2024 International Conference on Computing, Networking and Communications (ICNC): Communications and Information Security Symposium

On the Importance of Trust in Next-Generation Networked Cyber-Physical Systems

Nariman Torkzaban*, Anousheh Gholami*, and John S. Baras
Department of Electrical & Computer Engineering,
& Institute for Systems Research
Email: {narimant, anousheh, baras}@umd.edu

***Abstract*—With 5G and beyond promises to realize massive machine-type communications, a wide range of applications have driven interest in complex heterogeneous networked systems, including multi-agent optimization, large-scale distributed learning, 5G service provisioning, etc. This trend highlights the essence of seamless control, management, and security mechanisms to be in place for the next-generation networked cyber-physical systems (CPS). In this paper, we interpret *trust* as a relation among networked collaborating entities that can set forth a measure for evaluating the status of network components and secure the execution of the collaborative protocol. In this paper, we will first elaborate on the importance of trust as a metric and then present a mathematical framework for trust computation and aggregation within a network. We consider two use-case examples where trust can be incorporated into the next-generation networked CPS and improve the security of decision-making, i.e. i) federated learning (FL), and ii) network resource provisioning. Finally, we explain the challenges associated with aggregating the trust evidence and briefly explain our ideas to tackle them.**

***Index Terms*—Cyber-physical systems, trust model, trusted federated learning, trusted network service placement.**

I. Introduction

Given the escalating intricacy characterizing modern cyber-physical systems (CPS), the imperative to formulate an innovative framework for modeling, analyzing, and predicting their behaviors has become increasingly evident. This endeavor assumes heightened significance in light of recent advancements in the Internet of Things (IoT), coupled with the promises of 5G to facilitate extensive machine-to-machine (M2M) communications. In this evolving landscape, tightly coupled next-generation CPS devices are poised to engage in collaborative efforts, leveraging sophisticated sensing, computing, and communication capabilities on an amplified scale enabling them to realize a wide range of applications and use cases, involving data collection, processing, and decision-making, from healthcare, vehicular networks, and smart manufacturing, to 5G service provisioning, and content delivery. All these applications heavily rely on the constant exchange of collected raw data and processed information between the collaborating agents, as opposed to the traditional case where data were collected and processed at a centralized entity. Therefore, with the heterogeneity and the large scale of the CPS, as well as the paramount importance of devising a seamless management and control scheme dealing with privacy and security threats becomes a pivotal concern.

The fact that the information is crowd-sourced by the CPS agents, to a large extent, eliminates the risk of the existence of a single point of failure and contributes to the resilience of the network, but at the same time demonstrates the need to establish trust relationships between the agents that are exchanging information. More specifically, apart from ensuring the security of communications between the network agents, it is essential to answer the following questions: (i) whether an agent refuses to share its information with other agents due to privacy concerns or conflict of interest. (ii) whether an agent manipulates the received data before processing. (iii) whether an agent intentionally or unintentionally, shares incorrect information with the rest of the network? etc. [1] [2]; In other words, it is essential to establish to what extent each agent of the network can be trusted. Such mechanisms of trust contribute essentially to the resilience of networked cyber-physical systems (Net-CPS).

Within the context of Net-CPS, we interpret trust as a relation between network entities that may interact or collaborate in groups toward achieving various goals. These relations are set up and updated based on the evidence generated from the previous collaboration of the agents within a protocol. Suppose the collaboration has been contributive towards the achievement of a specific goal (positive evidence). In that case, the parties accumulate their trust perspective towards one another, and otherwise (negative evidence), trust will decrease between them. Trust estimates have input to decisions such as access control, resource allocation, agent participation, and so on. The method by which trust is computed and aggregated within the network may depend on the specific application, however similar to [2], we enumerate the central differences in the terminology of how the trust computation and aggregation are employed:

Centralized vs. Decentralized: Under *centralized* regime, all the network entities rely on a central trusted party that estimates the trustworthiness level of each entity and updates all the network nodes. In this sense, all the nodes are forced to agree on the degree to which each entity is trusted as dictated by the central provider. On the other hand, under the *decentralized* approach, each user is responsible for calculating its opinion on the level of trustworthiness for each entity it might be interested in. This distinction however is irrelevant to the fashion trust is computed and only relates to the semantics of trust. For instance, under a decentralized regime, a user may utilize a distributed approach for computing the trust of

* The identified authors contributed equally to this paper.

979-8-3503-7099-7/24/$31.00 ©2024 IEEE

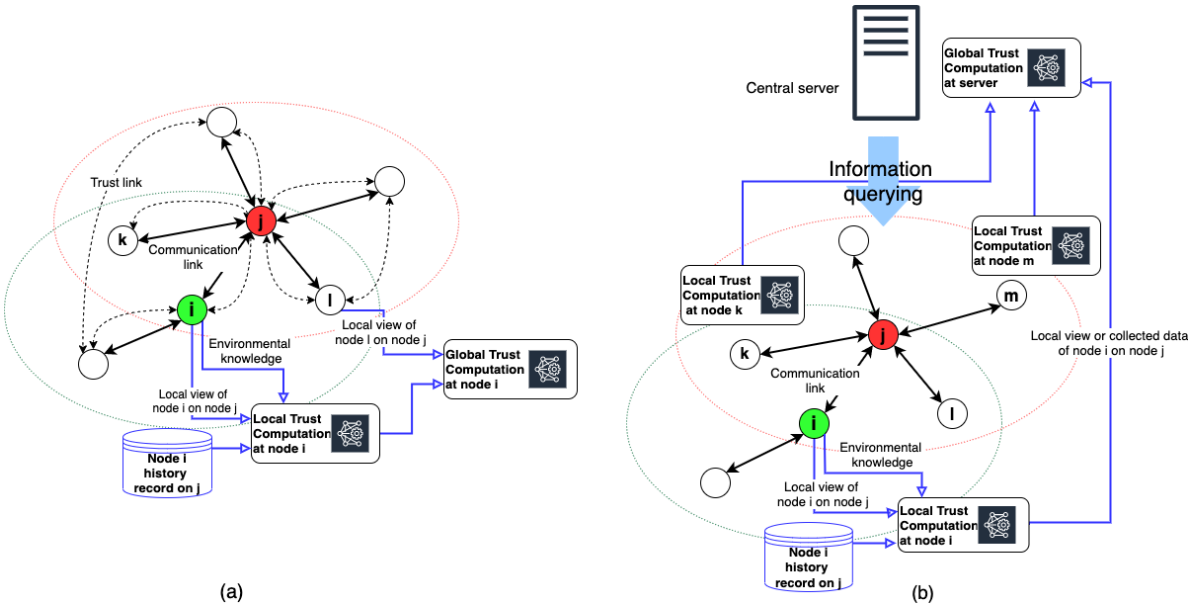

Fig. 1. Trust aggregation framework in (a) decentralized and (b) centralized regimes

its target.

Global vs. Local: *Local* trust is the opinion that a trustor node has towards a trustee generated depending on the first-hand evidence gathered based on local interactions, however, *global* trust is formed by combining the first-hand evidence and the opinions of other nodes about the specific trustee and is usually more accurate. The local exchange of the local observations is used towards obtaining global trust [3].

Proactive vs. Reactive: Under a *proactive* regime, the entities manage to keep the trust estimates updated, while under a *reactive* regime, the trust estimates are computed only when they are required. The proactive scheme is not communication efficient as a large bandwidth needs to be consumed to keep the trust values updated; therefore a reactive scheme is usually preferred unless the frequency by which trust decisions are made is comparable to the frequency of the local trust updates.

Direct vs. Indirect: Directed trust is obtained via interaction through direct communication with another agent. However, *indirect* trust is a trust relationship between two entities that have not interacted in the past. Establishing an indirect trust relationship heavily relies on the assumption that trust has the *transitivity* property which is not necessarily the case in any application.

The remainder of the paper is organized as follows. Section III describes the trust aggregation model, including the aggregation framework, the local, and the global trust models. In sections IV and V, we present two example use cases of the trust aggregation model in securing the next-generation multi-agent systems protocols. Finally, in section VI we present the challenges encountered while attempting to design realistic trust aggregation frameworks.

II. Related Work

Trust management, aggregation, and inference are multi-faceted concepts with widespread implications in networked systems. Within the context of ad-hoc networks, distributed trust models are envisioned that enable nodes to evaluate the trustworthiness of their peers based on observed behavior and interactions [4] [5] [6]. These models employ metrics like reputation, historical data, and network statistics to compute trust scores, allowing nodes to make informed decisions. Reputation systems are widely used in online communities, e-commerce, and peer-to-peer networks [7]. They allow users to assess the reliability of other participants by aggregating feedback and ratings. Trust is often correlated with reputation, making it a central element in decision-making processes [8]. In blockchain and distributed ledger technologies, consensus algorithms such as Proof of Work (PoW) and Proof of Stake (PoS) play a role in trust management. These algorithms facilitate agreement among nodes to validate transactions and maintain the integrity of the ledger. The application of social network analysis methods provides insights into trust relationships among individuals and entities in online and offline social networks [9]. Machine learning techniques are leveraged to improve trust management and inference. They allow for the development of predictive models based on historical data, aiding in trust evaluation and anomaly detection [10].

III. Trust Aggregation Model

A. Trust Aggregation Framework

In this section, we present two schemes for propagating and aggregating the trust estimates within a network of CPS devices as depicted in fig. 1. The first scheme corresponds to the case where there is no central entity involved in estimating the trustworthiness of the network agents, and

the nodes participate in direct computation of trust to obtain *local* trust estimates on the other peers, using the locally-available first-hand evidence they have gathered, the recorded history they have stored from the past observations, and the knowledge they obtain by sensing the environment. Once all agents form their local views, they will participate in the local exchange of their local trust estimates to form more accurate *global* values for the trustworthiness of the networked agents. Then, the obtained global trust model can be used in the corresponding trust-aware applications.

Within the second scheme, however, the global trust values are obtained indirectly. There exists a central trusted party that is constantly monitoring the network and communicating with the CPS agents to gather evidence on their state. The CPS agents may share their local view on their neighbors with the central entity which may be used in computing the trust estimates by the central entity. Once the central party calculates the trust values of the CPS agents using the information it has gathered, it will push the relevant information to each agent. The calculated trust values can be used by the central entity to perform centralized trust-aware decision-making or can be used by each agent to participate in local or distributed trust-aware protocols.

In what follows, we will formalize the above discussion to mathematically model the processing, propagating, and aggregation of the trust values. The components of our model mostly rely on the discussion in [3]. We model the network of agents at time instance k, as an undirected graph $\mathcal{G}^{(k)} = (\mathcal{N}^{(k)}, \mathcal{L}^{(k)})$ where $\mathcal{N}$ is the set of nodes and for $n, m \in \mathcal{N}^{(k)}$, $\mathcal{L}^{(k)}$ contains all links $(m, n)^{(k)}$ where agents m, and n can communicate with one another at time instance k. We denote this graph as the *communication graph* at time instance k. Let $\mathcal{N}_i^{(k)}$ be the set of neighbors of node i at time step k. Apart from the communication relationship, we define *local trust* relationships between nodes $i, j \in \mathcal{N}^{(k)}$. Let $\tau_{ij}^{(k)}$, and $t_{ij}^{(k)}$ be the local and global view of node i on trustworthiness of the node j at time instance k in respective order. We may ignore the index k in the following discussion for simplicity whenever doing so does not lead to any confusion.

B. Local Trust Model

To formalize the definition of local trust, let us define $X_{ij}^{(k)}$ to be a random variable denoting the reputation that node j has in the perspective of node i in time instance k. $X_{ij}^{(k)}$ follows a Beta distribution with parameters $\alpha_{ij}^{(k)}$, and $\beta_{ij}^{(k)}$. Moreover, define $r_{ij}^{(k)} = \alpha_{ij}^{(k)} - 1$, and $s_{ij}^{(k)} = \beta_{ij}^{(k)} - 1$ that determine the number of times up to round k, that node j's behavior is benign and malicious in perspective of node i, in respective order. The method for obtaining $r_{ij}^{(k)}$ and $s_{ij}^{(k)}$ relies on the particular scenario and will be explicitly outlined in the upcoming section. We let $\tau_{ij}^{(k)}$ to be precisely the expected value of the reputation random variable in the Beta system $X_{ij}^{(k)}$. Formally, we have:

$$f_{X_{ij}^{(k)}}(x; \alpha_{ij}^{(k)}, \beta_{ij}^{(k)}) = \left(\frac{\Gamma\left(\alpha_{ij}^{(k)} + \beta_{ij}^{(k)}\right)}{\Gamma\left(\alpha_{ij}^{(k)}\right)\Gamma\left(\beta_{ij}^{(k)}\right)}\right) \cdot \left(x^{\alpha_{ij}^{(k)}-1}\left(1-x\right)^{\beta_{ij}^{(k)}-1}\right) \quad (1)$$

$$\tau_{ij}^{(k)} = \mathbb{E}\left[X_{ij}^{(k)}\right] = \frac{r_{ij}^{(k)} + 1}{r_{ij}^{(k)} + s_{ij}^{(k)} + 2} \quad (2)$$

Intuitively, the evolution of r, and s parameters need to be in a way that the more recent information receives more relative importance compared to the older ones. Therefore, we define $0 < \rho_1 < \rho_2$ as forgetting factors to control the balance between old and new terms.

$$r_{ij}^{(k+1)} = \rho_1 r_{ij}^{(k)} + I_{ij}^{(k+1)} \quad (3)$$

$$s_{ij}^{(k+1)} = \rho_2 s_{ij}^{(k)} + 1 - I_{ij}^{(k+1)}, \quad (4)$$

Function $I_{ij}^{(k+1)} \in [0, 1]$ models the instantaneous perspective of node i on the behavior of node j in $(k+1)^{th}$ round.

C. Global Trust Model

At each instance, k, within the local trust model, each node i computes its local trust for all nodes $j \in \mathcal{N}_i$ in the communication graph. To make more accurate estimates, node i will need to take into account the opinions of other network nodes who have first-hand evidence of node j's behavior. Following the approach in [3], node i computes iteratively its global trust estimate for node j, i.e. $t_{ij}^{(k)}$ using the opinions of its neighbors as:

$$t_{ij}^{m} = \begin{cases} 1 & \text{if } i = j \\ \sum_{l \in \mathcal{N}_i, l \neq j} w_{il} t_{lj}^{m-1} & \text{if } i \neq j \end{cases} \quad (5)$$

where $w_{il} = \frac{\tau_{il}}{\sum_{l \in \mathcal{N}_i, l \neq j} \tau_{il}}$. In other words, node i pays more attention to the opinions of those of its neighbors whom it trusts more. We note again that the global trust computation is an iterative process that is going to be embedded in each iteration of the trust-aware protocol. Therefore, to avoid any confusion we have used the iteration counter m for this process. Here, we have dropped the superscript k as we assume the value of local trust remains constant within the loop of computing the global trust.

In the upcoming section, we will demonstrate two examples of the practical application of the trust framework mentioned earlier in bolstering the security of the next-generation CPS-based protocols. We will select as the use cases two challenging problems in the domain of collaborative multi-agent systems and network resource provisioning where either it is inherently challenging to ensure the security of the procedure, or security is a service requirement.

IV. Use Case I: Trusted Federated Learning

The recent advancements in AI/ML technologies have led to an increasing demand for the deployment of AI/ML-based solutions in a wide range of real-life applications such as AR/VR/XR, intelligent transportation, monitoring, and industry automation. Traditional AI/ML approaches involving a single machine that accesses all training data and performs

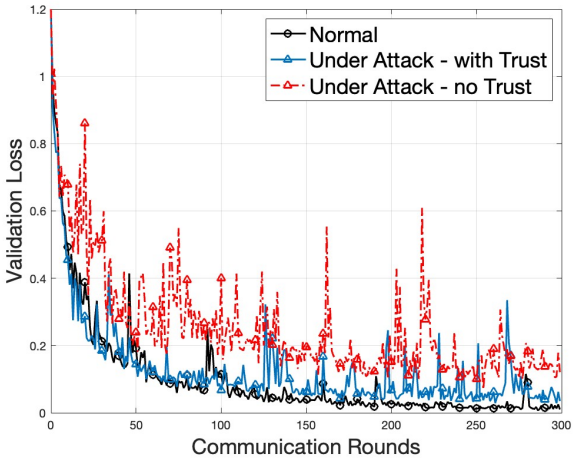

Fig. 2. Impact of trust on centralized FL

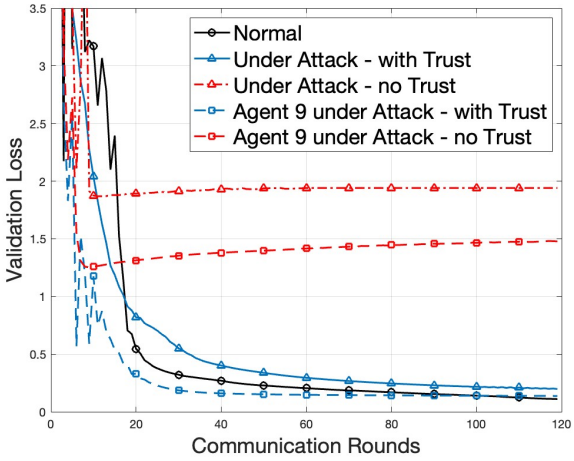

Fig. 3. Impact of trust on decentralized FL

model training in a central manner have been explored extensively in the literature. On the other hand, leveraging the computational power of multiple machines to accelerate training, handle large datasets, and address privacy has become a highly promising strategy. Nonetheless, ensuring security and privacy in the collaborative procedures of inference and training ML models across distributed devices or servers poses a complex challenge.

Federated learning (FL) is a recent learning paradigm that enables training a common model across multiple machines (agents), each performing the training process locally followed by an aggregation phase that combines the learned model parameters [11]. The model aggregation can be orchestrated at a central server (centralized FL) or performed in a decentralized manner (decentralized FL), e.g. through a consensus mechanism among participating machines to estimate the global model. The main steps of a FL algorithm are as follows:

- Local training over local data in the participating agents to obtain model updates (gradients),
- Sharing model updates with the central server in centralized FL or with the neighboring agents in decentralized FL,
- Model aggregation at the central server followed by broadcasting the updated global model in centralized FL, or model aggregation at agents using consensus method given the received messages from their neighbors in decentralized FL.

The above steps are repeated for a specified number of communication rounds, until convergence, or until the desired accuracy level is reached. It is important to note that while each agent in a decentralized FL system acts as a model aggregator, only the central server in a centralized FL performs model aggregation.

By distributing the local model training across multiple agents and using additional privacy-preserving mechanisms such as differential privacy, the privacy issue of traditional ML methods is alleviated. However, this approach remains susceptible to different security risks including data and model poisoning attacks. In these attacks, malicious agent(s) attempt to poison the global model parameters by sharing the poisoned local model with the central server in centralized FL or with the neighbor agents in decentralized FL. Trust-aware aggregation is a promising strategy to tackle the poisoning attacks in FL [12]. In this method, the received models of different agents are aggregated according to their trustworthiness. Particularly, at each round of the algorithm, the aggregator agent(s) uses the current message (model updates) received from the participating agents and the previous trust values to compute an updated trust vector for the agents following the steps explained in Section III. The new trust vector is incorporated in the aggregation process such that the messages received from agents with higher trustworthiness levels contribute more to the aggregated model.

We present the effectiveness of trust-aware mechanisms for both centralized and decentralized FL algorithms that utilize the centralized and decentralized trust aggregation frameworks respectively. Figure 2 illustrates the validation loss of the centralized FL setting in normal (unattacked) operation, under attack conditions without trust, and under attack conditions with a trust mechanism. We use the federated averaging (FedAvg) algorithm, a leading algorithm in centralized FL. The experiment involves training a convolutional neural network (CNN) across 100 agents using the MNIST dataset. At each round of communications, we assume that 20 agents are participating in the training. We also assume that in the system under attack, 20% of all agents are compromised and attempt to poison the learned model. We observe that while the global model of the attacked system without trust results in a learned model with degraded performance (increased validation loss), the trust-aware centralized FL algorithm closes the validation loss gap to the ideal unattacked system.

Moreover, we show the impact of trust on the decentralized FL mechanism. For this experiment, we use the trust-aware decentralized FL setting proposed in [12]. The

deployment setting is adopted from [13] that involves an actual CPS, where a number of radar devices employ Frequency Modulated Continuous Wave (FMCW) technology to survey their environment. This technology is used to monitor a collaborative workspace where both humans and robots operate together. The primary purpose of this system is to identify and trace the location of human operators as they move around a fenceless area in close proximity to a robotic manipulator, ensuring a safe separation between humans and robots. Figure 3 shows the validation loss averaged over all agents and the validation loss of a single agent for 120 communication rounds. We observe that the performance of the learned model by the trust-based decentralized FL converges to the normal unattacked system. This is in contrast to the attacked system without a trust mechanism which results in significantly higher validation loss compared to the trust-based and normal settings. Similarly, compared to the attacked system without trust, the performance of an example agent (agent 9) is enhanced significantly in the trust-based decentralized FL.

V. Use Case II: Trusted Network Embedding

The ability to virtualize the functionalities required by network services has revolutionized the field of computer networks. Network function virtualization (NFV) allows for decoupling the network functions from the traditionally complex dedicated hardware and realizing them through software. These virtualized network functions (VNFs) can be implemented on cheap off-the-shelf hardware instead of expensive proprietary servers. This not only reduces the cost of operation and shortens the time-to-market cycle for network providers but also allows for allocating the required infrastructure resources on demand at a much larger scale.

Each network service is a chain of network functions (e.g. routing, load balancing, firewall, intrusion detection, etc.) that are stitched together with logical links that determine the sequence in which the network packets need to be processed. Once virtualized, the network functions will have to be hosted by the servers of an NFV infrastructure (NFVI) that is controlled by a network controller and the traffic flowing between the VNFs will be mapped onto the infrastructure paths between the NFVI physical servers. This problem is termed service chain embedding [14] and is an instance of the more general virtual network embedding problem. To ensure the secure delivery of the network service, it is of paramount importance that the constituting VNFs are hosted by network servers that are trustworthy enough. More precisely, the notions of security, availability, and reliability are integrated into a single composite metric termed trust. In particular, each VNF requires a certain level of trust that is enforced by the network service template [15] and is incorporated into the service chain embedding problem formulation as a hard constraint.

Within the trust aggregation framework, each NFVI physical server is a network agent that communicates the evidence it has obtained to the central entity i.e. the central network controller (or the orchestrator). The controller computes the

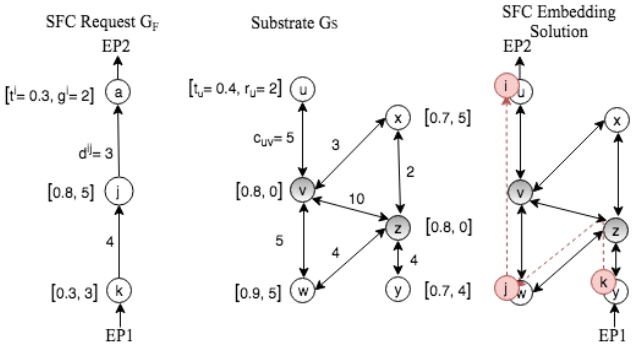

Fig. 4. Example of trust-aware network service placement

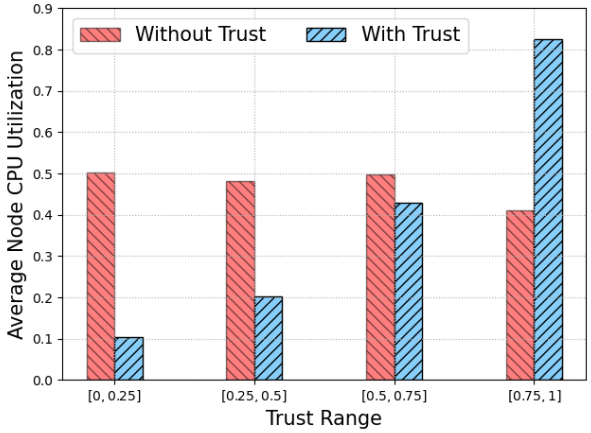

Fig. 5. Server utilization rate versus trust level

trust values according to the mathematical framework carried out in section III and updates the global trustworthiness of the physical servers. Each time the service chain embedding problem is solved, the new trust estimates of the physical servers are used in the corresponding constraints of the model. Fig. 5 shows how the trust-aware solution correctly allocates more load to more trusted servers when compared with the plain model. For a detailed analysis of the role of trust in SFC embedding the interested reader is referred to [15], and [14].

VI. Discussion & Challenges

Several challenges arise in practice when designing a trust aggregation framework. We enumerate the existing obstacles in the way of incorporating trust into the decision-making process for the next-generation networked systems and then propose our ideas for tackling these challenges.

A. Private Communication-efficient Trust Aggregation

In next-generation NetCPS, the information is crowd-sourced by the CPS devices. Further, we justified the importance of the existence of a framework for inferring and aggregating the trustworthiness of the networked agents and described two such *centralized* and *decentralized* structures. However, there are two major drawbacks to these schemes

concerning their communication efficiency and privacy. In the centralized scheme, the agents have to share their data with the central entity which is in charge of making the trust assignment decisions. However, on one hand, as the number of CPS devices increases and the data grows in size it will soon become bandwidth-inefficient to transmit the data to the central entity (especially, in proactive settings). On the other hand, it very well may be the case that the data gathered by or generated by a CPS agent contains sensitive or private information that the agent is not willing to share with another party. Especially, in wireless networks where it might take several hops for each agent to reach the central server; This not only increases the chance of privacy violation but also may increase the delay of communication between the agent and the central server. For the decentralized architecture, these issues are not resolved completely. Under these circumstances, the parties will have to communicate with one another and exchange their views regarding the trustworthiness of their fellow peers, rather than directly communicating with a centralized server. Although this may have the advantage of reducing the communication delays and lead to partial bandwidth efficiency, but may increase the net volume of information communication required to reach the same level of accuracy as in the centralized scheme. Moreover, the fact that the decentralized approach will have the devices share their opinions on the trustworthiness of other fellows, may make the privacy issue even worse. Additionally, as discussed in section I, due to potential competition between the networked agents, the parties might not be willing at all to respond to any queries by their fellow peers.

B. Trust Update Freshness

Trust is a metric that may evolve dynamically. Therefore, the resolution of updating the trust estimates in the network has to be adjusted carefully and tailored to the frequency of the trust-based decision-making; i.e. the most up-to-date trust estimates need to be available whenever required by the corresponding entities. In other words, it is of paramount importance that the trust information at the evaluating entities is fresh and an indicator of the current state of the network. We note that this argument is different from minimizing the latency of the trust estimates in the network and captures a separate requirement. *Age of information (AoI)*, is a recently-introduced metric that aims at maximizing the freshness of data in multi-server systems and has received a lot of attention in academia, due to its effectiveness in guaranteeing the freshness of status updates. We believe the path to an effective solution to the last challenge, crosses the AoI metric.

C. Quantifiable Trust

Another bottleneck in designing trust-aware mechanisms is the restrictiveness of quantifiable trust. Although there exists a large number of works in the literature concerning the notion of trust with various approaches, most of them maintain the qualitative perspective, so there are few works that study the quantification of trust. Among those few, most of them take the quantitative values for trust for granted without providing a detailed analysis of how to obtain such values. We believe this shortcoming stems from the abstractness of the notion of trust, the complexity of trust evaluation, and its application-specificity. Trust may be a composite of several metrics such as reliability, availability, resilience, adaptability, reputation, etc., and taking into account all these factors for all functions performed within the network will add to the complexity of trust evaluation. Moreover, it is impossible to define trustworthiness against all types of intrusions and security threats. Therefore, trust has to be viewed at the system level rather than a specific low-level preventive measure against specific attacks.

REFERENCES

[1] Y. Wang, "Trust quantification for networked cyber-physical systems," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 2055–2070, 2018.

[2] G. Theodorakopoulos and J. S. Baras, "On trust models and trust evaluation metrics for ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 318–328, 2006.

[3] X. Liu and J. S. Baras, "Using trust in distributed consensus with adversaries in sensor and other networks," in *17th International Conference on Information Fusion (FUSION)*, 2014, pp. 1–7.

[4] M. H. Junejo, A. A.-H. A. Rahman, R. A. Shaikh, K. M. Yusof, I. Memon, H. Fazal, and D. Kumar, "A privacy-preserving attack-resistant trust model for internet of vehicles ad hoc networks," *Scientific Programming*, vol. 2020, p. 8831611, 2020. [Online]. Available: https://doi.org/10.1155/2020/8831611

[5] W. Alnumay, U. Ghosh, and P. Chatterjee, "A trust-based predictive model for mobile ad hoc network in internet of things," *Sensors*, vol. 19, no. 6, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/6/1467

[6] N. Fan and C. Q. Wu, "On trust models for communication security in vehicular ad-hoc networks," *Ad Hoc Networks*, vol. 90, p. 101740, 2019, recent advances on security and privacy in Intelligent Transportation Systems. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S157087051830581X

[7] V. Kanchana Devi, "A comprehensive review on reputation-based trust management framework for cyber physical systems," in *International Virtual Conference on Industry 4.0*, R. J. Kannan, S. Geetha, S. Sashikumar, and C. Diver, Eds. Singapore: Springer Singapore, 2021, pp. 277–289.

[8] J. M. J. Valero, P. M. S. Sánchez, M. G. Pérez, A. H. Celdrán, and G. M. Pérez, "A review of "toward pre-standardization of reputation-based trust models beyond 5g"," *Actas de las VIII Jornadas Nacionales de Investigación en Ciberseguridad: Vigo, 21 a 23 de junio de 2023*, pp. 503–504, 2023.

[9] R. Ureña, F. Chiclana, and E. Herrera-Viedma, "Decitrustnet: A graph based trust and reputation framework for social networks," *Information Fusion*, vol. 61, pp. 101–112, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253519308097

[10] J. Qi, N. Zheng, M. Xu, X. Wang, and Y. Chen, "A multi-dimensional trust model for misbehavior detection in vehicular ad hoc networks," *Journal of Information Security and Applications*, vol. 76, p. 103528, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214212623001126

[11] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 1273–1282.

[12] A. Gholami, N. Torkzaban, and J. S. Baras, "Trusted decentralized federated learning," in *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*, 2022, pp. 1–6.

[13] S. Savazzi, M. Nicoli, and V. Rampa, "Federated learning with cooperating devices: A consensus approach for massive IoT networks," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4641–4654, 2020.

[14] N. Torkzaban and J. S. Baras, "Trust-aware service function chain embedding: A path-based approach," in *2020 IEEE Conference on NFV and SDN (NFV-SDN)*, 2020, pp. 31–36.

[15] N. Torkzaban, C. Papagianni, and J. S. Baras, "Trust-aware service chain embedding," in *2019 Sixth International Conference on Software Defined Systems (SDS)*, 2019, pp. 242–247.