” *Sensors*, vol. 23, no. 19, p. 8306, 2023.
- [9] Y. Sharma, D. Bhamare, N. Sastry, B. Javadi, and R. Buyya, “SLA management in intent-driven service management systems: A taxonomy and future directions,” *ACM Computing Surveys*, vol. 55, no. 13s, pp. 1–38, 2023.
- [10] C. Marche and M. Nitti, “Can we trust trust management systems?” *IoT*, vol. 3, no. 2, pp. 262–272, 2022.
- [11] D. Ferraris, C. Fernandez-Gago, R. Roman, and J. Lopez, “A survey on iot trust model frameworks,” *The Journal of Supercomputing*, vol. 80, no. 6, pp. 8259–8296, 2024.
- [12] P. Sun, S. Shen, Y. Wan, Z. Wu, Z. Fang, and X.-z. Gao, “A survey of iot privacy security: Architecture, technology, challenges, and trends,” *IEEE Internet of Things Journal*, 2024.
- [13] M. Amadeo, C. Marche, G. Ruggeri, S. Ranjbaran, and M. Nitti, “Composing digital twins for internet of everything applications: A user-centric perspective,” in *2024 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)*. IEEE, 2024, pp. 73–78.
- [14] J. John and J. S. K., “Predictive digital twin driven trust model for cloud service providers with fuzzy inferred trust score calculation,” *Journal of Cloud Computing*, vol. 13, no. 1, p. 134, 2024.
- [15] A. A. Adewuyi, H. Cheng, Q. Shi, J. Cao, Á. MacDermott, and X. Wang, “Ctrust: A dynamic trust model for collaborative applications in the internet of things,” *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5432–5445, 2019.
- [16] M. Nitti, R. Girau, and L. Atzori, “Trustworthiness management in the social internet of things,” *IEEE Transactions on knowledge and data engineering*, vol. 26, no. 5, pp. 1253–1266, 2013.
- [17] Z. Chen, R. Ling, C.-M. Huang, and X. Zhu, “A scheme of access service recommendation for the social internet of things,” *International Journal of Communication Systems*, vol. 29, no. 4, pp. 694–706, 2016.
- [18] S. Y. Hashemi and F. Shams Aliee, “Dynamic and comprehensive trust model for iot and its integration into rpl,” *The Journal of Supercomputing*, vol. 75, pp. 3555–3584, 2019.
- [19] C. Marche and M. Nitti, “Trust-related attacks and their detection: A trust management model for the social iot,” *IEEE Transactions on Network and Service Management*, vol. 18, no. 3, pp. 3297–3308, 2020.
- [20] C. V. L. Mendoza and J. H. Kleinschmidt, “A distributed trust management mechanism for the internet of things using a multi-service approach,” *Wireless Personal Communications*, vol. 103, pp. 2501–2513, 2018.
- [21] C. Esposito, O. Tamburis, X. Su, and C. Choi, “Robust decentralised trust management for the internet of things by using game theory,” *Information Processing & Management*, vol. 57, no. 6, p. 102308, 2020.
- [22] K. N. Qureshi, A. Iftikhar, S. N. Bhatti, F. Piccialli, F. Giampaolo, and G. Jeon, “Trust management and evaluation for edge intelligence in the internet of things,” *Engineering Applications of Artificial Intelligence*, vol. 94, p. 103756, 2020.
- [23] D. Wu and N. Ansari, “A trust-evaluation-enhanced blockchain-secured industrial iot system,” *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5510–5517, 2020.
- [24] K. N. Tripathi, A. M. Yadav, S. Nagar, and S. Sharma, “Retrust: Reliability and recommendation trust-based scheme for secure data sharing among internet of vehicles (ioV),” *Wireless Networks*, vol. 29, no. 6, pp. 2551–2575, 2023.
- [25] M. B. Al Zami, S. Shaon, V. K. Quy, and D. C. Nguyen, “Digital twin in industries: A comprehensive survey,” *IEEE Access*, 2025.