T-Safe: Trustworthy Service Provisioning for IoT-Based Intelligent Transport Systems

Prajnamaya Dass, *Student Member, IEEE,* Sudip Misra, *Senior Member, IEEE,* and Chandana Roy, *Student Member, IEEE*

***Abstract*—In this paper, we propose a trustworthy service provisioning scheme for Safety-as-a-Service (Safe-aaS) infrastructure in IoT-based intelligent transport systems. Typically, a Safe-aaS infrastructure provides customized safety-related decisions dynamically to multiple end-users using the concept of decision virtualization. We consider road transportation as the application environment of Safe-aaS to generate trustworthy decisions. On the other hand, the efficiency and accuracy of the decisions generated depend on the security, privacy, and trustworthiness of the participating sensor nodes and the route through which data transit. We propose a trust evaluation model to compute the trustworthiness of the data generated from these nodes. Further, we consider direct and indirect trust mechanisms for each of the sensor nodes and update their trust measures at regular intervals of time. Based on these measures, we evaluate the trust of each data item sourced from the network. We formulate an integer linear programming (ILP) model to select the optimal data for decision-making, while alleviating the effects of illegitimate sensor nodes. Further, we show that the formulated ILP problem is NP-hard and use a dynamic programming approach to solve the problem. Experimental results show that our proposed trust evaluation model exhibits more than 8% attack detection rate and 13% reduction in false attack detection rate in a network with 50% malicious nodes, compared to the benchmark schemes. The proposed trustworthy data selection algorithm outperforms different existing greedy approaches.**

***Index Terms*—Safety-as-a-Service, Safety services, Trust, Sensor network, IoT, Industrial IoT, Industrial safety, Road transportation**

I. Introduction

INTERNET of Things (IoT)-based solutions can provide improved safety to individuals and machines across various industries by integrating different advanced technologies [1], [2]. These solutions can assist in providing proper information to the customers in a timely manner by integrating edge computing technologies [3]–[5]. The Safe-aaS platform provides safety-related customized decisions as services to the end-users [6], [7]. Considering the road industry as one of the application scenarios, we propose a trustworthy safety service provisioning scheme for the Safe-aaS infrastructure. Efficient and error-free decision making depends on the security, privacy, and trustworthiness of the participating sensor nodes deployed for collecting and forwarding the sensed data [8], [9]. Due to the sensitivity of the road safety-related decisions, any incorrect or unauthentic data used for decision generation may lead to compromised road safety situations.

In this paper, we propose a trustworthy service provisioning scheme, T-Safe, based on the trustworthiness of the sensed data. In order to evaluate the trustworthiness of the sensed data, we first propose a trust evaluation model and then design a trust update method for the sensor nodes. Based on the role and network activity, we consider different trust parameters of the sensor nodes for trust evaluation. Thereafter, depending on the trust values of the nodes, we assess the trust for every unit of data collected from the nodes by considering the entire data path. Additionally, due to the high data analytics cost charged by the service providers, selecting only meaningful and trustful data for decision making and other processing is very crucial [10]–[12]. We apply a dynamic programming approach to solve this problem and achieve the trust level requirement with reliable and minimum volume of data.

A. Motivation

With the adoption of IoT in the transportation industry, vehicle riders have access to different road safety-related information. However, there always exist concerns related to the privacy, security, and trustworthiness of the information. Safe-aaS provides safety-related customized decisions for the end-users, as per their requests. As the trust of the data increases, the trustworthiness of the safety decisions improve. Therefore, the risk of the influence of attacking agents is minimized. Different existing trust management schemes consider different trust models to evaluate the trust parameters of each sensor node uniformly, irrespective of their role and participation. However, in various industrial applications such as the road transportation industry, the sensed data is transmitted to the cloud through multi-hop paths. Further, data transmission paths vary with time due to different aspects such as the presence of neighbor nodes, their trust values, and distance between the nodes. Therefore, trust evaluation of the sensed data collected typically at cloud is necessary. Additionally, various characteristics of the sensor nodes and network parameters of the other nodes, which take part in the decision generation process are required to be considered. Moreover, the trust values of the sensor nodes need to be updated in a timely manner, such that their performances and behaviors are reflected. The sensed data with their updated trust scores help in trustworthy data selection. These trust scores help in discarding the malicious and untrustful data and reducing the data processing costs. On the other hand, trustworthy data selection helps to maximize the effectiveness of the decisions, while considering the limitations

P. Dass and S. Misra are with the Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur, India. (e-mail: {prajnamaya, sudipm}@iitkgp.ac.in)

C. Roy is with the Department of Industrial and Systems Engineering, Indian Institute of Technology Kharagpur, India. (e-mail: chandanaroy@iitkgp.ac.in)

on trust requirement and data analytics cost, depending on the data volume. Considering the above mentioned trust-related constraints for decision making, we propose a trust model to assess the trustworthiness of each node. Unlike the existing trust-based approaches, we categorize the nodes as data source nodes and data forwarding nodes.

B. Contribution

In this work, we consider different aspects of the sensed data for Safe-aaS infrastructure, starting from the data source to the server, with an aim to provide trustworthy services for the decision-generation process. In particular, for road safety scenarios, where the cloud, typically, performs various data analytics and generate safety-related decisions, the proposed solution accounts for the following: a) non-identical trust evaluation for data source node and data forwarding node, b) window-based trust update method, c) assessment of data trust, and d) optimal selection of trustworthy data for decision making. In summary, the specific *contributions* of this paper are as follows:

- We propose a trust evaluation model to assess the trustworthiness of the participating nodes. We consider both the intrinsic characteristics and the roles of the nodes in making the data available at the cloud.
- We evaluate the direct and indirect trust values of the sensor nodes in each time slot, based on their participation in data forwarding and performances. Further, considering the window period, we propose a sliding window-based trust update method and update the trust values of the nodes.
- We assess the trustworthiness of data coming from each data source node based on the entire path the data transited through. We formulate the trustworthy data selection problem as an ILP problem and since the ILP is NP-hard to solve, we present a dynamic programming algorithm to solve the problem efficiently. Further, we evaluate the proposed trust model and optimal data selection algorithms.
- Extensive experimental results show that in a system with 50% malicious nodes, our proposed model, T-Safe, achieves more than 8% malicious node detection rate and 13% reduction in false detection, compared to the state-of-the-art. The data selection algorithm also achieves improved results when compared to different existing greedy approaches.

The remainder of this paper is organized as follows. In Section II, we discuss and analyze the related state-of-the-art. In Section III, we present the system architecture followed by the evaluation and trust update mechanism in Section IV. Section V solves the problem of trustworthy data processing with limited resources using the dynamic programming heuristic approach. In Section VI, we present the performance evaluation, and finally, conclude the paper in Section VII.

II. Related Work

In this Section, we discuss the existing researches done for the evaluation of trust and risk of the network. Trust is modeled as one of the important parameters in the domain of network security. Theodorakopoulos and Baras [13] evaluated the trust evidence in ad-hoc networks. Further, the authors modeled the trust evaluation process as a directed graph, where the nodes represent the entities and the edges as the trust relations. Additionally, they proved their proposed scheme as robust in the presence of the attackers in the network. Similarly, Can and Bhargava [14] designed an algorithm to compute the trustworthiness in a peer-to-peer system, based on past interactions. Additionally, the authors showed that the proposed model was capable of reducing 16 different malicious attacks. A machine learning-based trust evaluation model was proposed by Han *et al*. [15] for underwater acoustic sensor networks. The authors used the K-means algorithm and the support vector machine-based supervised learning for attack classification.

In a multi-cloud environment, various issues such as resource scaling, security, and service complexity may arise due to the centralized architecture. Wahab *et al*. [16] proposed a three-fold solution to the centralized cloud platform – trust-establishment architecture, bootstrapping method, and a hedonic coalition game, to provide a distributed form of trustworthy multi-cloud community. The authors performed experiments with real-life data sets and showed that the number of malicious services is minimized. Another game theoretic approach was proposed by Rani *et al.* [17] to detect malicious activities in an IoT environment. In the proposed scheme, the authors used a cluster formation game to promote the sensor nodes to participate in the game, and then, based on the minimum set of recommendations, maintain the trust value inside the clusters. Subsequently, for anomaly detection, the activity-based trust dilemma game is used.

On the other hand, wireless multimedia sensor network (WMSN) comprise of camera and scalar sensors, which may be prone to security issues. Considering these issues and topology management in WMSNs, Mali and Misra [18] designed a trust-based topology management scheme. They utilized the signal strength of the control packets to set up the distributed topology. In the recent work proposed by Wang *et al.* [19], a trust model for trustworthy data collection for mobile fog computing was proposed. In this work, the evaluated trust values of the nodes are used by the mobile fog nodes to selectively consider the data from the cluster heads. In an IoT environment, a massive volume of sensitive data is accessible in the ecosystem. Recently, Jayasinghe *et al*. [20] proposed an intelligent trust computational model by introducing the concept of trust for humans and services to avoid uncertainty and risks. Further, the authors computed the individual trust attributes and proposed a machine learning-based algorithm to categorize the extracted trust features and merge them to generate the final decision. In another recent work, Razaque *et al*. [21] proposed a trust-based model for risk management in the cloud using fuzzy mathematics and gray relational theory.

Synthesis: From the detailed analysis of the state-of-the-art, we find that a research lacuna exists on the trustworthiness of the data that undergo data analytics and decision-making processes. The existing trust evaluation approaches only consider the node trust, irrespective of the data either

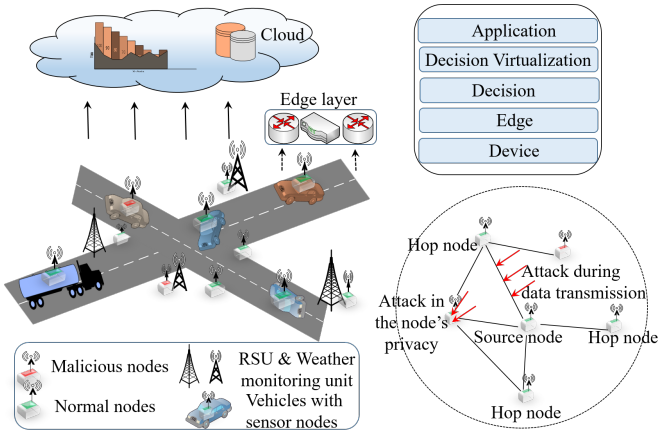


Figure 1: T-Safe: The system architecture

sensed or forwarded by them for further analysis at the cloud. However, in real scenarios, such as the road transportation industry, where sensitive safety decisions fully depend on the data collected, both node and data trust assessment are required, while considering the intrinsic characteristics of the nodes, their functionalities, and behaviors in the network. Further, considering the high cost charged by the cloud service providers for IoT data analytics [10]–[12], there exists a requirement to consider minimum data with higher trust values for the decisions to be effective and reliable.

III. System Architecture

We consider an intelligent transport system, which provides safety services to the on-road vehicle riders and other users, as shown in Figure 1. In our proposed scheme, we adopt the Safety-as-a-Service architecture [6] throughout the process of data collection and decision generation. Safe-aaS follows the concept of service-oriented architecture in which the end-users request safety-related services. The four main actors of Safe-aaS are the end-users, the sensor owners, the vehicle owners, and the safety service provider. Typically, a Safe-aaS infrastructure comprises five distinct layers – device, edge, decision, decision virtualization, and application.

Heterogeneous types of static and mobile sensor nodes are either deployed at a particular geographical location or on the vehicles in the device layer. These nodes at the device layer sense the data, and then, based on the time-sensitiveness, transmit them to the edge layer or cloud. Sensor nodes embedded in the vehicles send the sensed values along with the timestamp and present coordinates. Further, the decisions are generated from the primarily processed data in the decision layer. The timely reports collected from the cloud make the decisions more efficient. The decision virtualization layer maps the decision parameters requested by the end-users and the decisions to be provided to them. Multiple end-users receive the same decision simultaneously using the concept of decision virtualization. On the other hand, the application layer acts as the junction for communication among the end-users and the Safe-aaS architecture. The end-users register, select specific decision parameters, and pay rent through the web portal. Further, the end-users remain completely agnostic of the back-end process of decision generation.

Typically, in a practical scenario, the data sensed by the sensor nodes are either directly transmitted to the cloud or forwarded through the relay nodes, as shown in Figure 1. Further, the data sensed by the nodes are vulnerable to attacks and are prone to security threats. The integrity of this data may be reformed or modified by any attacker during the process of data transmission. In such a situation, the decisions generated from these sensor data may be misleading in nature. Therefore, we design a trustworthy decision generation scheme, while considering the trust of the data collected from the sensor nodes.

A. Mathematical Formulation

In the Safe-aaS architecture, heterogeneous sensor nodes are present in the device layer. We consider $\mathbb{S} = \{S_1, S_2, \cdots, S_n\}$ as the set of sensor nodes deployed at different geographical locations or vehicles. We define the tuple of time epochs $\{\varrho, \gamma\}$, where ϱ represents the data collection epoch, during which every node sends the sensed values to the cloud, and γ represents the decision epoch consisting of K data collection epochs. During data transmission, some sensor nodes act as data source nodes and the other intermediate nodes act as the nodes. Suppose, $\mathcal{S}$ and R represent the set of source nodes and relay nodes, respectively. Therefore, $\{\mathcal{S}, R\} \in \mathbb{S}$. Further, we consider that each of the sensor nodes possesses two trust values – *direct* and *indirect* trusts. Based on the role involved during data transmission, the trust values are assigned to the nodes.

Definition 1 Direct trust (π_d^j): Direct trust of node j is the trustworthiness of data source node denoted as a four-tuple $\langle S, D, E, A\rangle$, which are security and privacy parameters, data trust, energy trust, and data acceptance rate.

Definition 2 Indirect trust (π_i^j): Indirect trust of node j is the trustworthiness in data forwarding denoted as a triplet $\langle S, R, E\rangle$, which are security and privacy parameters, referral trust evaluated by neighbors, and energy trust.

We consider both the direct (π_d^j) and indirect trust (π_i^j) values within $[0, 1]$. For each decision epoch, these values are evaluated and updated for all the nodes. During a data collection epoch ϱ, the data sent by each node j is assessed and assigned a data trust value D_ϱ^j. After completing K such data epochs, the total data trust value for the entire K data collection epochs is computed. Considering the minimum data trust requirement D for reliable decision-making, we choose only the trustworthy data with minimum data volume for further processing. For the trust evaluation of sensor-enabled vehicles, we consider the vehicles which remain active for at least one decision epoch.

IV. T-Safe: Trust Evaluation Model

A. Trust Parameters

1) Security and Privacy: Security and privacy play a significant role in an industrial IoT environment, where starting from the data capturing end devices to the decision layer, everywhere there exists the possibility of threats and attacks.

In this scenario, we consider – confidentiality, integrity, and availability – known as the triad of a security architecture [22].

i) Confidentiality: We categorize the confidentiality requirements in a Safe-aaS architecture into *confidentiality at the endpoints* (C_e) and *confidentiality of communication* (C_m). The first parameter considers how the data collecting sensor nodes store the keys and other confidential data information in the device itself – termed as confidentiality of *data-at-rest*. The second parameter deals with the data encryption method to secure *data-in-motion*. Different end devices have different confidentiality mechanisms with varying security strength levels. Confidentiality, considering encrypted data storage and encrypted communication technique, is calculated as $S_C = C_e + C_m$, where C_e and C_m represent the level of the confidentiality mechanism used.

ii) Integrity: We consider *endpoint integrity* (i.e., I_e), which ensures the authentication of endpoints through mechanisms such as certificate-based authentication, password and key-based systems. On the other hand, *Integrity of communication* (i.e., I_m) reports the data modification during transit through mechanisms such as MAC and digital signatures. So, the integrity is expressed as $S_I = I_e + I_m$, where I_e and I_m represent the level of integrity mechanism used.

iii) Availability: Availability is computed based on the security and privacy primitives realized during data collection by the server against the service level agreement (SLA) and privacy level agreement (PLA) of the device. If, during a time period T, the successful number of times SLA and PLA followed is S_L, and the corresponding failure number is F_L, then the availability trust can be computed as $S_A = \frac{S_L}{S_L + F_L}$.

During a decision epoch, if a sensor node actively sends data for t number of times, then the security and privacy trust of the sensor node is expressed as $T_S = (S_C + S_I)\, t + S_A$.

2) Data Quality Trust: The sensed values from a node may be modified accidentally or intentionally over the communication channel or by any attacker. Additionally, due to sensor failures and the use of low-quality sensors, the collected data may not be trustworthy enough for real-time applications. However, this is the sole responsibility of the source node to send the correct values to the server. In a particular geographic location, for a certain period, data coming from all the sensor nodes for a common event possess temporal and spatial correlations. Data values related to the same event, in general, follows normal distribution [23]–[25]. For simplicity, we use the normal distribution to evaluate the data trust based on the similarities among the data values. The mean value of a set of sensor data is supposed to be the most trusted in this scenario. Therefore, we adopt the data trust mechanism, which is proposed by Han *et al.* [23]. To eliminate the dishonest data, first, the outliers are discarded using the median absolute deviation and then the mean value is computed. Finally, the data trust of each node is evaluated as [23]:

$$T_Q = 1 - 2\int_{\mu}^{v} f(x)dx = 2\int_{v}^{\infty} f(x)dx \tag{1}$$

where, $f(x)$ is the probability density function of the data item having the numerical value of v, represented as $f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$. Here, μ and σ represent the mean and standard deviation, respectively.

3) Energy Trust: After each decision epoch, all the sensor nodes inform their available residual energy to the server. A sensor node having a reputed trust value with no energy cannot involve itself for the sensing and transmission of data. However, a sensor node can pretend to have more energy to maintain its reputation. At time t, suppose a node reports its residual energy as E_t and after the time slot from t to $t+1$, it reports the value as E_{t+1}. The server has all the statistics for the node in terms of workload (L) in that duration and maximum energy consumption for each load (e). Besides, the server knows the charging behavior of each node and knows within that time, how much minimum energy (E^c) a node can absorb. If the energy value E_{t+1} is less than the minimum required energy for a node to operate smoothly in the next epoch, the energy trust is evaluated to zero. Otherwise, the truthfulness on energy (φ) is calculated as:

$$\varphi = \frac{E_{t+1}}{E_t - (L \times e) + E^c} \tag{2}$$

Considering the threshold for the truthfulness deviation E_Δ, the energy trust for a node is computed using Equation 3.

$$T_E = \begin{cases} \varphi, & \text{if } 0 \le \varphi \le 1 \text{ and } (1-\varphi) \le E_\Delta \\ 1-(\varphi-1), & \text{if } \varphi > 1 \text{ and } (\varphi-1) \le E_\Delta \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

4) Data Acceptance Trust: To assess the success rate of a node in terms of its data considered for the decision making process, the server keeps track of the node history and maintains the data acceptance (S_a) and rejection (S_r) counts. From these parameters, the trust on data acceptance can be expressed as $T_A = \frac{S_a}{S_a + S_r}$.

5) Referral Trust: Every sensor node in the network records the behaviors of its neighbor nodes. These parameters are the availability of nodes (i.e., *On* or *Off* based on the signal strength) and success and failure data delivery rates (S_j, F_j). After each decision epoch, a node sends the neighbor trust for all nodes placed one hop away. The success rate of node j considered for data transmission, calculated at each node z in the neighbor set Z is expressed as:

$$\chi(z,j) = \left(\frac{S_j}{S_j + F_j}\right) \Big/ \left(\frac{On_j}{On_j + Off_j}\right) \tag{4}$$

For other nodes except j, which are not considered for data transmission, are given the neighbor trust $\frac{On_j}{On_j + Off_j}$. The server receives neighbor trust for node j from all of its neighbor $\in Z$. Then, the referral trust for node j is computed based on the indirect trust of the neighbors as:

$$T_R = \frac{\sum\limits_{z \in Z} \pi_i^z \chi(z,j)}{\sum\limits_{z \in Z} \pi_i^z} \tag{5}$$

6) Involvement Trust: Each node in the network participates in forwarding the data one hop close to the server. The involvement trust specifies the number of times a node involved in data forwarding in an entire decision epoch, compared with the average involvement of the nodes in its neighborhood, which is expressed as:

$$T_I = \frac{\left(\sum_{\varrho=1}^{K} FC_{\varrho}^{j}\right)(|Z|+1)}{\sum_{z\in Z}\sum_{\varrho=1}^{K} FC_{\varrho}^{z} + FC_{\varrho}^{j}} \tag{6}$$

where, FC_{ϱ}^{j} represents the data forwarding count of node j for data epoch ϱ and the denominator values render the same for the neighbor nodes. For trust evaluation, nodes having $T_I \geq 1$ are considered with value 1.

B. Trust Evaluation

For trust evaluation of the nodes, we consider the intrinsic characteristics of the nodes and their timely behaviors. During network initialization, we consider the initial direct and indirect trust values as represented in Equation (7) and (8), respectively.

$$\pi_d^j = \eta_d(C_e + C_m + I_e + I_m) \tag{7}$$

$$\pi_i^j = \eta_i(C_e + I_e) \tag{8}$$

To normalize the trust values within $[0, 1]$, we use the system parameters η_d and η_i for direct and indirect trust values, respectively. The system parameter $\eta_d = \frac{1}{C_e^{max}+C_m^{max}+I_e^{max}+I_m^{max}}$, where, C_e^{max}, C_m^{max}, I_e^{max}, and I_m^{max} represent the maximum value of confidentiality and integrity trust values of a node, i.e., C_e, C_m, I_e, and I_m. Similarly, the mathematical expression of η_i is, $\eta_i = \frac{1}{C^{max}+I_e^{max}}$. After each decision epoch, both the direct and indirect trust values are evaluated, as depicted in Algorithm 1.

We use an array of linked lists, where all the trust values of a node are stored in the array and the neighbor node values are stored in their linked lists, respectively. In the trust evaluation algorithm, Steps 1-4 compute the data quality trust for each data source node. Since we evaluate the data quality trust based on the sensed values, we have considered the sensor nodes according to their types. In Step 6, the total data forwarding count for each node is computed. We find out the maximum data quality trust for each sensor node type in Step 8 and with reference to the maximum value, compute the data trust for each node in Step 10. Steps 11-17 compute the referral trust, involvement trust, and energy trust. Using the weighting factors, we compute direct and indirect trust in Step 20 and Step 22, respectively. Considering n number of nodes and decision epoch of size K, Steps 1-6 take $\mathcal{O}(nK)$ time. Steps 11-14 has a time complexity of $\mathcal{O}(Z)$. Due to the outer loop in Step 7 and Step 9, Steps 7-17 take $\mathcal{O}(nmZ) \leq \mathcal{O}(n^2)$. Both the loops starting at Step 18 and 21 take $\mathcal{O}(n)$ time in the worst case. Conclusively, the total algorithm complexity becomes $\mathcal{O}(nK + n^2 + n + n) \approx \mathcal{O}(n^2)$.

Algorithm 1 T-Safe Trust Evaluation

INPUTS: Trust values, neighbor referred trust values, and weighting factors.

OUTPUT: Direct trust T_d^j and Indirect trust T_i^j of each node j for a particular decision epoch

PROCEDURE:

1: **for** Each node type m **do**
2: **for** Each data epoch ϱ **do**
3: **for** Each node $j \in m$ **do**
4: Compute $T_Q^{j,\varrho} \leftarrow 2\int_v^{\infty} f(x)dx$
5: $T_Q^j \leftarrow T_Q^j + T_Q^{j,\varrho}$
6: $FC^j \leftarrow FC^j + FC_{\varrho}^j$
7: **for** Each node type m **do**
8: Compute $best \leftarrow \max_{j\in m}\left\{T_Q^j\right\}$
9: **for** Each node $j \in m$ **do**
10: Compute $T_Q^j \leftarrow \frac{T_Q^j}{best}$
11: **while** $z < |Z|$ **do**, z is from 1 to $|Z|$
12: $T_R^j \leftarrow T_R^j + T_i^z\chi(z, j)$
13: $T_i^Z \leftarrow T_i^Z + T_i^z$
14: $FC_Z \leftarrow FC_Z + FC_j$
15: $T_R^j \leftarrow T_R^j/T_i^Z$
16: $T_I^j \leftarrow FC^j(|Z|+1)/FC^Z + FC^j$
17: Compute T_E^j using the value of φ
18: **for** Each node j **do**
19: **if** $j \in \{\mathcal{S}\}$ **then** ▷ *$\mathcal{S}$ is the set of data source nodes*
20: $T_d^j \leftarrow (C_m^j + I_m^j)T_Q^j S_A^j w_q + T_E^j w_e + T_A^j w_a$
21: **if** $j \in \{R\}$ **then** ▷ *R is the set of relay nodes*
22: $T_i^j \leftarrow (C_e^j + I_e^j)T_I^j w_i + T_R^j w_r + T_E^j w_e$
23: **Return** T_d^j and T_i^j

C. Updating Trust Values

After each decision epoch, the trust values of the nodes must be updated to reflect their current network behaviors. Updating the trust values based on the current assessment and the last epoch trust value is not a good approach because a node might have performed very well in past epochs, except the last one. Hence, we use a weighted window-based approach, where all the previous trust values in that window are considered. Based on the trust values, we evaluate the cumulative change in the rate of trust towards the total sum trust value of the network. We consider the weighting parameter w_h and the length of the sliding window H, to update the trust values (π_d^j, π_i^j). Note that the the value of w_h decreases linearly towards the end of the window.

$$\pi_d^j = \frac{\sum_{h=1}^{H} \frac{T_d^j w_h}{\max_{j\in m}\{T_d^j\}}}{\sum_{h=1}^{H} w_h} \tag{9}$$

Similarly, the indirect trust value of a node j is updated. The updated trust values are then used for trustworthy data selection.

V. Trustworthy Data Selection

For each node $j \in S$, we have the direct trust value π_d^j and for each node $j \in R$, we have the indirect trust value π_i^j. At each data collection epoch, the data values from a node may travel different routes. Hence, based on the direct trust values of the data source nodes and the indirect trust values of the data forwarding nodes in the route, the data trust of node j for data epoch ϱ is computed as:

$$D_\varrho^j = \pi_d^j + min\left\{\pi_i^1, \pi_i^2, ..., \pi_i^p\right\} \tag{10}$$

where, $\{1, 2, ...p\}$ represent the set of all the nodes in the path through which data transited from node j to the server. During the process of data selection, the server may require data with trust values at least $\mathcal{T}$ and the data volume V should be as minimum as possible, considering the resource constraints. For each data epoch, the size of the data packet L_ϱ^j for each node j depend on the communication protocol, security mechanisms used, and the sensing parameters. The objective function for trustworthy data selection considering the data volume and the trust value is expressed as:

$$\min_{\beta_j} \sum_{j=1}^{n} \beta_j \sum_{\varrho=1}^{K} \alpha_\varrho^j L_\varrho^j \tag{11a}$$

$$\text{s.t.} \quad \sum_{j=1}^{n} \beta^j \sum_{\varrho=1}^{K} D_\varrho^j \geq \mathcal{T}, \tag{11b}$$

$$\forall j, \quad \beta_j, \alpha_\varrho^j \in \{0, 1\} \tag{11c}$$

where, β_j represents a binary decision whether to consider data of node j for decision making process and α_ϱ^j represents the data transmission flag of node j for the decision epoch ϱ.

If we reduce the problem (11a)-(11c) and find its dual then we get the 0-1 Knapsack problem, which is a well known combinatorial optimization problem proven to have NP-complete computational complexity [26]. It implies that the trustworthy data selection problem is an NP-complete problem. To solve this problem, we use a modified 0-1 Knapsack-based dynamic programming approach [27], which follows the tabulation method to solve the problem in pseudo-polynomial time. The tabulation method of dynamic programming follows a bottom-up approach to compute the optimal result. Starting from the initial cell of the table (a 2D array), in a row-major fashion, the algorithm compares the result in every step and fills each cell with optimal values (Step 11 in Algorithm 2). Finally, the last cell in the table gives the required optimal output of the algorithm. For applicability, we use absolute value of data trust, which do not have more effect on the results. In Algorithm 2, the initialization loops from Steps 1-4 take $\mathcal{O}(\mathcal{T} + n)$ time. In the bottom-up dynamic programming approach, we check for the required trust value $\mathcal{T}$ with minimum data volume possible, and hence, consider the values of each node. So, Steps 6-13 take $\mathcal{O}(\mathcal{T}n)$ time. In Steps 16-20, we find out the nodes giving the best solution with $\mathcal{O}(n)$ time. Considering the whole algorithm, the time complexity becomes $\mathcal{O}(\mathcal{T}n + \mathcal{T}n + n) \approx \mathcal{O}(\mathcal{T}n)$.

Algorithm 2 T-Safe Data Selection

INPUTS: $\mathcal{T}, n, \vartheta\,[1, ..., n]\,, D\,[1, ..., n]$
OUTPUT: V and $Selected\,[1, ..., n]$
PROCEDURE:

1: **for** $j \leftarrow 0$ to $\mathcal{T}$ **do**
2: $\quad Vol\,[0, j] \leftarrow \infty$
3: **for** $i \leftarrow 1$ to n **do**
4: $\quad Vol\,[i, 0] \leftarrow \infty$
5: $Net_\mathcal{T} \leftarrow 0$
6: **for** $i \leftarrow 1$ to n **do**
7: $\quad Net_\mathcal{T} \leftarrow min(Net_\mathcal{T} + D[i], \mathcal{T})$
8: $\quad$ **if** $Net_\mathcal{T} < \mathcal{T}$ **then**
9: $\quad\quad Vol[i, Net_\mathcal{T}, ..., \mathcal{T}] \leftarrow \infty$
10: $\quad$ **for** $j \leftarrow Net_\mathcal{T}$ **down** to 1 **do**
11: $\quad\quad$ **if** $Vol[i-1, j - abs(D[i])] + \vartheta[i] < Vol[i-1, j]$ **then**
12: $\quad\quad\quad Vol[i, j] \leftarrow Vol[i-1, j - abs(D[i])] + \vartheta[i]$
13: $\quad\quad\quad Dec[i, j] \leftarrow 1$
14: $V \leftarrow Vol[n, \mathcal{T}]$
15: $Net_\mathcal{T} \leftarrow \mathcal{T}$
16: **for** $i \leftarrow n$ **down** to 1 **do**
17: $\quad Selected[i] \leftarrow Dec[i, Net_\mathcal{T}]$
18: $\quad$ **if** $Dec[i, Net_\mathcal{T}] == 1$ **then**
19: $\quad\quad Net_\mathcal{T} \leftarrow Net_\mathcal{T} - D[i]$
20: **Return** V and $Selected[\,]$

Table I: Simulation Parameters

Number of nodes	500
Simulation area	$1000m \times 1000m$
Transmission range of a node	$100m$
Transmission and reception power	$24.75, 13.5mW$ [28]
Data rate	$40kbps$ [28]
Initial energy	$1J$ [29]
Sensor types	4
1 decision epoch	24 data epochs

VI. Performance Evaluation

A. Simulation Settings

We assess the performance of our proposed trust model and data selection algorithm using MATLAB R2018a. We considered an area of $1000m \times 1000m$ with 500 randomly deployed nodes, of which 100 nodes are selected as the data source nodes. We considered four types of sensor nodes to evaluate the data trust values. The different parameters considered for our experiments are mentioned in Table I.

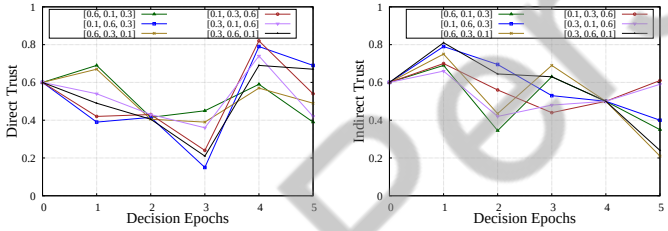
B. Attack Model

We consider three types of attacks, namely *data manipulation* (DM), *biased referral trust* (BRT), and *biased data forwarding* (BDF). Data source nodes perform DM attacks by sending incorrect sensed data values, which may affect the entire decision-making process. In BRT attack, the nodes refer biased trust values to their neighbors with an intention

to strengthen or weaken their indirect trust values. The data forwarding nodes or the relay nodes that choose selective nodes for data forwarding are named as BDF attackers. For each attack type, we define several rules and then use rule-based classification to detect the attacking nodes. Further, we create different attack vectors and verify the detection of the malicious nodes using the attack detection rules.

C. Results and Discussion

1) Effect of Weight Distribution on Trust Values: Trust evaluation results due to the use of different distributions of the weighting parameters are shown in Figure 2(a) and 2(b). For direct trust, we have three weighting parameters for data quality trust (w_q), energy trust (w_e), and data acceptance trust (w_a). Similarly, for indirect trust, along with the energy trust, we consider weights for data forwarding involvement (w_i), and referral trust (w_r). For an easy understanding of the figure, divide six lines into three groups, having one weighting parameter unchanged in each group. If we consider the weight distributions [0.6, 0.1, 0.3] and [0.1, 0.6, 0.3], where w_a remains unchanged in both, we can see the large gap between them in all the iterations, except the second iteration. This behavior shows that the data quality trust is very high compared to the energy trust, and the reverse situation occurs in the fourth iteration. In the second iteration of 2(a) and the fourth iteration of 2(b), it is worth mentioning that all the lines are very close due to the existence of almost close values of the trust parameters. Hence, there is no effect of the multiplication with the weight factors.

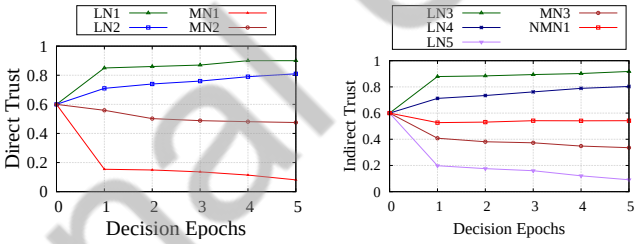


(a) Distribution on direct trust (b) Distribution on indirect trust

Figure 2: Effect of weight distribution on trust parameters

2) Trend of Trust Values in Legitimate and Malicious Nodes: Our proposed trust model considers both direct and indirect trust according to the node functionality. On the functionality basis, a data source node or a data forwarding node may act maliciously. In Figure 3, we present the trust value variation tendency of both legitimate nodes (LN) and malicious nodes (MN). In Figure 3(a), considering the weighting factors ($w_q = 0.6, w_e = 0.3, w_a = 0.1$) for direct trust, LN1 and LN2 have an increasing tendency in trust values. We can observe that LN2 has a slower increase compared to LN1, which is due to the lower data acceptance and energy trusts. On the contrary, the node MN1 has a sudden decreasing trend due to malicious behavior in data values. However, MN2 behaves maliciously with residual energy, and due to the low valued weighting factor for energy trust, the trend decreases slowly.

Similarly, in Figure 3(b), we consider the weighting parameters $w_i = 0.6$, $w_r = 0.3$, and $w_e = 0.1$ for involvement trust, referral trust, and energy trust, respectively. The legitimate nodes, LN3 and LN4, have growing indirect trust value. However, the reason for the gap between them is due to the lower referral trust of LN4. The node LN5 has low data forwarding involvement, and hence, the neighbor nodes recommend low referral trust. Low trust values for data forwarding involvement and referral trust decreases the indirect trust of the node in each decision epoch. In the plot for NMN1, we observe that the node has high forwarding involvement but due to the biased low trust recommendation by the neighbors, the indirect trust decreases. In this case, the neighbors are treated as malicious who are involved in the BRT attack. When a node is involved in biased data forwarding, low trust value is recommended by many of its neighbors and hence, incurs slow decrease in indirect trust like MN3.



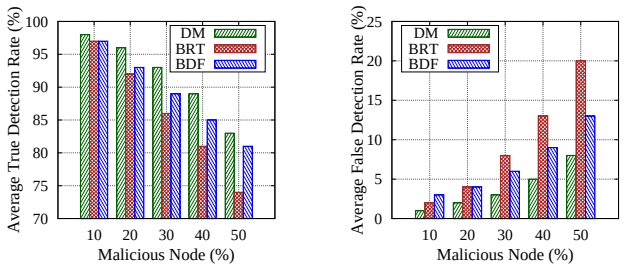
(a) Trend of direct trust value (b) Trend of indirect trust value

Figure 3: Trend of trust values in legitimate and malicious nodes

3) Malicious Node Detection Rate: To evaluate the system resistance against the malicious nodes, we inject different attack vectors to the system and verify the attack detection rate. In case of DM attacks, first, we adopt the median absolute deviation (MAD) rule to discard the outliers and then compute the mean value. Thereafter, we consider the nodes with $T_Q^j < 0.9$ as the attacker nodes, where T_Q^j represents the data quality trust of node j. From Figure 4(a), we observe that when the number of data manipulating nodes increases, the DM detection rate decreases. This is because increase in number of malicious nodes affects the data mean value, which eventually reduces the detection rate.

For BRT attack detection, first, we find out the nodes who refer biased trust values to their neighbors. Thereafter, to detect the attackers, we use the involvement trust and the referral trust value of the neighbor nodes. Considering a node j and its neighbor node z, to detect the malicious nodes, we take two cases – i) $\chi(z, j) = low$ when $T_R^j = high$ and $T_I^j = high$ and ii) $\chi(z, j) = high$ when $T_R^j = low$ and $T_I^j = low$. When most of the neighbor nodes recommend biased values for a node, the BRT detection rate decreases. In case of BDF attack detection, we classify the nodes, who are involved in $high$ data forwarding activity but possess low referral trust as the attackers. Here, when the trust value is less than 0.3, we label it as low and label it as $high$ when the trust value is greater than 0.7. However, our system fails to detect the nodes that have low forwarding involvement and high referral trust value. The attack detection thresholds are set while considering the

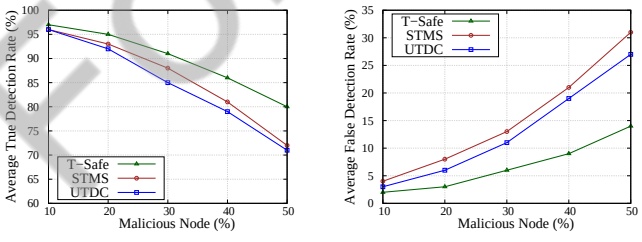
optimal rate of true positive and false negative attack detection. We evaluated for various threshold values like grid search tuning [30] and then, finalized the thresholds that yield optimal results.



(a) True malicious detection (b) False malicious detection

Figure 4: Malicious node detection

4) Attack Detection Rate: We compare the attack detection rate of T-Safe with two benchmark schemes – i) utility-based trustworthy data collection (UTDC) [19] and ii) a support vector machine-based trust model (STMS) [15]. Figure 5(a) and 5(b) illustrate the comparison of true attack detection rate and false attack detection rate between T-Safe, UTDC, and STMS. In Figure 5(a), we observe that UTDC has lowest attack detection rate. This is because UTDC fails to identify the BRT and BDF attacks due to the unavailability of the involvement trust parameter. However, in T-Safe, involvement trust and referral trust together help in achieving the highest detection rate. From Figure 5(b), we infer that the false attack detection rate is highest in STMS. This is due to the fact that the STMS approach uses the trust evidence collected from the sensor nodes and uses the K-means algorithm to classify the evidence into two classes, namely, *good* and *bad* nodes. Further, the classified data are used for training and fed to the SVM algorithm. However, it divides the sensor nodes with very close values into two classes, where legitimate nodes also fall into the attack class, which eventually increases the false detection rate. In T-Safe, first, we use the MAD rule to remove the outliers to reduce the affect of data manipulating nodes. Moreover, during referral trust computation, we use the indirect trust values of the nodes to reduce the bad influence from the malicious nodes, which helps in reducing the false detection rate of BDF and BRT attacks.



(a) True attack detection (b) False attack detection

Figure 5: Malicious node detection accuracy

5) Trust Value and Volume of Trustworthy Data: The server selects trustful data with threshold conditions on the data trust. For simplicity, we consider the number of data packets selected as the data volume. We compare the proposed scheme, T-Safe with first come first select (FCFS) approach based on node ID ordering, minimum data volume first (Greedy1), and maximum trust value first (Greedy 2), as shown in Figure 6. The Greedy 2 approach fails to give better result against our scheme as there may be some data with minimum volume remain unconsidered once the trust condition is satisfied. Increasing with the threshold value, all the schemes have closer outcomes because all the schemes traverse till end of the lists to satisfy the trust requirement.

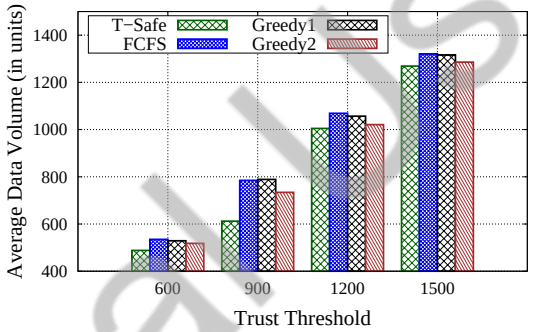


Figure 6: Data trust

VII. Conclusion

In this paper, we proposed a trustworthy decision generation scheme for intelligent transport system, where safety decisions need to be reliable and free from both internal and external attacking agents. We evaluated different parameters for trust assessment and assign both direct and indirect trusts for each node. Based on the trust values and data path, we evaluated the trust for each data sent by the nodes. Subsequently, we proposed a weighted window approach for trust updating after each decision epoch. Due to the huge data volume from the nodes, the server (or the cloud) has the option to select the minimum data volume satisfying the required trust threshold. Since the data selection with resource and trust level constraints is an NP-hard problem, we used a dynamic programming approach for optimal results. Experimental results showed that the proposed method can filter out malicious data to have trustworthy road safety decisions. The proposed scheme has better malicious node detection rate due to the individual assessment of direct and indirect trust parameters.

In this work, we considered trustworthy safety service provisioning, where all the operations are performed at the cloud server. However, for some real-time safety services which are handled by the edge devices, an instant trust assessment is required. Therefore, as an extension to this work, we plan to propose a delay-aware trust model for real-time safety service provisioning in intelligent transport systems.

References

[1] F. Fraile, T. Tagawa, R. Poler, and A. Ortiz, "Trustworthy industrial IoT gateways for interoperability platforms and ecosystems," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4506–4514, 2018.

[2] H. Xu, W. Yu, D. Griffith, and N. Golmie, "A survey on industrial internet of things: A cyber-physical systems perspective," *IEEE Access*, vol. 6, pp. 78 238–78 259, 2018.

[3] M. Aazam, S. Zeadally, and K. A. Harras, "Deploying fog computing in industrial internet of things and industry 4.0," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4674–4682, Oct. 2018.

[4] R. Oma, S. Nakamura, T. Enokido, and M. Takizawa, "An energy-efficient model of fog and device nodes in IoT," in *Proceedings of IEEE International Conference on Advanced Information Networking and Applications Workshops*, Cracow, Poland, 2018, pp. 301–306.

[5] W. Shi and S. Dustdar, "The promise of edge computing," *IEEE Computer*, vol. 49, no. 5, pp. 78–81, May 2016.

[6] C. Roy, A. Roy, S. Misra, and J. Maiti, "Safe-aaS: decision virtualization for effecting Safety-as-a-Service," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1690–1697, Jun. 2018.

[7] C. Roy, S. Misra, J. Maiti, and M. S. Obaidat, "DENSE: Dynamic edge node selection for Safety-as-a-Service," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[8] N. Mitton, S. Papavassiliou, A. Puliafito, and K. S. Trivedi, "Combining cloud and sensors in a smart city environment," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, pp. 1–10, Aug. 2012.

[9] F. H. Rahman, T.-W. Au, S. S. Newaz, W. S. Suhaili, and G. M. Lee, "Find my trustworthy fogs: A fuzzy-based trust evaluation framework," *Future Generation Computer Systems*, vol. 109, pp. 562–572, 2020.

[10] E. Dadashov, U. Cetintemel, and T. Kraska, "Putting analytics on the spot: or how to lower the cost for analytics," *IEEE Internet Computing*, vol. 18, no. 5, pp. 70–73, Oct. 2014.

[11] E. Siow, T. Tiropanis, and W. Hall, "Analytics for the internet of things: A survey," *ACM Computing Surveys*, vol. 51, no. 4, pp. 1–36, Jul. 2018.

[12] B. M. Balachandran and S. Prasad, "Challenges and benefits of deploying big data analytics in the cloud for business intelligence," *Procedia Computer Science*, vol. 112, pp. 1112–1122, 2017.

[13] 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, Feb. 2006.

[14] A. B. Can and B. Bhargava, "SORT: A self-organizing trust model for peer-to-peer systems," *IEEE Transactions on Dependable and Secure Computing*, vol. 10, no. 1, pp. 14–27, Jan. 2013.

[15] G. H. Y. He, J. Jiang, N. Wang, M. Guizani, and J. A. Ansere, "A synergetic trust model based on SVM in underwater acoustic sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 11, pp. 11 239–11 247, 2019.

[16] O. A. Wahab, J. Bentahar, H. Otrok, and A. Mourad, "Towards trustworthy multi-cloud services communities: A trust-based hedonic coalitional game," *IEEE Transactions on Services Computing*, vol. 11, no. 1, pp. 184–201, Jan. 2018.

[17] R. Rani, S. Kumar, and U. Dohare, "Trust evaluation for light weight security in sensor enabled internet of things: Game theory oriented approach," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8421–8432, Oct. 2019.

[18] G. Mali and S. Misra, "Trast: Trust-based distributed topology management for wireless multimedia sensor networks," *IEEE Transactions on Computers*, vol. 65, no. 6, pp. 1978–1991, Jun. 2016.

[19] T. Wang, L. Qiu, G. Xu, A. Kumar, and A. Liu, "Energy-efficient and trustworthy data collection protocol based on mobile fog computing in internet of things," *IEEE Transactions on Industrial Informatics*, vol. PP, pp. 1–1, 05 2019.

[20] U. Jayasinghe, G. M. Lee, T. Um, and Q. Shi, "Machine learning based trust computational model for iot services," *IEEE Transactions on Sustainable Computing*, vol. 4, no. 1, pp. 39–52, Jan. 2019.

[21] A. Razaque, M. Almiani, M. J. Khan, B. Magableh, A. Al-Dmour, and A. Al-Rahayfeh, "Fuzzy-gra trust model for cloud risk management," in *Proceedings of ACM International Conference on Software Defined Systems*, Rome, Italy, Jun. 2019, pp. 179–185.

[22] J. Andress, *The basics of information security: understanding the fundamentals of InfoSec in theory and practice*. Syngress, Elsevier, 2014.

[23] G. Han, J. Jiang, L. Shu, and M. Guizani, "An attack-resistant trust model based on multidimensional trust metrics in underwater acoustic sensor network," *IEEE Transactions on Mobile Computing*, vol. 14, no. 12, pp. 2447–2459, Dec. 2015.

[24] E. Elnahrawy and B. Nath, "Cleaning and querying noisy sensors," in *Proceedings of ACM International Conference on Wireless Sensor Networks and Applications*, New York, NY, USA, 2003, pp. 78–87.

[25] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *Proceedings of ACM International Symposium on Information Processing in Sensor Networks*, Berkeley, CA, USA, 2004, pp. 20–27.

[26] K. Bernhard and J. Vygen, "Combinatorial optimization: Theory and algorithms," *Springer, Third Edition*, 2008.

[27] H. Kellerer, U. Pferschy, and D. Pisinger, *Knapsack problems. 2004*. Springer, Berlin, 2003.

[28] F. Bouabdallah, N. Bouabdallah, and R. Boutaba, "On balancing energy consumption in wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 6, pp. 2909–2924, Jul. 2009.

[29] T. Ojha, S. Misra, N. S. Raghuwanshi, and H. Poddar, "DVSP: Dynamic virtual sensor provisioning in sensor-cloud based internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5265 – 5272, Jun. 2019.

[30] F. Hutter, L. Kotthoff, and J. Vanschoren, *Automated Machine Learning*. Springer, 2019.

Prajnamaya Dass (Student Member, IEEE) received his M.Tech from the Indian Institute of Technology (Indian School of Mines), Dhanbad, India, in 2016, and the B.Tech from Government College of Engineering, Kalahandi, Odisha, India, in 2013. Currently, he is working toward the Ph.D. degree in Department of Computer Science and Engineering at Indian Institute of Technology Kharagpur, India. His current research interests include security and trust management in Internet of Things (IoT).

Sudip Misra (Senior Member, IEEE) is a Professor and Abdul Kalam Technology Innovation National Fellow in the Department of Computer Science and Engineering at the Indian Institute of Technology Kharagpur. He received his Ph.D. degree in Computer Science from Carleton University, in Ottawa, Canada. His current research interests include Wireless Sensor Networks and Internet of Things. Professor Misra has published over 350 scholarly research papers and 12 books. He has won nine research paper awards in different conferences. He was awarded the IEEE ComSoc Asia Pacific Outstanding Young Researcher Award at IEEE GLOBECOM 2012, California, USA. He was also the recipient of several academic awards and fellowships such as the Faculty Excellence Award (IIT Kharagpur), Young Scientist Award (National Academy of Sciences, India), Young Systems Scientist Award (Systems Society of India), Young Engineers Award (Institution of Engineers, India), (Canadian) Governor General's Academic Gold Medal at Carleton University, the University Outstanding Graduate Student Award in the Doctoral level at Carleton University and the National Academy of Sciences, India – Swarna Jayanti Puraskar (Golden Jubilee Award), Samsung Innovation Awards-2014 at IIT Kharagpur, IETE-Biman Behari Sen Memorial Award-2014, and the Careers360 Outstanding Faculty Award in Computer Science for the year 2018 from the Honourable Minister for Human Resource Development (MHRD) of India. Thrice consecutively he was the recipient of the IEEE Systems Journal Best Paper Award in 2018, 2019, and 2020. He was awarded the Canadian Government's prestigious NSERC Post Doctoral Fellowship and the Alexander von Humboldt Research Fellowship in Germany. His team received the GYTI Award 2018 in the hands of the President of India for socially relevant innovations. Dr. Misra has been serving as the Associate Editor of different journals such as the IEEE Transactions on Mobile Computing, IEEE Transactions on Vehicular Technology, IEEE Transactions on Sustainable Computing, IEEE Network, and IEEE Systems Journal. He is the Fellow of the National Academy of Sciences (NASI), India, the Institution of Engineering and Technology (IET), UK, British Computer Society (BCS), UK, Royal Society of Public Health (RSPH), UK, and the Institution of Electronics and Telecommunications Engineering (IETE), India. Professor Misra is the distinguished lecturer of the IEEE Communications Society. He is the Director and Co-Founder of the IoT startup, SensorDrops Networks Private Limited (http://www.sensordropsnetworks.com). Further details about him are available at http://cse.iitkgp.ac.in/ smisra/.

Chandana Roy (Student Member, IEEE) is an Institute Scholar and is pursuing her Ph.D. from the Department of Industrial and Systems Engineering, Indian Institute of Technology Kharagpur, India. She received her M.Tech and B.Tech degree from National Institute of Technology Durgapur, India in 2012, and West Bengal University of Technology, India in 2010, respectively. The current research interests of Ms. Roy include Industrial Internet of Things, Wireless Body Area Networks, and Cloud Computing. She is a student member of the IEEE.