Computers, Materials & Continua
DOI: 10.32604/cmc.2023.031472
Article

TRUSED: A Trust-Based Security Evaluation Scheme for A Distributed Control System

Saqib Ali[1,*] and Raja Waseem Anwar[2]

[1]Sultan Qaboos University, Muscat, 123, Sultanate of OMAN
[2]Arab Open University, Muscat, 130, Sultanate of OMAN
*Corresponding Author: Saqib Ali. Email: saqib@squ.edu.om
Received: 18 April 2022; Accepted: 25 July 2022

Abstract: Distributed control systems (DCS) have revolutionized the communication process and attracted more interest due to their pervasive computing nature (cyber/physical), their monitoring capabilities and the benefits they offer. However, due to distributed communication, flexible network topologies and lack of central control, the traditional security strategies are inadequate for meeting the unique characteristics of DCS. Moreover, malicious and untrustworthy nodes pose a significant threat during the formation of a DCS network. Trust-based secure systems not only monitor and track the behavior of the nodes but also enhance the security by identifying and isolating the malicious node, which reduces the risk and increases network lifetime. In this research, we offer TRUSED, a trust-based security evaluation scheme that both, directly and indirectly, estimates each node's level of trustworthiness, incorporating the cumulative trust concept. In addition, simulation results show that the proposed technique can effectively identify malicious nodes, determine their node's trustworthiness rating, and improve the packet delivery ratio.

Keywords: Malicious; network lifetime; risk management; security; trust; untrustworthy

1 Introduction

Industrial control systems (ICS) are used for remotely managing industrial installations and facilities. Supervisory control and data acquisition (SCADA) systems, distributed control system configurations, and other smaller control systems are all included in the term ICS [1]. These systems are widely used for large, distributed industries such as oil and gas, transportation, manufacturing, and electric power generation facilities [2]. As these mission-critical operations require constant monitoring, distributed control systems provide an advanced mechanism for the remote command and control of industrial plants and processes.

Distributed Control System refers to the division of a major application into smaller submodules where each module carries certain specific application processes and allows communication between

these submodules. DCS is used for mission-critical industrial and manufacturing industries. It relies on decentralizing the control unit and establishing a shared network between the engineering stations. The DCS architecture consists of an engineering workstation, an operating station or Human Machine Interface (HMI), a process control unit or local control unit, intelligent devices, and a communication system make up. Sensors gather information and deliver it to input/output modules, where actuators help control process parameters [3]. The field control station receives input signals from the sensors. The system performs control calculations and outputs the signals to initiate compensatory actions. The field buses carry the results achieved upon processing and analyzing the input signals to the actuator devices. DCS distributes the control processing among the system's nodes, thus resulting in a reliable, fault-tolerant system.

While the DCS are reliable, scalable, and fault-tolerant, there is a rising concern regarding the security and accuracy of the data transmission. Due to the architecture's interconnectivity with the cyber world, it is prone to vulnerabilities, threats, and attacks. Some of the most common attacks on DCS-based environments include Distributed Denial of Service (DDOS), eavesdropping, man-in-middle, routing, malware, and Structured Query Language (SQL) injection attacks. The previous work discussed DCS technological environment followed by DCS environment-related threats and attacks [4]. This paper extends the last research work by further examining the risk management methodology by incorporating two additional components, namely, trust and reputation, where trust refers to the degree of confidence in the form of probability by which one entity will assume, or expect, the behavior of another entity [5–7], and reputation refers to a form of indirect trust where one entity relies on surrounding entities' observation and opinion about the target entity's behavior and reliability within a specified timeframe [8].

In this research we present a security evaluation scheme based on trust. The proposed TRUSED (Trust-based security evaluation scheme) intends to improve node trust by examining their direct and indirect interactions. The model distinguishes between untrustworthy nodes that should be segregated and trustworthy nodes that would be excellent for communication by combining direct and indirect trust.

In brief, our contributions can be summarized as follows:

- This research discusses the most efficient classic and novel approaches for establishing and evaluating trust and reputation in the distributed environment.
- The paper addresses the lack of trust-based security models for the DCS domain.
- The paper presents practical approach for trust-based security evaluation of the DCS environment.
- The model adopts the Probabilistic Bayesian approach for trust validation and node isolation. The Bayesian model is one of the most used architectural components for the numerical aggregation of past interactions and statistical trust computation [9–12].

The remainder of the paper is laid out as follows: The second section presents the most relevant previous work in the trust and reputation evaluation field. Section 3 covers the planned TRUSED scheme's design, whereas Section 4 describes its implementation. Section 5 discusses simulation and experiment results, followed by Section 6's concluding observations and recommendations for further work.

2 Related Works

The trust and reputation management models can be categorized into two types: centralized architecture and distributed architecture. In a centralized architecture, one central entity is responsible for assessing trust across all the nodes within a network. This central entity acts as the trust manager for the network by maintaining all the trust scores and responding to all the incoming requests from the devices [13]. This approach is beneficial in reducing the overhead cost. Centralized architecture is not, however, very fault-tolerant. If the central entity fails to function, it will disrupt the entire network. In the distributed architecture, all the nodes are responsible for calculating the trust values and maintaining scores. Because of processing overhead associated with all the nodes and, unlike centralized architecture, the load is shared among all the nodes [14,15]. In addition to this, the failure of one node does not drastically affect the entire network, thus making it more fault-tolerant than a centralized network. The distributed control systems have a similar mechanism where the controllers are geographically scattered and each controller acts as a standalone communicator. Based on DCS functional similarities, this paper will focus on distributed reputation architecture for establishing trust between the service receiver and the service provider. This section presents the various computational trust and reputational-based models, recent advances in research effort, and their limitations.

Various novel and classic quantitative approaches exist for evaluating trust and reputation in a distributed environment. This section presents the various computational trust and reputational-based models, recent advances in research efforts, and their limitations.

The fuzzy theory-based trust and reputation model provides a mechanism to overcome trust establishment and management issues among CPS, devices, and wireless sensors, using reputation in the Internet of Things (IoT) [16,17]. The model focuses on sensor nodes that consider the quality of Service (QoS) metrics [13]. In this model, each node takes direct trust and indirect reputation to determine the trustworthiness of other sensor nodes [18]. Direct trust is derived from direct observations, and indirect reputation is developed based on the recommendations of neighboring nodes. Based on the nodes' behavior in route discovery, maintenance, and data forwarding, the nodes are divided into two types, the malicious nodes that do not perform the package forwarding function and the nodes that do not participate in the route discovery phase.

The fuzzy model performs well in detecting the malicious nodes in Wireless Sensor Network (WSN). The model does not, however, address forged transactions and the effect of manipulated nodes. Thus, the model can be further optimized for discarding the influence of malicious nodes on the results and improvising the mechanism of updating trust**.**

On the other hand, Asiri et al. (2016) [17] propose a recommended-based trust and reputation model for overcoming the limitations of the Fuzzy theory-based trust model. The model protects against good-mouthing, bad-mouthing, and ballot stuffing attacks by constantly updating the weights and taking the history of behavior and rating quality into account for calculations. The model uses a probabilistic neural network framework to differentiate between trustworthy and malicious nodes [19]. The probabilistic neural network comprises a multi-layer architecture including input, hidden, pattern, and output layers [20].

The authors have introduced a distributed computing model where the computations are distributed between the nodes. The framework has defined the stronger nodes as the alpha nodes. These nodes are responsible for processing the computations [21]. The model also considers data sensitivity by taking an input parameter to specify the sensitivity level to differentiate between insensitive and confidential information and compute it accordingly [22,23]. It proposes a probability density function for calculating and distinguishing between untrustworthy and trustworthy nodes. Based on

the probability evaluated by the neural function, the rating matrix is updated by the alpha nodes accordingly [22].

Overall, the model has guaranteed better availability, energy preserving mechanisms, reduced computation overheads over publicly available information [20], and protection against good-mouthing, bad-mouthing, and ballot attacks. The model, however, cannot protect against attacks such as distributed denial of service (DDOS), man-in-the-middle (MIM), and wormhole. Lastly, as per current research, the model lacks real-life implementation.

The quantitative model introduced by Yu et al. (2017) [24] emphasizes the importance of establishing trust relationships among sensor nodes by efficiently analyzing the nodes' behavior. In this architecture, the main node calculates and analyzes the trust value to determine whether to transfer the packet to the next node or drop the packet [15]. The model considers packet forwarding capacity, repetition rate, consistency of the packet content, delay and integrity into consideration as trust factors [25,26]. The model utilizes the information entropy theory to determine each decision factor's weight. The reason behind incorporating the information entropy theory is to eliminate the inconsistency of the decision model because of the subjective weight setting. In addition to this, the model has also incorporated Dempster-Shafer (D-S) theory to deduce and synthesize the trust for calculating the indirect trust values [20].

The quantitative model promotes secure packet forwarding and eliminates the inconsistency of the decision model because of subjective weight setting by using information entropy theory consumption. In addition, the model uses less energy due to trust exchange among neighboring nodes, thus reducing latency and energy. However, the research lacks practical needs-driven implementation. The research could be extended to integrate practical demands including efficient technologies and lightweight trust management for further improvising the framework.

The CTrust model analyzes the trust of a node based on its current and past interactions with a node [27]. The key features of this model include the introduction of practical solutions for concerns regarding the establishment of trust and reputation. The model comprises mathematical functions for trust assessment, delay recommendations, and aggregation for computing partial trust values [28,29]. According to the model, the trust is accessed based on current and past interactions and recommendations, although the weightage for the interactions differs. The current interactions weigh more than the past interactions. The model has defined the trust maturity threshold as 250 direct interactions, after which the two nodes can interact without a recommendation from other nodes. The trust parameters can be defined based on the context at the runtime, and the truster has the authority to assign the weight to each parameter [30–32].

This model has several benefits, including the weighted trust parameters, weighted recommendation function, trust decay function, and a method for developing trust maturity and equilibrium between two nodes. The trust decay function allows the past trust to degrade or be eliminated over time, thus allowing more weight to the new trust scores [33]. In addition, it addresses general self-promoting and opportunistic attacks [34]. However, the model lacks a threshold scale for the parameters. In addition, it requires more computing and energy resources, high energy consumption issue has remained unresolved. This concern has also been addressed by Wang et al. (2020) [30] and Azad et al. (2020) [35] in their comparative reviews. Lastly, the model has not addressed the data privacy concerns and the risks that could arise from security threats and attacks. Overall, the model has adapted innovative approaches for tackling trust and reputation development issues. However, there is one significant gap that the authors can address, which relies on parameters' weight determination.

The Bayesian Model proposed by Teacy (2012) [9] places more weight on past interactions with trustees in a context like the present situation than on new interactions. The Bayesian model is a statistical trust and reputation model that relies on a beta probability density function based on interactions from neighbor nodes to obtain the preliminary information and direct interaction to get the posterior distribution to estimate the trust and reputation rating [36]. The Bayesian model is one of the most used architecture applications for the numerical aggregation of past interactions and statistical trust computation [9–12]. The model considers direct and indirect experience for evaluating the QoS parameters. The model comprises two components: the reputation model and the confidence model. The reputation model refers to the indirect experience where it considers the relationship between behaviors and observations of different models. The confidence model refers to the direct experience where the trustor's perception about a trustee's behavior is taken into consideration [37]. The benefits include the model's ability to determine the expected behavior of a vaguely known agent by observing similar trustees and thus not restricting the agent to relying on mutually shared trustees' observations. The feature helps the model overcome the agent's whitewashing issue where the agent tries to incriminate facts to eliminate their bad reputation. Overall, the Bayesian model provides a well-defined framework that can be applied to a wide range of domain specific applications.

The predictive model defines an alternate approach for determining direct and indirect trust among nodes for routing packets between the service provider and receiver [38]. The model uses the beta probabilistic distribution approach for calculating the direct trust and AMRA/GARCH for determining the future behavior of the nodes based on their past performances. The authors have taken the number of packets properly forward: the number of packets dropped and the number of packets falsely injected are the main parameters for evaluating the model. In addition, a routing mechanism has been defined to ensure safe and accurate delivery of the packets by using the trustworthy nodes within the destination's path [39]. The model is highly effective for predicting the trust values of multiple steps ahead in the series. The model efficiently detects the dropped or false injected packets and defines them as a blacklisted node. Most importantly, the model addresses various attacks including routing table overflow and resource consumption, DoS attack, sleep deprivation, and it also addresses spoofing attacks. However, this protocol alone is not suitable for highly dynamic networks. In addition, being a quantitative model, it cannot establish trust unless some behavior response gets recorded. Therefore the model requires vulnerable time to detect malicious activities.

Every trust and reputation model targets a set of risks and attacks associated with the distributed environment. When selecting the aggregation technique for the trust model, it is important to determine the major attacks that the proposed trust model must address to assist in mitigating the target risks associated with the environment. Table 1 summarizes the comparison of various trust and reputation models.

Table 1: Comparison of novel and classic trust & reputation models

Research	Computation schemes				Metrics
	Composition	Propagation	Aggregation	Update	
Bayesian model [9]	Quality of service	Distributed	Weighted mean and probability distribution	Event driven	Can be defined based on the context

(Continued)

Table 1: Continued

Research	Computation schemes				Metrics
	Composition	Propagation	Aggregation	Update	
Fuzzy model [16]	Quality of service	Distributed	Static weighted sum and fuzzy logic	Time driven	Packet Delivery Ration, Energy Consumption and End to End packet forwarding
Recommender based trust model [17]	Quality of service	Distributed	Probability density function	NA	Memory available, CPU, Severity flag, rating, packets ratio, rate of transmission, battery life and packets dropped
Quantitative model [24]	Quality of service	Distributed	Entropy theory	NA	Integrity, Delay, Packet content consistency and capacity of packet forwarding
CTrust [27]	Quality of service or social	Distributed	Dynamic weighted sum	Event driven	Can be defined based on the context at the runtime
Gaussian distribution-based comprehensive trust management system (GDTMS)–[32]	Quality of service	Distributed	Gaussian distribution-based	NA	Energy efficiency, transmission performance
Prediction based trust model [39]	Quality of service	Distributed	Beta probabilistic distribution & ARMA/GARCH	NA	Acknowledgment for packets forwarded successfully, packets dropped and falsely injected

3 Proposed Model

A trust-based security evaluation scheme (TRUSED) has been proposed and presented in this section. The components of TRUSED are discussed in the subsequent sections, where the following assumptions are made:

- Nodes are deployed randomly, without mobility, and communicate via a shared bi-directional wireless channel within their communication range.
- After deployment, no new nodes are added or removed, and they cannot be recharged after their initial energy has been exhausted.
- Initially, all the nodes are trustworthy and unknown, and only through communication between them will it be discovered whether or not they can be trusted.
- Malicious node attacks manifest as Denial of Service (DoS). Malicious nodes drop packets intentionally and adversely affect communication consistency.

3.1 Components of TRUSED

The trust-based security evaluation scheme (TRUSED) consists of two modules, namely the Direct Trust evaluation and Indirect Trust evaluation modules, consisting of different components as shown in Fig. 1. The following sub-sections elaborate on the detail of each of the components.

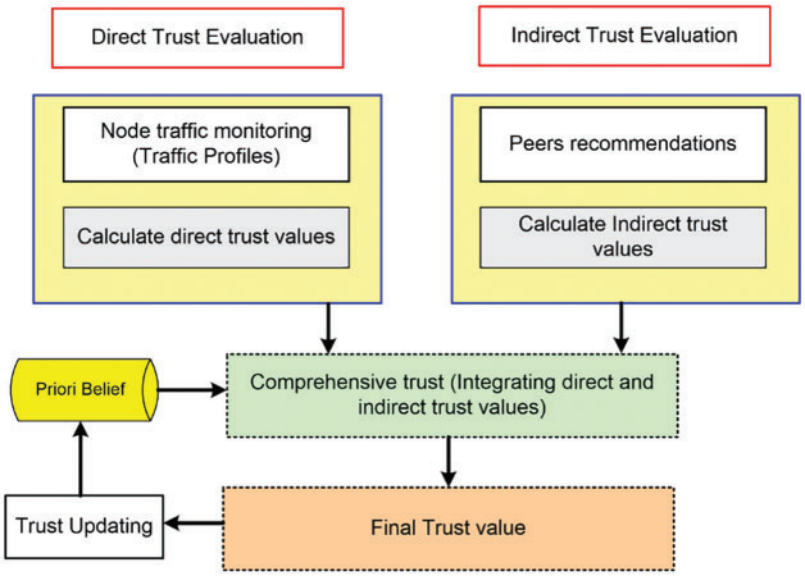

Figure 1: TRUSED–trust-based security evaluation scheme

3.2 Direct Trust Evaluation

Monitoring the communication behaviors of an evaluated node, such as packet forwarding or dropping behavior, which are stored in the form of traffic profiles at each node, determines the evaluation of a direct trust. In the same way, indirect trust is gained by recommendations from neighbors who have previously engaged with the node for which a recommendation is sought. Comprehensive trust is obtained by combining direct and indirect trust values, with the ultimate trust value of the node in question being checked and updated in accordance with the Probabilistic Bayesian theory [40].

A node's trust value is determined by the node's direct and indirect interactions with other nodes. As T_{AB}, denotes the trust value of the object node 'B' that is evaluated by subject node 'A' at a time 't', where the degree of trust is denoted as [0, 1], which indicates the node's trust level, either 0 or near to 0 and denoted as complete distrust while the value near to 1, represents the entire trust.

Moreover, the threshold value of trust is set to be 0.5, which indicates the node is malicious or untrustworthy [40]. A complete explanation of the trust-based security evaluation scheme is incorporated in this sub-section. Fig. 2 displays a hypothetical scenario in which subjective node A assesses the object node B's direct trust and receives indirect trust in the form of recommendations from other nodes.

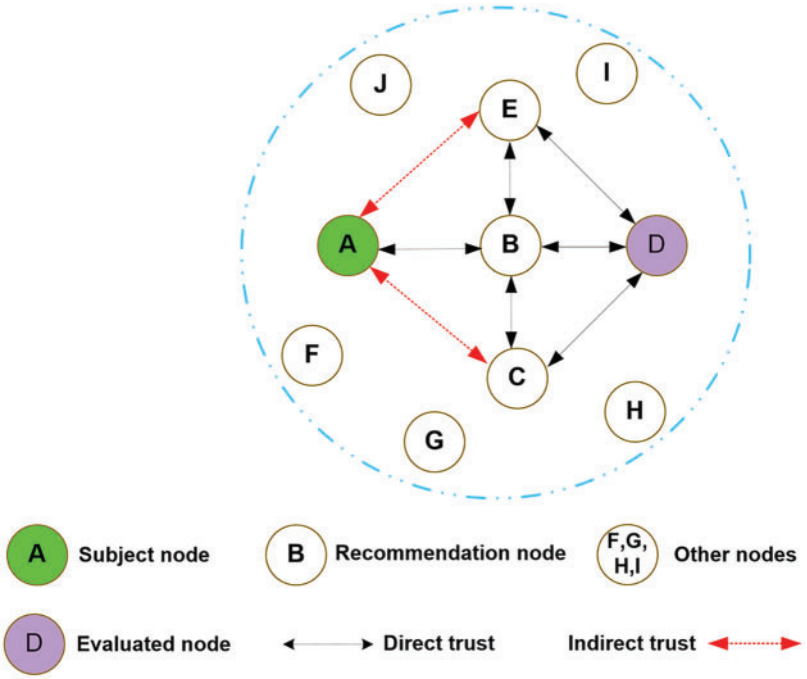

Figure 2: TRUSED–network topology scenario

The direct trust calculation is carried out through the transmission of data from node 'A' to node 'B' at the time 't', whereas the trustworthiness level of the node is obtained through the sent data packets, received data packets, and transit data packets which are stored and maintained at each node in the form of data traffic profiles (Tp) [40]. Moreover, the estimation of the node trustworthiness level is obtained, which is based on the probability of trust values such as:

- **Trust assessment of received data packets TRPA,B(t):** If node 'A' monitors node 'B' and confirms how many common acknowledgement data packets node 'B' delivers, the ratio of node 'B' received packets may be derived in the time period 't', according to the assumption. Due to the presence of malicious nodes on the network, packet loss may occur. There may be several malicious nodes depending on the distance between the source and destination, and packet loss can be catastrophic as a result. The *TRP* in Eq. (1) denotes the ratio of the packets received.

$$TRP_{AB}(T) = \frac{TRP_{AB}(t) - TRP_{AB}(t-1)}{TRPP_{AB}(t) + TRP_{AB}(t-1)} \tag{1}$$

- **Trust assessment of sent data packets $TSP_{A,B}(t)$:** As per the assumption, if node 'B' sends the data packets to node 'D', which is beyond the communication range of node 'A'. Although the sender cannot monitor packets that an intermediate node successfully sends to the next node, however, due to the broadcast nature of the wireless medium, any node in the range with promiscuous mode enabled can monitor the packets of other surrounding nodes. As a result, as indicated in Eq. (2), the sender can still keep track of the number of forwarded/repeated packets transmitted by the intermediate node.

$$TSP_{AB}(Tt) = \frac{TSP_{AB}(t)}{TSP_{AB}(t) + RPT_{AB}(t-1)} \tag{2}$$

where TSP_{AB} (T) indicates the data packet to be transmitted, albeit some data packets must be retransmitted because they were not received the first time. Because of the presence of a malicious node, packets are not being received or retransmitted. As a result, the retransmitted data packets were also considered and designated as RPT_{AB}.

- **Trust assessment of transit data packets TTPA,B(t):** The trust assessment for transit data is based on the time period 't', the number of packets that source node 'A' transmits to destination node 'K' through some intermediary node(s). It is difficult for a node to interact directly with a target node in a multi-hop environment. By involving the intermediate node, this communication is achievable. Only the sender node can assess the trustworthiness of node 'B' when the intermediary node 'B' changes its receiving array and likewise updates the 'A' node. Eq. (3) shows the transit trust evaluation of data packet TTP_{AB} (T) , at the intermediate node and at the receiving node:

$$TTP_{AB}(t) = \frac{TTP_{AB}(t) - TTP_{AB}(t-1)}{TTP_{AB}(t) + TTP_{AB}(t-1)} \tag{3}$$

In the equation, TTP_{AB} represents the total amount of sent data packets sent from node 'B' to node 'D', comprised of packets communicated and received at intermediary nodes and shared with node 'A'. The probability of the node trust estimation is based on trust values. However, with time more data traffic flow is accumulated, and therefore, the probability of trustworthiness is updated. Similarly, the trust formation between the nodes is also changed due to trust fluctuations. Therefore, Probabilistic Bayesian Theory estimates the trust degree of a node by calculating the number of successful and unsuccessful transmission of data packets [40].

3.3 Direct Trust (DT)

Multiple behavioral parameters 'β' must be developed with respect to multiple time intervals to ensure the proper behaviour of the proposed scheme under different circumstances. For instance, in one time period, node 'I' sends 2000 data packets towards node 'J' and 1000 packets are successfully received. Similarly, in another time period, node 'I' sends 1000 data packets towards node 'J' and successfully delivered 500 packets. In both scenarios the satisfactory ratio is 0.5, and therefore, in terms of percentage, both performed equally but, realistically, the former one is more realistic. Therefore, the behavioral parameters 'β' can be mathematically represented as shown in Eq. (4):

$$\beta = \frac{RE_{ij}(t) / (\mathrm{DR}_{\mathrm{AB}}(t) + RE_{AB}(t))}{\mathrm{RE}_{\mathrm{AB}}(t-1) / [\mathrm{DR}_{\mathrm{AB}}(t-1) + \mathrm{RE}_{\mathrm{AB}}(t-1)]} \tag{4}$$

where $\mathrm{RE}_{\mathrm{AB}}$ (t) denotes the number of data packets received at a specified time interval, and $\mathrm{DR}_{\mathrm{AB}}$ (t) displays the number of data packets dropped during transmission. As a result, $\mathrm{DT}_{\mathrm{AB}}$ (t), the evaluated direct trust, is represented as:

$$DT_{AB}(t) = \beta \times IT(T) \times [W_1 \times (1 - |TPA_{AB}(t)|) + W_2 \times |TSP_{AB}(t)| + W_3 \times (1 - |IT_{AB}(t)|) + (1 - TTP(t)) \times DT_{AB}(t-1) \tag{5}$$

In Eq. (5), the time interval of a data packet is denoted by Interval Time IT(t) whereas the action parameter 'β' considers the influence of time intervals. Similarly, for decision-making, a weighting method is utilized, and the overall trust value of each node is established by combining direct and indirect trust. The suggested approach includes weights because of their impact on minimizing the likelihood of incorrect recommendations signified by other nodes. As a result, W_1, W_2, W_3 are trust

values that must meet $W_1 + W_2 + W_3 = 1$ and are treated equally [41,42]. These weights, on the other hand, are independent of one another and can be adjusted differently depending on the scenario and application.

3.4 Indirect Trust (IDT)

When a pre-existing trust relationship between two nodes is not created via packet exchange or any other kind of interaction, it is referred to as indirect or recommended trust. If node 'A' trusts node 'B', and node 'B' trusts node 'C', then node 'A' trusts node 'C' indirectly. Similarly, trust might be intransitive, i.e., just because node 'A' trusts node 'B' and node 'B' trusts node 'C' does not mean node 'A' trusts node 'C'. Furthermore, this intransitive trust does not exclude the potential of trust information transfer [40]. However, the node 'N' calculates DT_{AB}^{direct} for evaluating the node 'D' and sends it to node 'A' as a recommendation of trust for the node 'B'. As a result, each time a suggestion from another node is updated, the chance of node 'B' being trustworthy or malicious is updated as well. The Bayesian estimate approach is used to determine the intensity of this belief. This estimate is based on the likelihood of an incidence based on the evidence available. Because the evidence is updated on a regular basis based on the amount of dropped packets, the posterior likelihood of each node being malicious or trustworthy is similarly updated on a regular basis. For the reason that Bayesian estimation is based on prior probability, each new probability is calculated and saved in the database to be used as a prior probability in the following round [40]. As a result, whenever a new probability is calculated, it is also saved in the database to be used as a prior probability in the next round, as shown in Eq. (6) [43].

$$\mathrm{P}(\mathrm{O} \mid \mathrm{E}) = \frac{\mathrm{P}(\mathrm{E}|\ \mathrm{O})\ \mathrm{P}(\mathrm{O})}{\mathrm{PE}} \tag{6}$$

Here, P(O | E) is the conditional probability where O denotes the occurrence and E denotes the evidence; thus, $P(O|\ \mathrm{E})$ denotes the likelihood of O assuming E is true. $P(\mathrm{O}\ |\mathrm{E})$ is the probability of O being true if E is true. $P(\mathrm{O})$ represents the prior probability and $P(E)$ is a normalizing constant that indicates the probability of E in all conditions. However, the Bayesian theorem, on the other hand, demands evidence in hand, which is not available at first but becomes available once the participating nodes begin communicating with one another and data traffic profiles are built. As a result, as illustrated in Eq. (7), the proposed approach for trust estimation is transferred to the Bayesian estimator.

$$P(U\ |SW_{nB}) = \frac{P(SW_{nB} \mid U)\ P(U)}{P(SW_{nB})} \tag{7}$$

In Eq. (7), the trust probability for the evaluated node 'B' is estimated, with the direct trust evaluation of node 'B' being provided through neighboring node 'N'. Here, (SW |U), denotes the node 'B' which is considered trustworthy, where P(U) reflects the prior probability discovered in the previous round. The normalization factor P(SWnB) represents the total probability in all situations. In the same way, Eq. (8) is calculating indirect trust ($IDT_{AB}^{indirect}$), which is based on the level of trustworthiness of a node 'B' as determined by its common neighbors of node 'A'. Furthermore, the number of surrounding nodes can vary in order to suit the recommendations for all neighbors.

$$IDT_{AB}^{indirect} = \frac{\sum_{n=0}^{N} P(B\ is\ trustworthy \mid Trust\ Value\ nB)}{N} \tag{8}$$

Eq. (9) derives both direct and indirect trust values from the DT and IDT trust values computed by Eqs. (7) and (8) respectively.

$$Total\ Trust\ \ (TT) = DT_{AB} + ITD_{AB} \tag{9}$$

The node's total trust value is computed by adding the direct and indirect trust values.

4 Implementation of the TRUSED Scheme

The efficacy of the proposed TRUSED scheme is assessed using the OMNET++ simulator due to its open-source and discrete nature with rich support of the graphical representation of a network. The nodes are deployed randomly within the network area of 100 m × 100 m. Furthermore, the simulation time for different studies range from 200 to 1200 s. A constant bitrate (CBR) traffic generator with a packet size of 50 bytes is utilized with the transport layer protocol (UDP). TRUSED's results are analyzed for trustworthiness, malicious node detection rate, detection accuracy, packet drop ratio, average network throughput, and packet delivery ratio.

IEEE 802.15.4 was utilized as the physical layer standard, and UDP was used as the transport layer protocol, with a constant bit rate (CBR) traffic generator and a data packet size of 50 bytes. The simulation length ranges from 200 to 1200 s, with a variable number of network nodes ranging from 10 to 50, randomly distributed over a 100 m × 100 m area. Table 2 shows a summary of simulation parameters.

Table 2: Simulation parameters and surrounding environment

Parameter	Value
Sensor field	100 m × 100 m
Node deployment	Random
Simulation time	200–1200 s
Network traffic type	CBR
Data packet size	50 Bytes
Physical standard	IEEE 802.15.4
Agent type	UDP
No. of nodes	10–50
Message queue type	Drop tail
Routing protocol	AODV

Moreover, Fig. 3 depicts the flow chart for the proposed TRUSED.

The algorithm for node trust calculation is as follows:

Input: Behavior characteristics collection (TRP_{AB}, TSP_{AB}, TTP_{AB})

Output: *Recommended/not-recommended for communication*

1: TRP = The average of the received packets assessment in the last interval
2: TSP = The average assessment of total sent packets
3: TTP = The average assessment of total packets (which are in transit) in the last interval

(Continued)

Algorithm Continued

4: $P\,(UN|\text{SPA}) = \dfrac{P(\text{SPA}|\ \text{UN})P\,(\text{UN})}{\text{P}\ \ (\text{RPA} + \text{SPA} + \text{TPA})}$
5: if $P(\text{UN}|\text{SPA}) > \text{Th}$
6: Mark as recommended for communication
7: else
8: Mark as not recommended for communication
9: Update the database for prior probability
10: end if

The algorithm steps are explained as follows:

- **Step 1:** The algorithm takes the nodes' behavioral characteristics such as received packet assessment, send packet assessment, and transit packet assessment as input values.
- **Step 2:** The model computes the input values and calculates the total trust value.
- **Step 3:** After computation, the model evaluates the total probability against the threshold to determine if the node is ideal for communication or if it's an untrustworthy node that should be isolated.

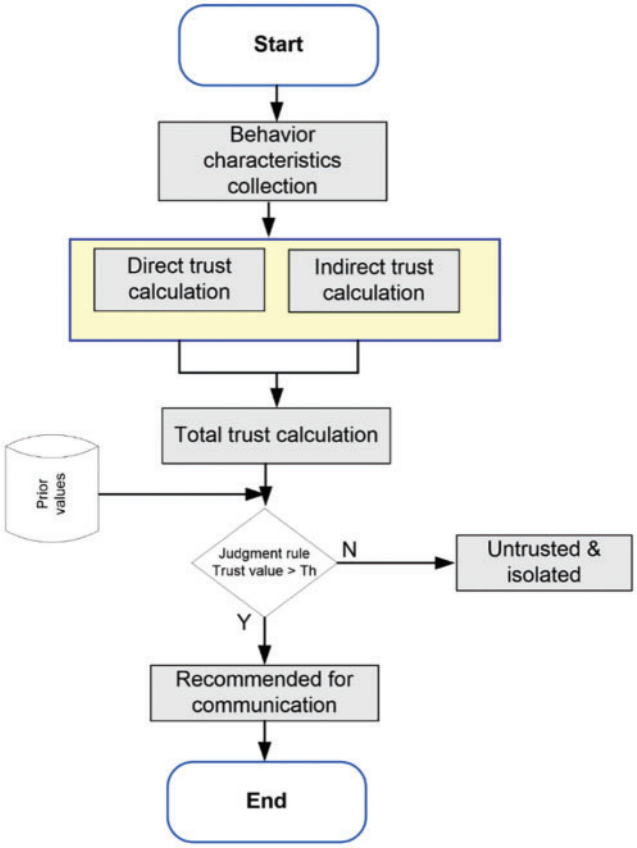

Figure 3: Flow diagram of the proposed TRUSED

5 Results and Discussion

To evaluate the performance of the proposed TRUSED scheme various performance parameters were considered and benchmarked against GDTMS [32] and CTrust [27], which are as follows:

- **Impact of trustworthiness:** This parameter depicts the impact of the trustworthiness level over time and in the presence of malicious nodes in the network.
- **Impact of the malicious node detection rate:** The detection rate parameter represents the malicious node detection ratio after implementation of the proposed scheme.
- **Impact of detection accuracy:** The actual detected percentage of the malicious nodes in lieu of false positive recommendations.
- **Impact of packet drop:** This parameter evaluates the number of drop packets.
- **Impact of the packet delivery ratio (PDR):** The PDR parameter describes the number of successfully received data packets in comparison to the total number of transmission packets that are expected to arrive at the receiver.
- **Average network throughput impact:** In the presence of malicious nodes, the average network throughput parameter is used to analyze the throughput and payload in bits per second (bps) during the entire session and divided by the total time of communicating nodes.

5.1 Trustworthiness Level

In the first scenario the performance of the proposed TRUSED is analyzed using the trustworthiness level of the nodes with respect to time, and the results show a steady performance when compared with GDTMS [32] and CTrust [27], shown in Fig. 4. TRUSED shows an increasing trend compared to the other two schemes, which is due to the consideration of direct and indirect trust computation. Furthermore, as time passes, more traffic problems become apparent, as do more traffic profiles, resulting in an increase in node trust. Moreover, isolation of a malicious and untrustworthy node increases the cooperation among nodes, which improves the trustworthiness level.

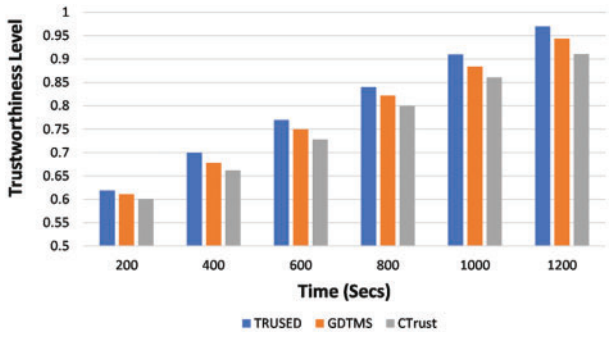

Figure 4: TRUSED–trustworthiness (with time)

5.2 Detection Rate

The second scenario investigates the amount of trust in the presence of hostile nodes. Fig. 5 shows that the proposed TRUSED has a declining character, with a changing number of malicious nodes ranging from 10% to 50% with a 10% increment. However, when compared to GDTMS [32] and CTrust [27], TRUSED's detection rate remains high. This is due to TRUSED's trust assessment in many orientations, such as sent, received, and transmitted data, and conducting the trust evaluation of each data packet, which gradually improves node trust.

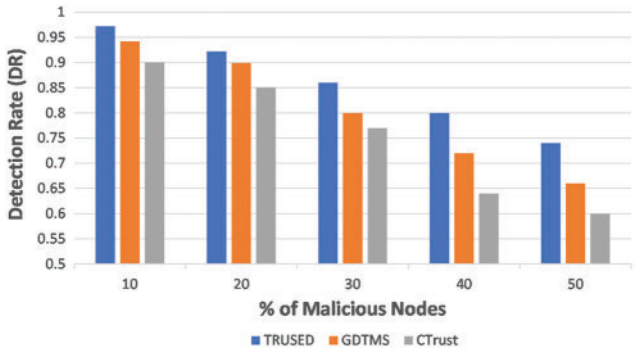

Figure 5: TRUSED–malicious node detection rate

5.3 Detection Accuracy

Fig. 6 depicts the proposed TRUSED's detection accuracy in the presence of different numbers of malicious nodes. In comparison to GDTMS [32] and CTrust [27], TRUSED performance is improved due to increased collaboration between nodes and a lower packet drop rate, which is attributable to the accuracy in detecting malicious nodes.

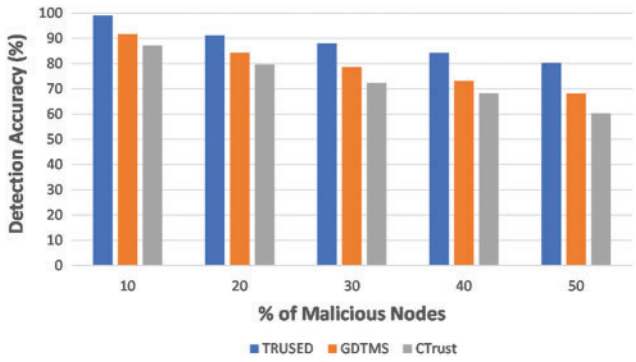

Figure 6: TRUSED–detection accuracy

5.4 Packet Drop Rate

As shown in Fig. 7, the proposed TRUSED has a lower packet drop rate than the other two schemes GDTMS [32] and CTrust [27], which is due to TRUSED's trust calculation and prediction capability, as TRUSED can predict a node's trust value and provides end-to-end trustworthy routes, resulting in a lower packet drop ratio and higher network throughput.

5.5 Packet Delivery Ratio (PDR)

Fig. 8 shows the packet delivery ratio of TRUSED's proposed scheme. As can be observed, TRUSED has a greater packet delivery ratio than the other two schemes, GDTMS [32] and CTrust [27] which is due to effective trust design and detection of malicious nodes. Malicious nodes spread false recommendations about the legitimate nodes and usually not cooperating in the communication process and drops the data packets which results in a less packet delivery ratio.

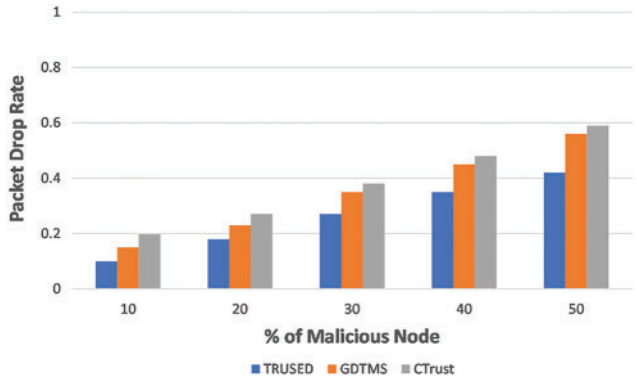

Figure 7: TRUSED–packet drop rate

Figure 8: TRUSED–packet delivery ratio

5.6 Average Network Throughput

Fig. 9 presents the results of average network throughput analysis performance of TRUSED, which is better than its counterparts GDTMS [32] and CTrust [27] respectively. Due to its trust evaluation skills and isolation of untrustworthy nodes, it can operate even in the presence of malicious nodes. Existing approaches can achieve good network performance, but precise detection of rogue nodes is difficult.

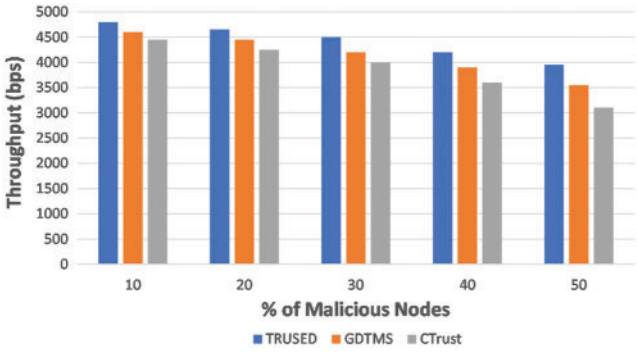

Figure 9: TRUSED–average network throughput

6 Conclusion and Future Work

Securing the Distributed Control Systems (DCS) is challenging due to their pervasive environment and complexity. Traditional network security mechanisms are ineffective regarding comprehensive solutions in securing DCS since they cannot detect the malevolent nodes. Trust-based security measures are helpful and substantial solutions and play a vital role in securing the DCS environment. In this paper, we proposed the TRUSED scheme that differentiates between the trustworthy and the untrustworthy nodes for promoting secure communication within a distributed environment. This scheme is designed to enhance security for Distributed Control Systems by monitoring the nodes and isolating the malicious ones. TRUSED is evaluated against a various performance characteristics, including trustworthiness, malicious node detection rate, detection accuracy, packet loss ratio, packet delivery ratio, and average network throughput, using the OMNET++ simulator. The suggested scheme's outcomes are compared to two existing models, GDTMS [32] and CTrust [27]. Overall, TRUSED performed better when compared to the benchmark criteria.

Funding Statement: The research that produced these findings received Project Funding from The Sultan Qaboos University, the Sultanate of Oman, under Research Agreement No [IG/EPS/INFS/21/04].

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] U. Patil, "Study of wireless sensor network in SCADA system for power plant," *International Journal of Smart Sensors and Ad Hoc Networks*, vol. 1, no. 2, pp. 41–44, 2011.

[2] C. Amarawardhana, K. S. Dayananada, H. Porawagama and C. Gamage, "Case study of WSN as a replacement for SCADA," in *2009 Int. Conf. on Industrial and Information Systems (ICIIS)*, Peradeniya, Sri Lanka, IEEE, pp. 49–54, 2009.

[3] S. Sennan, K. Kirubasri, Y. Alotaibi, D. Pandey and S. Alghamdi, "EACR-LEACH: Energy-aware cluster-based routing protocol for WSN based IoT," *Computers, Materials & Continua*, vol. 72, no. 2, pp. 2159–2174, 2022.

[4] S. Ali, "Cybersecurity management for distributed control system: Systematic approach," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 10091–10103, 2021. https://doi.org/10.1007/s12652-020-02775-5.

[5] M. B. Mansour, T. Abdelkader, M. Hashem and E. S. M. El-Horbaty, "An integrated three-tier trust management framework in mobile edge computing using fuzzy logic," *PeerJ Computer Science*, vol. 7, no. e700, pp. 1–24, 2021. https://peerj.com/articles/cs-700/.

[6] K. Karthikeyan and P. Madhavan, "Building a trust model for secure data sharing (TM-SDS) in edge computing using HMAC techniques," *Computers, Materials & Continua*, vol. 71, no. 3, pp. 4183–4197, 2022.

[7] W. Sherchan, S. Nepal and C. Paris, "A survey of trust in social networks," *ACM Computing Surveys (CSUR)*, vol. 45, no. 4, pp. 1–33, 2013.

[8] G. Fortino, L. Fotia, F. Messina, D. Rosaci and G. M. Sarné, "Trust and reputation in the internet of things: State-of-the-art and research challenges," *IEEE Access*, vol. 8, pp. 60117–60125, 2020.

[9] W. L. Teacy, M. Luck, A. Rogers and N. R. Jennings, "An efficient and versatile approach to trust and reputation using hierarchical Bayesian modelling," *Artificial Intelligence*, vol. 193, pp. 149–185, 2012.

[10] Y. Wang and J. Vassileva, "Bayesian network-based trust model," in *Proc. IEEE/WIC Int. Conf. on Web Intelligence (WI 2003)*, Halifax, NS, Canada, IEEE, pp. 372–378, 2003.

[11] J. Granatyr, V. Botelho, O. R. Lessing, E. E. Scalabrin, J. P. Barthès *et al.,* "Trust and reputation models for multiagent systems," *ACM Computing Surveys (CSUR)*, vol. 48, no. 2, pp. 1–42, 2015.

[12] A. A. P. Kazem, H. Pedram and H. Abolhassani, "BNQM: A Bayesian network based QoS model for grid service composition," *Expert Systems with Applications*, vol. 42, no. 20, pp. 6828–6843, 2015.
[13] A. Altaf, H. Abbas, F. Iqbal and A. Derhab, "Trust models of internet of smart things: A survey, open issues, and future directions," *Journal of Network and Computer Applications*, vol. 137, pp. 93–111, 2019.
[14] J. Guo, R. Chen and J. J. Tsai, "A survey of trust computation models for service management in internet of things systems," *Computer Communications*, vol. 97, pp. 1–14, 2017.
[15] I. Ahmad, K. L. A. Yau, M. H. Ling and S. L. Keoh, "Trust and reputation management for securing collaboration in 5G access networks: The road ahead," *IEEE Access*, vol. 8, pp. 62542–62560, 2020.
[16] D. Chen, G. Chang, D. Sun, J. Li, J. Jia *et al.,* "TRM-IoT: A trust management model based on fuzzy reputation for internet of things," *Computer Science and Information Systems*, vol. 8, no. 4, pp. 1207–1228, 2011.
[17] S. Asiri and A. Miri, "An IoT trust and reputation model based on recommender systems," in *2016 14th Annual Conf. on Privacy, Security and Trust (PST)*, Auckland, New Zealand, IEEE, pp. 561–568, 2016.
[18] F. Bao, R. Chen, M. Chang and J. H. Cho, "Hierarchical trust management for wireless sensor networks and its applications to trust-based routing and intrusion detection," *IEEE Transactions on Network and Service Management*, vol. 9, no. 2, pp. 169–183, 2012.
[19] M. Morshedi, J. Noll and R. Kari, "Building trustable remote monitoring and management systems," in *2018 IEEE/ACM Int. Conf. on Utility and Cloud Computing Companion (UCC Companion)*, Zurich, Switzerland, IEEE, pp. 213–219, 2018.
[20] M. Mohammadi, A. Al-Fuqaha, S. Sorour and M. Guizani, "Deep learning for IoT big data and streaming analytics: A survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2923–2960, 2018.
[21] Z. G. He, "Multi-parameter and time series based trust for IoT smart sensors," *International Journal of Network Security*, vol. 22, no. 4, pp. 589–596, 2020.
[22] I. U. Din, M. Guizani, B. S. Kim, S. Hassan and M. K. Khan, "Trust management techniques for the internet of things: A survey," *IEEE Access*, vol. 7, pp. 29763–29787, 2018.
[23] T. Ranathunga, R. Marfievici, A. McGibney and S. Rea, "A DLT-based trust framework for IoT ecosystems," in *2020 Int. Conf. on Cyber Security and Protection of Digital Services (Cyber Security)*, Dublin, Ireland, IEEE, pp. 1–8, 2020.
[24] Y. Yu, Z. Jia, W. Tao, B. Xue and C. Lee, "An efficient trust evaluation scheme for node behavior detection in the internet of things," *Wireless Personal Communications*, vol. 93, no. 2, pp. 571–587, 2017.
[25] A. Sharma, E. S. Pilli, A. P. Mazumdar and P. Gera, "Towards trustworthy internet of things: A survey on trust management applications and schemes," *Computer Communications*, vol. 160, pp. 475–493, 2020.
[26] S. F. A. Mon, S. G. Winster and R. Ramesh, "Trust model for IoT using cluster analysis: A centralized approach," *Wireless Personal Communications*, pp. 1–22, 2021. https://doi.org/10.1007/s11277-021-08401-7.
[27] A. A. Adewuyi, H. Cheng, Q. Shi, J. Cao, Á. MacDermott *et al.,* "CTRUST: A dynamic trust model for collaborative applications in the internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5432–5445, 2019.
[28] Z. Gao, C. Xu, H. Zhang, S. Li and V. H. C. de Albuquerque, "Trustful internet of surveillance things based on deeply represented visual co-saliency detection," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4092–4100, 2020.
[29] S. O. Ogundoyin and I. A. Kamil, "A trust management system for fog computing services," *Internet of Things*, vol. 14, pp. 100382, 2021. https://doi.org/10.1016/j.iot.2021.100382.
[30] T. Wang, L. Qiu, A. K. Sangaiah, A. Liu, M. Z. A. Bhuiyan *et al.,* "Edge-computing-based trustworthy data collection model in the internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4218–4227, 2020.
[31] W. Fang, N. Cui, W. Chen, W. Zhang and Y. Chen, "A trust-based security system for data collection in smart city," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, pp. 4131–4140, 2020.

[32] W. Fang, W. Zhang, W. Chen, Y. Liu and C. Tang, "TMSRS: Trust management-based secure routing scheme in industrial wireless sensor network with fog computing," *Wireless Networks*, vol. 26, no. 5, pp. 3169–3182, 2020.
[33] H. Liang, J. Wu, X. Zheng, M. Zhang, J. Li *et al.,* "Fog-based secure service discovery for internet of multimedia things: A cross-blockchain approach," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 16, no. 3s, pp. 1–23, 2020.
[34] S. M. Muzammal, R. K. Murugesan and N. Jhanjhi, "A comprehensive review on secure routing in internet of things: Mitigation methods and trust-based approaches," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4186–4210, 2020.
[35] M. A. Azad, S. Bag, F. Hao and A. Shalaginov, "Decentralized self-enforcing trust management system for social internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2690–2703, 2020. https://doi.org/10.1109/JIOT.2019.2962282.
[36] S. -R. Yan, X. -L. Zheng, Y. Wang, W. W. Song and W. -Y. Zhang, "A graph-based comprehensive reputation model: Exploiting the social context of opinions to enhance trust in social commerce," *Information Sciences*, vol. 318, pp. 51–72, 2015.
[37] N. Sardana, R. Cohen, J. Zhang and S. Chen, "A Bayesian multiagent trust model for social networks," *IEEE Transactions on Computational Social Systems*, vol. 5, no. 4, pp. 995–1008, 2018.
[38] A. Albeshri, "An image hashing-based authentication and secure group communication scheme for IoT-enabled MANETs," *Future Internet*, vol. 13, no. 7, pp. 166, 2021.
[39] 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, pp. 1467, 2019.
[40] R. W. Anwar, A. Zainal, F. Outay, A. Yasar and S. Iqbal, "BTEM: Belief based trust evaluation mechanism for wireless sensor networks," *Future Generation Computer Systems*, vol. 96, pp. 605–616, 2019.
[41] F. Afghah, A. Shamsoshoara, L. L. Njilla and C. A. Kamhoua, "Cooperative spectrum sharing and trust management in IoT networks," In: Charles A. Kamhoua, Laurent L. Njilla, Alexander Kott and S. Shetty, (Eds.), *Modeling and Design of Secure Internet of Things*, 1st ed., Wiley online library, pp. 79−109, 2020.
[42] W. Fang, M. Xu, C. Zhu, W. Han, W. Zhang *et al.,* "FETMS: Fast and efficient trust management scheme for information-centric networking in internet of things," *IEEE Access*, vol. 7, pp. 13476–13485, 2019.
[43] S. Iqbal, A. H. Abdullah, F. Ahsan and K. N. Qureshi, "Critical link identification and prioritization using Bayesian theorem for dynamic channel assignment in wireless mesh networks," *Wireless Networks*, vol. 24, no. 7, pp. 2685–2697, 2018.