IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 29 October 2023, accepted 20 November 2023, date of publication 30 November 2023,
date of current version 13 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3338550

RESEARCH ARTICLE

Security Provision by Using Detection and Prevention Methods to Ensure Trust in Edge-Based Smart City Networks

ABEER IFTIKHAR[1], KASHIF NASEER QURESHI[1,2], ALI A. ALTALBE[3,4], AND KHALID JAVEED[5], (Member, IEEE)

[1]Department of Computer Science, Bahria University, Islamabad 44000, Pakistan
[2]Department of Electronic and Computer Engineering, University of Limerick, Limerick V94-T9PX, Ireland
[3]Department of Computer Science, Prince Sattam bin Abdulaziz University, Al-Kharj 21589, Saudi Arabia
[4]Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 22252, Saudi Arabia
[5]Department of Computer Engineering, College of Computing and Informatics, University of Sharjah, Sharjah, United Arab Emirates

Corresponding author: Kashif Naseer Qureshi (kashifnaseer.qureshi@ul.ie)

The authors extend their appreciation to Prince Sattam bin Abdulaziz University for funding this research work through the project number (PSAU/2023/01/ 224252).

ABSTRACT The concept of smart cities is unique where the intelligent information technologies have used to form a heterogeneous with the help of edge and fog networks. Edge Computing (EC) has gained attention due to its distributive nature that brings computing resources closer to user devices for fast data communications. These networks are vulnerable to a variety of cyber threats due open network design and architecture and a lack of trusted computing resources. The Denial of Service (DoS) and Sybil attacks are common in these networks and cause of services degradation. Whereas in a Sybil attack, a node illegitimately claims multiple identities and escalate and launch more attacks in the networks. Identification and detection of malicious nodes and evaluating their trustworthiness is a challenging task. This paper proposes a novel Assertive Trust-based Efficient Technique (ATBET) for edge-based smart city networks by calculating the direct, indirect, penalized and cumulative trust. The trust is evaluated by using packet drop rate and the penalized over the transmission paths. The proposed solution detects the attacks and eliminates the malicious or selfish nodes from edge-based networks. The experiment results are conducted in simulation and compared proposed solution with state or the art solutions in terms of trustworthiness level, latency, packet delivery ratio, and network life span.

INDEX TERMS Sybil, trust, DoS, direct trust, indirect trust, cumulative trust, total trust, assertive trust, smart city networks, edge computing.

I. INTRODUCTION

Smart cities are based on new integrated Information and Communication Technologies (ICT) for more feasible, economical, and sustainable services. Cities have been changed from traditional communication systems to new advance integrated systems such as smart grids, smart homes, healthcare, intelligent transportation systems, and other smart computer technologies [1]. This concept has strengthened the data monitoring, managing, recognizing, and analyzing processes. All data communication processes in smart systems are dependent on local security measures by using appropriate security methods. These methods secure the devices from unauthorized access and direct or indirect harm [2]. Likewise, Edge computing (EC) networks are used to enhance computational capabilities and provide cost-effective and convenient services for the users. EC entails faster and greater data collection and provides better services for the users. Companies have adopted EC networks due to the provision of resources at the edge for quick processing. The EC networks also organize 75% of data at the edge side with low latency services and reduce the computational stresses of cloud servers [3]. Due

The associate editor coordinating the review of this manuscript and approving it for publication was Yassine Maleh.

to several new security threats, EC devices are vulnerable and compromised due to insufficient design and a lack of trusted computing methods and lead to several data leaks, rogue infrastructure, software and ICT based attacks [3]. The basic purpose of any security mechanism is to detect, prevent, and recover networks from malicious attacks. The security mechanism must fulfill the security requirements in terms of Confidentiality, Integrity, and Availability (CIA). The dynamics of EC-based smart city networks possess new requirements and considerations for network security [4].

Dijkstra's algorithm is used for pathfinding where nodes transfer their data to neighbor nodes and considering them trustworthy [5]. This type of solution reduces network delay, increases network lifetime, and improve energy efficiency. However, the weakness of these solutions is that they possess a high false positive rate and low detection accuracy rate for malicious nodes. The trust calculation is done through the trust metrics which depend on the threshold and identification of the secure paths. Most of the time, the solutions do not consider energy consumption and are prone to a high rate of Distributed Denial of Service (DDoS) attacks [6]. In DDoS, an attacker takes control of a tenant's virtual machine and disables another's web server [7]. These types of solutions are also unable to detect and eliminate multiple malicious nodes in one iteration which makes it vulnerable to attacks. These limitations lead to high computational complexity and bandwidth consumption. The existing networks have faced various challenges such as more computational power, utilizing resources in bulk, and consumption of high communication and storage resources. There is difficulty in detecting, identifying, and recovering malicious nodes and evaluating the trust among nodes for secure data transmission.

Sudden performance degradation of edge devices causes system collapse and makes the services inaccessible, leading to information breaches. These networks face various other external attacks like Sybil, impersonation, reflection, and physical attacks [5]. Malicious nodes in the network vary their behavior from time to time. It drops data packets and sometimes generates higher data delivery rates [8]. In a Sybil attack, an adversary tries to make a large number of nodes which operate as different nodes and may or may not be produced randomly [9]. To address these issues, there is a requirement to build a safe, resilient, and detection-efficient technique for dealing with malicious nodes in the network against cyber-attacks. Existing trust evaluation and security solutions are not up to the standard to defend against recent internal and external threats because these models are adopted as monitory measures against probable internal/external attacks [8], [10], [11]. The security-based evaluation approaches must provide dependability and usability. If there is a hostile node, it may launch more attacks to render the network vulnerable. To improve the node's degree of confidence in the network, a trust evaluation strategy is necessary. The existing solutions have suffered with high false positive rate and low detection accuracy rate for malicious nodes. The existing solutions also suffered with complexity issues due to intricate algorithms and policies which are challenging for administrators to configure troubleshoot. The trust based mechanism consume significant computational and network resources especially when the malicious activities exist in networks and lead to high operational costs and potential bottlenecks in network performance.

This paper proposes an Assertive Trust-based Efficient Technique (ATBET) to handle the identification, detection, and separation of malicious and selfish nodes from trustworthy nodes under the DoS/DDoS, Sybil, On-Off, Bad Mouthing attacks and computes trust evaluation metrics. The proposed solution is using assertive trust mechanism that detects more than one cyber attack and also the malicious nodes prevailing in the network in one iteration. It also detects and notifies more than one category of malicious nodes like Sybil and DoS during performing the registration in the network and also during data processing. The proposed technique applies complex trust tentacles to achieve better data threat detection accuracy and achieve the best worthwhile path. It increases the impact of the secure channels and enhances the security and Quality of Service (QoS) provisions in the smart city scenario. The other objectives of this paper are as follows:

- To develop a trust-based mechanism for trust evaluation of edge devices by identifying, notifying, and locating the malicious nodes.
- To distinguish the trust in inter-edge devices scenario using direct, indirect, penalized direct trust, cumulative trust, and total trust by using the trust threshold.
- To detect of malicious and selfish nodes from trustworthy nodes under the DoS/DDoS, Sybil, On-Off, Bad Mouthing attacks and computes trust evaluation metrics.

The rest of the paper is organized as follows: Section II, discusses the literature review to find the problem background. The design and development phases of the proposed solution are discussed in Section III. Section III-A3.a contains discusses the findings and discussion. The paper concludes with future direction.

II. RELATED WORK

Authors in [12], evaluated the nodes' reliability and trustworthiness in a distributive manner. The authors also proposed a recommendation-based trust model to investigate the nodes and share their trust tables regularly in the network. Node's trust values are obtained from neighbor nodes with a lower trust score. In an Internet of Things (IoT) network, a node updates trust scores by adding past trust scores to filtered trust weights and comparing the results to the Direct Trust (DT) value of the evaluating node. Based on the trust levels in the surrounding area, it relies on weight recommendations for trust evaluation to distinguish between malicious and honest nodes. It uses a technique that matches the examining node suggestion while seeking out differences between these recommendations, which is vulnerable to concerted attacks. This

strategy uses a theory-based trust detection methodology in the cloud, and it employs weight scores based on the service requests of adjacent nodes. Its advantage is that it formulates a generic framework for trust management that evaluates participating nodes' trustworthiness. However, the communication objects are avoided in diverse networks in this technique. Perhaps the approach is kept unverified against scalability to heterogeneous environments. It is unsuitable for EC networks, especially where mobility is required. It has moderate detection accuracy but has a loophole of detecting only a few types of IoT internal/external attacks such as Bad and Good Mouthing, DoS, Selected Forwarding, and On-Off. However, no external attacks are considered.

Authors in [13], proposed a remote attestation technique that depends on a trust mechanism-based cloud to certify network authenticity and enable credible access management among devices. This kind of technology ensures the node's credibility during data transmission over the internet in a secure network environment by enabling dependable connections between terminals and commanding server nodes. It tells the verifying node about the evaluator node's reliability. After receiving it, the authenticating node checks the status to make sure the sender is reliable and to see if the pertinent nodes adhere to the communication standards. This technique counters the Replay and On-Key attacks using authentication and identity-based methods. This technique is scalable and context-oriented to calculate trust level. The nodes intentionally behaving maliciously are used for calculating trust levels. This system improved the objectivity compared to threshold models, but it increased the overhead of the system.

Authors in [5], proposed a Activation Function-Based Trusted Neighbor Selection (AF-TNS), to identify malicious nodes by using dual and single-based link technologies. Nodes with dual links are more dependable, trustworthy, and secure. It emphasized trust by calculating trust using weights to identify malicious nodes. To accomplish this, each sensor node records its local vicinity and sends the results to the cluster for data aggregation. Then, using the majority voting mechanism, the trustworthiness of corresponding readings is measured as weights for significantly detecting hostile or selfish nodes. This solution is unsuitable for real-time or dynamic networks due to its single function.

The quantitative Trust Assessment (QTA) technique was proposed in [14] by using a Bayesian method. Trust measurements used are DT rate, statistical trust rate, and suggestion-based trust rate. Four phases are used to determine trust to identify and remove the malicious nodes from the network. Every node has the experience, knowledge, and recommendation to calculate the trust score and then permits the nodes in IoT-based networks. A Bayesian-based technique is employed in the parent node for trust calculation. This technique counters generic cyber-attacks along with direct attacks. This technique is scalable for adopting more IoT devices and more data it is energy efficient and suitable for the EC networks. However, it does not perform well on dynamic and real-time trust analysis. Hence it can only be performed in a static environment. It possesses high false rates and also has a low detection rate in real-time analysis.

Authors in [15], presented the RealAlert policy-based sensing technique for IoT-based networks. This technique evaluates the node's trustworthiness by using inconsistent network knowledge and contextual information. The trustworthiness of various devices in various circumstances is assessed by using policy foundation rules. Any new devices that appear with distinguishing features, bearing distinctive characters, or connecting differently are typically interpreted as an attack by a malicious or self-centered user who is maintaining an obsolete policy. It uses an approach based on network policy to prevent bad-mouthing and On-off attacks on IoT networks. This three-phased scheme detects malicious nodes that cannot provide services or computing resources to other network nodes. It only considers direct recommendations for trust calculation. It does not consider indirect recommendations and the context of service for calculating trust scores. However, this solution produces high false-positive rates and high latency rates.

Authors in [16], proposed an active trust acquisition mechanism to identify nodes' credibility and data collection reliability at a low cost. The receiver node sends information back to the sender node to confirm whether the packet is received or not. Information of multiple nodes at the same time reduces energy costs. Encoding of the verification explains which packet is received. The feedback mechanism of data routing increases trust evolution accurately and is swift. It is a robust technique that detects malicious entities having a higher deception prevention rate. It is a robust technique that quickly detects malicious nodes and has a high deception prevention of malicious node behavior. It updates latency due to Proof-of-Work and Proof-of-Stack. It counters black hole attacks only. However, this solution has low traffic prediction and detection accuracy not reported to the central node.

Authors in [17], proposed a Cipher text-stealing (in the case of any brief final layout data block) and Error Control Mechanism (LETCE), solution. In this solution, man-in-the-middle attacks related to unauthorized access are dealt with by using AES encryption with an error-control algorithm. Data is divided into blocks using Extended Tweaked Block Chaining (X-TBC) and each block is encrypted. X-TBC has a recursive nature, and each block depends on the previous block for encryption. The original layout is deleted from the computer if an attacker gets access to the computer; only an encrypted graphical design systems file is found. Only parties with the secret key can decrypt to attain the original graphical design systems file. It uses an activation technique with an error control mechanism. It exposes malicious nodes based on authentication-based trust values and, further, after identifying such isolates them over a network. However, a single activation function calculates trust levels in the unidirectional

communication network. It does not employ the data aggregation method for trust calculation and leads to high false detection rates and low accuracy detection.

Authors in [10], suggested a useful trust assessment technique named the Belief-based Trust Evaluation Mechanism (BTEM). It reduces internal security threats in Wireless Sensor Networks (WSN) by raising the bar for trustworthiness and dependability in data communication. The main factor considered when evaluating each communicating node is to check the effectiveness of malicious node detection. Each node uses more energy during the trust calculation method since this protocol focuses on high communication success rates. Eliminating rogue nodes and identifying DoS, defamation, and on-off attacks creates a safe connection between nodes. Values for Direct Trust (DT) Indirect Trust (IDT) and Recommended Trust (RT) are computed by using Bayesian estimation. The collected data is evaluated before identifying the malicious nodes and ambiguous data. The proposed framework uses three modules: firstly, the traffic monitoring module perceives the neighboring node based on its forwarding behavior by sending requests and response packets. Secondly, the trust evaluation module calculates DT and RT based on previous node interactions. Thirdly, the decision-making module compares the trust level of the nodes to the threshold values and excludes the malicious nodes. Simulation results showed a decrease in the delay and reduction of data throughput in detecting malicious nodes by using BTEM.

Authors in [18], proposed the trust mechanism routing protocol for Low power and Lousy Networks (LLN). The CTrust-RPL model is based on three layers: device, sink, and control. The device layer comprises sensor nodes and actuators. Nodes are not only sensing data from surrounding but also sensing data from neighboring devices in idle time and checking if the node in forwarding received data correctly and timely and marking it as a positive or negative node according to its behavior. The sink gets data from the device control plane and forwards it without processing. The device layer is provided with trust values obtained from the control layer. This layer's primary function is to propagate trust scores. The control layer calculates, accumulates, and updates trust values. Complex calculations of trust assessment are done to minimize computational, memory, and energy overload. This technique is resilient to bad-mouthing and advanced attacks in the IoT. It performs well when 40% of malicious nodes exist in the network, and it isolates them and gets disconnected through the sink node. It is efficient and has a high detection rate. However, once malicious nodes increase the precision & recall scores it cannot detect the node as a malicious node. It has high computational complexity and leads to more energy consumption issues.

The authors in [8], presented a novel Cumulative Trust (CT) and proposed an Analysis-based Economic Technique (CTBET) by focusing on multiple aspects of implementation and governance of safety in edge-based IoT networks. It is based on CT, DT, and IDT values of the available channels between the sender and receiver nodes. It calculates direct and IDT from corresponding nodes after focusing on packet transfer and drop rate. It imposes proper procedures for implementing the trust mechanism to improve the node's security and data privacy. It is capable of handling On-Off, DoS, and Bad-Mouth attacks, as well as isolating malevolent nodes in the network. A threshold is set according to which the normal, hostile, and selfish nodes are detected. Malicious nodes are isolated from a path, updated the routing tables. System performance is assessed by using a simulator which proves the metric node's life span. The nodes became more reliable, and the data delivery ratio improved i.e. metrics level of trustworthiness and data delivery ratio have increased. Furthermore, the end-to-end delay is decreased. This solution has achieved high detection accuracy, high end-to-end life span, and cost-effectiveness. However, the security calculation is done through the trust metrics where the identification of the secure path is not considered and leads to energy consumption, high computational complexity, and too much bandwidth.

Authors in [19], presented a Blockchain (BC)-based Multi-mobile Code-driven Trust Mechanism (BMCTM). Fog Computing (FC) requires a distributed mechanism to assure data security, privacy, transactions, and trust. The BC technology provides a security solution for IoT systems. Consensus runs among fog nodes for adding a new communication node into a network or to detect a compromised and malicious node and isolate it. The proposed system is based on two main layers; the device layer containing sensor nodes that communicate to exchange data to accomplish the desired task and the fog layer comprising fog nodes responsible for managing sensor nodes. A subjective Logic Framework (SLF) is implemented to calculate trust value. Compromised nodes are restricted and cannot update or edit the trust value of a trusted node. It uses subjective logic to counter the Grey and Blackhole attacks. It adopts an Open-source distributed trust management simulator where data is provided in CSV format for detailed analysis. Despite a high detection rate and good QoS management, this technique cannot be applied to other trust management attacks like ballot stuffing, and badmouthing. Moreover, it only considers direct observations to compute trust and does not consider the QoS factors to give real-time trust scores.

Authors in [11], presented the Path Association-based Trust Management (PATM) scheme for the IoT networks. It is a trust management scheme that considers the history of the link in the routing path from source to destination while weighing the trust values of a given node. The node's trust value corresponded to the trust value of the link from another node to that node. A link is discouraged from being selected as part of a routing path if the history of the routing path selection suggests its active selection. To achieve this, the node DT value is weighted accordingly. The greater the number of link selections in the previous cycles, the lesser the weight given to the DT value of the corresponding node. It aims to prevent the trustworthy nodes from overloading

and protect them from potential DoS attacks. However, a high false detection rate is observed in the first iteration.

Authors in [20], proposed a LightTrust management solution for Industrial IoT networks. LightTrust uses a centralized trust agent to create and execute trust certificates that allow nodes to communicate for a specific period without the need for an execution trust calculation. To keep the current trust level of aggregation/propagation, the trust agent has additionally kept a trusted database. The system handles trust certificates without doing trust assessments to interchange services across devices. This approach quantified additional direct trust findings in terms of collaboration and compatibility. However, IDT is seen using trust recommendations. For small cell networks, this technique provides a distributed federated trust algorithm based on merge-splitting criteria. The trust connections between small cell networks are described by the definitions. This connection is made possible through a network of social trust, which is determined by determining the shortest path.

Authors in [21], proposed Trust2Vec as a trust management solution for extensive IoT networks. Large IoT systems can manage trust relationships with the system's help, and malicious device assaults may be minimized. It employs a network structure to establish trust between devices, and one of its most important phases involves to identifying the device and creating random walk algorithms. It uses trust connections in clusters to find malicious nodes. The fundamental innovation of the proposed system is a random-walk algorithm for navigating trust connections and a parallelization approach for attack detection. It uses a random-walk mobility technique to navigate trust connections and a parallelization mechanism to identify attacks. It is possible to expand the jobs by adding the TM for data entities. In this technique, the large-scale attacks against the trust launched by several hostile units are repelled by it. To recognize attacks like self-promotion and defamation, authors offered an embedding population identification approach that recognizes and blocks communities of damaging nodes.

A. DISCUSSION AND FINDINGS

Various security schemes and solutions, including cryptography, authentication, and trust evaluation mechanisms, provide a certain level of security. Still, they lack true identification of threats within the network due to weak design and high computation complexities. Most techniques [16], [17], [18], [19], [20], [21] do not emphasize DT and IDT based on true network factors and possess increased communication and computational costs. Hence, PATM [11], CTBET [8] and BTEM [10] are effective for data integrity, using DT and IDT methods considering essential network metrics serving as a foundation for the proposed trust evaluation scheme at edge nodes. This paper attempts to improve the security flaws of these studies and offer optimal robust and more secure solutions than these protocols. The solution will include an effective trust evaluation mechanism at edge nodes to ensure data integrity, usability, and secrecy by evaluating DT, IDT, CT, and AT based on real-time complexity considerations to identify selfish and malicious nodes. From problem investigation, it is learned that different researchers in the field of EC and Smart city networks promote the lifespan of edge nodes and provide countless security but almost all schemes are vulnerable to two or more attacks. These techniques discussed in the literature review face challenges as all require highly complicated computational power, utilize resources in bulk, and consume high computation/communication and storage resources to statically detect, identify, and recover malicious nodes while countering external attacks and trust evaluation, among nodes for secure data transmission against internal/external attacks. Edge nodes are small, lightweight, resource-restricted, and possess low computational power. Hence, such technical limitation makes them unsuitable for relying on several security mechanisms in trust and authentication. Despite being lightweight, some schemes offer static time solutions but compromise on the intensity of security. Hence, a real-time, advanced, and intelligent system to tackle these threats/attacks and protect smart city networks through the implementation of trust evaluation technique for inter-node communication in edge-based smart cities to maintain a steadiness between resource efficiency, power computation, dynamicity, and security standards.

III. PROPOSED ASSERTIVE TRUST-BASED TECHNIQUE

This section elaborates proposed Assertive Trust-based Efficient Technique (ATBET) to handle the identification, detection, and separation of malicious and selfish nodes from trustworthy nodes under the attacks DoS/DDoS, Sybil, On-Off, Bad Mouthing and computes trust evaluation metrics. It calculates the assertive trust based on direct/indirect interactions, employs penalty to enhance trust, processes Total Trust (TT) to identify malicious /selfish nodes, and notifies other network nodes to update their routing tables and remove malicious nodes from the network.

A. COMPONENTS OF ATBET

ATBET is a trust management scheme that calculates a given node's trust values, considering the association history of its connectivity in the network's routing path from source to destination. The purpose of ATBET is not only to prevent the trustworthy nodes from being attacked and overloaded but also to protect them from potential DoS, DDoS, and Sybil attacks which makes them creating a flooded situation over the network and also a breach of sensitive information being transferred to the outside world, especially to hackers. The node's trust value corresponds to the trust value of the link from another node in the network to that node. A link is discouraged from being selected as part of a routing path if the history of the routing path selection suggests its active selection. The edge node's Direct Trust (DT), Indirect Trust (IDT), and Cumulative Trust (CT) are weighted accordingly. Each node checks its successor node for connection to the destination node by sending 'Hello' messages. In case of

no direct connection to a destination node, successor nodes search their successor nodes for the link with the destination and routing table maintained.

Figure 1 illustrates connectivity among edge nodes, from transmitter to receiver. nodes that are not linked to another node are unable to communicate directly. The nodes' trust values determine the routing choice from source to destination. Beta reputation-based systems calculate node trust value.

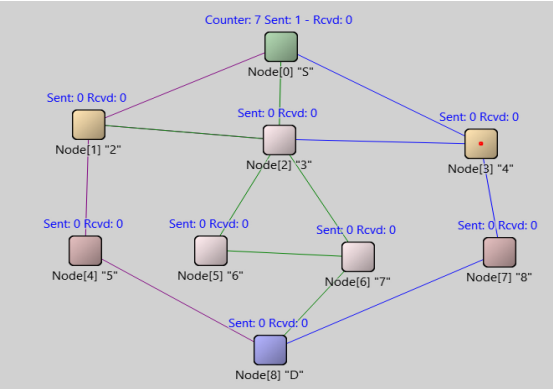

FIGURE 1. Network topology.

The probability density function of future events as a function of previous observation is known as reputation. Based on historical trust values, trust-based systems employ reputation models to calculate the predicted value and assess the node's belief in the other node. Due to the inference of previous knowledge in posterior behavior prediction, the beta reputation model effectively calculates the node's trust compared to the statistical models because of the inference of prior information in posterior behavior prediction. Reputation depends upon cooperative or non-cooperative transactions among nodes.

1) TIER 1: ATTACKING MODEL

Sybil, DoS, DDoS, On and Off, and the Bad Mouthing attacks are considered. In a DoS Attack,data in bulk is forwarded by malevolent and selfish nodes, putting a strain on the network resources handled [22]. Further, DoS is launched through power depletion, in which the intruder constantly demands packets from the nodes that intend to exhaust their battery life [23]. In a DDoS attack, a group of nodes act maliciously and block and jam the network. In a Sybil attack, a single node assumes many identities to control other nodes. An adversary's duplication of edge nodes is known as a node replication attack [24]. In a bad-mouthing attack, a node gives the network's neighboring nodes incorrect values and recommendations. Collecting IDT via DT of neighboring nodes is identified by communication parameters and mitigates this attack by monitoring packet loss and delaying rates, end-to-end latency, and trustworthiness level [25]. In a Replay attack, nodes replay old data obtained by any node, get a timestamp, and the nodes reject all messages arriving after a certain duration. This duration is a transmission time, the maximum time the node takes to send the data. In an On-Off attack, node behaviors are corrupted and provide unusually erroneous DT values to mislead the communicating nodes bearing malicious intent to create a network blockage. It is mitigated by employing network lifetime span, level of trustworthiness, time-oriented packet delays, data drop rate, and data rate characteristics. For miscellaneous like in conflicting behavior attacks, malevolent nodes behave inconsistently with other network nodes, causing other nodes to provide contradicting suggested trust levels for the targeted node.

TABLE 1. Various scales of trust measurement.

Value	Degree of trust
1	Complete Trust
0.9	Very High Trust
0.8	High Trust
0.7	High Medium Trust
0.6	Low Medium Trust
0.5	Low Medium Mistrust
0.4	High Medium Mistrust
0.3	High Mistrust
0.2	Very High Mistrust
0.1	Complete Mistrust
0	High Alert: No Trust

In this network, malicious, bad, and selfish nodes are kept static, capable of maligning the behavior of neighboring nodes within the range of communication. The attacking model avoids sending and forwarding data to the relevant nodes and occasionally creates data flooding commands with the bulk of messages or retransmits the same bulk of messages to the targeted nodes to cause network link failure. We assume that X(n) represents the node's behavior as a random variable. A malicious node is identified by frequent retransmission, flooding, and packet loss:

$$X(n) = \begin{cases} 1, & \textit{if } n \textit{ relays packet} \quad \text{where } n \epsilon N \\ 0, & \textit{if } n \textit{ drops packet} \\ -1, & \textit{if } n \textit{ compromise packet} \\ & 3 \text{ States (Drop/Data rate)} = -1/0/1 \end{cases} \tag{1}$$

Different scales are used for trust measurement as shown in Table 1 [26].

The values of edge nodes are calculated in each iteration of the trust evaluation mechanism in which the communicating nodes assign the values to trust variables. During trust evaluation, if trustworthy communication is detected with collaborating nodes then depending upon the level of trustworthiness a trust score from 0.5 to 1 is assigned where 1 is notified as a fully trustworthy interface. Correspondingly, once a malicious node is detected in the neighboring nodes, the trust-seeking edge node assigns a trust score between 0 −

0.4 where 0 is considered a fully malicious interface whereas 0.4 donates a low middle level of mistrust. Trust scores are assigned to trust variables that are considered for the calculation of direct, indirect, penalized, cumulative, and total edge node trust over another corresponding and collaborative edge node. Therefore, using total trust computation, malicious and non-malicious nodes are classified based on the interactions between each internal and exterior edge node [26].

a: NODE RELAYING OR RETRANSMISSION

Nodes that possess forwarding and relaying rates not exceeding 0.5 are highly dependable when it comes to communication, whereas nodes crossing this threshold may be viewed as malevolent or selfish causing their status to switch to 1.

b: NODE DROPPING PACKET

A node can be considered malevolent or selfish when its drop rate exceeds 0.5 and its corresponding X(n) state is set to 0. Communication can be done safely with nodes whose drop rates are at or below 0.5 since they are deemed dependable.

Attack models for trust assessment are used to identify malicious, compromised, selfish nodes capable of manufacturing and flooding bogus traffic throughout the network and generating inaccurate trust values for malicious nodes. Trust models are invented and built to improve and update application security. Nodes, trust values, data items, and the attacking and evaluation model are constantly threatened by adopting attack types and related protection resiliency.

2) TIER 2: TRUST AND BEHAVIOR ANALYSIS MODEL

This tier deals with monitoring the data packet transmission/reception on multiple nodes. This tier is also recording the behavior of nodes in terms of data rate/throughput, as well as packet loss rate. Each node has a packet profiler that records each node's behavior and crucial Information Packet Traffic (IPT) information. Furthermore, the examination of the trustworthy data transfer path. Here, direct and IDT are computed, and the two are combined to form forceful trust i.e. $AT_{(IDT,DT,CT,TT)}$ BET.

a: ESTIMATION OF THE NODE'S TRUST

The trustworthiness of corresponding and neighboring nodes is estimated based on data rate and packet drop rate in a real-time domain. It evaluates the CT of participating nodes by considering the node's IDT as well as DT values, the record and profile of IPT created and stored at each node, and shared among each other in a specific time interval. This is composed of various types of packet information such as several Packets Sent (PS) from the transmitting node, Time taken during Packets Sent (TPS) from the transmitting node, and several Packets Received (PR) at the receiving node. The trust levels of nodes influence routing choices from the source to the destination. Beta reputation-based systems calculate a node's trust. The probability density function of future events as a function of historical observation is called reputation (R). Predicted value and node's belief on other nodes are calculated by reputation models in trust-based systems by using past trust values. The beta reputation model effectively calculates the trust value of the nodes compared to the simple statistical models because of the inference of prior information in posterior behavior prediction. R depends upon cooperative or non-cooperative transactions among the nodes. Network topology is shown in Figure 2.

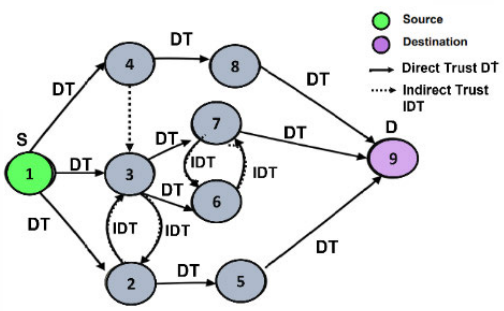

FIGURE 2. Initial stage for ATBET.

Trust establishes the reputation through an assessment based on the network traffic parameters. Trust is a belief or confidence in the reliability, integrity, and honesty of a person, organization, or entity. It is the willingness to rely on someone's actions, decisions, and promises with the expectation that they will act responsibly and dependably. Reputation is the collective opinion or perception that others have about a person, organization, or entity based on their past actions, behavior, and interactions. It is an evaluation of the trustworthiness, credibility, and competence of that entity. Overall, trust and reputation are intertwined, and a positive reputation is often the result of earning and maintaining trust through reliable and ethical actions. Strong trustworthiness fosters a positive perception and can lead to various benefits, while a lack of trust can be detrimental to an individual's or organization's standing in the eyes of others. Trust estimated based on the QoS parameters identifies the real-time trust established hence the trust creates reputation through parametric estimation of any node through direct and indirect estimations.

The risk of cooperation and noncooperation amongst nodes varies from 0 to 1. Value 1 symbolizes the most cooperative behavior, though value 0 represents the minor collective behavior. A Bayesian framework revises transaction ratings [27]. A Beta probability contains a density function representing binary event distributions of likelihood. It offers a sound mathematical basis for integrating comments and representing the system nodes' reputation scores. The advantage of the Beta reputation model is flexibility, simplicity, and its foundation in the statistics theory. A and β represent the two binary functions. The value of beta potability is always between 0 and 1 [11]. Reputation (C) or Probability (p) of the behavior of node B for node A is expressed as $R_{ab} = P = \text{Beta}_{(\alpha+1,\beta+1)}$.

In Equation 2, the Partial Trust (PAT) trust value of node B is calculated by node A as equal to E(p) / E(R_{ab}) or expectation of probability of behavior. A is the number of cooperative transactions, and β is the number of non-cooperative transactions between the nodes. Every node in the system saves the number of cooperative and non-cooperative attributes of a reputation for all its successor nodes from source to destination in its memory for each interaction [6], [11].

$$PAT_{ab} = E(p) = E(\text{Rab})\,\text{Beta}(\alpha+1, \beta+1) = \frac{\alpha+1}{\alpha+\beta+1} + \frac{(\alpha-1)}{\alpha^2+\beta^2+2} \quad (2)$$

DT is investigated and established between directly connected nodes from node S to 3 and S to 2, based on various network traffic metrics/analytical parameters between various destination and intermediate nodes. These analytical parameters are defined as:

a. ***PR (Packets Received)***: Total number of packets a recipient node receives.
b. ***PS (Packets Sent)***: Total number of packets a transmitting node sends.
c. ***TPR (Time of Reception of Packets)***: Duration of a receiving node's packet reception from a transmission node.
d. ***TPS (Time of Sending of Packets)***: Time taken to send or forward packets to the receiver.
e. ***PDR (Packets Drop Rate).*** The proportion of packets lost while transmitting from one node to another, i.e. from transmitter to receiver.
f. ***PDtR (Packets Data Rate)***. A total number of packets exchanged in a given period across a single node-to-node link.
g. ***Communication Trust (COT)***. The ratio is the expected value of the probability distribution describing the reputation of communication between two nodes.
h. ***Data Trust (DAT)***. The ratio is the probability distribution's expected value describing the data's reputation between two nodes.

Communication Trust (COT) depends upon cooperative or non-cooperative communication between two nodes. Attacks on communication, such as selective forwarding attacks, make the nodes act less cooperatively. Using the beta distribution, one may forecast cooperative and non-cooperative behavior between nodes in time T which is given in Equation 3. In this equation, αc is the number of cooperative communications and βc is the number of non-cooperative communications between nodes. The αc increments by 1, with each cooperative communication, and βc increments by 1, with each non-cooperative communication between nodes [6].

$$COT_{ab} = E(p) = E(\text{Beta}(\alpha+1, \beta+1)) = \frac{\alpha c+1}{\alpha c+\beta c+1} + \frac{(\alpha c-1)}{\alpha c^2+\beta c^2+2} \quad (3)$$

In Data Trust (DAT), if the nodes are operating in a normal scenario, the perceptual information in the neighborhood nodes stays constant and consistent. Data follows Gaussian distribution. Data attacks compromise the nodes by significantly differing data in neighborhood nodes. T-test is performed for each Δt in T time to check whether the data sequences of two nodes A and B are from the same population or not. The similarity of data is checked by giving a threshold value $\rho = 0.05$ in the T-test. If the result of the hypothesis is less than 0.05, then the data is significantly different in nodes. DAT is calculated by using beta distribution in Equation (4) [10], [11].

$$DAT_{ab} = E(p) = E(\text{Beta}(\alpha+1, \beta+1)) = \frac{\alpha d+1}{\alpha d+\beta d+1} + \frac{(\alpha d-1)}{\alpha d^2+\beta d^2+2} \quad (4)$$

In Equation 4, the αd is the total number of times data is not significantly different, and βd is the number of times data is quite different between nodes. αd increments by 1, each time the perceptual data is not significantly different between two nodes, and βd increments by 1. These factors are used to build trust and analytically evaluate the inter-node relationships' trustworthiness. Path Detector (PD) is a structure that keeps track of the node linkage entries. Each node's trust value is computed for determining DT by using path linkages from PDs. The packet information profiler analyzes and records analytical parameters and trust contributing parameters. The IPT data structure contains the true behavior based on the analytical parameters at each node. S assesses the trustworthiness of nodes 2, 3, and 4 based on their packet data rates and packet loss rates and then validates it by comparing the results to the data contained in the IPT. If the comparison is valid, the node is regarded as a trustworthy node. Next, the node determines the degree of reliability of the nodes that are directly connected to it. The alternate path is chosen once the first path has been completed.

b: CALCULATION OF DIRECT TRUST (DT)

Six factors are calculated during packet transmission and reception to evaluate each node's trustworthiness. In addition, an estimate is made of whether the node is likely to be malevolent, self-centered, or trustworthy. Threshold values i.e. TS and TR are maintained for each node, which is compared to if a node successfully sends all or most of the packets and the receiving node receives all or most of the packets. When the PS exceeds the TS and the PR exceeds the TR, nodes are deemed trustworthy, and their values are stored in the IPT. When processing trusted path detection, other nodes utilize these values to calculate or estimate the IDT based on the DT values of other adjacent nodes. Similarly, a node is labeled malevolent once the communicating node forwards packets to another node that is less in amount than the TS or TR threshold value. It implies that dropout packets are not being transmitted due to the transmitter node's ill intentions, or that erroneous information about the packet flow in the IPT has been recorded, indicating incorrect estimations of

the number of packets received and stored in the IPT. These indicators identify the transmitter node as a malicious node.

The packet sent and received times are considered by the formula used to determine the distance between two nodes (DT) [8], [11].

$$DT_{AB} = \left(\left| \frac{PR_{t-1} - PR_t}{PS_{t-1} - PS_t} - \frac{PDR_t \times TR_t}{PS_t \times TS_t} + \frac{PDR_{t-1} \times TR_{t-1}}{PS_{t-1} \times TS_{t-1}} \right| \right) \times (\mathrm{PAT} + \mathrm{COT} + \mathrm{DAT}) \quad (5)$$

where $\mathrm{PDR_t} = (\mathrm{PS_t} - \mathrm{PR_t})$, and $\mathrm{PDR_{t-1}} = (\mathrm{PS_{t-1}} - \mathrm{PR_{t-1}})$, and all the DTs are stored at each node and shared with the IPT. PDRt is the Packet Drop rate for the current iteration, whereas PDRt-1 is the Packet Drop Rate of the consecutive previous transaction. It controls the extreme drop rates in two different transactions, balances the DT calculation, and avoids the negative trend.

c: CALCULATION OF THE PENALIZED DIRECT TRUST (PDT)

Over usage and overburdening edge, nodes are susceptible to DoS/DDoS attacks. Hence, ATBET introduces a Penalizing Factor named Penalized Trust (PDT) in the DT values of the nodes. It encourages routing through alternative trustworthy nodes based on the link association in the selected routing path. It discourages and prevents a prolonged association of a particular path over an extended period. A Sybil suffered node creates multiple identities, and DoS suffered generates attacks on the trustworthy nodes based on the internode path association profile. Alternative paths are selected on historic comparative trust values of corresponding nodes. Penalization does not make nodes malicious. DT values of the nodes are not penalized below or equal to a threshold of PTH = 0.625. Equation 6 presents the relationship between DT and PDT [11].

$$\mathrm{PDT(n)} = \mathrm{DT(n)} \quad (6)$$

If node DT value <= 0.625 (PTH) PDT = DT of cycle (n)

If node DT value > 0.625 (PTH) Penalize or Regain PDT

Two cases determine DT penalization or regain in PDT:

Case 1 (Penalizing the DT): A DT of a particular node calculated by its predecessor node is penalized if its corresponding path is present in the selected routing path, from node S to D, in the previous cycle (n – 1). Penalization value equals the number of cycles a particular path chooses for routing between S and D. DT penalization [11] is given in Equation 7.

$$\mathrm{PDT(n)} = \mathrm{DT(n)} - \mathrm{K}\mu \quad (7)$$

PDT in Equation 7 is equal to DT minus Penalizing Factor $\mathrm{K}\mu$ in Equation 8. Penalizing Factor avoids overloading the trustworthy nodes. $\mathrm{K}\mu$ temporarily decreases the node's DT value to select different paths. K is initialized by 0. If the link exists in the selected routing path in the previous cycle, K increases by 1 in its last value each time of its presence. μ is the Penalizing Weight and has a constant value, which remains the same in the penalization process. If PDT becomes less than PTH = 0.625, then PDT is the value calculated in the previous cycle [8], as given in Equation 8.

$$\mathrm{PDT_{(n)}} = \mathrm{PDT(n-1)} \quad (8)$$

If DT is not penalized to the limit, PT reduces below PTH = 0.625 (as PDT is 75% of the CT value of the node ($0.75 \times 0.625 = 0.54$)). Equations (5) and (7) are used to avoid falsely changing the status of nodes in any case. The packet information profiler, IPTs, and PDs are updated accordingly.

Case 2 (Regaining the DT): Once the selected route path from S to D modifies and the link is not part of the previous cycle, nodes regain their trust values. Equation 7 is reused with a variance in that the value of K decreases to restore reduced DT values. In Penalizing Factor $\mathrm{K}\mu$, the value of K decreases by 1, from its previous value, in each cycle of the link's absence in the previously selected routing path. The minimum value K can achieve is 0. This is useful for regaining nodes' trust values and their re-selection in the routing path. The value of μ remains the same for regaining trust as it is done in penalizing trust. Both the packet information profiler and PDs are updated respectively.

d: CALCULATING THE INDIRECT TRUST (IDT)

The recommended trust for a subject node is gathered from its neighboring nodes to calculate IDT. IDT has the transitive property and is situated between two less associative nodes. The two nodes are transitively linked to recommender nodes. For other nodes, the DT of one node equates to the IDT. Due to two evident factors—first, a lack of knowledge about the node's behavior due to decreased communication among sensors, and second, the need to combine recommendations and DT scores to obtain a comprehensive trust score—recommender nodes may necessitate IDT evaluation. The advised trust [8], [26], i.e., IDT, is given in Equation 9.

$$IDT_R = \sum_{k=A}^{H} \binom{n}{k} PDT^H x \alpha^{n-k} x \beta^{n-k-1} \quad (9)$$

In Equation 9, n is the total number of nodes employed in the network, k is the penalty constant already used in Equation 7 and Equation 8, α is the number of cooperative transactions, and β is the number of non-cooperative transactions between the nodes. PDT is the DT calculated after scrutiny for penalty trust.

e: CALCULATION OF THE CUMULATIVE TRST (CT)

CT integrates trust calculated by the nodes' direct interaction and the recommender nodes' recommendations. CT ensures that the trust value of the nodes is not solely dependent on their direct interactions but also on the recommendations received from commonly connected nodes. After successful verification performed on $\mathrm{PDT_{S,3\ and}}$ $\mathrm{IDT_{S,3}}$ collectively, CT is calculated among nodes i.e. nodes S and 3 through the

TABLE 2. Direct/indirect paths between node S and D.

Path Details	
Path-1	S – 4 – 8 – D
Path-2	S – 3 – 7 – D
Path-3	S– 3 – 6 – 7 – D
Path-4	S – 2 – 5 – D
Path-5	S– 2 – 3 – 7 – D
Path-6	S– 2 – 3 – 6 – 7 – D
Path-7	S – 4 – 3 – 7 – D
Path-8	S– 4 – 3 – 6 – 7 – D
Path-9	S– 4 – 3 – 2 – 5 – D

formula given in Equation 10.

$$CT_{(IDT,PDT)}PDT_{S,3} = \frac{PDTS,3 + IDTS,3}{2} \quad (10)$$

The computed $CT_{(IDT,PDT)}PDT$ is updated in IPT and distributed to all nodes nearby, whether directly connected to the source node or not. The records at IPT have been updated for estimation and confirmation during path identification and selection, and all of these computations are being done for all likely paths connecting nodes S and D. This process is carried out up until node D when the CT is calculated. IPT updates CT distributed among the network nodes. Because it depends on the nodes' direct interaction, the trust that depends on the DT of the nodes carries more weight. Recommended Trust is unavoidable as all recommendations are used without filtering for holistic and accurate trust calculation. Nodes without recommender nodes must have PT equal to PTH = 0.625 (0.625∗0.8+0) = 0.5 for them to be normal. Various steps involved in calculating the five direct and four available indirect paths are illustrated in Table 2 and displayed jointly in Figure 3.

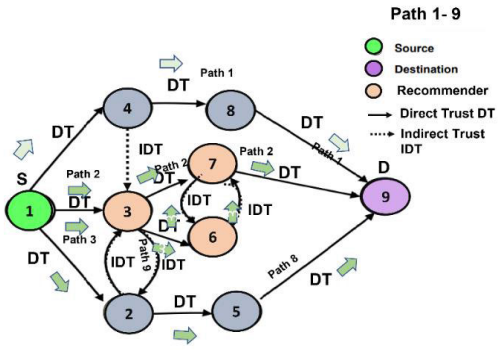

FIGURE 3. Path S – D for trust calculation.

f: CALCULATING THE TOTAL TRUST (TT)

Nodes in a network are susceptible to change, therefore the trust among nodes has to be re-evaluated in each cycle or iteration. In every time slot/cycle, the trust values of the nodes in a system are recalculated, and the previous trust value is also considered. Equation 11 presents TT [11] or the node's trust value in the cycle.

$$TT(n) = (\mu_{n-1}.TT(n-1)+\mu_n.CT(n)) + (\mu_{n-2}.TT(n-2) + \mu_{n-1}.CT(n-1))/2 \quad (11)$$

TT refers to the total trust value achieved by a node in consecutive running cycles. TT value in a cycle depends upon the TT of the consecutive two previous cycles and CT of the successive two cycles, including the current and last cycle. In Equation 11, n = 1, 2,…, k is the number of time slots per cycle. $\boldsymbol{\mu}_{n-1}$ is defined as the probability accuracy factor of the node's past iteration trust value, μ_n is a probability accuracy factor of the current CT value for the current cycle, and μ_{n-2} is the probability accuracy factor of the node's consecutive previous trust value, and μ_{n-1} is the probability accuracy factor of the successive last CT value for the current cycle. The importance of CT values in the consecutive two cycles starting from the current cycle is more than the TT or trust value of the previous successive cycles.

The probability accuracy factor of μ_{n-2} is θ, and μ_{n-1} is 1- θ. μ_n is 2- θ. θ is defined as an Aging Factor that is used to reduce and balance the previous trust values. The value of θ is 0.1, so the previous consecutive trust values possess a 10% and 20% probability accuracy factor, whereas the new and most recent trust has an 80% probability accuracy factor [11]. The following are considered and enable the detection of a malicious node in Figure 4:

If node TT value < 0.5	**node is malicious**	**Removal from the routing path**
If node TT value > 0.5	**node is trustworthy worthier**	**Take part in the path-selection procedure**

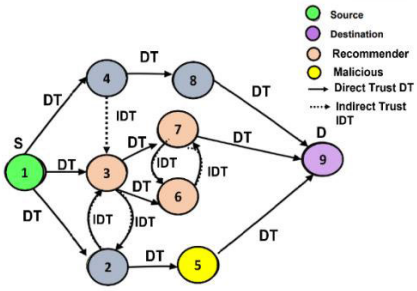

FIGURE 4. Malicious node detection.

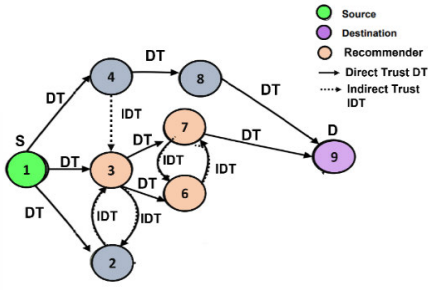

FIGURE 5. Malicious node removal.

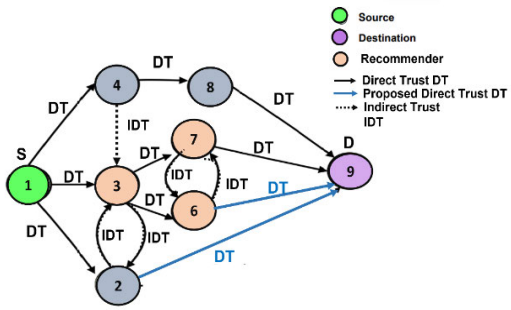

FIGURE 6. Balancing network.

In Figure 4, DoS/DDoS and Sybil attacks affect the performance of node 5. node 2 calculates the TT of node 5, which is less than 0.5. node 5 is marked as a malicious node, and removed from the current routing table. Node 5 removal from the network and the modified routing table are updated in the IPT and PD structure. In Figure 5, node 5 is removed only, but not it's any other connective nodes like node 2. Node 2 is not connected to D directly but it has a connection through node 3, node 6, or node 7 to D. For the next trust evaluation cycle, two possible direct connections to D will be attempted by nodes 2 and 6 (being a Recommender Node) which are not directly connected or linked with D. Further, two new routes are being updated in IPT and PD Structure and duly notified to all edge nodes shown in Figure 6 after completion of the current complete cycle and will be considered in next phase.

3) TIER 4: ADDITIVE TRUST METRIC MODEL

This module uses regression factors to determine identifiers and the ability to retain the more trustworthy path before rectifying it. This approach uses additive metrics to corroborate the projected probability of a path and performs risk analysis to ensure error-free and successful packet delivery. Therefore, the most optimized routing path is selected from S to D after isolating and filtering malicious nodes. Path selection depends upon the Additive Path Metric and Average Path Metric. These parameters are added together with certain weights to find the Composite Path Metric. The path bearing the minimum Composite Path Metric is selected for routing.

a: CALCULATING THE ADDITIVE METRICS

This submodule initially finds the most optimized routing path selected from node S to D after initial filtering/isolation of malicious nodes. The path selection depends upon metrics and further estimation of probability. The Composite Path Metric is initially evaluated by estimating the Additive Metric Count and Average Path Trust Metric. These parameters are added with certain probability accuracy factors to find a Composite Path Metric. The path with the minimum Composite Path Metric is selected for initial routing.

i) ADDITIVE METRIC COUNT

Additive Path Metric (APM) is measured to find a path with the nodes' minimum hop count and high trust value. Equation 12 presents the APM of the path as used in [11].

$$\text{APM} = \sum_{Sn-1}^{m} \sqrt{1/TT} \tag{12}$$

In Equation 12, the APM of a path is equal to the sum of the inverses of TT/trust values of the trustworthy nodes in a path. Sn = 1,2.... M. Here, Sn is the number of trustworthy nodes in a path ranging from 1 to m. Path with low trust value nodes or a greater number of nodes yields a higher APM value and is avoided for routing.

ii) AVERAGE PATH TRUST METRIC

An inverse of the average trust values of the nodes in a particular path is considered for the selection of the best path for routing. Average Path Trust Metric (APTM) is calculated as in Equation 13.

$$\text{APTM} = \frac{\sum_{Sn-1}^{m} \sqrt{TT}}{m} \tag{13}$$

This metric provides the inverse of the average TT/trust value of the trustworthy nodes m in the path through standard deviation. Such deviated average metric is considered so that a path with additional hop counts scoring the highest values of trust is taken into account for routing.

iii) ASSESSMENT OF COMPOSITE PATH

The assessment of Composite Path parameters (CptPM) is an aggregate sum of APM and APTM [11] as given in Equation 14.

$$\text{CptPM} = \sqrt{\varphi.\text{APM}\gamma.\ \text{APTM}} \tag{14}$$

The probable accuracy factor φ for APM and factor γ for APTM have an equal value of 0.5. The selected routing path in any cycle is the path with the minimum CptPM value. This nominated routing path has the highest trust value and lowest hop count. Trust values of the nodes in the system and CptPM values of the routing paths from node S to D are re-calculated per cycle. Paths are compared, and the path with the lowest CptPM value is nominated for routing.

b: VERIFICATION OF IDENTIFIED PATHS AND FILTRATION FOR NOMINATION

Furthermore, using regression factors, this module calculates identities and capabilities to retain a more trustworthy path and then corrects it. For error-free and successful packet transmission, this method can employ additive metrics to confirm the estimated path probability and perform risk analysis.

4) TIER 5: DECISION-MAKING MODEL

Records of packet profiler and further substantiated and verified results of an additive metrics module, including various additive metrics components and its cross verification, with

priority-wise nominated paths, are processed by this module to conclude the probability of participating nodes whether trust worthier, malicious, or selfish nodes. It compares the trust values of DT, PDT, IDT, CT, and then TT of all participating nodes deployed on various possible paths are analyzed, evaluated, and further verified while comparing with thresholds. Normally, trust values range from 0 to 1; It is considered trustworthy if the PDT-based TT is more than 0.5. If a node's trust value is equal to one, that node is regarded as the most trustworthy. If a predicted trust value is less than 0.5, such a node is considered selfish behavior that may lead to becoming a malicious node in the future; if less than 0.3, nodes are marked malicious; if equal to zero then the node is most malicious node possessing worst behavior, or producer of the most attacking packets and must be causing an unending data flooding situation on the network, and such nodes must be stopped functioning in the network and initially to be removed from the routing table.

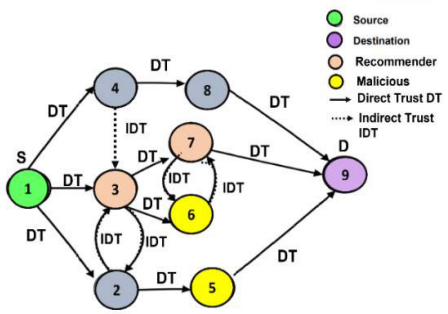

FIGURE 7. Nominating malicious node.

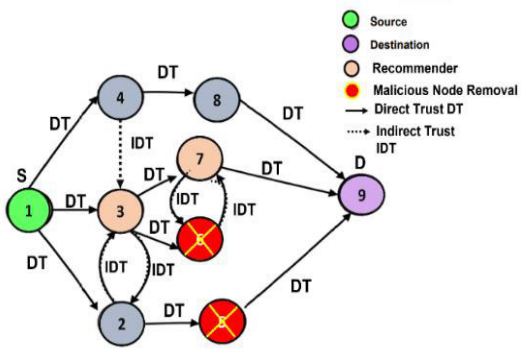

FIGURE 8. Elimination of the malicious nodes.

Further, this module detects the highest score TT and compares it with the full path threshold ranging from 0 to 1 after checking each node's value on the different paths between S and D. If TT is greater than or equal to 0.5, the path is regarded reliable; otherwise, it is considered unsuitable for transmission. Path trust values within IPT have been modified and are ready for selection in the following module. Malicious/misbehaving nodes are marked for partial or complete network shutdown, termination, or elimination, and IPT is updated for such bad and selfish nodes with consecutive lowest trust score, complete cessation of any connectivity with such node, including shutting down malicious or attacked or attacker node.

ATBET detects more than 1 malicious node i.e. multiple nodes using DT and IDT between two corresponding nodes i.e. transmitter and receiver. Additionally, after scrutiny from the DT and IDT calculations, the CT and TT filter out and identify the malicious node again and those nodes are marked malicious and eliminated from the Routing Table. Other communication statistics, such as data rate and drop rate based on the number of packets sent and received, and the cost of time essential for both traffic metrics and related features, are included in the trust evaluation. If a certain node receives a communication request from another node in a network, an edge node can calculate the trust value of the requested node. ATBET calculates the possible trust paths between any nodes naming them as Sender / Source and Receiver/Destination. ATBET calculates in real-time paths and further trust estimations between any nodes desirous to perform any future transaction and identifies malicious nodes while estimating trust.

Following are the steps for the ATBET algorithm:

Step 1: Determine the most likely pathways between nodes S and D.
Step 2: Estimate each path that exists between S and D, select a node, calculate DT
Step 3: Calculate PDT based on DT, and adjust DT
Step 4: Select neighboring nodes and compute IDT
Step 5: Compute CT based on DT and IDT of nodes in that path
Step 6: Compute TT using CT for each path
Step 7: Review thresholds; if below a threshold, mark node normal; alternatively, mark malicious, and update IPT. Repeat steps 2–6 till all paths' trust has been calculated and IPT has been updated
Step-8: Nodes marked malicious based on TT are disconnected and eliminated from the routing table, recalculate and sort the remaining paths based on TT
Step 9: Calculate the additive metrics and calculate CPTM
Step 10: Thresholds are used for evaluating nodes; if they are below a certain threshold, a node is considered normal; otherwise, considered malicious, and IPT is updated. Rep step 9 till all paths' trust has been calculated and IPT is updated.
Step 11: Notify S about the best paths available, and perform the transaction after encryption and hashing of the data packet.
Step 12: Removal of malicious node, IPT is updated, and S will notify all nodes to update their routing tables.

IV. EXPERIMENT RESULTS OF ATBET

Experimental results are obtained in terms of duration, period, malicious nodes linked to internal or external cyber-attacks, rates of detecting malevolent nodes, rates of accurate

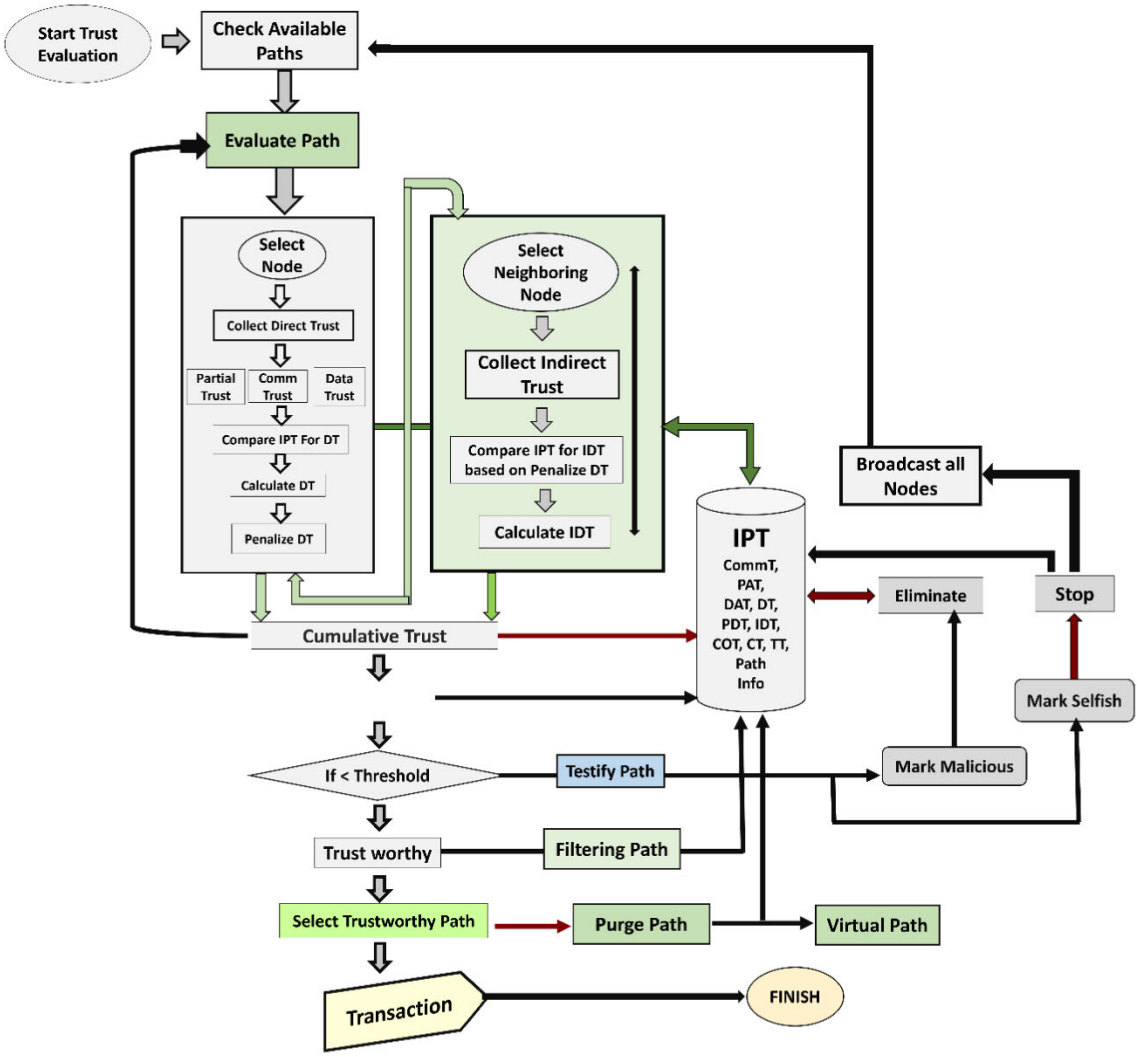

FIGURE 9. Flow diagram of ATBET.

detection of malicious nodes, rates of identifying false positives against malicious nodes, average packet delays, average data rates, an entire network's many edge nodes, along with their trust values with metrics. OMNET+ is used as a discrete event simulator with a graphical user interface that represents the network nodes and their behavior. It is a free and open-source program for modeling traffic, protocols, queues, multiprocessors, distributive hardware, verifying hardware-based designs, and evaluating performance attributes of complex software-based networks. It is an object-oriented integrated discrete event network.

Its comprehensive descriptive capacity to incorporate internal communication and detect different threats makes it suitable for evaluating smart city network trust [28]. OMNET++ provides efficient tools to define the structure of the existing smart city-based systems possessing characteristics that include topology description language and hierarchically assembled modules, where modules are instances of module types and communicate with messages over channels containing customizable module parameters. Tool models are often named networks composed of systems and sub-modules having recursive integration [29]. ATBET uses the Random waypoint Model for mobility. Various mobility models can be employed to simulate the movement of nodes (devices, sensors, edge, fog, cloud, mobile, etc.) within the network. Mobility models help researchers and developers understand how nodes move and interact with each other in a simulated environment, which is crucial for evaluating the performance and effectiveness of trust mechanisms. In general, Two common mobility models are used in such scenarios which are Random waypoint and Random walk models. The Random Waypoint Model is a widely used mobility model in many wireless and mobile networking simulations and is suitable for edge computing environments [30].

The network layer is responsible for the best suitable route selection based on the trust evaluation mechanism ATBET being performed by the application layer in terms of different network parameters by calculating DT and IDT values

TABLE 3. Parameters for simulation in OMNET++.

SNo	Parameters	Values
1	Field size	100 m x 100 m
2	Simulation time	100–1000 s
3	Agent type	UDP
4	Physical Standards	IEEE-802. 15. 4
5	No of nodes	10, 20, 30,……, 100
6	Tx power	J 11
7	Queue type	Drop tail
8	Node deployment	Random
9	Traffic type	CBR
10	Packet size	50 Bytes
11	Traffic load	CBR
12	Initial power	100 J
13	Rx power	J 11
14	Routing protocol	AODV
15	Mobility Model	Random Waypoint

with different mathematical verifications already discussed in different tiers of the mechanism. The physical layer errors do not hamper the trust model as the predictive model provides a reliable data packet via applying the encryption along with the trusted path selected for ensured establishment of communication countering the effects of DDoS / DoS, Sybil, Bad Mouthing, and On-Off attacks. Table 3 describes the simulation parameters.

A. ASSUMPTIONS

The following are the assumptions for the deployment of ATBET in edge-based smart city networks: All nodes have stable links. Nodes are static. Trust propagation is decentralized. Basic information about successor nodes is stored in the routing table. Initially, nodes are assigned a trust value equal to 0.5 in cycle 0 or the 1st cycle. A node becomes malicious when the trust value is below 0.5. Trust is updated in each cycle based on time. Alternate paths with trustworthy nodes are present in the network. Only DoS, DDoS, and Sybil attacks are considered in this paper. These attacks are simulated on the node part of the selected routing path in the previous cycle. Network topology remains the same throughout all the cycles. The weights of all the parameters can change with the environment and applications.

B. PERFORMANCE EVALUATION

Performance parameters used to evaluate the proposed mechanisms are as follows: -

- ***Impact of Trustworthiness level of a Node:*** It shows an accurate detection percentage of malicious/false reporting/selfish nodes after processing through ATBET.
- ***Impact of Detection Rate:*** It presents several detected malicious nodes in ratio instead of in percentage.
- ***Impact of Detecting Accuracy:*** It indicates the correct detection percentage of malicious nodes based on the number of false positive recommendations.
- ***Detection of False Positive Rate:*** It is calculated by dividing the number of false positive outcomes by the total number of negatives. The best false positive rate is zero, while the worst is one.
- ***Impact on Entire Network Lifetime:*** This metric evaluates how long an edge-based network remains operational and alive. It counts the number of times a node consumes its energy, stops working, or enters a deadlock condition.
- ***Impact of Average Throughput:*** This metric analyzes the packet data rate (total payload over the entire session divided by total duration) of communicating nodes coexisting with hostile nodes in the network.
- ***Delay Analysis:*** It examines the technique's end-to-end latency and measures the time packets take from node S to D in the presence of hostile nodes.
- ***Impact of Average Packet Delay:*** It analyzes persistent packet delay among communication nodes in the existence of malicious nodes during network transmission.

C. TRUSTWORTHINESS LEVEL OF A NODE

This parameter exhibits the accuracy percentile of malevolent nodes. It is based on data collected at specific observation intervals of simulator execution changing from 100s to 1000s, between the trust threshold values of 0.5 and 1.0. A percentage increase is used to calculate the ratio of erroneous reporting nodes while comparing threshold values. In the first experiment, the trustworthiness of reliable nodes is compared to malicious nodes among the current edge nodes.

The results of ATBET are compared with BTEM [10], CTBET [8], and PATM [11] and it is illustrated in the graph in Figure 10. ATBET, with several edge nodes coexisting with hostile nodes, shows a positive inclination in comparison with the other three techniques, and as the period increases, In ATBET, a higher trust level was gained as a result of its predictive behavior and avoidance of harmful nodes right after timely detection. It also studies erroneous reporting; faster and more accurate identification of malicious/selfish nodes provides more trustworthiness than its five competitors.

In an alternative scenario, a network's proportion of malicious nodes among total edge nodes is used to measure the system's trustworthiness. It estimates that the average rise of the malicious nodes is close to 15%, with the proportion of edge nodes found malicious ranging from 10 to 50%. Time intervals ranging from 100 to 1000 seconds are used to monitor various sets of nodes.

D. IMPACT OF DETECTION RATE

This parameter displays the proportion of malicious nodes discovered instead of considering in percentage. The proposed design is used to simulate DoS, DDoS, Sybil, Badmouthing, and On-Off attacks with varying proportions of malicious nodes, from 10% to 50% with upholding 10% increments, demonstrating a positive trend in the ratio of the detection of trust. In a different scenario, the network of edge nodes' rate of malicious node identification among trustworthy nodes has been simulated. Compared to other mechanisms, the ATBET mechanism achieves a high rate of

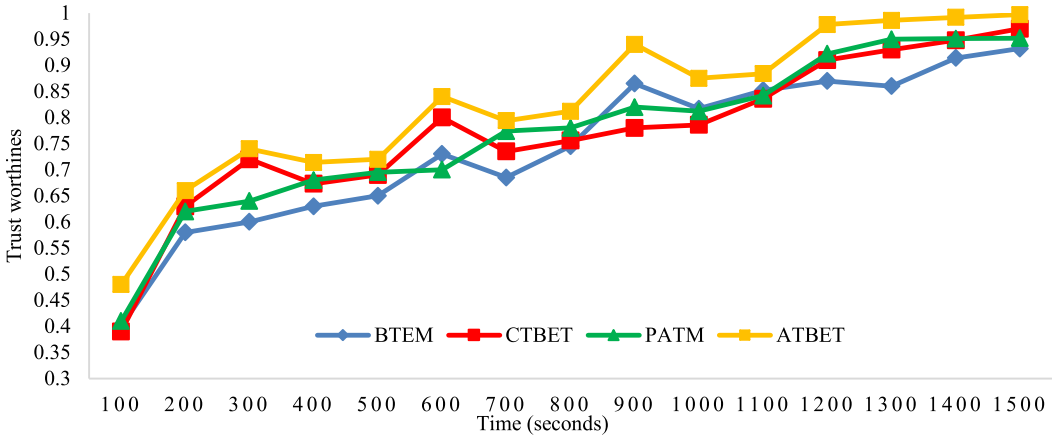

FIGURE 10. Trustworthiness with advancement in time.

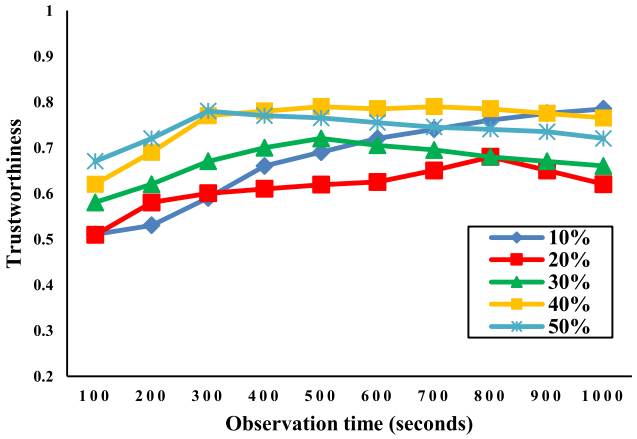

FIGURE 11. Trustworthiness as untrustworthy/selfish nodes increase.

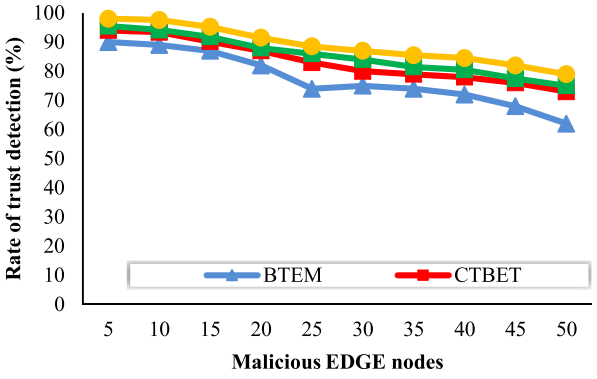

FIGURE 12. Detection rate of malicious nodes against internal and external attacks.

success in identifying harmful and hostile nodes and identifying trustworthy nodes, as shown in Figure 11, which depicts the trust detection rate in the presence of malicious nodes. With a 5% annual growth rate, the fraction of malicious nodes ranges from 5 to 50 percent. ATBET is in comparison to BTEM [10], CTBET [8], and PATM [11]. It shows that ATBET produces more detection rate of trust in nodes than its counterparts 26%, 29%, and 35% due to its ability to track data delivered, received, and transferred information while assessing the trustworthiness of each packet of data. It establishes that the number of malicious nodes is closely correlated with the number of false positives, demonstrating that the ratio of malicious nodes increases as the number of nodes increases. Therefore, as the number of edge nodes in smart city networks increases, it becomes harder to detect malicious nodes.

E. IMPACT OF DETECTING ACCURACY

This metric presents the percentage of malevolent nodes being accurately identified after using the indicated strategy. based on a few suggestions that turn out to be false positives. Edge nodes network analyzes the precise rate of detection of malevolent and selfish nodes in the network. It is built on a predetermined number of false positive predictions. Figure 13 depicts that ATBET is a mechanism that is more efficient in comparison with BTEM [10], CTBET [8], and PATM [11] in percentages 38%, 42%, and 48%, respectively. It is factual that this metric assesses the accuracy while considering trust built and enhanced collaboration among interacting nodes with the lowest packet loss rate.

F. DETECTION OF FALSE POSITIVE RATE

This parameter measures the frequency of false positive attacks. It is a measure obtained as the sum of all the negatives divided by total false positive outcomes. Value 0.0 is considered the best false positive rate, whereas 1 is considered the worst respectively. The impact of the trust levels of trustworthy and malicious nodes on the proportion of false positives that are examined against attacks like DoS/DDoS, Sybil, Bad Mouthing, and On-Off creates a different situation. When compared to alternative trust systems, the False Positive detection rate for malicious nodes is greater while that for trustworthy nodes is less valuable like BTEM [10],

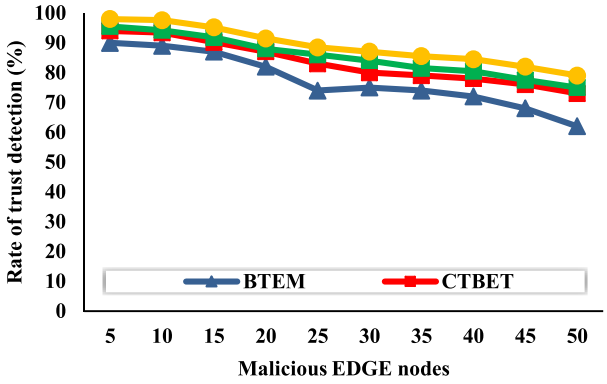

FIGURE 13. Rate of accurate detecting of malicious nodes against internal attacks.

CTBET [8], and PATM [11] in percentages 34.21%, 41.67%, 44.52%, and 53.56 %, respectively, shown in Figure 20.

G. IMPACT ON THE ENTIRE NETWORK LIFETIME

This metric examines the operational and life duration of an edge-based smart city network. Any edge node's first energy drain, which prevents it from being able to transmit the packet, is measured. The network is impacted by the loss of resources and services when a node due to excessive use faces drainage of energy and switches off. The lifespan of a network may be determined by testing the effects of malicious nodes that quickly deplete the energy of edge nodes, which can result in a network jam and stop all network activity. Nodes in a network use less energy than equivalent systems, which increases the network's lifespan. Figure 14 displays the results of evaluating the network lifespan across 20 malicious nodes within the network, with a proportion of 301 seconds being shown in BTEM [10], 325 seconds in CTBET [8], and 385 seconds in PATM [11], ATBET preserves time is 446 seconds.

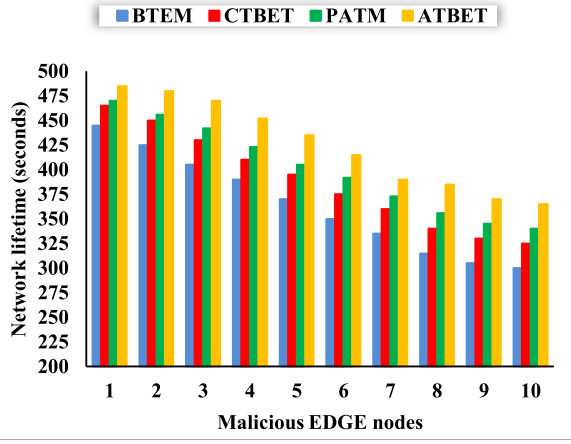

FIGURE 14. Network lifetime (seconds).

H. IMPACT OF AVERAGE PACKET DELAY

It examines packet delays that continue to occur between communication nodes when malicious nodes are present during transmission over the network. It evaluates the effectiveness of ATBET considering the decrease in packet latency while coexisting with malicious nodes.

The average packet delay across numerous malicious nodes is examined differently. This parameter evaluates how well the suggested scheme performs when malicious nodes intrude and disrupt communication in the network since there is less packet latency. This plan, nevertheless, has a greater throughput than comparable trust schemes. i.e. like, BTEM [10], CTBET [8], and PATM [11], respectively shown in Figure 17, though investigating an increase from 5 to 50% in the volume of malevolent nodes. This method takes into account each node (both trustworthy and monovalent nodes) participating in the networks and their amount of energy.

TABLE 4. Performance of ATBET against viable protocols with increasing or decreasing trend.

Techniques	Throughput (%)	Life Time of Network (s)	Rate of Detection (%)	Detection Accuracy (%)	Rate of False Positives (%)	Latency (s)
BTEM	↑190	↑301	↑26	↑38	↑34.21	↓0.038
CTBET	↑205	↑325	↑29	↑42	↑41.67	↓0.029
PATM	↑242	↑385	↑32	↑48	↑44.52	↓0.021
ATBET	↑285	↑446	↑35	↑56	↑53.46	↓0.017

I. DELAY ANALYSIS: END-TO-END

This parameter examines the suggested approach's end-to-end latency in the presence of selfish/malicious nodes. The amount of time a packet needs to travel across a linked network from host to destination is also known as one-way delay (OWD). Different numbers of malicious nodes are used to compute end-to-end delay and one-way packet delay also called latency. It figures out how long it typically takes for node D to receive a packet sent from a host node S. When this parameter is compared to other competing trust techniques, it is discovered that while the first 10% of malicious nodes demonstrate the same performance levels, there is a difference as the number of malicious nodes approaches 20%. Figure 18 depicts a significant difference due to the ability and capacity of ATBET, to choose the trustworthy nodes possessing a high level of energy i.e. bearing lower consumption of energy and less computational complication), and further avoid malicious nodes while communicating in smart city networks.

Table 4 clearly shows that an ATBET outperforms the benchmark protocols by achieving an increase of 213% in throughput, 345 seconds in network lifetime, 38% in the rate of detection, 48% in detection accuracy, 47% in the rate of false positive rate, and decrease by 45% in delay rate in the

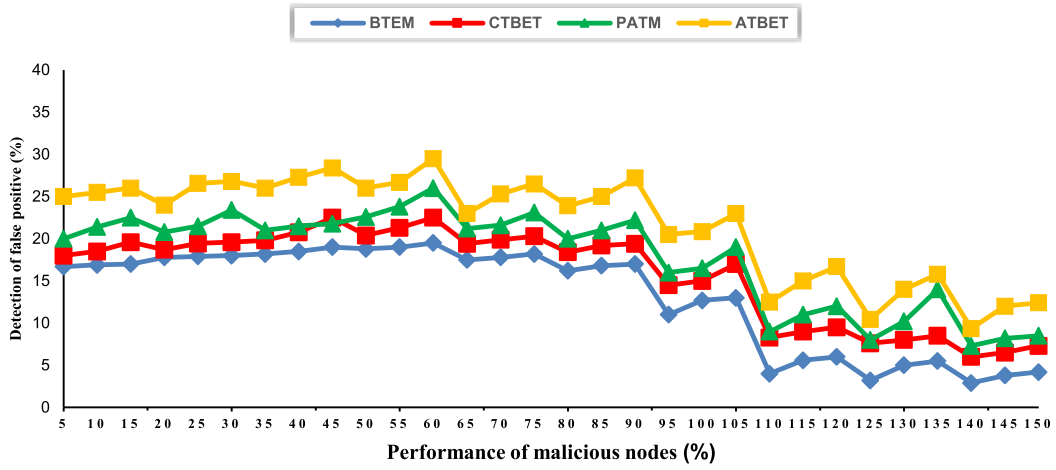

FIGURE 15. Rate of false positive between 5 to 150.

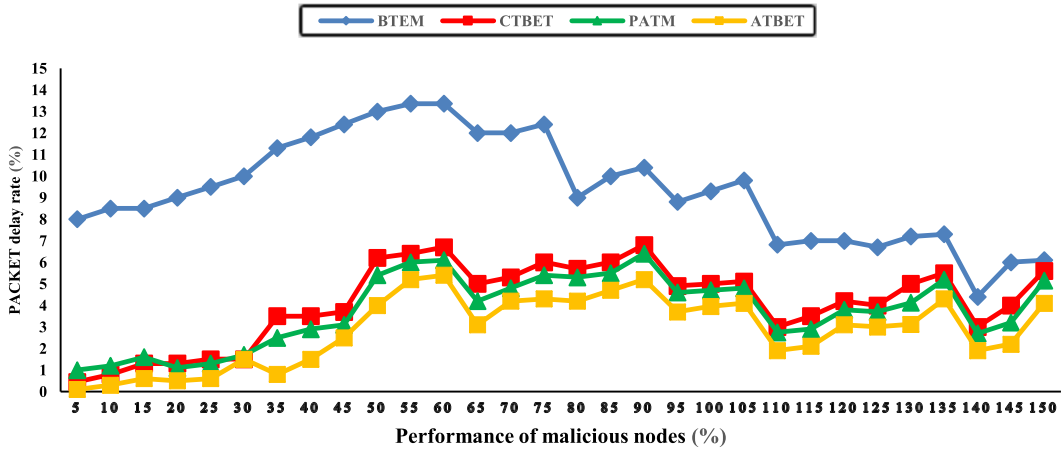

FIGURE 16. Ratio of average delay in nodes between 5 and 150.

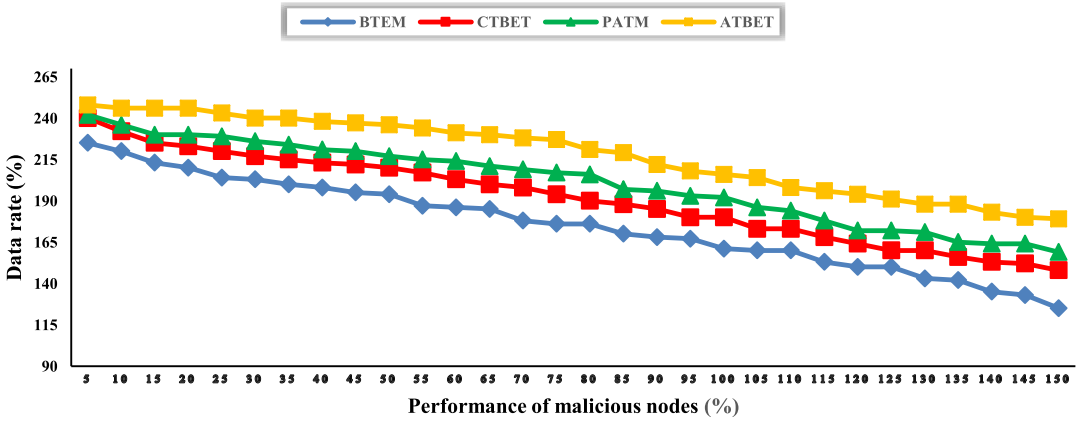

FIGURE 17. Average data rate ratio between nodes 5 and 150.

comparative techniques BTEM [10], CTBET [8] and PATM [11] respectively.

The proposed ATBET achieved the better performance in terms of malevolent nodes detection, identification of false positive rate against malicious nodes, average packet delays, average data rates along with their trust values with metrics. The simulation is used for evaluation where proposed solution has achieved a higher rate of data transmission and low rate of packet delay and better performance in the presence of malicious nodes. Results further demonstrated that

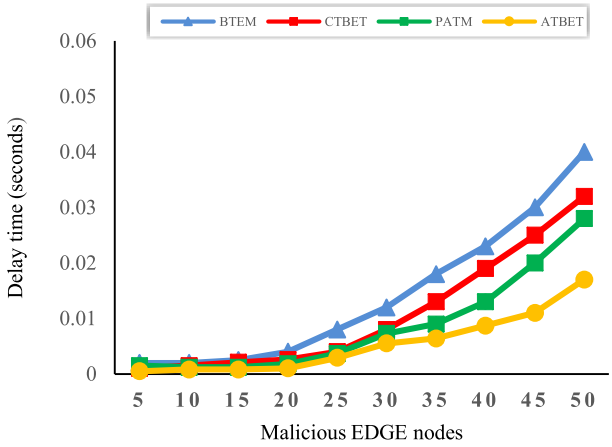

FIGURE 18. Delay analysis: End-to-end delay.

a proposed solution has a lower false positive detection rate while somewhat raising trustworthiness in the presence of Sybil, DoS, On-Off, Bad Mouthing based attacks, compared to the BTEM, CTBET and PATM.

V. CONCLUSION

An Assertive Trust-Based Efficient Approach (ATBET) for smart city networks is presented in this research. The main challenges in designing efficient, stable, robust total trust-based techniques capable of enhancing any network's security features have been addressed and determined. ATBET is an attempt to find the solutions to the technical issues and gaps discussed in related work and enhance the trust technique performance compared to the existing techniques. The interaction quality measure is used to collect direct and indirect (DT and IDT) trust values of the edge-based nodes to correlate the data over time, pick the trustworthy nodes, and further the trustworthy channel for data transfer. It also identifies and detects the malicious nodes twice in a single iteration and further successfully eliminates them from routing tables, and all nodes are updated to disconnect from that malicious node. ATBET is simulated using OMNET++, and the performance of several aspects is compared to existing trust evaluation-based methodologies. Simulation results depict that ATBET has achieved a higher rate of data transmission, achieve a low rate of packet delay and better performance is achieved in the presence of malicious nodes than all communicating edge nodes in the smart city. Results further demonstrate that a proposed approach has a lower false positive detection rate while somewhat raising trustworthiness, proving that this method is more resistant to Sybil, DoS, On-Off, Bad Mouthing based attacks, compared to the BTEM [10], CTBET [8] and PATM [11]. ATBET has been verified and our simulation results illustrate that the system works effectively in a hostile environment and that ATBET promptly converges towards sound trust values in comparison to the conventional trust calculation techniques.

The proposed approach will be refined in the future to identify various internal and external attacks that cause nodes to be malevolent, selfish, and inadequate/ contradictory, such as mouthing, Routing, Ballot Stuffing, Conflicting Behavior, Routing attacks, Camouflages rivals, and wormhole attacks. ATBET is a robust technique and is capable of evaluating the trust dynamically but still has the potential limitation for some routing points that may exist in the various interconnected networks, Hence, for future work, the optimized routing mechanism will be employed to enhance this trust evaluation mechanism and will be deployed on the fog environment. In addition, this approach will be evaluated using time-based analysis for various service composition applications to weed out service-oriented attacks and architectural attacks. ATBET will assure reliable remote connectivity, prevent rogue nodes, and function effectively with fewer resources. Another future path is to take into account other QoS characteristics, such as latency in high-speed networks, storage for restricted EC devices, and energy restrictions, to confirm the relevance of the ATBET. Additionally, the number of edge nodes, as well as their respective servers and intermediate nodes, will be examined using these attributes.

REFERENCES

[1] H. B. Sta, "Quality and the efficiency of data in 'smart-cities,'" *Future Gener. Comput. Syst.*, vol. 74, pp. 409–416, Sep. 2017.
[2] R. Y. Clarke, "Business strategy: IDC government insights' smart city maturity model—Assessment and action on the path to maturity," Int. Data Corp. (IDC) Government Insights, Bus. Strategy GI240620, Alexandria, VA, USA, 2013.
[3] L. Munn, "Staying at the edge of privacy: Edge computing and impersonal extraction," *Media Commun.*, vol. 8, no. 2, pp. 270–279, Jun. 2020.
[4] Y. A. Qadri, A. Nauman, Y. B. Zikria, A. V. Vasilakos, and S. W. Kim, "The future of healthcare Internet of Things: A survey of emerging technologies," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 1121–1167, 2nd Quart., 2020.
[5] O. AlFarraj, A. AlZubi, and A. Tolba, "Trust-based neighbor selection using activation function for secure routing in wireless sensor networks," *J. Ambient Intell. Humanized Comput.*, pp. 1–11, Jun. 2018.
[6] K. N. Qureshi, A. Iftikhar, S. N. Bhatti, F. Piccialli, F. Giampaolo, and G. Jeon, "Trust management and evaluation for edge intelligence in the Internet of Things," *Eng. Appl. Artif. Intell.*, vol. 94, Sep. 2020, Art. no. 103756.
[7] P. Ranaweera, A. D. Jurcut, and M. Liyanage, "Survey on multi-access edge computing security and privacy," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 1078–1124, 2nd Quart., 2021.
[8] K. N. Qureshi, A. Iftikhar, S. N. Bhatti, F. Piccialli, F. Giampaolo, and G. Jeon, "Trust management and evaluation for edge intelligence in the Internet of Things," *Eng. Appl. Artif. Intell.*, vol. 94, Sep. 2020, Art. no. 103756.
[9] J. Jang-Jaccard and S. Nepal, "A survey of emerging threats in cybersecurity," *J. Comput. Syst. Sci.*, vol. 80, no. 5, pp. 973–993, Aug. 2014.
[10] R. W. Anwar, A. Zainal, F. Outay, A. Yasar, and S. Iqbal, "BTEM: Belief based trust evaluation mechanism for wireless sensor networks," *Future Gener. Comput. Syst.*, vol. 96, pp. 605–616, Jul. 2019.
[11] A. Ahmed, K. N. Qureshi, M. Anwar, F. Masud, J. Imtiaz, and G. Jeon, "Link-based penalized trust management scheme for preemptive measures to secure the edge-based Internet of Things networks," *Wireless Netw.*, pp. 1–23, Apr. 2022.
[12] C. V. L. Mendoza and J. H. Kleinschmidt, "A distributed trust management mechanism for the Internet of Things using a multi-service approach," *Wireless Pers. Commun.*, vol. 103, no. 3, pp. 2501–2513, Dec. 2018.
[13] B. Gong, Y. Zhang, and Y. Wang, "A remote attestation mechanism for the sensing layer nodes of the Internet of Things," *Future Gener. Comput. Syst.*, vol. 78, pp. 867–886, Jan. 2018.
[14] Q. Lin and D. Ren, "Quantitative trust assessment method based on Bayesian network," in *Proc. IEEE Adv. Inf. Manage., Communicates, Electron. Autom. Control Conf. (IMCEC)*, Oct. 2016, pp. 1861–1864.

[15] W. Li, H. Song, and F. Zeng, "Policy-based secure and trustworthy sensing for Internet of Things in smart cities," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 716–723, Apr. 2018.
[16] Ch. Rupa, R. Patan, F. Al-Turjman, and L. Mostarda, "Enhancing the access privacy of IDaaS system using SAML protocol in fog computing," *IEEE Access*, vol. 8, pp. 168793–168801, 2020.
[17] D. Gountia and S. Roy, "Design-for-trust techniques for digital microfluidic biochip layout with error control mechanism," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 19, no. 3, pp. 1570–1582, May/Jun. 2021.
[18] T. ul Hassan, M. Asim, T. Baker, J. Hassan, and N. Tariq, "CTrust-RPL: A control layer-based trust mechanism for supporting secure routing in routing protocol for low power and lossy networks-based Internet of Things applications," *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 3, p. e4224, Mar. 2021.
[19] N. Tariq, M. Asim, F. A. Khan, T. Baker, U. Khalid, and A. Derhab, "A blockchain-based multi-mobile code-driven trust mechanism for detecting internal attacks in Internet of Things," *Sensors*, vol. 21, no. 1, p. 23, Dec. 2020.
[20] I. U. Din, A. Bano, K. A. Awan, A. Almogren, A. Altameem, and M. Guizani, "LightTrust: Lightweight trust management for edge devices in industrial Internet of Things," *IEEE Internet Things J.*, vol. 10, no. 4, pp. 2776–2783, Feb. 2023.
[21] S. Dhelim, N. Aung, M. T. Kechadi, H. Ning, L. Chen, and A. Lakas, "Trust2Vec: Large-scale IoT trust management system based on signed network embeddings," *IEEE Internet Things J.*, vol. 10, no. 1, pp. 553–562, Jan. 2023.
[22] H. Wang, L. Xu, and G. Gu, "FloodGuard: A DoS attack prevention extension in software-defined networks," in *Proc. 45th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw.*, Jun. 2015, pp. 239–250.
[23] D. Singh, B. Kumar, S. Singh, and S. Chand, "A secure IoT-based mutual authentication for healthcare applications in wireless sensor networks using ECC," *Int. J. Healthcare Inf. Syst. Informat.*, vol. 16, no. 2, pp. 21–48, Apr. 2021.
[24] B. A. Pratomo, P. Burnap, and G. Theodorakopoulos, "BLATTA: Early exploit detection on network traffic with recurrent neural networks," *Secur. Commun. Netw.*, vol. 2020, pp. 1–15, Aug. 2020.
[25] N. Karthik and V. S. Ananthanarayana, "A hybrid trust management scheme for wireless sensor networks," *Wireless Pers. Commun.*, vol. 97, no. 4, pp. 5137–5170, Dec. 2017.
[26] A. Altaf, H. Abbas, F. Iqbal, F. A. Khan, S. Rubab, and A. Derhab, "Context-oriented trust computation model for industrial Internet of Things," *Comput. Electr. Eng.*, vol. 92, Jun. 2021, Art. no. 107123.
[27] J. Jiang, S. Hua, G. Han, A. Li, and C. Lin, "Controversy-adjudication-based trust management mechanism in the Internet of underwater things," *IEEE Internet Things J.*, vol. 10, no. 3, pp. 2603–2614, Feb. 2023.
[28] P. Palensky, E. Widl, and A. Elsheikh, "Simulating cyber-physical energy systems: Challenges, tools and methods," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 44, no. 3, pp. 318–326, Mar. 2014.
[29] A. Varga and R. Hornig, "An overview of the OMNeT++ simulation environment," in *Proc. 1st Int. Conf. Simul. Tools Techn. Commun., Netw. Syst. Workshops*, 2008, p. 60.
[30] S. S. Chandra and S. Pallamsetty, "Impact of mobility on power consumption in RPL," *I-manager's J. Wireless Commun. Netw.*, vol. 6, no. 4, p. 23, 2018.

ABEER IFTIKHAR received the M.S. degree in information security from Bahria University, Islamabad, with a focus on security and trust solutions for edge computing and smart city networks, where she is currently pursuing the Ph.D. degree with the Computer Science Department, under the supervision of Dr. Kashif Naseer Qureshi. Her Ph.D. thesis was titled Security Provision by Using Detection and Prevention Methods to Ensure Trust in Edge-Based Smart City Networks. She is involved in a security-related domain to design new trust mechanisms, threat-preventive solutions, and privacy-preserving solutions in smart cities, edge computing, and high-speed networks.

KASHIF NASEER QURESHI received the Ph.D. degree from the University of Technology Malaysia (UTM), in 2016. He holds the dual master's degrees in computer science and information technology from Reputable Universities. He is an Associate Professor of Cyber Security with the Department of Electronic and Computer Engineering, University of Limerick, Ireland. He is also actively involved in the Cyber Skills project, a HEA-HCI Pillar three initiative Ireland. He is the Co-Principal Investigator in Cyber Reconnaissance and Combat project funded by higher education commission. He has published various high-impact factor papers in international journals and conference proceedings and served on several conferences IPCs and journal editorial boards. He has number of book chapters and five edited books in Springer, CRC and Elsiver Publishers related to Cybersecurity, Privacy and Trust architectures. He has also a part of various research projects related to wireless communication, routing and CyberSecurity domains in the U.K., China, Ireland, Malaysia, Canada, Dubai, Vietnam, and Pakistan. His research interests focus on the security, trust and privacy concerns for Internet of Everything (IoE), Internet of Vehicles (IoV), Electronic Vehicles (EV) charging management planning and recommendation systems, and the Internet of Things (IoT) and use cases implementation in wireless and wired networks. He is an active member of Lero, the Science Foundation Ireland Research Centre for Software at the University of Limerick (UL). His name is included in top 2% Scientist for consecutive three years from Stanford university, USA.

ALI A. ALTALBE received the M.Sc. degree in information technology from Flinders University, Australia, and the Ph.D. degree in information technology from The University of Queensland, Australia. He is currently an Associate Professor with Prince Sattam bin Abdulaziz, King Abdulaziz University, Saudi Arabia.

KHALID JAVEED (Member, IEEE) is currently an Assistant Professor with the Department of Computer Engineering, College of Computing and Informatics, University of Sharjah, Sharjah, United Arab Emirates.