Hindawi
Journal of Healthcare Engineering
Volume 2022, Article ID 5625897, 10 pages
https://doi.org/10.1155/2022/5625897

Research Article

A Novel Edge-Based Trust Management System for the Smart City Environment Using Eigenvector Analysis

G. Nagarajan,[1] Serin V. Simpson,[2] K. Venkatachalam,[3] Adel Fahad Alrasheedi,[4] S.S. Askar,[4] Mohamed Abouhawwash,[5,6] and Parthasarathi P[7]

[1]*Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology, Chennai, India*
[2]*Department of Computer Science and Engineering, SCMS School of Engineering and Technology, Kerala, India*
[3]*Department of Computer Science and Engineering, CHRIST (Deemed to be University), Bangalore 560074, India*
[4]*Department of Statistics and Operations Research, College of Science, King Saud University, Riyadh 11451, Saudi Arabia*
[5]*Department of Mathematics, Faculty of Science, Mansoura University, Mansoura 35516, Egypt*
[6]*Department of Computational Mathematics, Science, and Engineering (CMSE), Michigan State University, East Lansing, MI 48824, USA*
[7]*Department of Computer Science and Engineering, Bannari Amman Institute of Technology, Erode, India*

Correspondence should be addressed to K. Venkatachalam; venkatachalam.k@ieee.org

Received 16 January 2022; Revised 6 April 2022; Accepted 5 May 2022; Published 26 May 2022

Academic Editor: Senthil kumar

The proposed Edge-based Trust Management System (E-TMS) uses an Eigenvector-based approach for eliminating the security threats present in the Internet of Things (IoT) enabled smart city environment. In most existing trust management systems, the trust aggregation process completely depends on the direct trust ratings obtained from both legitimate and malicious neighboring IoT devices. E-TMS possesses an edge-assisted two-level trust computation approach for ensuring the malicious free trust evaluation of IoT devices. The E-TMS aims at removing the false contribution on aggregated trust data. It utilizes the properties of the Eigenvector for identifying compromised IoT devices. The Eigenvector Analysis also helps to avoid false detection. The analysis involves a comparison of all the contributed trust data about every single connected device. A spectral matrix will be generated corresponding to the contributions and the received trust will be scaled based on the obtained spectral values. The absolute sum of obtained values will contain only true contributions. The accurate identification of false data will remove the effect of malicious contributions from the final trust value of a connected IoT device. Since the final trust value calculated by the edge node contains only the trustworthy data, the prediction about the malicious nodes will be accurate. Eventually, the performance of E-TMS has been validated. Throughput and network resilience are higher than the existing system.

1. Introduction

A smart city environment has been established by utilizing the capabilities of edge computing-assisted IoT networks [1, 2]. The edge computing-assisted IoT network provides a collaborative computing facility with the help of a wide range of heterogeneous smart devices. Such a heterogeneous environment has the highest risk of being vulnerable to security attacks. Such networks require a robust trust management mechanism for maintaining a good device trust level. Trust management helps to keep users with increasing numbers. The traditional cloud-based trust evaluation approaches are incapable to analyze the context-aware trust relationships among connected IoT devices [3, 4]. The heterogeneity as well as the large size of the network became the prime reasons for the performance degradation of the centralized cloud servers. The centralized cloud server can work efficiently with smaller networks. But, it is hard to serve large-scale networks with centralized architecture. In such cases, the centralized server cannot offer real-time support to time-dependent applications. Also, it is not possible to make context-aware decisions for all the connected devices by a single cloud server.

Edge computing has been introduced to achieve context-aware data analysis among a large number of tiny IoT devices [5, 6]. The distributed architecture of edge computing-assisted networks is more vulnerable than the traditional cloud-based centralized architecture. Since the majority of the data will be processed near the end devices, the IoT network requires several data processing units (edge servers) at the edge of the network. That in turn increases the opportunities of the attackers to intrude on the network [7, 8]. All the security threats associated with the cloud server will be experienced at each tiny edge server. In other words, the attackers will utilize the vulnerabilities of edge servers to intrude on the network. Thus, the data aggregation process, as well as the control information management, must be done in a secure environment. The trust of each device and communication must be evaluated in regular intervals by using a robust trust evaluation framework. Thus, a scalability and mobility-aware universal trust mechanism needs to be incorporated with an IoT-enabled smart city environment. The paper mainly deals with the following aspects.

(i) Contributing a robust mechanism to evaluate the trustworthiness of smart city devices

(ii) Contributing a two-level trust assessment approach for increasing accuracy

(iii) A method for the direct assessment of device trust level based on the occurred events

(iv) An event assessment approach for computing the trust value indirectly at the edge nodes

(v) Contributing an edge-driven Eigenvector-based approach for identifying the false trust contribution and malicious free aggregation of individual trust values

(vi) Contributing an Eigenvector method to identify and isolate the malicious entities in the smart city environment

The following sections of this paper will give a detailed idea about the proposed E-TMS approach. The next section checks the requirement of a robust trust-based approach by analyzing the currently functioning approaches in IoT-enabled smart networks. Section 3 gives an overview of the need for research in this area. The proposed Edge-based Trust Management System has been detailed in Section 4. The performance analysis and the comparative study have been included in Section 5. The conclusion and the future scope of research in this area have been discussed in Section 6.

2. Related Work

Wang et al. [9] introduced a recommendation-dependant system to take decisions for network management. The proposed work evaluates the trust of each entity in the smart city environment for excluding the malicious entities from the recommendation process. The computing node will accept the recommendations only from trustworthy IoT devices. The proposed work aims to utilize the trust-based recommendation mechanism to secure the network from various security threats. If the recommendation system considers only the trustworthy nodes, it can produce a reliable outcome. The proposed system evaluates each node based on the trust values. The trust will be calculated by the trust aggregation process. But the trust aggregation process does not possess an intelligent mechanism to eliminate the impact of malicious contributions. Thus, the selection of entities to participate in the recommendation system is vulnerable.

ElRahman and Alluhaidan [10] introduced a blockchain-based approach to secure healthcare IoT systems. The proposed framework designs a trust model to prevent data leakage. Most of the data involved with the healthcare systems will be related to personal health information. Such sensitive data needs to be handled carefully. The proposed system initially builds ontologies for the IoT network. The ontology-based IoT-enabled healthcare system utilizes semantic references to find cognitive relationships. Upon creating the ontologies, the framework applies blockchain technologies to secure the IoT network. Blockchain technology offers sensor data integrity to the perception layer, authentication service to the network layer, privacy-preserving schemes to the middleware layer, and mechanisms to ensure the overall security of the devices in the application layer. The overall operational complexity of the proposed approach is quite high. To enhance the performance of the edge servers, it is always adequate to employ only lightweight algorithms.

Adewuyi et al. [11] designed a recommendation dependant approach to evaluate the network entities. The system receives recommendations from all the registered entities to finalize the recommendation trust. Upon finalizing the recommendation trust, the proposed framework applies the belief function to estimate the trustworthiness of the evaluated trust. The output of the belief function indicates the willingness of each node to trust the recommendation trust. Thus, the nodes need not blindly believe the recommendations. Each node will act based on the output of the belief function. Thus, the recommendation trust cannot make changes directly to the existing trust relationships. Since the recommendation trust also includes the contribution from malicious nodes, the evaluation performed by the belief function may not be accurate always.

Fang et al. [12] introduced a fog-based approach for ensuring data integrity. The proposed method uses a source anonymity algorithm to make the source node undetectable to malicious nodes. Also, it integrates RSA digital signature to preserve the confidentiality of data. It follows a randomly delayed transmitting scheme to reduce energy consumption. But the overall framework lacks an intelligent approach to isolate the involvement of malicious nodes from the execution of subsidiary methods. Manimurugan et al. [13] introduced a machine learning-based approach for detecting malicious nodes. The work has been introduced to prevent unauthorized access to network resources. The method evaluates each entity by gathering necessary information from the neighboring nodes. The Deep Belief Network predicts the behavior of each network entity by analyzing the

individual contributions. Since the mechanism accepts trust contributions also from malicious nodes, the malicious node can make a large impact on the output. By utilizing this limitation, a malicious node can continue in the network for a long period.

Most of the trust management mechanisms in the IoT platform mostly adopt the contributive approach which accepts the recommendations from both legitimate and malicious nodes. All those systems are not concerned about malicious contributions. Such malicious contributions can mislead the network.

3. Problem Statement and System Architecture

A smart city environment holds several heterogeneous tiny end devices. Due to economic constraints, it is not possible to deploy resource-rich devices at the bottom layer to execute complex computations [14, 15]. Thus, a smart city environment highly relies on cloud/edge paradigms to fulfill both its operational and security needs [16, 17]. In most of the existing trust evaluation mechanisms, the edge server will aggregate the trust information from the connected IoT devices. But, the existing mechanisms usually do not possess an intelligent method to identify and eliminate the false contribution from the malicious nodes. This work mainly aims at identifying such untrustworthy contributions. The overall trust in E-TMS will be computed by considering the individual trust values obtained from direct as well as indirect evaluations. The direct trust will be obtained from the neighboring nodes based on the node's behavior toward a set of network events. The indirect trust will be computed depending on the node's involvement in network management. E-TMS performs an Eigenvector Analysis on the aggregated final trust values to detect the misleading contributions. The proposed two-level evaluation approach could produce the exact reflection of a node's behavior on the final trust value. Based on those observations, an edge node can confirm the malicious behavior of a connected device.

The architecture of the proposed E-TMS is shown in Figure 1. The cloud data center is responsible for performing all the complex computations. The edge nodes will be placed near the end devices. The edge node can fulfill all the required real-time computational needs of the smart city environment. The end devices will perform the individual trust assessments about the neighboring nodes and the edge servers will aggregate the same. The edge servers are also responsible for identifying and eliminating malicious contributions. The proposed architecture balances the computational overload of both cloud servers and the end devices by placing the real burden on the edge servers. A detailed report will be shared with the cloud server, whenever it is required.

4. Proposed System

E-TMS uses Eigenvector-based malicious identification approach for identifying and eliminating the malicious nodes from the smart city environment. The trust

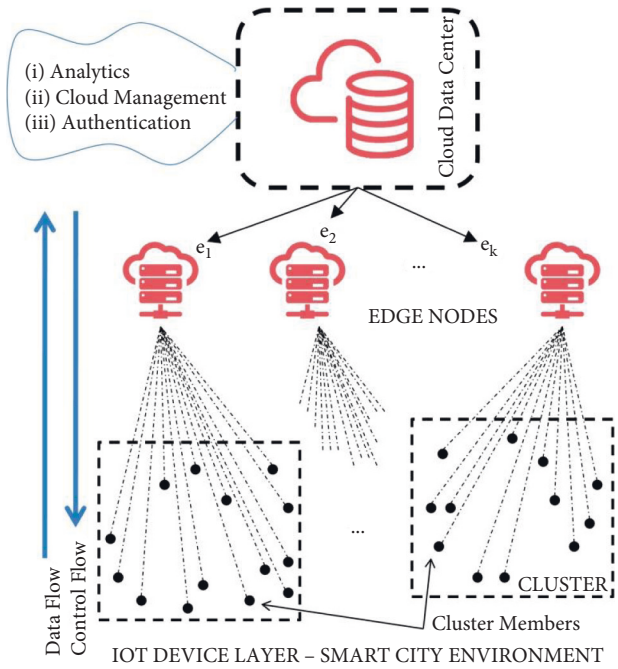

Figure 1: System architecture.

management system proposed in E-TMS uses a two-level trust assessment approach for generating the final trust about a node.

(i) Level 1: Edge Independent Direct Assessment

(ii) Level 2: Edge-Based Indirect Assessment

The final event-based trust values will be sent to the edge node for performing the Eigenvector computations to remove false contributions. The nodes which contribute malicious data will be included in the Do-not-Consider-List (DCL) and all the listed malicious entities will not be further considered.

Definition 1 (trustworthy node). The trustworthy node will behave legitimately to all the network events. Such a node will do its best to avoid packet drops and forward each packet to the desired next hop. A trustworthy node will hold the latest DCL packet, and all the routing decisions will be carried out only based on the available information in the DCL. Such nodes will strictly follow the rules associated with the node joining procedure. Also, the trustworthy node will perform all the optimizations required for maintaining the residual energy at a satisfying level. Any malicious interruptions can mislead the network entities from the above etiquettes. A trustworthy node must be able to withstand all such malicious interventions.

4.1. Event-Based Trust Assessment. All nodes in the network will compute event-based trust (ET_{ij}) about all other nodes in their occupying cluster. All such assessed values along with their own self-assessed score will be sent to the edge node for identifying the malicious nodes. The event-based

Table 1: Event-based score allotment: direct trust.

Events	Score
Correct forwarding of the offered packet	+1
Dropping an offered packet	−1
Reception of updated DCL packet from "n_j"	+1
Reception of old DCL broadcast form "n_j"	−1
Timely reply for a hello packet	+1
Route request for a node listed in DCL	−1

Table 2: Event-based score allotment: indirect trust.

Events	Score
Leaving a cluster without notifying the edge node	−1
Leaving the cluster in a proper way	+1
Violation of node joining procedure	−1
Approved node joining	+1
Residual energy	−1 or +1
Based on acknowledgment	−1 or +1

trust assessment involves two levels, edge independent direct assessment and edge-based indirect assessment. Both assessments will consider different network events for computing the trust score.

4.1.1. Edge Independent Direct Assessment (Level 1). All nodes will compute the event-based direct trust (DT_{ij}) of their neighboring nodes and that will be saved in the Local Trust Table (LTT).

Definition 2 (direct trust). It can be defined as the trustworthiness of that node toward the neighboring nodes. A neighboring node can compute the same by considering all the events occurring directly between them. The predicted response for every event will be identified at the initial level. The neighboring node will observe the evaluating node for a certain period. All the responses of the observed node for the occurring events will be examined closely. The prioritized or nonprioritized score can be assigned to all the responses. Direct trust can be formulated based on the obtained score values.

The direct trust can be computed using

$$DT_{ij} = \frac{\text{DirectTrustScore}_{\text{ij}}}{\text{TotalnumberofEventsConsidered}}. \quad (1)$$

The events considered for calculating the direct trust have been selected based on data transmission, DCL distribution, neighbor discovery process, and path determination. The events associated with the above-mentioned actions have been listed in Table 1. All the events which can produce a significant impact on results are considered for the direct trust evaluation. The events which have occurred in the desired fashion will contribute a positive score to the direct trust evaluation, and all undesired events will contribute a negative score. The initial Direct Trust Score will be assigned as "0" for newly joined nodes. Based on the involvement in the network, the Direct Trust Score of a neighboring node will be incremented or decremented by the assessing node.

The following events will be considered by node "n_i" during the direct assessment of neighboring node "n_j".

The Local Trust Table will be shared with the edge node, and further updates will be communicated at regular intervals. Based on the same, the edge server will construct a Global Trust Table (GTT) where each column represents the direct trust about a single node contributed by the neighboring nodes.

4.1.2. Edge-Based Indirect Assessment (Level 2). The node "n_i" will compute the event-based indirect trust of other cluster members by considering some network events. The events considered for the calculation of indirect trust have been listed in Table 2.

Node movement, node joining procedure, residual energy, and acknowledgment process have been considered for calculating the indirect trust. The neighboring node will compare the residual energy with a value that has been explicitly derived based on the application, to determine the score (+1 or a −1). Since the data about all the above-listed events are obtained from the edge node, the assessment is considered an indirect assessment.

Definition 3 (indirect trust). The indirect trust of a node can be defined as the measure of desirability in general network events. A node can calculate the indirect trust of its neighboring nodes by obtaining the necessary information from the monitoring authority (connected edge node). All the general network events can be considered for this evaluation. Since a normal network entity does not have access to the log data of general events, the data need to be obtained from the connected edge node. Thus, the evaluation completely depends upon the data provided by a third entity. Thus, the evaluation has been termed an indirect evaluation.

The event-based indirect trust (IT_{ij}) of node "n_j" can be computed using

$$IT_{ij} = \frac{\text{IndirectTrustScore}_{\text{ij}}}{\text{TotalnumberofEventsConsidered}}. \quad (2)$$

Both direct trust and indirect trust are equally significant while computing event-based trust.

4.1.3. Event-Based Trust (ET_{ij}). In order to calculate the event-based trust, the direct trust values (DT_{ij}) about the assessed node (n_j) will be obtained from the GTT. The edge node will send the values listed in the column corresponding to the assessed node. After getting the direct trust values, the assessing node (n_i) will do the following computations to nullify the effect of malicious contributions.

$$\text{Avg} = \sum_{i=1}^{m} DT_{ij}. \quad (3)$$

As an initial step, the average value of all the received direct trust values about node "n_j" will be computed. It includes the contributions from "m" contributing nodes that

have the direct connectivity (neighbors) with node "n_j". The deviation of each Direct Trust Value from the average value will be computed and listed as follows:

$$\text{dev}_{ij} = \left|\text{Avg} - DT_{ij}\right|. \tag{4}$$

$$\begin{aligned}\text{DeviationList} &= \left(\text{dev}_{1j}, \text{dev}_{2j} \ldots\ldots \text{dev}_{\text{mj}}\right)\\ \text{LargestDeviation, } L\ D &= \text{Max}\left(\text{dev}_{1j}, \text{dev}_{2j} \ldots\ldots \text{dev}_{mj}\right).\end{aligned} \tag{5}$$

The largest deviation value among the obtained deviations can be represented as *LD*. The weight values for nullifying the effect of malicious contributions can be computed using the following equations:

$$\text{diff}_{ij} = \left|(L\ D + 0.001) - \text{dev}_{ij}\right|. \tag{6}$$

$$\text{DifferenceList} = \left(\text{diff}_{1j}, \text{diff} \ldots\ldots \text{diff}_{mj}\right). \tag{7}$$

$$\text{Avg.Difference} = \frac{\sum_{i=1}^{m} \text{diff}_{ij}}{m}. \tag{8}$$

$$\text{WeightValue, } w_{ij} = \frac{\text{diff}_{ij}}{\text{Avg.Difference}}. \tag{9}$$

The received direct trust values will be multiplied with the corresponding weight values, and the average of obtained results will be the malicious free average of received direct trust values (MFDT_{ij}) about node "n_j":

$$\text{MFDT}_{ij} = \frac{\sum_{i=1}^{m} DT_{ij} \times w_{ij}}{m}. \tag{10}$$

Thus, the malicious contributions cannot tamper with the individual trust assessment process. The node "n_i" can compute the final event-based trust (ET_{ij}) of node "n_j" using equation (12):

$$ET_{ij} = 0.5 \times IT_{ij} + 0.5 \times \text{MFDT}_{ij}. \tag{11}$$

Each node will assess the event-based trust value of all the nodes in the same cluster. The obtained results will be shared with the connected edge server. Further, a trust aggregation process will be carried out at the edge server to examine the malicious behavior. Since the malicious nodes are also allowed to send the trust values, the edge server needs to be more efficient to identify the malicious contributions.

4.2. Trust Aggregation. Both trust aggregation and the process of finding the malicious nodes will be done at edge nodes for reducing the computational overhead at individual IoT devices. The received event-based trust will be stored as a $(n \times n)$ matrix (RT) at the edge node:

$$RT = \begin{bmatrix} ET_{(1,1)} & ET_{(1,2)} & \ldots & ET_{(1,n)} \\ ET_{(2,1)} & ET_{(2,2)} & \ldots & ET_{(2,n)} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ ET_{(n,1)} & ET_{(n,2)} & \ldots & ET_{(n,n)} \end{bmatrix}. \tag{12}$$

The i^{th} row of matrix RT includes the trust contributions of "n" number of cluster members about i^{th} IoT device. Similarly, the column of matrix RT includes the trust contributions of a single IoT device in the cluster about all other cluster members. Thus, the row average of matrix RT represents the relative trust of a single IoT device.

$$\text{RelativeTrust}_i = \frac{\sum_{j=1}^{n} ET_{(i,j)}}{n}. \tag{13}$$

The relative trust includes the contribution from both legitimate as well as malicious nodes. Thus the edge node cannot conclude the malicious behavior of a cluster member simply based on the row average value/relative trust. Thus, an Eigenvector-based malicious node identification approach has been introduced in the next section.

4.3. Eigenvector-Based Malicious Node Identification. The effect of trust contributions from the malicious nodes needs to be nullified for getting the actual trust value of individual cluster members. Here, we are applying a vector-based malicious identification approach for excluding the false trust contribution from the malicious nodes. We consider each trust contribution as an independent vector. In order to construct orthogonal vectors, the input matrix must be a symmetric matrix. Thus, it is required to construct a real symmetric matrix corresponding to the matrix RT. The device trust values received about a single node and the device trust values contributed by a single node will possess some unique patterns. Thus, the symmetric matrix must be capable enough to hold all such properties of the parent matrix (RT). The covariance matrix of any matrix will be symmetric. The covariance matrix is defined as a matrix that is able to show the covariance between each pair of elements in a matrix. The covariance matrix of RT can be represented as follows.

$$CoRT = \begin{bmatrix} \text{CoRT}_{(1,1)} & \text{CoRT}_{(1,2)} & \ldots & \text{CoRT}_{(1,n)} \\ \text{CoRT}_{(2,1)} & \text{CoRT}_{(2,2)} & \ldots & \text{CoRT}_{(2,n)} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \text{CoRT}_{(n,1)} & \text{CoRT}_{(n,2)} & \ldots & \text{CoRT}_{(n,n)} \end{bmatrix}. \tag{14}$$

where $\text{CoRT}_{(i,j)}$ will be same as $\text{CoRT}_{(j,i)}$ for all $i \neq j$. As per the probability theory and statistics, the diagonal elements of the covariance matrix (CoRT) can be computed using equation (16).

$$\mathrm{CoRT}_{(i,j)} = \left(\frac{1}{n}\sum_{i=1}^{n} ET^{2}_{(i,j)}\right) - \left(\frac{1}{n}\sum_{i=1}^{n} ET_{(i,j)}\right)^{2}. \quad (15)$$

Also, the covariance of nondiagonal elements will be computed using equation (17).

$$\mathrm{CoRT}_{(i,j)} = \frac{1}{n}\left(\sum_{k=1}^{n} ET_{(k,i)} ET_{(k,j)}\right) - \left(\sum_{k=1}^{n} ET_{(k,j)}\right). \quad (16)$$

Since each element in the covariance matrix has been computed by considering the covariance of each element in the matrix with other elements, all the properties of the parent matrix will be cloned effectively to the resulting matrix (CoRT). The Eigenvector and spectral values corresponding to the trust values received from the cluster members can be computed as follows. The characteristic equation can be represented as

$$|\mathrm{CoRT} - \lambda I| = 0, \quad (17)$$

where "CoRT" is an $(n \times n)$ matrix. "λ" represents the spectral values corresponding to the trust values received from a single node. "I" represents the identity matrix in the order of "CoRT". The characteristic equation (17)will be n^{th} degree polynomial in "λ". While solving (17), we will get "n" spectral values $\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$. The linear homogeneous system with respect to the (17) can be represented as

$$\begin{aligned} (CoRT * X - \lambda * X) &= 0 \\ CoRT * X &= \lambda * X, \end{aligned} \quad (18)$$

where X is an $(n \times 1)$ column matrix and $X \neq 0$ (i.e, nonzero vector). The matrix "X" is known as Eigenvector. Since the multiplication with identity matrix results in the same value, λ in equation (19) can be represented as a product of "λ" and "I".

$$\begin{aligned} (CoRT - \lambda * I)\, X &= 0, \\ (CoRT * X - \lambda * \mathrm{I} * \mathrm{X}) &= 0. \end{aligned} \quad (19)$$

We can solve the above-mentioned linear system (19) corresponding to each value of "λ". While solving the same for each value of "λ", we will get a nonzero Eigenvector (X_i) with order $(n \times 1)$. A spectral matrix can be constructed by including each "X_i", corresponding to all λ values.

$$\text{Spectral Matrix, SM} = [\mathrm{X}_1, \mathrm{X}_2 \ldots\ldots\ldots \mathrm{X}_\mathrm{n}]. \quad (20)$$

Here, X_i represents the Eigenvector corresponding to "λ_i".

$$X_i = \begin{bmatrix} v_{(1,i)} \\ v_{(2,i)} \\ \cdot \\ \cdot \\ v_{(n,i)} \end{bmatrix}. \quad (21)$$

The spectral matrix can be expanded by substituting the values for X_1 to X_n. The spectral matrix of order $(n \times n)$ after substituting the individual values is shown in

$$SM = \begin{bmatrix} v_{(1,1)} & v_{(1,2)} & \cdots & v_{(1,n)} \\ v_{(2,1)} & v_{(2,2)} & \cdots & v_{(2,n)} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ v_{(n,1)} & v_{(n,2)} & \cdots & v_{(n,n)} \end{bmatrix}, \quad (22)$$

where $v_{(1,1)}$ represents the i^{th} Eigenvector value corresponding to the i^{th} spectral values (λ_j). A transformation process has been applied to the CoRT matrix for getting the SM matrix. It is a process of scaling the received trust value corresponding to the obtained spectral values. The obtained values inside the spectral matrix represent the direction of each individual trust data. The mathematical operations applied to the received trust eliminate the effect of malicious contribution. Absolute row sums of the spectral matrix are the malicious free scalar values (MFSVs) of received trust corresponding to individual nodes.

$$\mathrm{MFSV}_i = \sum_{j=1}^{n} \left|v_{(i,j)}\right|, \quad (23)$$

where MFSV_i represents the malicious free scalar values of i^{th} node. The row average (RA_i) of obtained MFSV_i value represents the actual trust value of i^{th} node.

$$\mathrm{RA}_i = \frac{\mathrm{MFSV}_i}{n}. \quad (24)$$

The RA_i represents the aggregated trust of i^{th} node, which has been evaluated by considering the contributions of "n" number of nodes. The Eigenvector-based operations on received trust remove the effect of malicious trust contributions from the compromised nodes. Since the aggregated trust value of i^{th} node (RA_i) contains only the true trust contributions, the RA_i value can be used for the detection of malicious nodes inside the network. An Aggregated Trust Threshold (ATT) has been fixed to 0.2 based on the repeated simulation results for identifying the malicious nodes inside the network. Nodes having RA_i value less than ATT can be marked as malicious and will be included in DCL. The updated DCL packet will be circulated among the network entities at a regular time interval. Thus, the local copies of DCL stored at each network entity will be replaced with the updated list without any delay. A legitimate network node will initiate a communication only after verifying the trustworthiness of the recipient entity with the DCL stored in the local memory. This approach will eliminate the chances of the inclusion of malicious nodes in new communication. Thus, the proposed method can ensure the complete isolation of malicious nodes with the help of DCL.

5. Comparison and Analysis of Experimental Results

The performance of E-TMS has been examined with the help of network simulator NS 2.35. Table 3 summarizes the network conditions introduced for setting up the simulation environment. Since the IoT devices are mobile in the

Table 3: Simulation parameters.

Parameters		Scenario 1	Scenario 2
Physical layer	S. propagation	Two-ray ground	
	Antenna model	Omniantenna	
Mac layer	Mac protocol	802.11	
	Link bandwidth	1 MB	
Simulation	Size of network field	1000 m × 1000 m	1000 m × 1000 m
	Rate (Mbs)	0.1	0.1
	Packet size (B)	1000	1000
	Traffic type	CBR	CBR
	Duration (s)	600	600
	Speed (m/s)	25	25
	Number of nodes	25/50/75/100/125	100
	Load	500 Kb	1000–6000 Kb
Queue	Type	DropTail/PriQueue	
	Size	50	
NS2 version		2.35	
Processor		Intel processor 3 GH	
Operating system		Ubuntu 16.04 LTS	

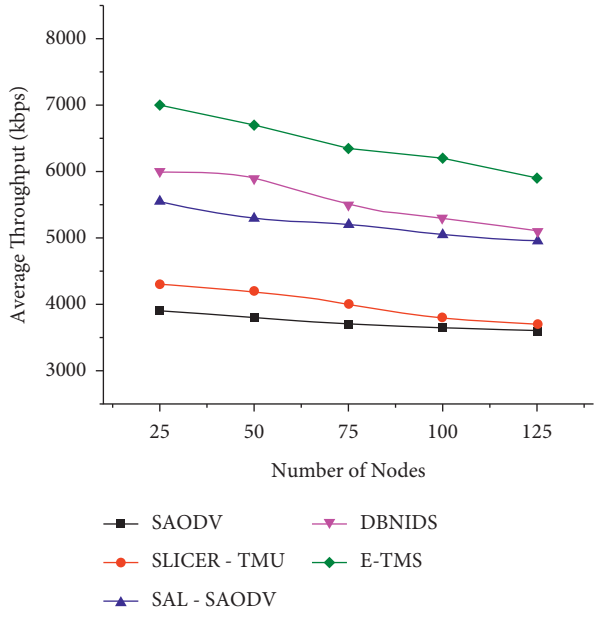

Figure 2: Average throughput (scenario 1).

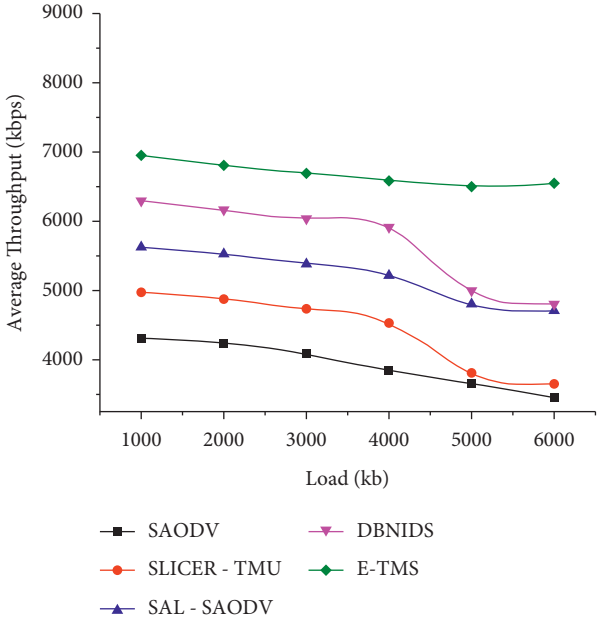

Figure 3: Average throughput (scenario 2).

network field, the direction of the signals cannot be predicted. Thus, the antenna must have the ability to accept the signals in 3600. Thus, the simulation environment uses an omnidirectional antenna in the physical layer. In real-time systems, the use of an omnidirectional antenna increases the possibility of receiving interferences from all directions. Due to this reason, performance degradation may be experienced in real-time systems.

The experimental setup examines the performance of the proposed method in two different aspects. Initially, the performance metrics have been calculated with respect to varying network load (as mentioned in scenario 2 under Table 3). Further, the evaluation proceeds with a constant network load for a different number of nodes in the same network field (as mentioned in scenario 1 under Table 3). The second evaluation environment has been introduced to study the behavioral changes of E-TMS under different network conditions. In order to compare the obtained results, the works, SAODV [18], SLICER-TMU [19], SAL-SAODV [12], and DBNIDS [13] have also been evaluated under the same network conditions. The efficiency has been evaluated based on average throughput, network resilience, and packet delivery ratio (PDR) [20–22]. The throughput can be defined as the number of successful receptions during a stipulated interval. The resilience value is a ratio of unsuccessful packet deliveries and the number of initiations. It

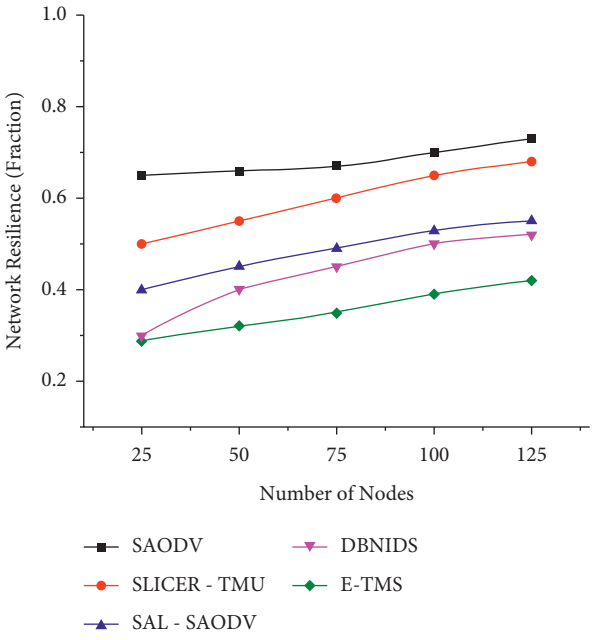

Figure 4: Network resilience (scenario 1).

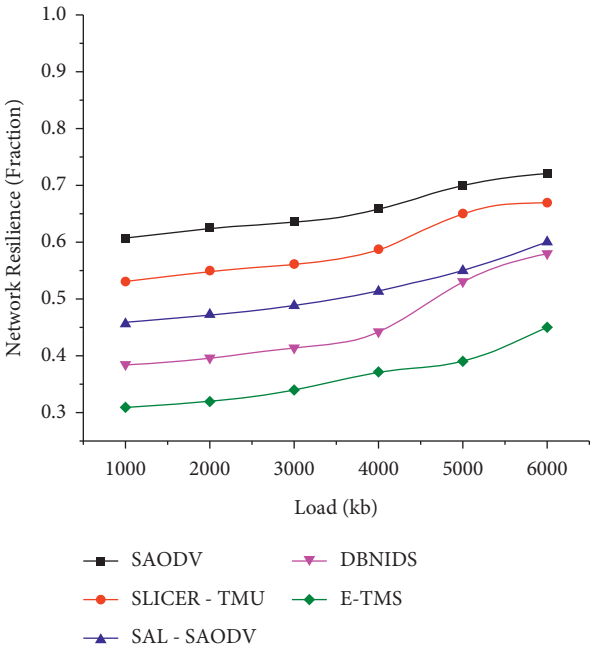

Figure 5: Network resilience (scenario 2).

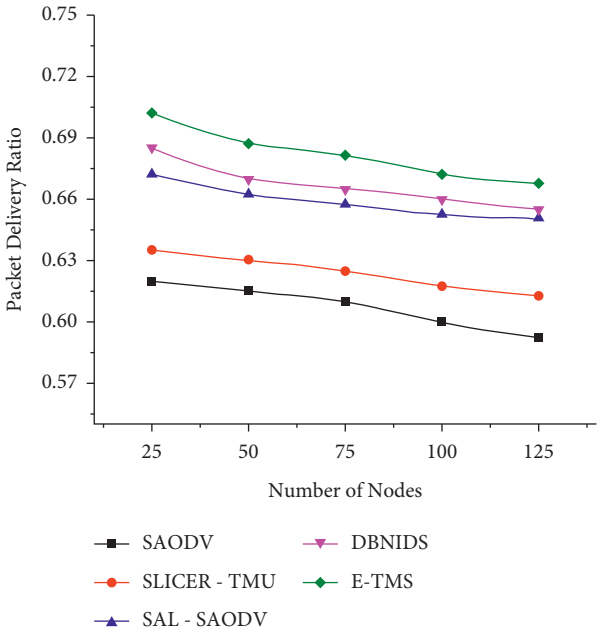

Figure 6: Packet delivery ratio (scenario 1).

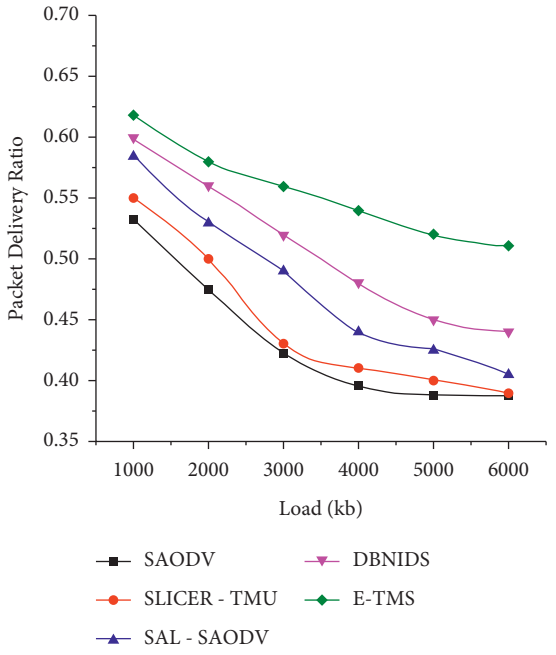

Figure 7: Packet delivery ratio (scenario 2).

gives the exact measure of unsuccessful packet delivery attempts. PDR is the measure of successful packet deliveries with respect to the total communication initiations in a stipulated time interval.

The average throughput under the varying number of nodes and varying network load has been evaluated and plotted in Figures 2 and 3 It is the count of successfully received packets at the receiver side. E-TMS could achieve better throughput by the proper identification of malicious nodes in both scenarios.

Figures 4 and 5 represent the network resilience assessed in both scenarios. E-TMS utilizes the features of Eigenvector for identifying the malicious trust data contributions during the trust aggregation process. The proper isolation of compromised entities will avoid the chances of having unsuccessful communication links. The lower resilience of E-TMS indicates that only a minimal number of compromised communications has been experienced during the assessed time interval.

The packet delivery ratio (PDR) based comparison under scenario 1 has been plotted in Figure 6. A better packet

Table 4: Analysis of proposed work.

Work name	Significance	Methodology for identifying the malicious trustdata contributions	Methodology	Significance/limitations
E-TMS	(i) Eigenvector-based approach for eliminating the malicious contributions	Present	(i) Malicious free aggregated trust value evaluation	(i) Two-level trust evaluation approach
SAODV [18]	(i) Resistant toward routing attacks	Nil	(i) Enhancement of path determination	(i) Introduced only to secure AODV
SLICER-TMU [19]	(i) Prevention of identity-based attacks	Nil	(i) Secure authentication mechanism	(i) Vulnerable to malicious trust contributions
SAL-SAODV [12]	(i) Power-aware approach	Nil	(i) Architectural enhancement	(i) Fog-based approach
DBNIDS [13]	(i) Malicious attack detection	Nil	(i) Deep belief neural network-based approach	(i) Method accepts trust data contributions from both malicious as well as legitimate nodes

delivery ratio can be achieved only when the network becomes malicious-free. A good trust management system can ensure the trustworthiness of the network. The proposed E-TMS experiences a linear decrease in PDR under the given network conditions. But, it could maintain higher PDR by the incorporation of a two-level event-based trust assessment mechanism.

Figure 7 shows the PDR of 5 works based on the varying network load. The network load has been increased to 6000 kb. E-TMS maintains a stable packet delivery ratio even with the higher load. By incorporating a good load balancing mechanism, the network can withstand the burden of a higher load. But, the attacks from maliciously compromised nodes will destroy the harmony between the increased load and packet delivery. E-TMS could ensure a good PRD count by removing all the impurities from the network. Further findings of E-TMS over the existing works have been included in Table 4.

The existing approaches for trust data aggregation fail to identify the malicious contributions. Such contributions have the capacity to mislead the network [23–29] if the network does not possess an intelligent approach to identify the same. The proposed E-TMS approach has the ability to remove malicious contributions. Thus, it could outperform the existing approaches in identifying the maliciously compromised nodes. The experimental result justifies the above statement.

6. Conclusion

The proposed trust management system, E-TMS, addresses the issues associated with Direct Trust Management Systems. Both legitimate and malicious nodes will contribute the trust data about their neighboring nodes. The aggregated trust value about a node may become inaccurate due to the presence of malicious contributions. E-TMS uses an Eigenvector-based approach for eliminating the malicious contributions while aggregating the individual trust contributions about a node. Rather than completely depending on the direct assessment, E-TMS possesses a two-level trust evaluation approach by considering both direct trust and indirect trust. As per the experimental results, E-TMS could outperform other existing trust management systems by the proper identification and elimination of malicious contributions, emerging from maliciously compromised nodes. In the future, a real-time trust management system can be developed using a machine learning system.

Data Availability

All the required data used to support the findings of the study are available within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of the paper.

Funding

This project is funded by King Saud University, Riyadh, Saudi Arabia.

Acknowledgments

Research Supporting Project number (RSP-2021/323), King Saud University, Riyadh, Saudi Arabia.

References

[1] B. Li, Q. He, F. Chen, H. Jin, Y. Xiang, and Y. Yang, "Auditing cache data integrity in the edge computing environment," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 5, pp. 1210–1223, 2021.

[2] T. Qiu, J. Chi, X. Zhou, Z. Ning, M. Atiquzzaman, and D. O. Wu, "Edge computing in industrial internet of things: architecture, advances and challenges," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2462–2488, 2020.

[3] P. Akaber, B. Moussa, M. Ghafouri, R. Atallah, B. L. Agba, and C. M. Assi, "CASeS: concurrent contingency analysis-based security metric deployment for the smart grid," *IEEE*

Transactions on Smart Grid, vol. 11, no. 3, pp. 2676–2687, 2020.

[4] X. Jia, D. He, N. Kumar, and K.-K. R. Choo, "A provably secure and efficient identity-based anonymous authentication scheme for mobile edge computing," *IEEE Systems Journal*, vol. 14, no. 1, pp. 560–571, 2020.

[5] B. Hajimirzaei and N. J. Navimipour, "Intrusion detection for cloud computing using neural networks and artificial bee colony optimization algorithm," *ICT Express*, vol. 5, no. 1, pp. 56–59, 2019.

[6] A. Shenfield, D. Day, and A. Ayesh, "Intelligent intrusion detection systems using artificial neural networks," *ICT Express*, vol. 4, no. 2, pp. 95–99, 2018.

[7] K. Cao, S. Hu, Y. Shi, A. Colombo, S. Karnouskos, and X. Li, "A survey on edge and edge-cloud computing assisted cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 11, pp. 7806–7819, 2021.

[8] S. Hameed, S. A. Shah, Q. S. Saeed et al., "A scalable key and trust management solution for IoT sensors using SDN and blockchain technology," *IEEE Sensors Journal*, vol. 21, no. 6, pp. 8716–8733, 2021.

[9] B. Wang, M. Li, X. Jin, and C. Guo, "A reliable IoT edge computing trust management mechanism for smart cities," *IEEE Access*, vol. 8, pp. 46373–46399, 2020.

[10] S. A. ElRahman and A. S. Alluhaidan, "Blockchain technology and IoT-edge framework for sharing healthcare services," *Soft Computing*, vol. 25, no. 21, pp. 13753–13777, 2021.

[11] A. A. Adewuyi, H. Cheng, Q. Shi, J. Cao, A. MacDermott, and X. Wang, "CTRUST: a dynamic trust model for collaborative applications in the Internet of Things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5432–5445, 2019.

[12] W. Fang, W. Zhang, J. Xiao, Y. Yang, and W. Chen, "A source anonymity-based lightweight secure AODV protocol for fog-based MANET," *Sensors*, vol. 17, no. 6, p. 1421, 2017.

[13] S. Manimurugan, S. Al-Mutairi, M. M. Aborokbah, N. Chilamkurti, S. Ganesan, and R. Patan, "Effective attack detection in internet of medical things smart environment using a deep belief neural network," *IEEE Access*, vol. 8, pp. 77396–77404, 2020.

[14] B. Li, Q. He, F. Chen et al., "Cooperative assurance of cache data integrity for mobile edge computing," *IEEE Transactions on Information Forensics and Security*, vol. 16, no. 2021, pp. 4648–4662, 2021.

[15] X. Ding, R. Lv, X. Pang et al., "Privacy-preserving task allocation for edge computing-based mobile crowdsensing," *Computers & Electrical Engineering*, vol. 97, p. 107528, 2022.

[16] H. Liu, P. Zhang, G. Pu, T. Yang, S. Maharjan, and Y. Zhang, "Blockchain empowered cooperative authentication with data traceability in vehicular edge computing," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4221–4232, 2020.

[17] Y. Wu, P. Tian, Y. Cao, L. Ge, and W. Yu, "Edge computing-based mobile object tracking in internet of things," *High-Confidence Computing*, vol. 2, no. 1, p. 100045, 2022.

[18] S. Lu, L. Li, K.-Y. Lam, and L. Jia, "SAODV: a MANET routing protocol that can withstand black hole attack," in *Proceedings of the international Conference on Computational Intelligence and Security*, vol. 2, pp. 421–425, Beijing, China, 2009.

[19] F. Qiu, F. Wu, and G. Chen, "Privacy and quality preserving multimedia data aggregation for participatory sensing systems," *IEEE Transactions on Mobile Computing*, vol. 14, no. 6, pp. 1287–1300, 2015.

[20] V. Gupta, K. C. Santosh, R. Arora, and T. Ciano, "Khairul shafee kalid and senthilkumar mohan. Socioeconomic impact due to COVID-19: an empirical assessment," *Information Processing & Management*, vol. 59, no. 2, 2022.

[21] V. S, J. A, S. R et al., "Multi-modal prediction of breast cancer using particle swarm optimization with non-dominating sorting," *International Journal of Distributed Sensor Networks*, vol. 16, no. 11, Article ID 155014772097150, 2020.

[22] A. K. Yadav, K. Singh, A. Ahmadian, S. Mohan, S. B. Hussain Shah, and W. S. Alnumay, "EMMM: energy-efficient mobility management model for context-aware transactions over mobile communicationficient mobility management model for context-aware transactions over mobile communication," *Sustainable Computing: Informatics and Systems*, vol. 30, Article ID 100499, 2021.

[23] P. Kumaresan, M. Prabukumar, and E. Barathkumar, "Smart home: energy measurement and analysis," in *Proceedings of the International Conference on Emerging Trends in Information Technology and Engineering (Ic-ETITE)*, pp. 1–5, IEEE, Vellore, India, Feb 2020.

[24] Y. B. Sundaresan, A. K. Jaiswal, and P. Kumaresan, "A low cost prototype for multiple access security system," *International Journal of Applied Engineering Research*, vol. 11, pp. 7907–7913, 2016.

[25] S. A. Iraj, "A Novel Method of motor imagery classification using eeg signal," *Artificial Intelligence in Medicine*, vol. 103, p. 101787, 2020.

[26] K. Yasoda, R. S. Ponmagal, K. S. Bhuvaneshwari, and K. Venkatachalam, "Automatic detection and classification of EEG artifacts using fuzzy kernel SVM and wavelet ICA (WICA)," *Soft Computing*, vol. 24, no. 21, pp. 16011–16019, 2020.

[27] P. Prabu, A. N. Ahmed, K. Venkatachalam, S. Nalini, and R. Manikandan, "Energy efficient data collection in sparse sensor networks using multiple Mobile Data Patrons," *Computers & Electrical Engineering*, vol. 87, Article ID 106778, 2020.

[28] V. R. Balaji, M. S, M. Rajesh Babu, M. Kowsigan, and V. K, "Combining statistical models using modified spectral subtraction method for embedded system," *Microprocessors and Microsystems*, vol. 73, Article ID 102957, 2020.

[29] A. C. J. Malar, M. Kowsigan, N. Krishnamoorthy, and S. E. K. Karthick, "Multi constraints applied energy efficient routing technique based on ant colony optimization used for disaster resilient location detection in mobile ad-hoc network," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 3, pp. 4007–4017, 2020.