Multidirectional Trust-Based Security Mechanisms for Sinkhole Attack Detection in the RPL Routing Protocol for Internet of Things

Sopha Khoeurt[1], Chakchai So-In[2], Pakarat Musikawan[3] and Phet Aimtongkham[4*]

[1]Applied Network Technology (ANT), Department of Computer Science, College of Computing, Khon Kaen University, Khon Kaen, Thailand. sopha.k@kkumail.com,
Orcid: https://orcid.org/0009-0007-2809-2399

[2]Applied Network Technology (ANT), Department of Computer Science, College of Computing, Khon Kaen University, Khon Kaen, Thailand. chakso@kku.ac.th,
Orcid: https://orcid.org/0000-0003-1026-191X

[3]Advanced Intelligent Interdisciplinary Integration (AIII), Department of Computer Science, College of Computing, Khon Kaen University, Khon Kaen, Thailand. pakamu@kku.ac.th,
Orcid: https://orcid.org/0000-0001-5315-751X

[4*]Applied Network Technology (ANT), Department of Computer Science, College of Computing, Khon Kaen University, Khon Kaen, Thailand. phetim@kku.ac.th,
Orcid: https://orcid.org/0000-0001-5289-1149

Received: May 15, 2023; Accepted: June 30, 2023; Published: September 30, 2023

Abstract

The Internet of Things (IoT) has gained popularity in recent years by connecting physical objects to the Internet, enabling innovative applications. To facilitate communication in low-power and lossy networks (LLNs), the IPv6-based routing protocol for LLNs (RPL) is widely used. However, RPL's lack of specified security models makes it vulnerable to security threats, particularly sinkhole attacks. Existing sinkhole attack detection techniques suffer from high detection delays and false positives. To overcome these limitations, in our research we propose a multidirectional trust-based detection approach for sinkhole attacks in the RPL routing protocol. Our model introduces a novel architecture that considers trust in parent, child, and neighbor directions, reducing detection delays. We enhance detection efficiency and reduce false positives by combining fuzzy logic systems (FLSs) and subjective logic (SL). Additionally, we introduce a new trust weight variable derived from Shannon's entropy method and multiattribute utility theory. We adaptively adjust the SL coefficient based on network conditions, replacing the constant coefficient value of SL theory. Our approach is compared to the most recent techniques, and we assess different indicators, such as false-positive rate, false-negative rate, packet delivery ratio, throughput, average delay, and energy consumption. Our results demonstrate superior performance in all these metrics, highlighting the effectiveness of our approach.

Keywords: Sinkhole Attack, RPL Attack, IoT, Fuzzy Logic System, Subjective Logic.

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), volume: 14, number: 3 (September), pp. 48-76 DOI: *10.58346/JOWUA.2023.I3.005*

*Corresponding author: Applied Network Technology (ANT), Department of Computer Science, College of Computing, Khon Kaen University, Khon Kaen, Thailand.

1 Introduction

The Internet of Things (IoT) connects gadgets, buildings, vehicles, instruments, and other items with electronics, circuits, software, and sensors (Khan, J.Y., 2019) (Darabkh, K.A., 2022). Connectivity allows data gathering and exchange (Suo, H., 2012) (Haq, S.U., 2022). Through remote object sensing and control using existing network infrastructure, IoT enables a more direct integration of the real world with computer-based systems (Hossain, M.S., 2019) (Kumar, B.S., 2022). Cisco expected 100 billion IoT devices by 2025 (Abbas, N., 2019) (Adekanbi, M.L., 2021).

A brand-new unique network type known as a low-power and lossy network (LLN) has emerged as a result of the development of IoT technologies (Ghaleb, B., 2018). Resource-constrained devices, which have low levels of memory, power, and computation, as well as communication links with low bandwidth and short transmission ranges, which among other things cause low throughput, high packet loss, and high end-to-end latency, are some of the restrictions that LLNs have. Consequently, the IPv6 Routing Protocol for LLNs (RPL) was developed in 2012 by the IETF Working Group as a network layer and common routing protocol (Winter, T., 2012).

A specific routing protocol called RPL was created for networks with little power, little computing power, and lossy communication links (Vasseur, J., 2011). RPL is a distance-based routing protocol that enables users to create logical routing topologies known as DODAG structures with a router or sink node that has an internet connection; by use of multihop connections via the sink node, normal nodes are linked to the internet (Winter, T., 2012). This routing is susceptible to several security risks, including rank attacks, wormhole attacks, clone ID attacks, blackhole attacks, and sinkhole attacks (Jahangeer, A., 2023) (Zahra, F., 2022) (Burange, A.W., 2021); among these attacks, sinkholes are the most dangerous (Zaminkar, M., 2020) because they advertise a false route to the sink node, divert traffic to a node under the attacker's control, bypass traditional security measures and drop transmitted data packets (Rehman, A.U., 2019). Because of the widespread use and adoption of IoT devices (Maraveas, C., 2022), security is the most important factor for IoT devices that use the RPL routing protocol, particularly for preventing sinkhole attacks (Karthikeyan, M.K.P.P.P., 2019) (Tahir, S., 2019).

Recently, several approaches have been proposed to detect sinkhole attacks, such as trust-based, rank-based and machine learning-based approaches (Patel, B., 2021) (Zaminkar, M., 2020) (Prathapchandran, K., 2021). However, these approaches have limitations that should be considered; for example, the trust-based approach uses a static coefficient value, which makes it inefficient in different network conditions. Rank-based approaches have a high rate of false positives due to the use of a single trust metric to evaluate malicious nodes. Thus, this approach fails to capture all the relevant factors that differentiate legitimate nodes from malicious nodes. The machine learning approach uses trust metrics to evaluate malicious nodes, which are not powerful against sinkhole attacks because they need a specific amount of time to detect malicious nodes. This approach requires a training model that cannot be deployed in IoT nodes. Due to the various directions of nodes in an RPL network, such as parent, child, and neighbor nodes, malicious nodes can be located in any direction (Lodhi, M.A., 2015) (Mayzaud, A., 2016). Therefore, these proposed approaches have the same limitations when considering one direction of trust detection (the parent direction), which increases the malicious detection delay.

For these reasons, it is essential to create novel techniques and approaches to detect and avoid sinkhole attacks on the RPL network. Moreover, we propose a multidirectional trust-based detection (MDTrust-RPL) approach for sinkhole attack detection in the RPL routing protocol using a novel architecture that accounts for the parent, child, and neighbor trust directions, thereby capturing the

different forms of directional trust in an RPL network. The neighbor trust direction is one of the most significant among these three directions because malicious neighbor nodes can become the child or parent of another node; consequently, trust in neighbor nodes is crucial to avoid side effects in future communication and reduce the malicious detection delay. In addition, to increase the detection efficiency and decrease the number of false positives, we present a hybrid detection method that combines fuzzy logic systems (FLSs) and subjective logic (SL). Then, we propose a new trust weight variable based on Shannon's entropy method and multiattribute utility theory. The constant coefficient value of SL theory is replaced with an adaptive coefficient that dynamically adapts to network conditions.

Due to the multiple-parameter requirement of our proposed method, we adopt an FLS (Zadeh, 1965) to make accurate multiparameter choices (Tang, J., 2019), which is also more computationally efficient because it does not need model training (Hentout, A., 2023). Additionally, the FLS employs weights (likelihoods) of probable outcomes, unlike a typical logical system, in which a statement is either true or false. On the other hand, a probabilistic logic that permits inference from ambiguous and partial data is subjective logic (Jøsang, A., 1998). By using subjective logic, it is possible to simulate real-world scenarios more correctly and ascertain that the conclusions accurately represent the ignorance and uncertainty that inherently arise from partially uncertain input arguments (Jøsang, A., 2016). The following are the key contributions of the MDTrust-RPL model:

- We propose a multidirectional trust-based detection architecture that considers parent, child, and neighbor directions.
- We propose a hybrid of FLSs and subjective logic for malicious node detection.
- We propose a trust weight variable derived from Shannon entropy and multiattribute utility theory as the FLS input.
- We propose an adaptive coefficient for subjective logic theory.

The remainder of the paper is arranged as follows: the related work, the network and assumption models, the proposed multidirectional trust-based model (MDTrust-RPL), the simulation results and discussion, and the conclusion and future work are presented sequentially.

2 Related Work

Researchers have conducted extensive studies to find solutions for IoT network security using the RPL routing protocol in response to the expanding trends in the IoT area. An overview of current IoT research using the RPL protocol is provided in this section.

Alzubaidi et al., 2018 (Alzubaidi, M., 2018) proposed a hybrid technique for locating irregular neighboring nodes in RPL networks. This technique consists of two steps: the detection of unusual nodes and the identification of nodes involved in sinkhole attacks. Two nodes are used in this method to passively gather data on nearby nodes, analyze it, and detect suspicious nodes. A sinkhole node can be found by passively comparing the query data with the two nodes. The research gap for this technique is that two passive nodes are busier than other nodes due to increased traffic while collecting and sending query data to evaluate malicious nodes, which consumes more energy.

The use of rank-based approaches has been suggested as a different strategy for identifying sinkhole and clone ID attacks in RPL networks (Mirshahjafari, S.M.H., 2019). This method combines earlier research on the effects of rank attacks in an IoT network environment (Raza, S., 2013) with the real-time detection of clone ID attacks (Salehi, S.A., 2013) by changing the decision condition to evaluate malicious or normal nodes to reduce the detection time and the amount of false alarms. Similar

approaches have been proposed by (Zaminkar, M., 2020), using node ranking and rating as a security measure to prevent sinkhole attacks. To identify a sinkhole attack, this model has two components; the first identifies malicious nodes based on rank distance, and the second identifies them by using the average data packet transmission rate. However, a high false positive rate is the key limitation of these two proposed techniques because they use one metric to evaluate sinkhole nodes, which fails to capture all of the key criteria that distinguish genuine nodes from malicious nodes. Additionally, this approach enables a child node to recognize a malicious parent; when there is a malicious child, this approach takes more time to recognize it and sometimes fails.

To improve the previous rank-based approach (Zaminkar, M., 2020), the DSH-RPL method was proposed (Zaminkar, M., 2021). This method aims to detect sinkhole attacks and consists of four phases. The production of a reliable RPL is the first phase based on energy, trust, and integrity. The second phase is attacking detection based on the ranking and ratio of packet delivery. The malicious nodes were quarantined in the third phase, and data were transferred using encryption in the fourth phase. However, this approach has limitations in terms of the rate of false positives because it relies on the evaluation of malicious nodes based on a single trust metric. Furthermore, the second phase, which utilizes the packet delivery ratio (PDR) as a detection technique for identifying suspicious nodes, may prove inefficient when the network consists of a high number of malicious nodes. This inefficiency arises from the fact that the sink calculates the PDR based on the broadcast messages by the sink to all leaf nodes and then waits for a reply message from each leaf node to calculate the individual PDR for each route. It subsequently compares the individual PDR values to the average PDR to determine if a suspicious node with a low PDR on a particular route is malicious. However, this technique fails when all routes contain a malicious node, as all the data sent between the source and destination will fail.

In addition, a similar technique has been proposed to mitigate the impact of a sinkhole attack, namely, DSTIDS (Patel, B., 2021). This proposed technique is divided into two parts. The first part records the negative and positive observations at a particular node. The second part uses subjective logic to make decisions regarding trusted and malicious nodes through a sink node. However, the drawback of this technique is that it uses a static coefficient in the subjective logic, which makes it inefficient in different network conditions.

To overcome the limitations of the rank-based technique (Zaminkar, M., 2020) and hybrid technique (Alzubaidi, M., 2018), the proposed model, RFTrust, was proposed (Prathapchandran, K., 2021). This model provides a secure trust-based network for the IoT and is intended to identify sinkhole attacks by using direct and indirect trust with random forest algorithms and subjective logic. However, the limitations of this proposed model are that it increases the malicious detection delay because it uses parameters that are not the most greatly affected by sinkhole attacks and need a specific amount of time to detect malicious nodes and that it considers only one-directional parent trust, making it difficult for this technique to identify malicious child and neighbor nodes.

Recently, the concept of eavesdropping (overhearing) has been regarded as a new technique for recognizing sinkhole attacks (Jamil, A., 2021). Using this technique, a child node can hear what its parent is transmitting. The node parent is an attacker if the child node sends many packets but receives no response from the parent node. However, this mechanism causes a high end-to-end delay because the overhearing concept requires every node to wait for ongoing transmissions to complete before they can overhear and capture packets. Additionally, this technique consumes more energy because every node must stay awake and listen to all nearby transmissions. Furthermore, the overhearing technique cannot recognize malicious child and neighbor nodes because it considers only malicious parents.

Table 1: Summary of Related Work to Detect Sinkhole Attacks in the RPL Routing Protocol

Authors	Parameter	Evaluation Metrics	Simulators	Research Gap
Alzubaidi et al. 2018	Rank value	Energy Consumption (EC), Detection Accuracy (DA), False Positive Rate (FPR), and True Positive Rate (TPR).	Contiki/Cooja	Two passive nodes consumed more energy than normal node.
Mirshahjafari et al. 2019	Rank value	TPR	Contiki/Cooja	Increase the high false-positive rate
Zaminkar et al. 2020	Rank distance and average packet transmission	Detection Rate (DR), PDR, FNR, FPR, Throughput, and Packet Loss Rate.	NS-3	Increase the high FPR and consider only malicious parent nodes.
Zaminkar et al. 2021	Energy, trust, integrity, rank, inconsistency, and PDR	DR, FNR, FPR, and PDR	NS-2	Increase the high FPR and consider only malicious parent nodes.
Patel et al. 2021	Parent rank and node rank	FNR, FPR and PDR	Contiki/Cooja	High malicious detection delay and using a static coefficient.
Prathapchandran et al. 2021	Packet delivery ratio, average delay, energy consumption, and honesty	DA, PDR, throughput, average delay, EC, FPR, and FNR	Contiki/Cooja	High malicious detection delays and considers only malicious parent nodes.
Jamil et al. 2021	Overhearing packet transmission	PDR	Contiki/Cooja	High end-to-end delay and low PDR.

3 Network and Assumption Models

In this section, we describe the network and assumption models.

Network Model

In this research, similar to other recent work on sinkhole attacks in RPL networks, there is only one sink node, which is typically located at the network's end node (to be connected to the internet) (Prathapchandran, K., 2021) (Zaminkar, M., 2020) (Patel, B., 2021). This node has a power supply. Source nodes, which are sensor nodes triggered by specific events (such as sensing and preparing for data transmission), send data to the sink node at a specific time via subsequent or nearby nodes. Our network assumptions are based on the following:

The IoT environment serves as the primary criterion for the network model.

- One high-capacity device serves as the sink node in an IoT network. RPL assumes that the root node it employs as its underlying routing protocol is safe and cannot be used by an attacker since the commander oversees it. This node notifies all other IoT nodes in the network and sends them information about the malicious node when a sinkhole attack is discovered in the network.

- RPL Nodes: Along with the sink node, RPL nodes are uniformly distributed at random within the square network area, and after the RPL nodes are deployed, there is no mobility and no clear location information.
- RPL nodes are small and have limited resource availability. Through sensing, observing, updating, and processing, these nodes can become exhausted.
- If a node launches a sinkhole attack, it is considered a malicious node; every node can observe the behavior of its parent, child, and neighbors.

Assumption Model

In this subsection, we describe the two required components and the efficiency of the model we propose to detect sinkhole attacks in multiple directions on RPL networks.

i. *DODAG Information Object (DIO)*: A DIO message is a message used in RPL routing to build and maintain a routing topology. This message contains information about the network's topology, including the RPL instance ID, version number, node ranks, and other related information (Vasseur, J., 2011). These messages are sent periodically by a node to its neighbors to maintain the routing topology of the network. Therefore, our system model uses the DIO message as the primary means of processing the evaluation and detection techniques, and we modify this message by adding more individual information on nodes into this control message: the PDR, dropped packet rate (DPR), number of route entries (NRE), parent rank (PR) and number of DAO output messages (DAO). Figure 1 illustrates the DIO message communication that contains this additional information.

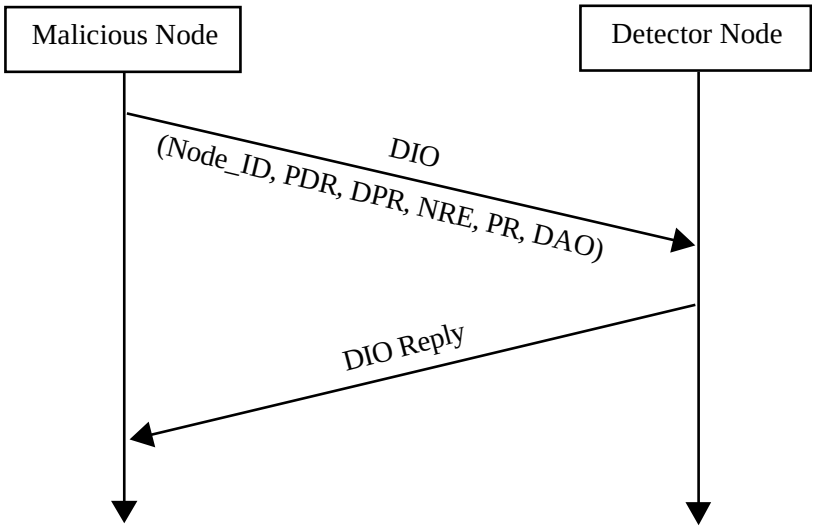


Figure 1: DIO Message Communication

ii. *Blacklist Database (Backlist DB)*: The blacklist DB is the second component, which stores the IDs of malicious nodes after detection by our proposed model.

Figure 2 illustrates our suggested model's effectiveness in detecting sinkhole attacks in different directions in RPL networks. For instance, nodes N1 and N8 are malicious nodes; node N1 has node N4 as a child and node N3 as a neighbor node, and node N8 has node N7 as a parent node. In our proposed model, nodes N4 and N3 can detect node N1 as malicious by using the parent and neighbor trust directions, whereas node N7 can detect node N8 as malicious by using the child trust direction.

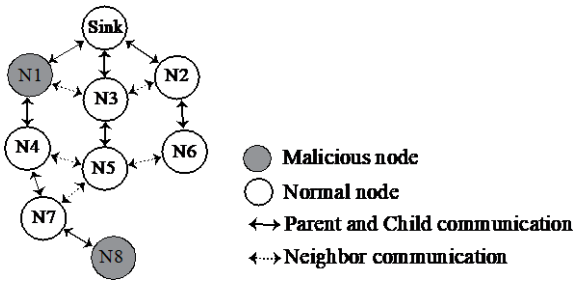


Figure 2: Sinkhole Attacks on Different Locations and the Efficiency of Our Detection Model

4 Proposed Multidirectional Trust-Based Model (MDTrust-RPL)

In this section, the proposed MDTrust-RPL is detailed.

IoT nodes in the RPL network consist of parent, child, and neighbor nodes (Mayzaud, A., 2016) (Shreenivas, D., 2017). Hence, a sinkhole node can become a parent, child, or neighbor of any normal node in this network. Therefore, our method MDTrust-RPL is proposed to evaluate and detect this attack by classifying three types of trust: Trust I (hybrid-based parent trust), Trust II (fuzzy logic-based child trust), and Trust III (fuzzy logic-based neighbor trust). The architecture of MdTrust-RPL is described below, and the following section details our three types of trust.

Figure 3 illustrates the architecture of the MDTrust-RPL model. The MDTrust-RPL process is based on the received DIO message from a node in the RPL network and then checks whether the message was sent from a blacklisted node, in which case it will ignore that message. Otherwise, if the DIO message was sent from a parent node, it will use the parent trust technique, a hybrid between FLSs and subjective logic (Trust I); if the DIO message was sent from a child node, it will use the child trust technique-based FLS (Trust II); otherwise, it will use the neighbor trust technique-based FLS (Trust III).

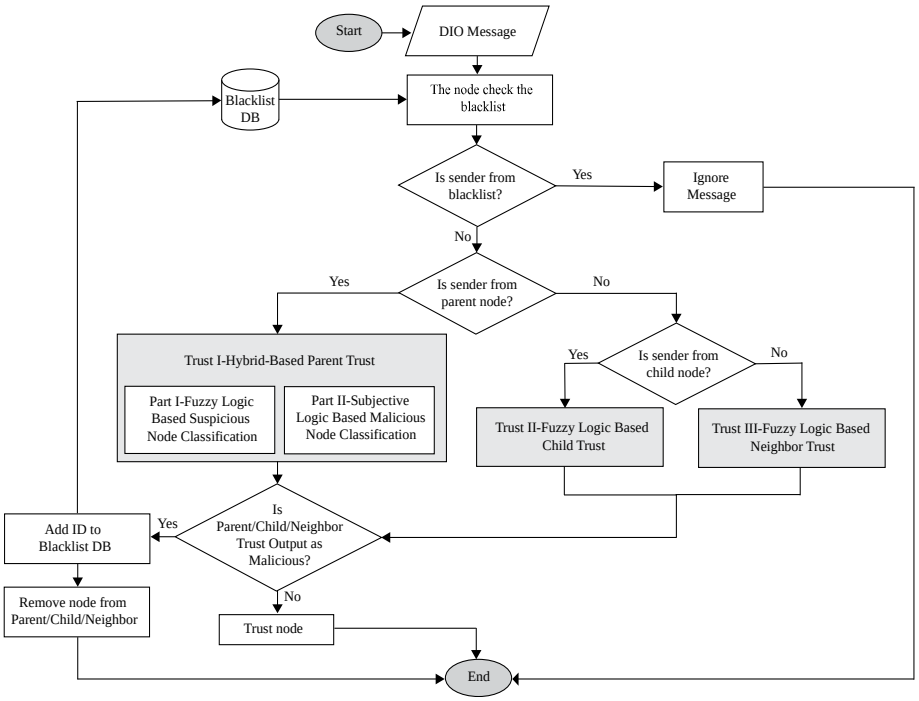


Figure 3: MDTrust-RPL Architecture

Trust I: Hybrid-Based Parent Trust

This section details the hybrid-based parent trust used to evaluate a node's parent.

In our method, after building the network, each child node in the RPL network records and stores the ID of its parent and then monitors abnormal behavior based on four trust metrics: trust weight (*TW*), *RANK*, packet delivery ratio of the node (*PDRN*), and packet delivery ratio of the parent (*PDRP*). This trust method is divided into two parts: part I is suspicious node classification, which applies an FLS (Mayzaud, A., 2016) to determine the adaptive weight based on a membership function to classify suspicious or normal nodes, and part II is malicious node classification, which applies SL theory to classify malicious and trusted nodes. The architecture of hybrid-based parent trust is described below, as well as in the following subsection, which details the two parts of suspicious and malicious node classification.

Figure 4 below illustrates the process of the parent trust architecture. This process is divided into two parts: suspicious node classification and malicious node classification. The first part starts by determining the *TW* variable based on three input parameters, the *DPR*, average end-to-end delay (A*EED*) and energy consumption (*EC*), which are input into Shannon's entropy method and then processed with multiattribute utility theory. After obtaining the *TW*, we combine it with the other three variables, consisting of *RANK*, *PDRN* and *PDRP*, and then input it into the FLS to obtain the weight output indicating whether a node is normal or suspicious. In the second part, we take the value of the FLS weight output (*FSCW*) and record positive and negative observations; if the *FSCW* weight indicates a suspicious node, this increases the negative observations (*NO*), and otherwise it increases the positive observations (*PO*). We use the FSCW value to compute the adaptive coefficient value (*K*) of SL by combining it with two other input variables, the *DAO* message and *NRE*. Then, this is input into a new FLS to yield another FLS weight output, the *K* coefficient. When we have obtained the *PO*, *NO*, and *K* values, we compute three weights of SL (*Wb*, *Wd*, *Wu*), and finally, malicious and trusted node decisions are made by comparing the three weights of SL; if Wd > Wb × Wu, the parent node is considered malicious; otherwise, it is a trusted node.

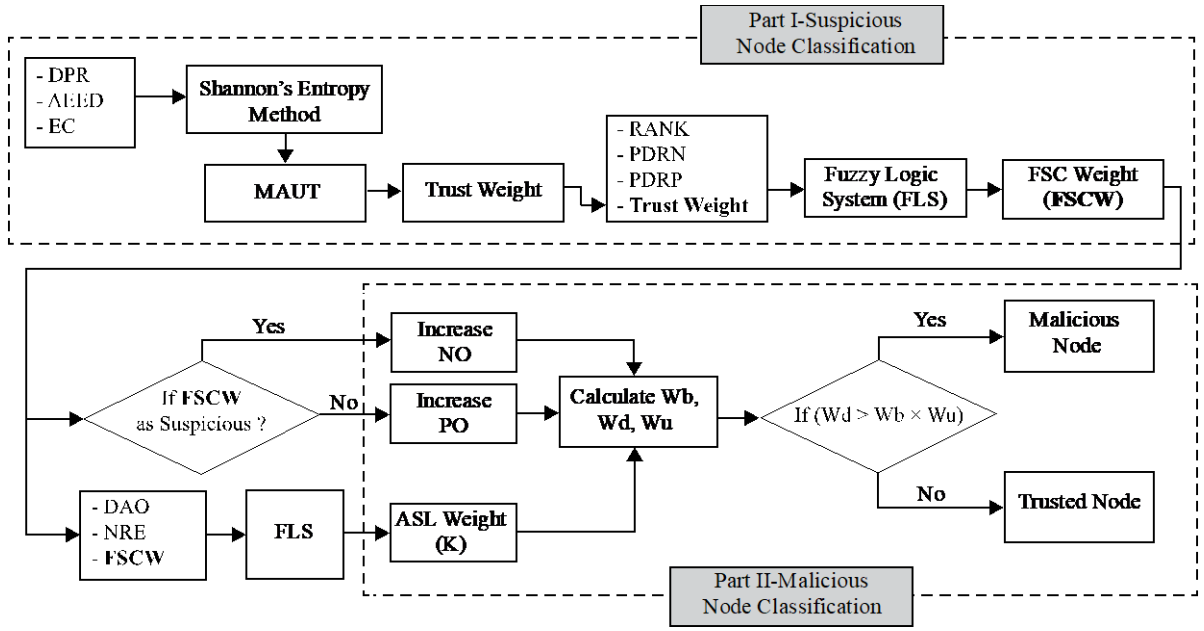


Figure 4: Hybrid-based Parent Trust Architecture

Part I-Fuzzy Logic-Based Suspicious Node Classification (FSC)

In this subsection, the suspicious node classification-based FLS is discussed, which obtains the adaptive output weight behavior of the existing parent nodes that indicates whether they are normal or suspicious. By mimicking the human thought process, the FLS provides an excellent solution to many control problems (Heng, S., 2020). Figure 5 illustrates the four primary components of the FLS: fuzzification, fuzzy rules, fuzzy inference, and defuzzification (Balakrishnan, B., 2017). The following steps describe the fuzzy logic component.

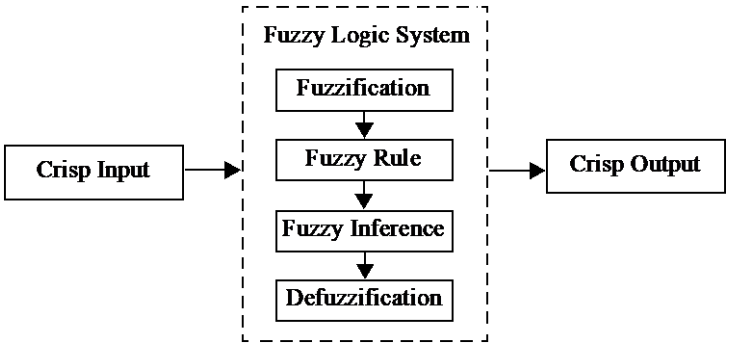


Figure 5: Fuzzy Logic System

- **Fuzzification**

The process of "fuzzification" involves assigning a variable of numerical input to fuzzy sets that have varying membership levels. Examples of the linguistic variables, the membership functions, and the fuzzification approach are given in the following subsections for each metric.

Linguistic Variables (Fuzzy Sets)

In contrast to other variables, the linguistic variable's values are words. There are two fuzzy sets for each input metric and the linguistic terms “Low” and “High”. Details regarding the input are provided in the following, such as the *TW*, *RANK*, and *PDR*, where the PDR consists of the *PDRN* and *PDRP*.

i. Trust Weight

The TW is the weight of trust between a node and its parent in an RPL network. In our case, TW is derived from Shannon’s entropy method and multiattribute utility theory (MAUT) as an FLS input parameter based on three important metrics: the *DPR*, *EC*, and A*EED*. The following details Shannon’s entropy method and MAUT, as well as the effects of sinkhole attacks on the three metrics and the calculation equation.

A. Shannon’s Entropy Method

Shannon entropy is a concept that is used to measure the uncertain occurrence of specific events given partial information about the system; in particular, this established approach is used to determine the weights for multiattribute decision-making problems (Shannon, C.E., 2009). The original Shannon entropy process can be stated as the following steps:

Step 1. Normalize the decision matrix (P_{ij})

$$P_{ij} = \frac{x_{ij}}{\sum_{i=1}^{m} x_{ij}} \qquad (1)$$

where x_{ij} is a value in the matrix that is used for normalization (i=1,2, …, m, j=1, 2, …, n, where n is the number of columns and m is the number of rows in the matrix) and $\sum_{j=1}^{m} x_{ij}$ is the sum of all values in row *i* and column *j*.

Step 2. Compute the entropy (h_i)

$$h_i = -(ln\, m)^{-1} \sum_{i=1}^{m} P_{ij}\, ln\, P_{ij}\,, j = 1,2, \dots, n \quad (2)$$

Step 3. Compute the degree of diversification (d_j)

$$d_j = 1 - h_j\,, j = 1,2, \dots, n \quad (3)$$

Step 4. Calculate the entropy criteria weight (W_j)

$$w_i = \frac{d_j}{\sum_{j=1}^{n} d_j}\,, j = 1, 2, \dots, n \quad (4)$$

B. Multiattribute Utility Theory

After applying Shannon's entropy method, we must order the weights to determine them in interval form. There are several ways to rank interval data, but in our proposed method, we employ MAUT (Jansen, S.J., 2011), which is an excellent way to formulate problems involving intuitive decision-making and multiple attributes. This is a normative approach based on how to value an entity, that is, to determine which entities are good, normal, and bad. The MAUT approach can be summarized as follows:

Step 5. Determine the normalized utility criteria value (u_i) as

$$u_i = \frac{x_{val} - x_{min}}{x_{max} - x_{min}} \quad (5)$$

where x_{val} denotes the current value of x, x_{min} is the minimum value of x and x_{max} is the maximum value of x.

Step 6. Compute the final value of each criterion (u) as

$$u = \sum_{j=1}^{n} w_i u_i \quad (6)$$

where w_i is the entropy criteria weight and u_i is the normalized utility criteria value.

In our approach, each attribute weight is calculated using Shannon's entropy method, and MAUT is utilized to determine the final weight of all attributes as a single weight, which is then input into the fuzzy logic system.

- *The DPR* is one of the crucial metrics that we can use to detect a sinkhole attack node. The sinkhole attack node will drop all packets that are sent across it when a blackhole assault is coupled and will drop some packets when coupled with a selective forward attack. Equation (7) is used to calculate the DPR.

$$DPR = \frac{PR - PF}{PR} \times 100 \quad (7)$$

where *PR* denotes packets received and *PF* denotes packets forwarded.

- *The AEED* includes *any* possible delay time experienced during retransmission, propagation and path detection (Prathapchandran, K., 2021). The AEED will grow since a sinkhole attack generates a large amount of traffic around the malicious node; the AEED can be calculated as follows.

$$AEED_t = \frac{\sum_{i=1}^{n} (TRP_i - TSP_i)}{TNP} \quad (8)$$

where TRP_i is the time for receiving the packet at the *i-th* time, TSP_i denotes the time for sending the packet at the *i-th* time and *TNP* is the total number of packets received.

- The *EC* is a crucial trust metric that is used to evaluate a node's parent. Typically, a sinkhole node uses the most energy in the network because many neighboring nodes choose this node as the parent node after this node broadcasts bogus information in an effort to attract its neighbors. This node uses more energy than necessary because there is so much traffic across this sinkhole node (Prathapchandran, K., 2021). High energy consumption is the term used to describe a node that uses more energy than is necessary.

Radio Transmission (Transmit), *Radio Listening (Listen)*, *CPU Power* and *Low Power Mode (LPM)* all contribute to energy consumption. Transmit and listen time denotes the period a radio is transmitting and receiving when the MCU is on; CPU time refers to the period the radio is on; and low-power mode refers to the period the radio is off (Mirshahjafari, S.M.H., 2019). The initial energy setup is 3 V (Kharche, S., 2016), and the node's energy consumption is calculated using (9).

$$EC\ (mJ) = (Transmit \times 19.5\,mA + Listen \times 21.5\,mA + CPU\ time * 1.8\,mA + LPM \times 0.0545\,mA) \times 3\,V \div (32768) \quad (9)$$

ii. Rank

This metric considers the node's ranking as well as the source message sender's ranking. The network's ranking will shift from high to low if sinkhole attacks are present, which is similar to the rank of sink nodes. In this case, we can use rank distance to predict sinkhole attacks when the rank comparison is far from normal. According to (Zaminkar, M., 2020), the normal or abnormal ranking of a node is calculated by the difference in ranking between the node and its parent (*DNP_Rank*) and the difference in ranking between the node and the source message sender (*DNS_Rank*). The calculation of rank prediction uses equations 10, 11, and 12.

$$DNP\_Rank = |PR - NR| \quad (10)$$

$$DNS_{Rank} = |SMSRank - NodeRank| \quad (11)$$

$$RANK = |DNS\_Rank - DNP\_Rank| \quad (12)$$

where *PR* denotes the parent rank, *NR* is the node rank and *SMSRank* refers to the source message sender rank.

iii. Packet Delivery Ratio (PDR)

The PDR is a measure of the efficiency of a communication network in delivering packets of data from a sender to a receiver; this is calculated by dividing the total number of transmitted packets by the total number of packets that reach their destination without error. A high packet delivery ratio indicates that the network is functioning well and that most packets are being successfully transmitted, while a low packet delivery ratio indicates that there are problems with the network that are causing some packets to be lost; it is also an important metric for identifying sinkhole attacks. The PDR calculation uses (13).

$$PDR_{(t)} = \frac{TPR_{(t)}}{TPS_{(t)}} \quad (13)$$

where *TPR(t)* denotes the total number of packets received at the destination at time "*t*" and *TPS(t)* denotes the total number of packets sent by the source node at time "t".

Membership Functions

The membership levels for a fuzzy set vary from 0 to 1, and the membership function (MF) improves the graphic depiction of the fuzzy set. The selection function of membership is influenced by numerous

factors, in addition to thorough simulations based on prior research, trial and error methods, and particular requirements. As fuzzy logic frequently uses two unique functions, L-functions and R-functions derived from the trapezoidal function, they are employed in our method. Equations 14 and 15 are used to compute the MF level, and Figure 6 shows a graphical representation of the four input metrics' membership functions and the parent trust suspicious node decision (FSC) output variable's MF.

$$low(x;a,b) = \begin{cases} 1, & x \le a \\ (b-x)/(b-a), & a < x < b \\ 0, & x \ge b \end{cases} \quad (14)$$

$$high(x;a,b) = \begin{cases} 1, & x \ge b \\ (x-a)/(b-a), & a < x < b \\ 0, & x \le a \end{cases} \quad (15)$$

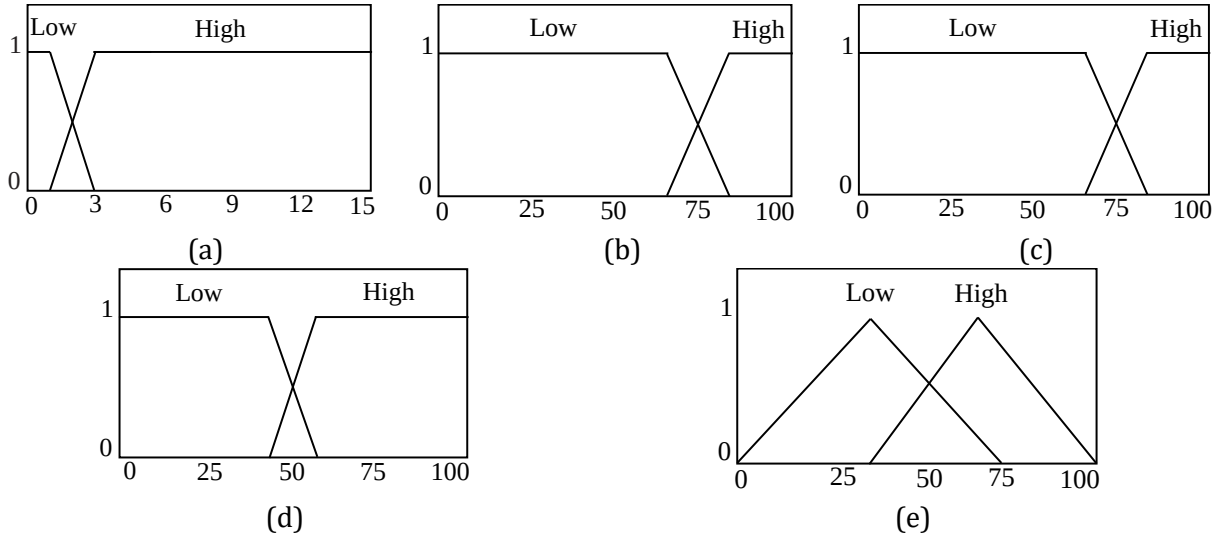


Figure 6: Membership Function: (a) RANK Distance, (b) Packet Delivery Ratio of the Node (PDRN), (c) PDR of the Parent (PDRP), (d) Trust Weight (TW), and (e) FSC Weight (FSCW)

Fuzzy Rules

This section lays out the guidelines for determining an output MF's MF level. Our initial experiment with every potential rule (probability distribution model) served as the basis for defining the rules and their corresponding conditions. The quantity of distinct input variables and MF phases determines the total number of rules.

IF-THEN rules are used throughout this stage of fuzzy logic to develop our proposed method. We generated 16 (2^4) rules based on two MFs (L and H) and four input variables. The fuzzy sets for the output variables "L" and "H" are found in the final column of Table 2's fuzzy rule base.

Table 2: Fuzzy Rule for Parent Trust Phase (L Represents Low and H Represents High)

No	IF (RANK)	AND (PDRN)	AND (PDRP)	AND (TW)	THEN (FSCW)
1	L	H	H	L	H
2	L	H	H	H	H
3	L	H	L	L	H

4	L	H	L	H	H
5	L	L	H	L	L
6	L	L	H	H	L
7	L	L	L	L	L
8	L	L	L	H	H
9	H	H	H	L	L
10	H	H	H	H	H
11	H	H	L	L	L
12	H	H	L	H	H
13	H	L	H	L	L
14	H	L	H	H	L
15	H	L	L	L	L
16	H	L	L	H	H

Fuzzy Inference

In this step, we employ the Mamdani Inference System (Mamdani, E.H., 1975). In particular, the identical fuzzy sets generated by each rule will be integrated using a fuzzy aggregation operator to create a single output fuzzy set; the logical operators "AND" and "OR" stand for "Minimum" and "Maximum,", respectively. For instance, the "Average" output fuzzy set incorporates numerous rules. Using the "Maximum" operator, we aggregate the results in our design on the presumption that four rules are matched. Given that four samples are matched for the "Average" output, the following formula illustrates how to calculate the "Average" fuzzy set output:

$$Average = Maximum\begin{pmatrix} Minimum(H_{RANK}, H_{PDRN}, H_{PDRP}, L_{TW}), \\ Minimum(H_{RANK}, H_{PDRN}, H_{PDRP}, H_{TW}), \\ Minimum(H_{RANK}, L_{PDRN}, H_{PDRP}, L_{TW}), \\ Minimum(H_{RANK}, L_{PDRN}, H_{PDRP}, H_{TW}) \end{pmatrix} \quad (16)$$

Defuzzification

This phase clarifies the defuzzification process and offers a thorough illustration of how to calculate the parent quality.

Defuzzification is the final stage of the FLS. In our proposed method, we use this stage to determine whether the current parent node is a suspicious or normal node by creating a single crisp value from the combined fuzzy output. The center of gravity (CoG) algorithm (Landi, G., 2002) is used as a defuzzification procedure to determine the aggregated rule's center (fuzzy set output). Equation 17 below illustrates the CoG calculation.

$$CoG = \frac{\sum_{i=1}^{N} A_i \times \bar{x}}{\sum_{i=1}^{N} A_i} \quad (17)$$

where N indicates the number of matched rules, A_i indicates the predicate truth in the domain and $\bar{x}$ represents the *i-th* rule's domain value or the fuzzy set's center of gravity for each output.

The process for determining the parent nodes as suspicious or normal using fuzzy logic is demonstrated in the following. Let us say, for example, that the following values are present in node the N4, which is the child of node N1 in Figure 2: RANK = 3, PDRN = 68, PDRP = 89 and TW = 47. The RANK metric fuzzy set is "H" according to the membership functions presented in Figure 6(a); thus, using equation (15), the membership function's outcome is 1. Additionally, the PDRN metric includes "L" and "H" fuzzy sets, and when using equations (14) and (15), the membership function returns 0.85

and 0.15. Considering that the membership function result is 1, after inserting 89 into equation (15), the PDRP metric is likewise found to contain fuzzy sets, such as "H", according to the membership functions presented in figure 6(c). The final metric, TW, has two fuzzy sets, "L" and "H", based on the membership functions presented in figure 6(d); hence, the membership function's outputs are 0.8 and 0.2 for that metric when applying formulas (14) and (15). With reference to Table 2's fuzzy rule basis, there are four fuzzy rules, 9, 10, 13 and 14, for the output quality that are precisely matched with the input's combinations of fuzzy sets. There are four rules that trigger the output fuzzy sets labeled "L" and "H". Using aggregation, for output quality, we find that the fuzzy rules, specifically rules 9, 10, 13, and 14, have values of 0.15, 0.15, 0.8, and 0.2, which belong to the "L", "H", "L", and "L" fuzzy set outputs, respectively. The aggregated values of the "H" and "L" distinctive output quality fuzzy sets in the penultimate phase are 0.15 and 0.8, respectively, especially when exact aggregation is used, as indicated in equation (16). By using equation (17) to substitute the aforementioned values and the output quality membership functions' center of gravity technique, the Defuzzifier crisp output (CoG) is accurately determined to be 32.89. When we obtain the *CoG* value, we compare this value with the middle value (between 0 and 100) that we use as the value of the threshold. When the value of *CoG* is less than the middle value, we consider that the weight of the parent is low, which we can consider suspicious; otherwise, it is high, which we can consider normal. Therefore, the parent node is evaluated, and based on this result, node N4 will consider its parent node (N1) to be suspicious.

Part II: Subjective Logic-Based Malicious Classification

The following section details the adaptive coefficient subjective logic with the FLS and malicious node classification using SL.

Adaptive Coefficient Subjective Logic with the FLS (ASLF)

In this stage, the *K* coefficient value of SL is considered by using the FLS to obtain the dynamic value. The use of the FLS in this subsection is similar to that in the above section on trust, except it involves only fuzzification, which includes linguistic variables, and the membership function and fuzzy rule evaluation detailed below.

Fuzzification

The following details the two components of fuzzification.

A. Linguistic Variables (Fuzzy Sets)

The input metric in this phase is the same as that in the suspicious node classification phase in terms of having two fuzzy sets and two linguistic terms, namely, "Low" and "High". However, in this phase, we have two input variables: the number of destination advertisement object (*DAO*) messages and the *NRE* of each node. The following section gives further details regarding the two input variables.

i. DAO Message

In the RPL network, to advertise routing information to its parent, the node is sent DAO messages. Afterward, that message is forwarded until it reaches the root node by going from the parent to its own parent, and so on. However, when a sinkhole attack occurs, the attacker node intercepts the DAO messages and falsely claims to have the quickest route to the root node in its advertising; this leads to an increased number of DAO messages being routed toward the attacker node and a decrease in the

number sending DAO messages to the network's other reputable nodes. Therefore, monitoring the different numbers of DAO messages every minute in the network can help to detect sinkhole attacks.

ii. Number of Route Entries (NRE)

In RPL networks, nodes maintain paths to the sink node and maintain routing tables to establish. When a sinkhole attack occurs, the attacker node attracts and reroutes the traffic toward itself with a claim to be the fastest route to the root; this leads to an increased NRE for the attacker node and a decrease in the NREs of other legitimate nodes. Therefore, monitoring the number of route entries can help detect sinkhole attacks in the network. For this reason, we select these two metrics as our input variables and combine them with the FSCW.

B. Membership Functions

The membership levels in this phase follow those of the suspicious node classification phase, in which the fuzzy set membership level varies from 0 to 1. Figure 7 shows a graphical representation of the MFs of the three-input metrics and the output weight of the ASLF MF; equations 14 and 15 are used to compute the MF level.

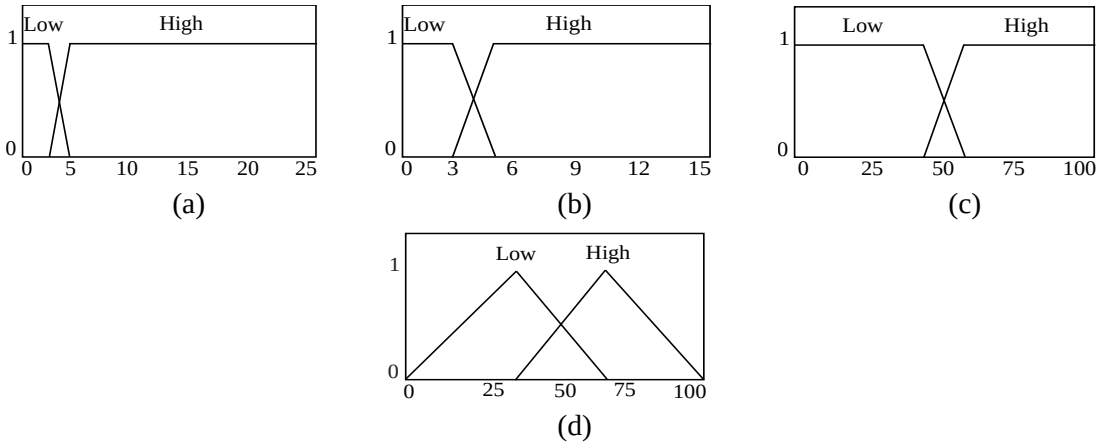


Figure 7: Membership Function: (a) the DAO, (b) the Number of Route Entries (NRE), (c) the FSCW, (d) the Output Weight of ASLF (AW)

Fuzzy Rules

In this phase, 8 (2^3) fuzzy rules were generated based on two MFs (L and H) and three input variables. In the rule base of fuzzy in Table 3, the last column contains the fuzzy sets for the output variables "L" and "H".

Table 3: Fuzzy Rules for the Adaptive SL Theory Coefficient

No	IF (DAO)	AND (NRE)	AND (FSCW)	THEN (AW)
1	H	H	L	H
2	H	H	H	H
3	H	L	L	H
4	H	L	H	L
5	L	H	L	H
6	L	H	H	L
7	L	L	L	L
8	L	L	H	L

The steps of the FLS in this part and the FLS in the suspicious node classification part have been detailed. Therefore, the remaining component follows the FLS in the suspicious classification phase. We assume that the FLS process has been performed. Thus, when we obtain the FLS output (AW), we take this value as the K coefficient of subjective logic, as detailed in the following section.

Subjective Logic

In this part, trusted and malicious nodes are classified based on the output result of the FLS performance in the suspicious node classification phase.

A statement is going to be classified as either true or untrue for the purpose of determining whether or not to trust it. However, we can only have an opinion on it, as it is difficult to know for sure whether it is true or not. SL is used in this phase to classify a parent as a trustworthy or malicious node. SL was derived from the Dempster-Shafer theory (Jøsang, A., 2001) (Jøsang, A., 1998); it abandons the additivity principle of probability theory, which states that all pairwise exclusive possibilities have a sum of probabilities that must equal one. This method has the advantage of allowing for the explicit expression of doubt about the probabilities—that is, the absence of evidence supporting any particular probability—by giving the entire frame a belief mass. Based on the opinion triangle, the trust values of SL are referred to as *Wb* (weight of belief), *Wd* (weight of disbelief) and *Wu* (weight of uncertainty). These variables have values that range from 0 to 1, and their total must equal 1, i.e., $Wb + Wd + Wu = 1$. The equations below are used to compute the three weights of SL.

$$Wb = \frac{PO}{PO + NO + K} \quad (18)$$

$$Wd = \frac{NO}{NO + PO + K} \quad (19)$$

$$Wu = \frac{K}{PO + NO + K} \quad (20)$$

where *PO* is the observations of positive, *NO* is the observations of negative and *K* is a coefficient value that is obtained from an adaptive coefficient SL with the FLS (*AW*), as described above (4.1.2.1).

The overall pseudocode for trust computation and malicious node detection is explained below and shown in algorithm 1.

The first line of the algorithm declares the malicious list as empty, the second line initializes *PO*, *NO* equals zero, and *K* is determined by adaptive subjective logic with fuzzy logic (*ASLF*). Line three is the while loop that processes the received DIO message, and line four is used to check that the DIO message was not sent from a node on the malicious list before going to line five to determine whether the node is suspicious or normal using the FLS. Lines six to ten determine whether to increase the *NO* or *PO* values based on the FLS output weight; if the FLS output weight is less than 50, we increase *NO*, and otherwise, we increase *PO*. Line 11 computes the weights of belief (*Wb*), disbelief (*Wd*), and uncertainty (*Wu*), and lines 12–14 determine the parent node's trust or maliciousness by comparing *Wb* with *Wd* and *Wu*. Additionally, line 15 is line 4's end condition, and line 16 is the end of line 3's while loop.

Algorithm 1: Trust Computation and Malicious Detection

1 Malicious List = ∅
2 **Initialize** PO = 0, NO = 0, K = ASLF
3 **While** DIO received D**o**
4 **If** DIO Received ∉ Malicious List **Then**

```
5            Compute Suspicious or Normal using FLS
6            If Suspicious (FLS output < 50) Then
7                  NO = NO + 1
8            Else
9                  PO = PO +1
10           End If
11           Compute W_b, W_d, W_u for parent node using Eqs. (18), (19), (20)
12           If (W_d > W_b × W_u) Then
13               Consider parent node as malicious
                 Add to malicious list
14           End If
15       End If
16   End While
```

Trust II: Fuzzy Logic-Based Child Trust (FL-CTrust)

In this section, FL-CTrust is detailed.

After building the network scenarios, each IoT parent node in the RPL records the ID of its child nodes and then begins to monitor the behavior of its child node when a DIO message is received. This trust approach uses the FLS technique to find the adaptive output weight behavior of a child node and identify it as a trusted or malicious node (Hui, H., 2019). The use of the FLS in this section is similar to that in the above trust section, except it involves only fuzzification (linguistic variables and the membership function) and fuzzy rule evaluation. The following describes the two steps of the FLS that are different: fuzzification and the fuzzy rules.

Fuzzification

The following details the two components of the fuzzification of child trust.

A. Linguistic Variables (Fuzzy Sets)

In this phase, we have two input variables, hop count (HC) and number of DIO output messages (DIO), and we have two fuzzy sets and two linguistic names, “Low” and “High”, which are the same as in the suspicious node classification phase. Further details about the two input variables are given below.

- *HC*: This indicates how many hops the destination or the DODAG root requires to be reached from the sender node (Darabkh, K.A., 2022). The parent node in the RPL network always has a lower HC than the child nodes, as well as every child node delivers packets to the sink via its parent. However, when the network experiences a sinkhole attack, the HC of the child node will become lower than that of its parent. Therefore, we can use this metric to predict sinkhole attacks.
- *Number of DODAG Information Object* (DIO): These are messages generated by the nodes in an RPL network. In normal network operation, the nodes are generated, and the number of DIO messages should remain relatively stable over time, assuming that the network topology remains unchanged. However, if a sinkhole attack occurs, where a malicious node attracts excessive traffic by pretending to be the route that leads closest to the root node, it may result in an abnormal increase in the number of DIO messages observed compared to that produced by normal nodes. The number of DIO messages is computed as the current DIO minus the DIO at the previous period divided by the number of neighbor nodes.

B. Membership Function

Figure 8 shows a graphical representation of how the two input metrics' membership functions work; the child trust decision output variable's MF and equations (14-15) are used to compute these MF levels.

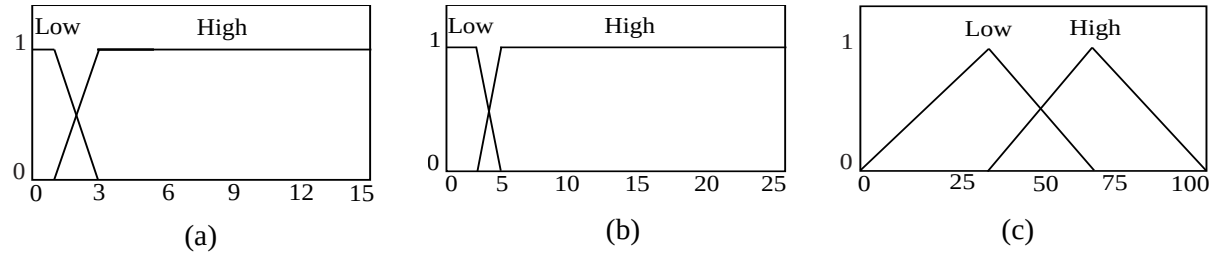


Figure 8: Membership Function: (a) the Hop Count, (b) the DIO Output Messages, (c) the Child Weight (CW)

Fuzzy Rules

The method for ascertaining the MF level of an output MF for the purpose of evaluating child trust is outlined in this section. L and H are the two membership functions and two crisp inputs that make up this FLS phase. Thus, there are 4 (2^2) possible fuzzy rules for this phase, as listed in Table 4.

Table 4: Fuzzy Rules of Child Trust

No	IF (HC)	AND (DIO)	THEN (CW)
1	L	L	H
2	L	H	H
3	H	L	H
4	H	H	L

The different components of the FLS in this child trust phase and the FLS in the suspicious node classification phase were described earlier. Therefore, the remaining component concerns the suspicious node classification phase of the FLS. We assume that the FLS process has been performed. Thus, when we obtain the FLS output (*CW*), we compare this value with the middle value (between 0 and 100) of the FLS output range that we use as the value of the threshold. When the *CW* value is less than the value of the middle, the weight of the child node is low enough that we can consider the child node to be malicious; otherwise, the child's weight is high enough that we consider the child node to be trustworthy.

Trust III: Fuzzy Logic-Based Neighbor Trust (FL-NTrust)

This section details FL-NTrust.

In FL-NTrust, the RPL node can monitor the behavior of all neighboring nodes, except child and parent nodes, using the FLS to predict the adaptive output weight. The FLS in this section also follows that in the above section (suspicious node classification) except for fuzzification (linguistic variables and the membership function) and fuzzy rule evaluation. The following details the two steps of the FLS that are different from those of the FLS in the other phases.

Fuzzification

The following details the two components of the fuzzification of neighbor trust.

A. Linguistic Variables (Fuzzy Sets)

In this phase, we have three fuzzy sets and three linguistic names, namely, “Low”, “Medium” and “High”, which are different from the others (parent and child trust phases), and we have two input variables for neighbor trust: the rank difference between the neighbor node and the average ranking of all neighbors (RDNA) and the rank difference between the neighbor node and its parent (RDNP). Further details about the two input variables are given below.

To obtain the RDNA and RDNP values, each RPL node records data from neighboring nodes, such as the node rank, node ID, parent rank, and parent ID, when it receives a DIO message and stores them in a table. Equations (21) and (22) are used to compute RDNA and RDNP; note that before inputting RDNA and RDNP into the FLS, our proposed method performs normalization to obtain a value in the range -100 to 0 or 0 to 100.

$$RDNA = NR - ARA \quad (21)$$

$$RDNP = NR - PRN \quad (22)$$

where *NR* denotes the neighbor rank, *ARA* is the average rank of all neighbor nodes and *PRN* denotes the parent rank of the neighbor node.

B. Membership Function

The membership levels in this section follow those in the suspicious node classification section, which form a fuzzy set of values from 0 to 1. In this phase, our method employed the L-functions, triangular functions, and R-functions for the membership selection function. Equations (23-25) below are used to compute the MF level, and Figure 9 shows a graphical representation of the two-input metrics’ MF and the neighbor trust decision output variable’s MF.

$$low(x;a,b,c) = \begin{cases} 1, & x \le a \\ (b-x)/(b-a), & a < x < b \\ 0, & x \ge b \end{cases} \quad (23)$$

$$medium(x;a,b,c) = \begin{cases} 0, & x \ge c \\ \frac{(x-b)}{c-b}, & b < x < c \\ 1, & x = a \\ \frac{(x-a)}{b-a}, & a < x < b \\ 0, & x \ge a \end{cases} \quad (24)$$

$$high(x;a,b,c) = \begin{cases} 1, & x \ge c \\ (x-b)/(c-b), & c < x < c \\ 0, & x \le b \end{cases} \quad (25)$$

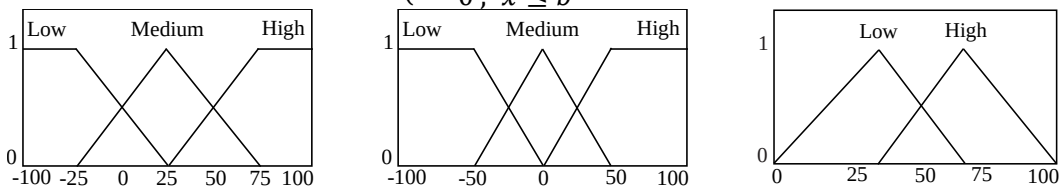


Figure 9: Membership Function: (a) the RDNA, (b) the RDNP, (c) the Neighbor Weight (NW)

Fuzzy Rules

This section outlines the formula for calculating an output MF's MF level to evaluate neighbor trust. The L, M, and H membership functions make up this FLS phase’s three membership functions and two crisp inputs. As a result, Table 5’s list of feasible fuzzy rules for this phase includes 9 (3^2).

Table 5: Fuzzy Rules for NT-FLS (M Represents Medium)

No	IF (RDNA)	AND (RDNP)	THEN (NW)
1	L	L	L
2	L	M	L
3	L	H	H
4	M	L	H
5	M	M	H
6	M	H	H
7	H	L	H
8	H	M	H
9	H	H	H

The differences between the FLS in this neighbor trust phase and the FLS in the suspicious classification phase were described previously. Hence, it remains to follow the FLS in the suspicious classification phase. We assume that the FLS process has been performed. When we obtain the FLS output (NW), we compare this value with the middle value (between 0 and 100) of the FLS output range that we use as the value of the threshold. When the NW value is less than the value of the middle, the weight of the neighbor node is low enough that we can consider the neighbor malicious; otherwise, the weight of the neighbor node is high enough that we consider the neighbor node trustworthy.

After using the trust architecture by considering parent, child and neighbor trust, the RPL node adds the IDs of malicious nodes to the blacklist when parent, child or neighbor nodes are detected as malicious to avoid future communication and then forward the IDs of malicious nodes to the sink node. In this case, the sink node can alert all nodes in the network of the existence of malicious nodes, and every node will avoid malicious nodes. Attacks through sinkholes will thus be avoided.

5 Performance Evaluation

The simulation setup, results, and discussion of our proposed MDTrust-RPL approach to detecting sinkhole attacks are provided in this section.

Simulation Setup

The ContikiOS 3.0 and Cooja simulators were used to evaluate the MDTrust-RPL model. In addition, the SoS-RPL (Zaminkar, M., 2020) and RFTrust models (Prathapchandran, K., 2021) were assessed by the Cooja simulator. Z1 (sensor nodes) was the mote type utilized by the MDTrust-RPL model. Table 6 shows the simulation parameters for the proposed MDTrust-RPL model, which was tested with two different scenarios with a total simulation time of 3600 s and a set data packet size of 64 bytes.

- Scenario 1: Randomly deploy 50 nodes in a network area of 100 m × 100 m and increase the number of malicious nodes from 10% to 50%.
- Scenario 2: Randomly deploy 100 nodes in a network area of 100 m × 100 m and increase the number of malicious nodes from 10% to 50%.

For each experiment, our simulation was run 10 times, after which an average number was chosen to represent the final experimental result, and the standard deviation was calculated using a 95% confidence interval (CI).

Six key performance criteria were assessed to contrast the suggested method with the related techniques SoS-RPL (Zaminkar, M., 2020) and RFTrust (Prathapchandran, K., 2021).

- The proportion of legitimate nodes that were mistakenly identified as malicious nodes in relation to all valid nodes is known as the false positive rate (FPR) (Prathapchandran, K., 2021).

$$FPR = \frac{FP}{FP + TN} \qquad (26)$$

where *TN* is the true negative number and *FP* is the false positive number.

- The false negative rate (FNR) measures the proportion of harmful nodes that were mistakenly identified as normal nodes in comparison to all malicious nodes (Prathapchandran, K., 2021).

$$FNR = \frac{FN}{FN + TP} \qquad (27)$$

where the values of TP and FN are the number of true positives and false negatives, respectively.

- A communications network's ability to deliver data packets from a sender to a receiver is measured by the PDR. It is calculated by taking the all number of transmitted packets and dividing it by the packets number that reach their destination successfully (Prathapchandran, K., 2021).
- The total number of data packets carried through a communication channel in one time unit, or the average quantity of data successfully transferred each second, is known as the throughput. Bits per second (bit/s or bps) are the standard units used to describe it (Prathapchandran, K., 2021).
- The average delay, also known as latency, is a measure of the packet's time of data to be transmitted by a sender to a receiver in a communications network. It is determined by dividing the total time required to deliver all packets by the number of packets delivered. Given that it can significantly affect user experience, the average delay is a crucial indicator for assessing a network's performance (Prathapchandran, K., 2021).
- Energy consumption refers to the amount of electrical power used by the node over a specified period and is expressed in millijoules (mJ). The energy consumption is computed using equation 9.

Table 6: Lists the MDTrust-RPL Model's Suggested Simulation Parameters

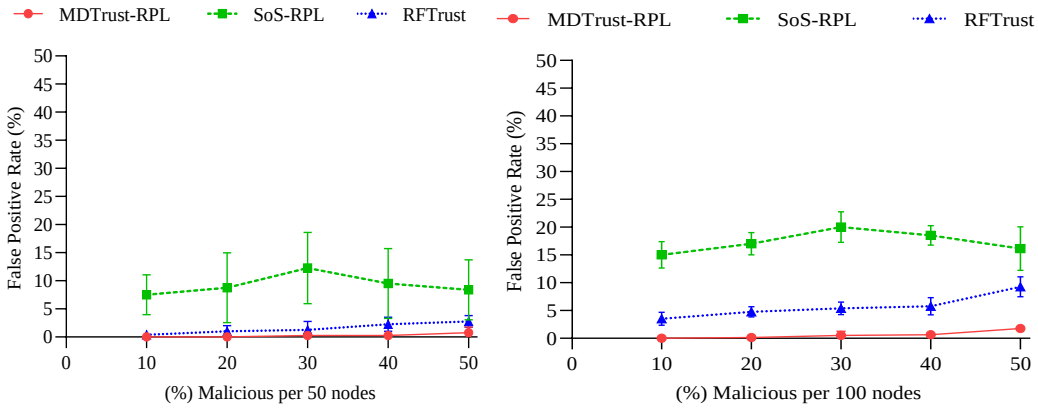
System Parameters	**Values**
Number of sink nodes	1
Number of nodes	50, 100
Percentage of malicious (%)	10, 20, 30, 40, 50
Time of Simulation	3600s
Network coverage area	100 m ×100 m
Mote type	Z1 mote
Communication range	50 m
Data packet size	64 bytes

Simulation Results and Discussion

This subsection presents and discusses the simulation results. The efficiency was evaluated with respect to the FPR, FNR, PDR, throughput, average delay and energy consumption.

In two separate scenarios, Figure 10 displays the rate of false positives for the MDTrust-RPL, SoS-RPL, and RFTrust models for varying percentages of malicious nodes. In both cases, the malicious number increased from 10% to 50%. The average FPR of the proposed MDTrust-RPL model was 0.75% and 1.75% for a 50% malicious rate, while the SoS-RPL model reached a maximum, at a 30% malicious rate, of 12.25% and 20%. Additionally, the RFTrust model reached a maximum of 2.75% and 9.25% for 50% malicious nodes in the two scenarios.

The proposed model in scenario 1 has a lower FPR than that in scenario 2 because scenario 1 has fewer nodes and each node has a higher chance of connecting with trustworthy nodes. The network may have more reliable communication paths, and the trust-based detection mechanism can successfully identify malicious nodes since trust metrics can properly distinguish between malicious and nonmalicious nodes, resulting in a decreased false positive rate. Scenario 2 increases the node number while keeping the same malicious percentage, making the network denser. Malicious nodes and routes that cross through them are more likely when there are more nodes. The trust-based detection method makes it difficult to appropriately evaluate each node's behavior and identify malicious nodes due to the increased density. The other reason that our proposed model has a lower FPR than the other models in both scenarios is because the proposed MDTrust-RPL model uses a hybrid technique as a double inspector to evaluate malicious nodes and uses multiple trust metrics that combine significant trust metrics that are the most affected by sinkhole attacks. RANK, PDRN, PDRP and the new trust metric TW are used for evaluating parent nodes, DIO and HC for evaluating child nodes, and RDNA and RDNP for evaluating neighbor nodes. Therefore, based on these significant metrics, our proposed MDTrust-RPL model specifically detects sinkhole attack behavior and has a lower rate of false positives than SoS-RPL and RFTrust. In contrast, SoS-RPL uses one metric, rank distance, to identify sinkhole nodes. SoS-RPL performs highly confusing detection (many false positives) because a single metric cannot capture all the relevant factors that differentiate legitimate nodes from malicious nodes. RFTrust measures the PDR, delay, energy, and honesty trust metrics, but these metrics are affected by both normal and sinkhole nodes when sinkhole attacks occur in the network, making it difficult to classify the different behaviors of trusted and sinkhole nodes. Thus, RFTrust has a higher rate of false positives than the proposed MDTrust-RPL model.



(a) (b)

Figure 10: a. False Positive Rate of 50 Nodes vs. Malicious Nodes (%), b. False Positive Rate of 100 Nodes vs. Malicious Nodes (%)

The FNRs of the MDTrust-RPL, SoS-RPL, and RFTrust approaches are depicted in Figure 11 under varying percentages of malicious nodes in both situations. The percentage of malicious nodes rises to 50% from 10%, and the proposed MDTrust-RPL model has average false negative rates of 0.50% and 1.25%, while the rates for SoS-RPL are 3.85% and 5.88%, and those of the RFTrust models are 28.25% and 19.75%, respectively. Comparing the proposed approach to existing models, the proposed MDTrust-RPL approach has a lower false negative rate because our proposed model uses multidirectional trust detection; this enables our model to detect malicious nodes faster than the other two models, which use the one-directional trust detection technique.

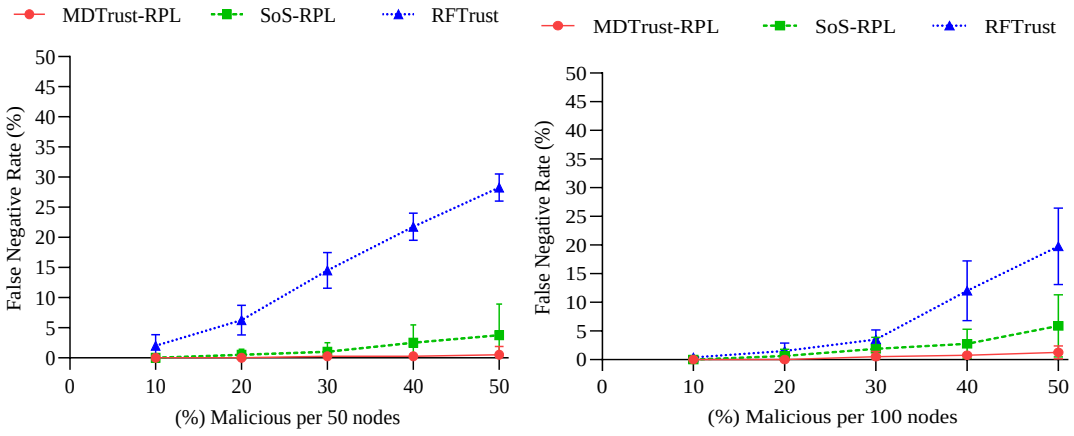


Figure 11: a. False Negative Rate of 50 Nodes vs. Malicious Nodes (%), b. False Negative Rate of 100 Nodes vs. Malicious Nodes (%)

Figure 12 shows the PDRs in scenarios 1 and 2 of the MDTrust-RPL, SoS-RPL, and RFTrust models. The proposed approach outperforms the other two models in both scenarios in terms of PDR. It has a 90.67% and 79.74% packet delivery ratio even with 50% malicious nodes in the two situations, compared to 65.02% and 43.52% for SoS-RPL and 63.83% and 36.22% for RFTrust. The proposed MDTrust-RPL model uses multidirectional trust detection to identify malicious parent, child, and neighbor nodes by classifying their behavior as parents, children, and neighbors. Therefore, this approach can quickly identify and eliminate sinkhole nodes. The proposed MDTrust-RPL model therefore boosts PDR by routing trusted nodes, unlike the SoS-RPL and RFTrust models, which consider only one-direction detection that focuses on malicious parent nodes. Thus, the parent node takes longer or is unable to identify a sinkhole node when it is a malicious child or neighbor node, resulting in a lower PDR for these two models.

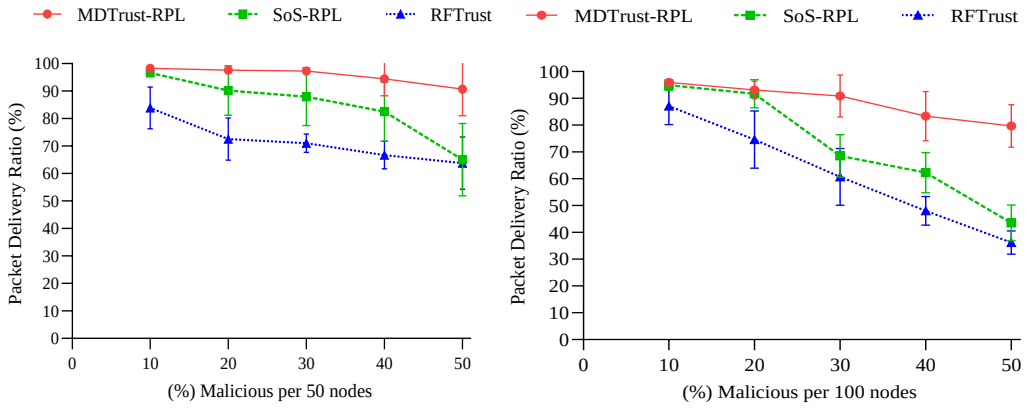


Figure 12: a. Packet Delivery Ratio of 50 Nodes vs. Malicious Node (%), b. Packet Delivery Ratio of 100 Nodes vs. Malicious Node (%)

In both instances, the MDTrust-RPL model has a greater throughput than the SoS-RPL and RFTrust models (Figure 13). The proposed MDTrust-RPL model has higher throughput than the other models (SoS-RPL and RFTrust) because MDTrust-RPL uses a multidirectional detection technique by including

a hybrid paradigm between FLS and SL and a ranking-based FLS detection method for trust prediction. The MDTrust-RPL model accurately identifies harmful nodes, removing sinkhole nodes from the network. Thus, the MDTrust-RPL model has good average throughput.

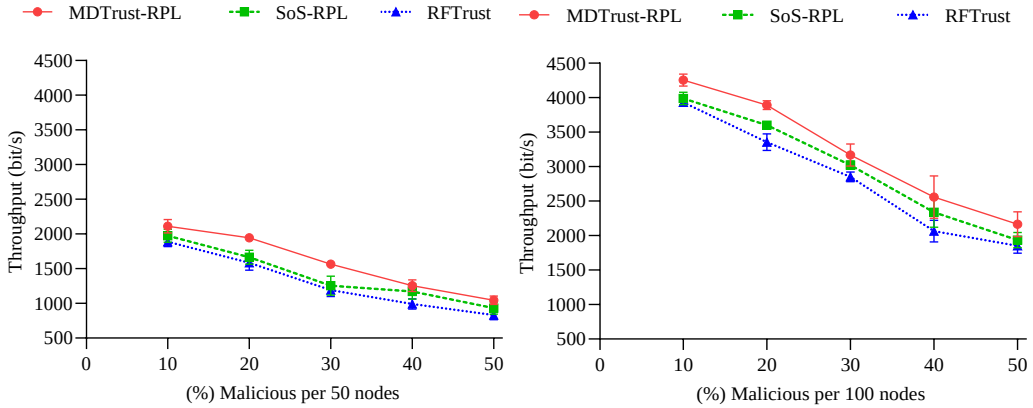


Figure 13: a. Throughput of 50 Nodes vs. Malicious Node (%), b. Throughput of 100 Nodes vs. Malicious Node (%)

The impact of different paradigms (MDTrust-RPL, SoS-RPL, and RFTrust) with various malicious node numbers on the average latency is illustrated in Figure 14 below. The suggested MDTrust-RPL model locates and eliminates sinkhole nodes at an earlier stage than the other models, resulting in a reduced average latency for the proposed model than for the other models. The proposed MDTrust-RPL model's average delay is reduced, while in the other models, the presence of sinkholes persists, increasing the average delay.

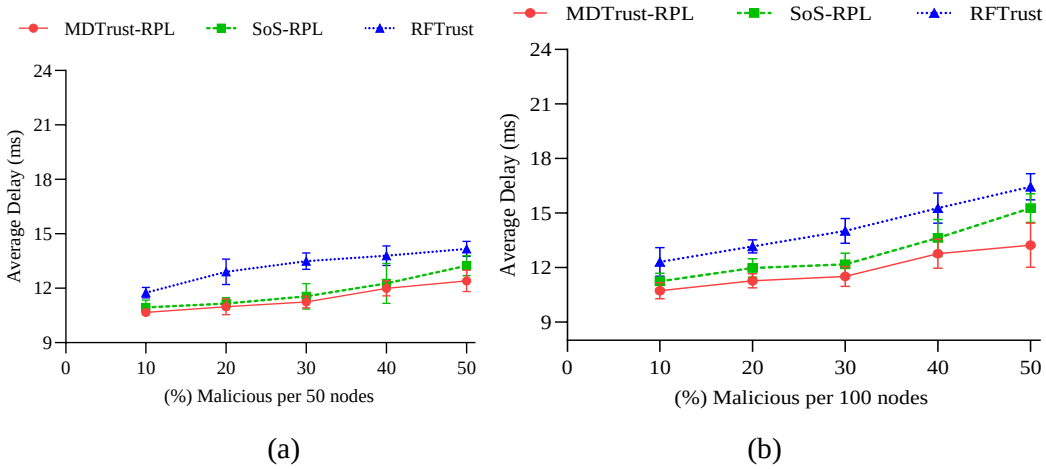
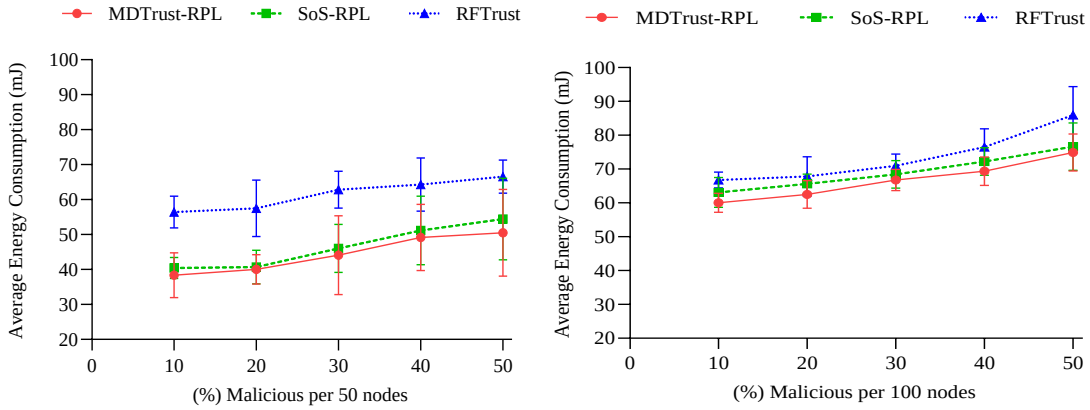


Figure 14: a. Average Delay of 50 Nodes vs. Malicious Node (%), b. Average Delay of 100 Nodes vs. Malicious Node (%)

The average energy use is the final comparison. The average energy usage of various models with various numbers of nodes is illustrated in Figure 15. The energy consumption of all models rises as the proportion of malicious nodes rises. The suggested MDTrust-RPL model uses less energy than the other models, despite this. The MDTrust-RPL model consumed 50.48 mJ and 74.89 mJ in scenarios 1 and 2 for a 50% attacker rate, SoS-RPL consumed 54.43 mJ and 76.65 mJ, and RFTrust consumed 66.55 mJ

and 85.97 mJ. The reason that the proposed MDTrust-RPL model consumed less energy than the other two models is that the MDTrust-RPL model uses a unique methodology that can detect the presence of sinkhole attacks in any location in the RPL network faster than the SoS-RPL and RFTrust models because these two models consider only one-directional detection and take more time to detect malicious attacks in other directions in the network. Due to this limitation, the SoS-RPL and RFTrust models consume more energy because sinkhole attacks persist longer in the network.



(a) (b)

Figure 15: a. Average Energy Consumption of 50 Nodes vs. Malicious Node (%), b. Average Energy Consumption of 100 Nodes vs. Malicious Node (%).

In summary, the simulation results of our proposed MDTrust-RPL are better than those of SoS-RPL and RFTrust on the six-comparison metrics in both scenarios. The relevance of these six comparisons is that they show the effect of FPR and FNR on PDR, throughput, average delay, and EC; hence, our technique's consideration of the FPR and FNR is significant. The proposed method uses a hybrid technique and combines multiple metrics that are the most effective for evaluating node behavior to detect sinkhole nodes, giving the proposed MDTrust-RPL a lower FPR than SoS-RPL and RFTrust. Furthermore, the proposed multidirectional trust-based detection method is crucial because it can improve the efficiency of detection faster than one-direction detection, which gives our proposed method a lower FNR, higher PDR, higher throughput, and lower EED and EC than the other two models (SoS-RPL, RFTrust). Another feature of the discussion concerns the varying number of malicious nodes. As shown in our results, when the percentage of malicious nodes increases, the network's performance increases or decreases; the reason for this is that as the number of malicious nodes in an RPL network increases, the potential for malicious activity increases. With a larger number of malicious nodes, there are more entities capable of launching attacks or engaging in disruptive activities, which leads to an increase in FPR, FNR, EED, and EC while decreasing PDR and throughput.

6 Conclusions and Future Work

The security and dependability of RPL networks are seriously threatened by sinkhole attacks, and detecting them is critical to ensuring the continued operation of these networks. In this research, the proposed MDTrust-RPL detects sinkhole attacks under the RPL routing protocol. Our method has a novel architecture that considers parent, child, and neighbor trust directions, thereby capturing the various forms of directional trust in an RPL network. We note neighbor trust as a significant factor among these three directions because malicious neighbor nodes can become child or parent nodes; hence,

trust in neighbor nodes is significant in avoiding side effects for future communication and reducing the malicious detection delay. In addition, we present a hybrid detection system that combines an FLS and SL to increase detection efficiency and reduce false positives. We also propose a new trust weight variable based on Shannon's entropy method and multiattribute utility theory. In addition, the constant coefficient value of SL theory is replaced with an adaptive SL coefficient that dynamically adapts to network conditions.

Our technique is lightweight and does not require significant modifications to the existing RPL network infrastructure; therefore, it can be easily deployed in real-world scenarios to enhance the security of RPL networks against sinkhole attacks. The experimental evaluation compares the proposed technique with existing techniques for an increasing malicious node rate of 10% to 50%. The results show that the proposed technique achieves an average lower false positive rate (11% and 4.38%), false negative rate (3.94% and 23.13%), average delay (1.44 ms and 2.49 ms), and energy consumption (2.36 mJ and 13.06 mJ) and a higher packet delivery ratio (30.93% and 35.18%) and throughput (170 bits and 262 bits) at a malicious node rate of 50% compared with those of SoS-RPL and RFTrust.

In future work, we will extend the current research and focus on the possibility of implementing our proposed scheme on real devices as well as in dynamic or mobile networks.

Author Contributions: Conceptualization, methodology, formal analysis, investigation, and writing original draft, S.K.; writing-review and editing: S.K., C.S-I., P.A., and P.M. All authors have read and agreed to the published version of the manuscript.

Acknowledgments: This work was supported by a Royal Scholarship under Her Royal Highness Princess Maha Chakri Sirindhorn Education Project to the Kingdom of Cambodia and by the College of Computing, Khon Kaen University, Thailand.

Conflicts of Interest: The authors declare no conflicts of interest.

References

[1] Abbas, N., Asim, M., Tariq, N., Baker, T., & Abbas, S. (2019). A mechanism for securing IoT-enabled applications at the fog layer. *Journal of Sensor and Actuator Networks*, *8*(1), 1-18.

[2] Adekanbi, M.L. (2021). Optimization and digitization of wind farms using internet of things: A review. *International Journal of Energy Research*, *45*(11), 15832-15838.

[3] Alzubaidi, M., Anbar, M., Chong, Y.W., & Al-Sarawi, S. (2018). Hybrid monitoring technique for detecting abnormal behaviour in rpl-based network. *Journal of Communication*, *13*(5), 198-208.

[4] Balakrishnan, B., & Balachandran, S. (2017). FLECH: fuzzy logic-based energy efficient clustering hierarchy for nonuniform wireless sensor networks. *Wireless Communications and Mobile Computing*, *2017*.

[5] Burange, A.W., & Deshmukh, M.V. (2021). Detection of Rank, Sybil and Wormhole Attacks on RPL based Network using Trust Mechanism. *CEUR Workshop Proceedings*, 3283, 152-162.

[6] Darabkh, K.A., Al-Akhras, M., Ala'F, K., Jafar, I.F., & Jubair, F. (2022). An innovative RPL objective function for broad range of IoT domains utilizing fuzzy logic and multiple metrics. *Expert Systems with Applications*, *205*.

[7] Darabkh, K.A., Zomot, J.N., Al-qudah, Z., & Ala'F, K. (2022). Impairments-aware time slot allocation model for energy-constrained multi-hop clustered IoT nodes considering TDMA and DSSS MAC protocols. *Journal of Industrial Information Integration*, *25*.

[8] Ghaleb, B., Al-Dubai, A.Y., Ekonomou, E., Alsarhan, A., Nasser, Y., Mackenzie, L.M., & Boukerche, A. (2018). A survey of limitations and enhancements of the ipv6 routing protocol

for low-power and lossy networks: A focus on core operations. *IEEE Communications Surveys & Tutorials*, *21*(2), 1607-1635.

[9] Haq, S.U., & Abbas, A.M. (2022). Advancements in Intrusion Detection Systems for Internet of Things Using Machine Learning. *In IEEE 5th International Conference on Multimedia, Signal Processing and Communication Technologies (IMPACT)*, 1-5.

[10] Heng, S., Aimtongkham, P., Vo, V.N., Nguyen, T.G., & So-In, C. (2020). Fuzzy adaptive-sampling block compressed sensing for wireless multimedia sensor networks. *Sensors*, *20*(21), 1-29.

[11] Hentout, A., Maoudj, A., & Aouache, M. (2023). A review of the literature on fuzzy-logic approaches for collision-free path planning of manipulator robots. *Artificial Intelligence Review*, *56*(4), 3369-3444.

[12] Hossain, M.S., Rahman, M., Sarker, M.T., Haque, M.E., & Jahid, A. (2019). A smart IoT based system for monitoring and controlling the sub-station equipment. *Internet of things*, *7*.

[13] Hui, H., An, X., Wang, H., Ju, W., Yang, H., Gao, H., & Lin, F. (2019). Survey on Blockchain for Internet of Things. *Journal of Internet Services and Information Security (JISIS)*, *9*(2), 1-30.

[14] Jahangeer, A., Bazai, S.U., Aslam, S., Marjan, S., Anas, M., & Hashemi, S.H. (2023). A Review on the Security of IoT Networks: From Network Layer's Perspective. *IEEE Access*.

[15] Jamil, A., Ali, M.Q., & Alkhalec, M.E.A. (2021). Sinkhole Attack Detection and Avoidance Mechanism for RPL in Wireless Sensor Networks. *Annals of Emerging Technologies in Computing (AETiC)*, *5*(5), 94-101.

[16] Jansen, S.J., Coolen, H.C., & Goetgeluk, R.W. (Eds.). (2011). *The measurement and analysis of housing preference and choice*. Springer Science & Business Media.

[17] Jøsang, A. (2001). A logic for uncertain probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *9*(03), 279-311.

[18] Jøsang, A. (2016). Subjective logic, *3*, 1–8. Cham: Springer.

[19] Jøsang, A., & Knapskog, S. (1998). A metric for trusted systems. *In Proc. 21st NIST-NCSC National Information Systems Security Conference*, 16-29.

[20] Karthikeyan, M.K.P.P.P., & Swathy, M.A. (2019). A Research of Dealing with Sinkhole Attacks to Prevent Security in WSN. 5, 2143–2148.

[21] Khan, J.Y. (2019). Introduction to IoT Systems. *Internet of Things (IoT): Systems and Applications*.

[22] Kharche, S., & Pawar, S. (2016). Node level energy consumption analysis in 6LoWPAN network using real and emulated Zolertia Z1 motes. *In IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, 1-5.

[23] Kumar, B.S. (2022). Smart farming system using ai. *13*, 419–428.

[24] Landi, G. (2002). Properties of the center of gravity as an algorithm for position measurements. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, *485*(3), 698-719.

[25] Lodhi, M.A., Rehman, A., Khan, M.M., & Hussain, F.B. (2015). Multiple path RPL for low power lossy networks. *In IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob)*, 279-284.

[26] Mamdani, E.H., & Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International journal of man-machine studies*, *7*(1), 1-13.

[27] Maraveas, C., Piromalis, D., Arvanitis, K.G., Bartzanas, T., & Loukatos, D. (2022). Applications of IoT for optimized greenhouse environment and resources management. *Computers and Electronics in Agriculture*, *198*.

[28] Mayzaud, A., Badonnel, R., & Chrisment, I. (2016). A Taxonomy of Attacks in RPL-based Internet of Things. *International Journal of Network Security*, *18*(3), 459-473.

[29] Mirshahjafari, S.M.H., & Ghahfarokhi, B.S. (2019). Sinkhole+ CloneID: A hybrid attack on RPL performance and detection method. *Information Security Journal: A Global*

Perspective, *28*(4-5), 107-119.

[30] Patel, B., & Shah, P. (2021). Direct Neighbour Sink Reputed Trust Based Intrusion Detection System to Mitigate Sinkhole Attack in RPL for IoT Networks. *Journal of Engineering Science & Technology Review*, *14*(1), 38-45.

[31] Prathapchandran, K., & Janani, T. (2021). A trust aware security mechanism to detect sinkhole attack in RPL-based IoT environment using random forest–RFTRUST. *Computer Networks*, *198*.

[32] Raza, S., Wallgren, L., & Voigt, T. (2013). SVELTE: Real-time intrusion detection in the Internet of Things. *Ad hoc networks*, *11*(8), 2661-2674.

[33] Rehman, A.U., Rehman, S.U., & Raheem, H. (2019). Sinkhole attacks in wireless sensor networks: A survey. *Wireless Personal Communications*, *106*, 2291-2313.

[34] Salehi, S.A., Razzaque, M.A., Naraei, P., & Farrokhtala, A. (2013). Detection of sinkhole attack in wireless sensor networks. *In IEEE international conference on space science and communication (IconSpace)*, 361-365.

[35] Shannon, C.E., (2009). Probability and Information Theory. *Applied and Numerical Harmonic Analysis*, 63–99.

[36] Shreenivas, D., Raza, S., & Voigt, T. (2017). Intrusion detection in the RPL-connected 6LoWPAN networks. *In Proceedings of the 3rd ACM international workshop on IoT privacy, trust, and security*, 31-38.

[37] Suo, H., Wan, J., Zou, C., & Liu, J. (2012). Security in the internet of things: a review. *In IEEE international conference on computer science and electronics engineering*, *3*, 648-651.

[38] Tahir, S., Bakhsh, S.T., & Alsemmeari, R.A. (2019). An intrusion detection system for the prevention of an active sinkhole routing attack in Internet of things. *International Journal of Distributed Sensor Networks*, *15*(11), 1-10.

[39] Tang, J., Yu, S., Liu, F., Chen, X., & Huang, H. (2019). A hierarchical prediction model for lane-changes based on combination of fuzzy C-means and adaptive neural network. *Expert systems with applications*, *130*, 265-275.

[40] Vasseur, J., Agarwal, N., Hui, J., Shelby, Z., Bertrand, P., & Chauvenet, C. (2011). RPL: The IP routing protocol designed for low power and lossy networks. *Internet Protocol for Smart Objects (IPSO) Alliance*, *36*, 1-20.

[41] Winter, T., Thubert, P., Brandt, A., Hui, J., Kelsey, R., Levis, P., & Alexander, R. (2012). *RPL: IPv6 routing protocol for low-power and lossy networks*.

[42] Zadeh, (1965). Fuzzy Sets. *Computer Civil Build Engineering Proceedings*, *353*, 1562–1569.

[43] Zahra, F., Jhanjhi, N.Z., Brohi, S.N., Khan, N.A., Masud, M., & AlZain, M.A. (2022). Rank and wormhole attack detection model for RPL-based internet of things using machine learning. *Sensors*, *22*(18), 1-17.

[44] Zaminkar, M., & Fotohi, R. (2020). SoS-RPL: securing internet of things against sinkhole attack using RPL protocol-based node rating and ranking mechanism. *Wireless Personal Communications*, *114*(2), 1287-1312.

[45] Zaminkar, M., Sarkohaki, F., & Fotohi, R. (2021). A method based on encryption and node rating for securing the RPL protocol communications in the IoT ecosystem. *International Journal of Communication Systems*, *34*(3).

Authors Biography

Sopha Khoeurt

Sopha Khoeurt is an M.S. CSIT. student in the College of Computing, Khon Kaen University, Thailand. He received a B.S. degree from the University of South-East Asia, Cambodia, in 2016 and a teacher with higher education degree specialization in informatics from the National Institute of Education, Cambodia, in 2019. He is a lecturer in the Faculty of Science and Technology at the University of South-East Asia. His research interests include cybersecurity, machine learning, and computer networking. Email: sopha.k@kkumail.com, Orcid: https://orcid.org/0009-0007-2809-2399

Chakchai So-In

Chakchai So-In is a Professor of Computer Science in the Department of Computer Science, Khon Kaen University, KK, TH. He received B.Eng./M.Eng. degrees from Kasetsart University, BKK, TH in 1999/2001 and M.S./Ph.D. degrees from Washington University in St. Louis, MO, USA in 2006/2010, all in Computer Engineering. He has interned with Cisco Networking Academy (CNAP-NTU, SG), Cisco Systems (Silicon Valley, USA), WiMAX Forums (USA), and Bell Labs (Alcatel-Lucent, USA). His research interests include computer networking and the internet, wireless and mobile networking, the Internet of Things, wireless sensor networks, signal processing, cybersecurity, cyber-physical systems, and applied intelligent systems. He has served as an associate editor for IEEE Access, PLOS ONE, Wireless Networks, WCMC, PeerJ (CS), and ECTI-CIT and as a committee member/reviewer for many journals/publishers such as IEEE, Elsevier, Springer, Wiley, IET, Inderscience, IEICE, and ETRI; and conferences such as GLOBECOM, ICC, VTC, WCNC, ICNP, ICNC, and PIMRC. He has authored/coauthored over 100 international (technical) publications, including some in IEEE JSAC, IEEE TCCN, IEEE/CAA, IEEE Commun./Wireless Commun. Mags, IEEE IoT J., IEEE System J., COMNET, MONET, and ESWA; and 10 books, including Mobile & Wireless Nets with IoT, Computer Network Lab., and Network Security Lab. He is also a senior member of IEEE and ACM. Email: chakso@kku.ac.th, Orcid: https://orcid.org/0000-0003-1026-191X

Pakarat Musikawan

Pakarat Musikawan received his B.S. degree in information technology from Ubon Ratchathani University, Thailand, in 2010. He received his M.S. and Ph.D. degrees in computer science from Khon Kaen University, Thailand, in 2012 and 2020, respectively. He is currently a lecturer in the Department of Computer, College of Computing, Khon Kaen University, Khon Kaen, Thailand. His research interests include machine learning, artificial intelligence, computational intelligence, soft computing, applied intelligence, and knowledge-based systems. Email: pakamu@kku.ac.th, Orcid: https://orcid.org/0000-0001-5315-751X

Phet Aimtongkham

Phet Aimtongkham is currently a lecturer and researcher at the Department of Computer Science, College of Computing, Khon Kaen University, Thailand. In 2020, he has recent received a Ph.D. in Information Technology from the Department of Computer Science, Khon Kaen University, Thailand, where he also received the B.S. and M.S. degrees in 2013 and 2017, respectively. His research interests include computer networking, Internet of Things (IoTs), multimedia networks, cybersecurity, and machine learning.
Email: phetim@kku.ac.th, Orcid: https://orcid.org/0000-0001-5289-1149