



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, L. L. Piras and M. Nitti, "Trust Beyond Perfection: Managing Uncertainty in the Internet of Digital Twins," *ICC 2026 - IEEE International Conference on Communications*, Glasgow, United Kingdom, 2026, pp. 1-6.

© 2026 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/ICC59461.2026.11586834>

When citing, please refer to the published version.

Trust Beyond Perfection: Managing Uncertainty in the Internet of Digital Twins

Claudio Marche
University of Cagliari, DIEE
Cagliari, Italy
claudio.marche@unica.it

Luca Luciano Piras
University of Cagliari, DIEE
Cagliari, Italy
lucaluciano.piras@unica.it

Michele Nitti
University of Cagliari, DIEE
Cagliari, Italy
michele.nitti@unica.it

Abstract—Digital Twins (DTs) are increasingly adopted in IoT systems as autonomous agents capable of providing services and interacting within distributed ecosystems. However, most trust management solutions rely on overly idealized assumptions: service evaluations are always accurate, and benevolent nodes never make mistakes. These assumptions limit the applicability of such models in realistic scenarios, where uncertainty and imperfection are the norm. This paper proposes a trust management model tailored to more realistic DTs, where both service providers and requesters may introduce errors. Providers may fail due to physical or software limitations; requesters cannot rely on objective ground truth to evaluate outcomes. To capture this complexity, we introduce a feedback mechanism based on divergence from aggregated results, rather than binary correctness. We define an IoT scenario where DTs dynamically offer and request services. A trust computation model is applied to distinguish truly malicious nodes from those affected by faults, reducing false positives. Simulations confirm improved trust accuracy and system resilience under realistic conditions.

Index Terms—Digital Twins, Trust Management Systems, Feedback Evaluation, Internet of Things.

I. INTRODUCTION

The increasing adoption of Digital Twins (DTs) as autonomous agents in IoT ecosystems has paved the way for more intelligent, distributed, and adaptive systems. DTs are no longer confined to static digital replicas of physical entities but are evolving into collaborative agents capable of exchanging knowledge, making decisions, and delivering services across heterogeneous infrastructures [1]. However, despite their growing relevance, current deployments are still dominated by vertical, application-specific DTs, often isolated and tailored for narrow objectives [2]. The transition toward horizontal integration, where DTs dynamically interact, cooperate, and share resources, introduces new opportunities and new challenges, particularly concerning trust management.

In traditional trust management systems (TMSs) for IoT, the evaluation of a node's behaviour typically relies on idealised feedback, assuming perfect observations and attributing any deviation from expected behaviour to malicious intent [3]. Such simplifications overlook the complexity of real-world environments, where errors may originate from benign causes, such as sensor noise, outdated models, or misaligned contextual data [4]. In this scenario, overly rigid trust models risk excluding useful nodes simply because they operate under suboptimal conditions.

This paper addresses these limitations by focusing on scenarios where both trust requesters and providers operate under uncertainty. On the one hand, requesters can only evaluate a service based on aggregated or subjective perceptions, in the absence of a known ground truth; on the other, providers may offer degraded or incorrect outputs due to limitations in sensing accuracy, computational models, or environmental noise. These imperfections are intrinsic to DT-based systems and must not be mistaken for malicious behavior.

To this end, we introduce a novel trust framework tailored for realistic DT scenarios, where errors and uncertainties are explicitly modelled, and feedback is inherently imperfect. Our scenario builds on a cooperative DT network where each node provides and consumes services, i.e. DTs are considered prosumers, and trust evolves through distributed feedback mechanisms based on observed outcomes and social context. We focus on inter-DT interactions, a largely underexplored area, where collaboration among DTs is crucial for composing services, aggregating knowledge, and enabling robust IoT applications [5]. The main contributions of this work are:

- 1) A feedback model tailored to imperfect and decentralized evaluations, where each requester computes trust based on divergence from aggregated service results, accounting for both observational noise and systemic ambiguity.
- 2) A trust mechanism robust to uncertainty, capable of detecting malicious behaviors while minimizing false positives due to honest but inaccurate providers, preserving collaboration in noisy real-world environments.

In the following, Section II introduces the concept of DTs and discusses the state of the art of TMSs, whereas Section III describes the reference scenario. Moreover, Section IV illustrates the policies to evaluate the feedback, and Section V proposes a trust model that takes into account these realistic aspects. Finally, Section VI analyses the impact of this uncertainty on the model and presents the performance, while Section VII draws conclusions.

II. BACKGROUND

A. The concept of Digital Twin

A DT can be defined as the digital counterpart of a physical entity, dynamically mirroring its properties, state, and

behavior through data-driven models [6]. This representation is maintained in continuous synchronization with the real world, enabling real-time interaction between the two domains. DTs are thus not merely static digital replicas but evolve in parallel with their physical counterparts, continuously assimilating information and providing analytical or predictive elaboration.

Initial DT developments have mainly followed vertical architectures related to specific domains, emphasizing synchronization [7]. These approaches aim to ensure representativeness and consistency. For instance, in the automotive sector, [2] surveys DT implementations centered on battery management and intelligent charging in electric vehicles. Similarly, [8] introduces a DT framework for smart buildings, focusing on air quality control and thermal comfort management. Despite their effectiveness, such vertical systems often operate as isolated systems with limited interoperability and scalability.

To address these limitations, early standardization initiatives have emerged, notably the ISO 23247 framework [9], which proposes a reference architecture for DTs. The model identifies three fundamental entities: the Device Communication Entity, in charge of data collection and command exchange with the physical device; the Digital Twin Entity, responsible for data processing, simulation, and optimization; and the User Entity, which hosts user applications and interacts with DT services. Although the ISO framework represents a significant step toward DT systems, several open issues remain unresolved, particularly concerning their reliability and the exchange of data among interconnected DTs.

B. Trust Management Systems

Existing TMSs can be broadly classified according to the type of information used for trust evaluation. The first category encompasses models based on parameters, where trust is derived from the static characteristics of providers. These include objective indicators such as service latency, availability, or energy efficiency, as well as contextual or social factors describing the node's role in the network [10]. A second group of models builds trust from historical behavior, using past interactions as the primary input for evaluation. These systems continuously update trust scores through learning mechanisms or statistical aggregation of feedback [11]. Some models integrate distributed computation or blockchain records to prevent tampering, thus providing transparency and resilience against manipulation [12]. The third class relies on models based on recommendations, where nodes exchange opinions to estimate the reliability of unknown nodes. These frameworks often weight recommendations according to the credibility of the recommending node, sometimes supported by reputation management to mitigate false information [13]. While this approach accelerates trust propagation across the network, it may also amplify erroneous evaluations if not carefully evaluated. Despite their differences, most existing TMSs share common limitations. They typically assume perfect service evaluations and tend to treat all deviations from expected behavior as malicious. However, in realistic DT environments,

where errors could occur, these simplifications can lead to misclassifying benevolent as untrustworthy.

Moreover, in general, state-of-the-art TMSs primarily focus on the detection of malicious behaviors, aiming to prevent uncooperative nodes from compromising the network [14]. The literature identifies a variety of trust-related attacks, which can be divided into static and dynamic types depending on whether malicious behavior remains constant or varies over time. Static attacks deliver poor services, while dynamic ones alternate between cooperative and malicious behavior to avoid detection. Both forms disrupt the reliability of networks and highlight the need for TMSs capable of operating in real-world DT scenarios, affected by errors and attackers.

III. SCENARIO

The scenario is modeled as a set of interconnected DTs denoted by $\mathcal{N} = \{n_1, n_2, \dots, n_i, \dots, n_I\}$, where I represents the total number of nodes in the network and n_i a generic DT. The resulting structure can be represented as an undirected graph $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$, where $\mathcal{E} \subseteq \{\mathcal{N} \times \mathcal{N}\}$ is the set of edges. Each edge identifies a bidirectional communication between two DTs. For each node n_i , it is defined the set of its direct neighbors as $\mathcal{N}_i = \{n_j \in \mathcal{N} : (n_i, n_j) \in \mathcal{E}\}$, i.e., the DTs with which n_i can directly interact. Moreover, given two DTs n_i and n_j , it is defined $\mathcal{H}_{ij} = \{n_z \in \mathcal{N} : n_z \in \mathcal{N}_i \cap \mathcal{N}_j\}$ as the set of their common neighbors. Every DT in the network can provide one or more services. Let $\mathcal{S}_j$ denote the set of services offered by node n_j . When a DT n_i requests a particular service s_h , a discovery mechanism identifies the subset of possible providers $\mathcal{P}_h = \{n_j \in \mathcal{N} : s_h \in \mathcal{S}_j\}$. The requester then needs to decide which provider to select among those in $\mathcal{P}_h$, based on their trust level. The trust value is derived from the analysis of different providers, as will be illustrated in the next sections of this manuscript, and computed locally by each node using the feedback gathered from past interactions, so that different DTs may hold distinct opinions about the same peer depending on their individual experience. After each service interaction k between requester n_i and provider n_j , the requester assigns a feedback value $f_{ij}^k \in [0, 1]$, where 1 indicates full satisfaction and 0 a completely unsatisfactory outcome. The set of all feedback values related to the interactions between n_i and n_j can thus be expressed as $\mathcal{F}_{ij} = \{f_{ij}^1, f_{ij}^2, \dots, f_{ij}^{K_{ij}}\}$, with K_{ij} being the last transaction between them. In the following, to avoid overloading the notation, the superscript k is omitted for variables related to interactions; the index is explicitly shown when required, such as in expressions summing over interactions.

An illustrative example of the proposed scenario is reported in Figure 1. In this example, the DT n_1 issues a request for service s_7 . The Service Discovery component analyzes the network, considering only neighboring nodes in a decentralized scenario or the entire network in a centralized one, and identifies the set of possible providers. The Node Selection phase then chooses the most suitable nodes among these candidates, which may also be selected based on their

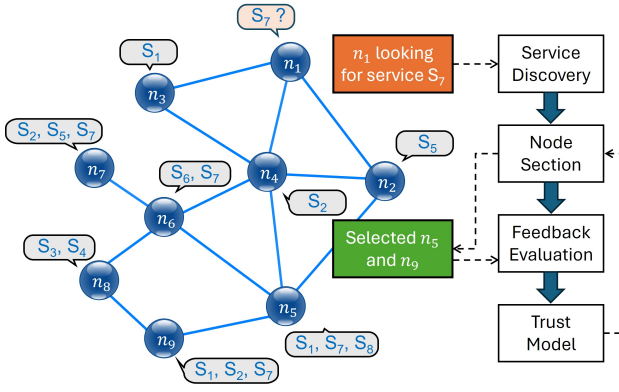


Fig. 1. Reference scenario.

trust levels, as will be illustrated in the next section. In this example, n_5 and n_9 are chosen to provide service s_7 . After the service delivery, the Feedback Evaluation module assesses the quality of the received services and produces a feedback score for each provider. Specifically, the Node Selection and the Feedback Evaluation are modeled as realistic components that introduce overheads (i.e., retrieving the same data from multiple nodes) and uncertainties (i.e., non-binary feedback), influencing trust updates and subsequent service interactions. Indeed, these feedback values are then processed by the Trust Model, which updates the trust levels of the involved nodes.

As illustrated in section II, each DT can exhibit either benevolent or malicious behavior. Benevolent DTs are designed to cooperate by correctly providing the requested services; however, even these entities are not immune to inaccuracies. Indeed, realistic DTs may produce outputs affected by errors due to hardware limitations, communication noise, or data processing inaccuracies. Consequently, even cooperative DTs can introduce uncertainty into the system, leading to errors in the provided service values. In this context, the service providing error is defined as the deviation between the actual service value generated by a DT and the corresponding ground truth value during transaction k , denoted as v_{true} . The error can be modeled as a Gaussian distribution centered around the true value, representing the likelihood that a DT provides a service value close to or far from the expected one. Let δ be the error factor associated with the service provided by a benevolent DT. The probability function of the provided value v_{ij} , representing the service delivered by node n_j to requester n_i , can be expressed as:

$$Prob(v_{ij}) = e^{-\frac{(v_{ij} - v_{true})^2}{2\sigma^2}} \quad (1)$$

where the parameter σ controls the spread of the Gaussian curve: smaller values of σ correspond to highly precise DTs with low measurement errors, while higher values represent less reliable but still benevolent DTs. Specifically, σ is related to the probability of error p of the provider through the expression $\sigma^2 = \delta \cdot p$, where δ is a scaling factor that determines the magnitude of the error variance. The ground truth value is

located at the center of the distribution, corresponding to the expected service output.

In the next section, the feedback evaluation mechanism is discussed along with the node selection phase, highlighting that inaccuracies in interactions derive not only from errors introduced by service providers but also from those arising on the requester side, where the received service is evaluated and scored.

IV. FEEDBACK EVALUATION MECHANISM

Given a service request for s_h issued by requester n_i , the Service Discovery component returns the candidate set of providers $\mathcal{P}_h = \{n_j \in \mathcal{N} : s_h \in \mathcal{S}_j\}$, from which the Node Selection chooses the subset to query. Specifically, providers are first sorted according to their trust values T_{ij} , and the most trustworthy ones are selected, corresponding to a fixed percentage of the top-ranked nodes, while ensuring a minimum number of providers is always included.

Once the subset of providers is defined, each selected DT n_j delivers its service output v_{ij} in response to the request of n_i . Let $\mathcal{C}_h \subseteq \mathcal{P}_h$ denote the set of providers actually selected during transaction k for service s_h . After all responses are collected, the requester computes a reference value, referred to as the pseudo ground truth v^* , which serves as a benchmark for evaluating the received results. This pseudo ground truth is obtained by aggregating all service values provided in the same transaction, using one of several possible strategies. In line with previous work [15], the pseudo value can be computed as the average or the median of the collected values. In addition, two variants are introduced: the *trusted average* and the *trusted median*. In the trusted average, each contribution v_{ij} is weighted according to the current trust value of the provider:

$$v^* = \frac{\sum_{n_j \in \mathcal{C}_h} T_{ij} \cdot v_{ij}}{\sum_{n_j \in \mathcal{C}_h} T_{ij}} \quad (2)$$

Similarly, the trusted median is obtained by applying the median operation only to the subset of the most trusted nodes.

As soon as the pseudo ground truth v^* is computed, each provider's output v_{ij} is compared against it to derive a feedback score $f_{ij} \in [0, 1]$. This evaluation follows a triangular function centered at v^* , where the feedback assumes its maximum value when the received service exactly matches the pseudo ground truth and decreases linearly as the deviation increases. Let $d_{ij} = |v_{ij} - v^*|$ denote the absolute deviation between the provided and the reference values. Then, the feedback function is defined as:

$$f_{ij} = \begin{cases} 1 - \frac{d_{ij}}{\tau}, & 0 \leq d_{ij} \\ 0, & d_{ij} > \tau, \end{cases} \quad (3)$$

where τ is a parameter defining the maximum acceptable deviation from the pseudo ground truth. Within this range, the feedback decreases linearly from 1 (perfect match) to 0 (divergent output). This formulation allows the system to adapt

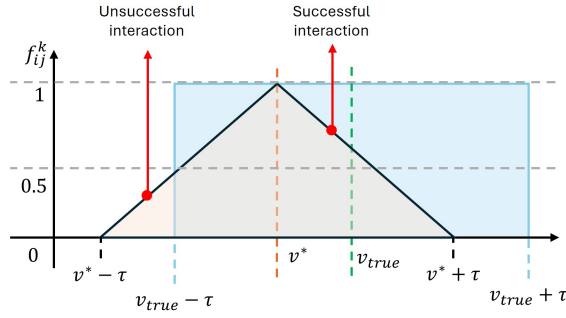


Fig. 2. Triangular feedback function used for the feedback evaluation. The blue region represents the application accuracy determined by τ .

the evaluation depending on the application or service type. For instance, applications requiring high precision can use smaller τ values, whereas more tolerant contexts may adopt larger thresholds.

As illustrated in Figure 2, the pseudo ground truth v^* is used as a reference to evaluate the received services, even though the actual true value v_{true} is unknown to the requester. The difference between these two values represents the evaluation error introduced by the requester in the feedback computation process. This misalignment may lead to inaccurate assessments: for example, an interaction that is unsuccessful from the application standpoint (i.e., $|v_{ij} - v_{true}| > \tau$) may still receive a non-zero feedback; this situation is highlighted by the limit case in the figure, with a large separation between v^* and v_{true} . These variations can originate from the uncertainty of benevolent providers, whose outputs are affected by sensing or computation errors, but significant discrepancies are generally caused by malicious behaviors. In this context, a trust management model is required to mitigate these effects, isolating malicious nodes and preventing them from being selected as future service providers.

V. TRUST MANAGEMENT SYSTEM

As illustrated in Section III, once the feedback values f_{ij}^k are generated, the TMS makes use of them to estimate the reliability of each provider. The model combines direct observations and neighbor recommendations, and its purpose is twofold: to reduce the impact of errors produced in the requester and provider sides, and to identify malicious entities, excluding them from future service selections. After each interaction k , the requester n_i updates its trust toward provider n_j as follows:

$$T_{ij} = \theta_{ij} R_{ij} + (1 - \theta_{ij}) D_{ij} \quad (4)$$

where $\theta_{ij} \in [0, 1]$ controls the influence of the recommendations and direct components. This parameter decreases exponentially with the number of interactions between i and j , following $\theta_{ij} = e^{-\lambda K_{ij}}$, with $\lambda > 0$ and K_{ij} representing the total number of interactions. Consequently, recommendations dominate during the initial phase, while the direct component becomes more relevant as experience accumulates.

The recommendation component R_{ij} aggregates the opinions that requester n_i receives from its neighbors $\mathcal{N}_i$, as follows:

$$R_{ij} = \frac{\sum_{z \in \mathcal{N}_i} T_{iz} T_{zj}}{\sum_{z \in \mathcal{N}_i} T_{iz}}, \quad (5)$$

with the convention $R_{ij} = 0.5$ if no valid recommendations are available. This weighted mechanism ensures that poorly trusted recommenders have minimal impact on the overall trust estimation.

On the other hand, the approach of the direct component evaluation relies on a clustering approach, where each node maintains a centroid representing the typical behavior of peers. The direct component D_{ij} captures both long-term and short-term behavioral trends of the provider. The long-term value represents the historical mean of all feedback:

$$L_{ij} = \frac{1}{K_{ij}} \sum_{k=1}^{K_{ij}} f_{ij}^k, \quad (6)$$

while the short-term value focuses on recent interactions through an exponentially decreasing weight function:

$$N_{ij} = \sum_{k=1}^{K_S} w_k f_{ij}^k, \quad (7)$$

where K_S is the size of the window used to emphasize the most recent interactions. The weight w_k follows a geometric distribution so that the most recent feedback values have a higher influence. Specifically:

$$w_k = \rho(1 - \rho)^{k-1} + \frac{\varepsilon_{res}}{K_S}, \quad (8)$$

where $\rho \in (0, 1]$ controls the decay rate, and the residual term ε_{res} compensates for the truncated tail of the infinite geometric series:

$$\varepsilon_{res} = \sum_{r=K_S+1}^{\infty} \rho(1 - \rho)^{r-1}. \quad (9)$$

Let $\mathbf{x}_{ij} = (L_{ij}, N_{ij})$ denote the behavior point mapped within $[0, 1]^2$. Requester n_i maintains a centroid $\mathbf{c}_i$ representing the center of the trustworthy cluster. The centroid is updated only when the feedback is above a reliability threshold TH, to avoid the influence of malicious responses:

$$\mathbf{c}_i = \begin{cases} \mathbf{c}_i^{(k-1)} + \mu(\mathbf{x}_{ij} - \mathbf{c}_i^{(k-1)}), & \text{if } f_{ij}^k > \text{TH} \\ \mathbf{c}_i^{(k-1)}, & \text{otherwise} \end{cases} \quad (10)$$

with in the first interaction $\mathbf{c}_i^0 = (0.5, 0.5)$ and $\mu \in (0, 1]$. The direct trust D_{ij} is then computed from the normalized Euclidean distance between the current behavior point and the centroid:

$$D_{ij} = 1 - \frac{\|\mathbf{x}_{ij} - \mathbf{c}_i\|_2}{\sqrt{2}} \quad (11)$$

with $D_{ij} \in [0, 1]$.

VI. PERFORMANCE EVALUATION

A. Simulation Setup

The goal of this Section is to evaluate the behavior of the proposal under different network conditions and malicious behaviors. The model is tested considering DTs mirroring temperature devices as a single representative service in order to accelerate simulations. This is done just to showcase the system's behaviour under steady-state conditions, but it is extensible to other service types. The setup employs a portion of a well-known IoT network [16] comprising 791 nodes, in which these are interconnected through predefined links. Furthermore, the nodes have been partitioned into 27 groups. This is achieved using a Dirichlet Process (DP) [17], due to its ability to produce clusters with a skewed size distribution, with a concentration parameter set to $\alpha = 0.8$, resulting in 27 clusters. Each group corresponds to a unique location and, to provide a realistic background, the network is enhanced with real data from 27 different cities in the US [18].

Each physical device generates data according to Section III, using a scaling factor ($\delta = 1$) and varying the parameter p to emulate devices with different accuracies. Feedback values are computed through a triangular function with $\tau = 1.7$, which allowed obtaining a not too negative feedback value (around 0.4) with a variation of $\pm 1^\circ\text{C}$. Moreover, regarding the trust model described in Section V, the trust value T_{ij}^0 between each pair of nodes is initially set to 0.5, while all the parameters are kept constant across simulations to ensure comparability. In particular, λ is set equal to 0.35, which assigns a meaningful weight to recommendations while ensuring that, after five interactions, their impact is reduced to less than 10% of that of direct interactions. The parameter $\rho = 0.5$ controls the short-term component of the trust computation related to the last $K_S = 10$ interactions, whereas $\mu = 0.5$ determines the influence of each individual interaction on the centroid update. Finally, the threshold defining the minimum feedback required to update the centroid is set to $TH = 0.4$.

The simulations have been conducted emulating the interactions among DTs, where each simulation comprises 300,000 queries. Specifically, query-based interactions have been run, where each DT (the requester) queried a group of DTs (the providers) for the desired service. The providers are then selected as the top 25% most trusted nodes when the group size is sufficiently large; otherwise, a fixed minimum number of providers is enforced. Finally, malicious nodes are modeled to provide incorrect services by reporting values from a Gaussian distribution whose mean is offset from the ground truth. This offset is randomly chosen from a range of $[-5, +5]$.

B. Performance

The initial set of simulations is designed to compare four aggregation strategies for computing the pseudo ground truth: simple average (AVG), trust-weighted average (T-AVG), median (MED), and trust-weighted median (T-MED). Two values of error probabilities have been tested, with $p \in \{0.1, 0.4\}$, together with two percentages of malicious nodes, i.e. 40%

TABLE I
PERFORMANCE COMPARISON OF AGGREGATION METHODS.

Malicious	p	AVG		T-AVG		MED		T-MED	
		init.	steady	init.	steady	init.	steady	init.	steady
40%	0.1	0.40	0.19	0.40	0.15	0.15	0.10	0.15	0.11
	0.4	0.48	0.30	0.48	0.26	0.30	0.22	0.32	0.22
70%	0.1	0.59	0.45	0.59	0.36	0.39	0.19	0.40	0.16
	0.4	0.60	0.49	0.60	0.42	0.52	0.33	0.55	0.30

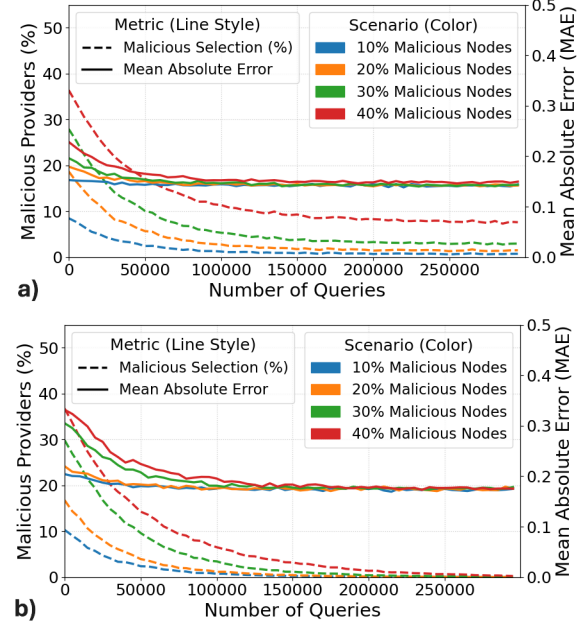


Fig. 3. Comparison of the malicious provider selection ratio and the MAE for two minimum provider settings: (a) 10 providers and (b) 5 providers.

and 70%. Accuracy is quantified via the Mean Absolute Error (MAE) between the unknown true value v_{true} and the estimated pseudo ground truth v^* in the initial phase (on the first 10 queries per requester) and the steady-state phase. Table I summarizes the results, from which a pattern can be observed: accuracy is degraded as either the error probability or the percentage of malicious providers increases. Nevertheless, strategies based on the median are found to be more robust, in which the influence of attackers is attenuated as these are treated as outliers, limiting their effect on the pseudo ground truth. When trust weights are incorporated, robustness is further improved in the steady state, since contributions from low-trust providers are poorly weighted. For these reasons, the trust-weighted median is adopted in the next simulations as the default aggregator for computing the pseudo ground truth.

The second experiment, in Figure 3, examines the model's filtering ability, comparing two configurations with a minimum of 10 providers in the selection mechanism (Figure 3a) and 5 ones (Figure 3b), both with $p = 0.2$. The results reveal a clear trade-off between filtering effectiveness and aggregation accuracy. The second configuration excels at filtering: the percentage of selected malicious providers drops to zero, as the low threshold allows the system to discard them. Conversely,

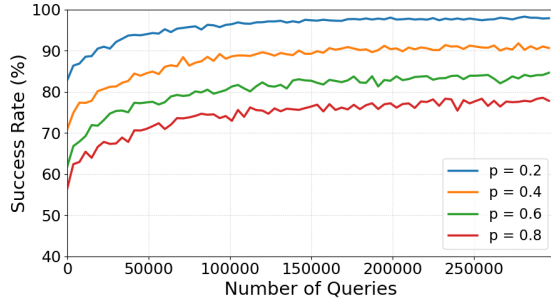


Fig. 4. Success rate for different error probabilities p in a network with 40% malicious nodes.

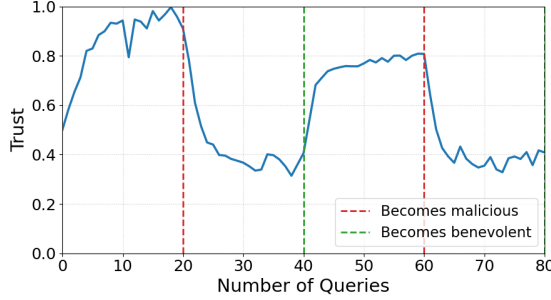


Fig. 5. Trust for a single dynamic provider alternating behaviors every 20 queries.

the first configuration achieves a lower MAE. This result occurs because the trust-weighted median benefits from the larger statistical sample. This larger sample is more effective at decreasing the effects of inaccuracies, even though the higher number of providers forces the inclusion of a few malicious nodes, discarded by the trusted approach.

To further test the model, the success rate is evaluated in a network with 40% malicious providers, and with a minimum provider threshold equal to 5. The simulation is analyzed in terms of success rate, where a query is successful if $|v_{true} - v^*| < 0.5$. As shown in Figure 4, for all values of p the success rate rises rapidly at first and then stabilizes at the last queries. This behavior indicates that performance is no longer constrained by malicious DTs, which are effectively discarded, but by the probability of error of benevolent nodes.

Finally, the model is tested against dynamic malicious nodes. A single provider is programmed to alternate between benevolent and malicious behavior every 20 queries. As shown in Figure 5, trust rises rapidly during the initial benevolent phase (0–20 queries) and then drops after the first switch (20–40). When benevolent behavior resumed (40–60), trust recovered but remained below its earlier maximum, indicating persistence of negative history and a more conservative re-admission of the provider, thereby protecting the system against dynamic attacks and reducing the opportunity for repeated attacks.

VII. CONCLUSIONS

This paper introduced a trust model for Digital Twin systems operating under realistic conditions, where errors from

both providers and requesters are possible. By leveraging aggregated feedback rather than binary evaluations, our model reduces false positives and supports more inclusive trust management. Simulation results confirm improved accuracy and resilience, even in noisy environments. Future work will explore adaptive strategies and real-world deployment.

ACKNOWLEDGMENT

This work has been supported by Fondazione di Sardegna, research project “DITTO - Digital Twin as The bridge between meTaverse and internet Of things”, CUP F23C25000300007, call 2023.

REFERENCES

- [1] L. U. Khan, W. Saad, D. Niyato, Z. Han, and C. S. Hong, “Digital-twin-enabled 6g: Vision, architectural trends, and future directions,” *IEEE Communications Magazine*, vol. 60, no. 1, pp. 74–80, 2022.
- [2] G. Bhatti, H. Mohan, and R. R. Singh, “Towards the future of smart electric vehicles: Digital twin technology,” *Renewable and Sustainable Energy Reviews*, vol. 141, p. 110801, 2021.
- [3] C. Marche and M. Nitti, “A binary trust game for the internet of things,” *IoT*, vol. 2, no. 1, pp. 50–70, 2021.
- [4] M. Aaqib, A. Ali, L. Chen, and O. Nibouche, “Iot trust and reputation: a survey and taxonomy,” *Journal of Cloud Computing*, vol. 12, no. 1, p. 42, 2023.
- [5] R. Minerva, G. M. Lee, and N. Crespi, “Digital twin in the iot context: A survey on technical features, scenarios, and architectural models,” *Proceedings of the IEEE*, vol. 108, no. 10, pp. 1785–1824, 2020.
- [6] A. Talkhestani *et al.*, “An architecture of an intelligent digital twin in a cyber-physical production system,” *at-Automatisierungstechnik*, vol. 67, no. 9, pp. 762–782, 2019.
- [7] D. M. Botín-Sanabria, A.-S. Mihaita, R. E. Peimbert-García, M. A. Ramírez-Moreno, R. A. Ramírez-Mendoza, and J. d. J. Lozoya-Santos, “Digital twin technology challenges and applications: A comprehensive review,” *Remote Sensing*, vol. 14, no. 6, p. 1335, 2022.
- [8] L. Hadjidemetriou, N. Stylianidis, D. Englezos, P. Papadopoulos, D. Eliades, S. Timotheou, M. M. Polycarpou, and C. Panayiotou, “A digital twin architecture for real-time and offline high granularity analysis in smart buildings,” *Sustainable Cities and Society*, vol. 98, p. 104795, 2023.
- [9] “Iso 23247-2:2021 automation systems and integration - digital twin framework for manufacturing - part 2: Reference architecture,” 2021.
- [10] 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.
- [11] F. Ullah, A. Salam, F. Amin, I. A. Khan, J. Ahmed, S. A. Zaib, and G. S. Choi, “Deep trust: A novel framework for dynamic trust and reputation management in the internet of things (iot)-based networks,” *IEEE Access*, vol. 12, pp. 87 407–87 419, 2024.
- [12] 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.
- [13] 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.
- [14] S. A. Alhandi, H. Kamaludin, and N. A. M. Alduais, “Trust evaluation model in iot environment: A comprehensive survey,” *IEEE Access*, vol. 11, pp. 11 165–11 182, 2023.
- [15] C. Marche and M. Nitti, “Can we trust trust management systems?” *IoT*, vol. 3, no. 2, pp. 262–272, 2022.
- [16] C. Marche, L. Atzori, V. Pilloni, and M. Nitti, “How to exploit the social internet of things: Query generation model and device profiles’ dataset,” *Computer Networks*, vol. 174, p. 107248, 2020.
- [17] A. Chowdhury, A. Pal, A. Raut, and M. Kumar, “Kihcdp: an incremental hierarchical clustering approach for iot data using dirichlet process,” *IEEE Access*, vol. 12, pp. 56 019–56 032, 2024.
- [18] Hourly weather data 2012–2017 [accessed: Oct. 1, 2025]. Kaggle. [Online]. Available: <https://www.kaggle.com/datasets/selfishgene/historical-hourly-weather-data>