Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.064414

ARTICLE

NADSA: A Novel Approach for Detection of Sinkhole Attacks Based on RPL Protocol in 6LowPAN Network

Atena Shiranzaei[1,*], Emad Alizadeh[2], Mahdi Rabbani[3], Sajjad Bagheri Baba Ahmadi[4,*] and Mohsen Tajgardan[5]

[1]Department of Computer Engineering (Khash), University of Sistan and Baluchestan, Zahedan, 9816745845, Iran
[2]Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan, 8415683111, Iran
[3]Canadian Institute for Cybersecurity (CIC), University of New Brunswick (UNB), Fredericton, NB E3B 5A3, Canada
[4]School of Computing, Engineering & Physical Sciences, University of the West of Scotland, Paisley, PA1 2BE, UK
[5]Department of Electrical and Computer Engineering, Qom University of Technology, Qom, 3718146645, Iran

*Corresponding Authors: Atena Shiranzaei. Email: ashiranzaei@eng.usb.ac.ir;
Sajjad Bagheri Baba Ahmadi. Email: sajjad.bagheri@uws.ac.uk

Received: 15 February 2025; Accepted: 19 June 2025; Published: 30 July 2025

ABSTRACT: The sinkhole attack is one of the most damaging threats in the Internet of Things (IoT). It deceptively attracts neighboring nodes and initiates malicious activity, often disrupting the network when combined with other attacks. This study proposes a novel approach, named NADSA, to detect and isolate sinkhole attacks. NADSA is based on the RPL protocol and consists of two detection phases. In the first phase, the minimum possible hop count between the sender and receiver is calculated and compared with the sender's reported hop count. The second phase utilizes the number of DIO messages to identify suspicious nodes and then applies a fuzzification process using RSSI, ETX, and distance measurements to confirm the presence of a malicious node. The proposed method is extensively simulated in highly lossy and sparse network environments with varying numbers of nodes. The results demonstrate that NADSA achieves high efficiency, with PDRs of 68%, 70%, and 73%; E2EDs of 81, 72, and 60 ms; TPRs of 89%, 83%, and 80%; and FPRs of 24%, 28%, and 33%. NADSA outperforms existing methods in challenging network conditions, where traditional approaches typically degrade in effectiveness.

KEYWORDS: Internet of Things; security; RPL; intrusion detection; sinkhole attack detection; RSSI

1 Introduction

The Internet of Things (IoT) is a transformative technology that facilitates connectivity among numerous physical objects across diverse domains, including smart healthcare, transportation, cities, grids, and homes [1]. Fig. 1 illustrates a generalized IoT architecture encompassing smart transportation, homes, and communities, where smart devices communicate via gateway nodes over the Internet [2,3]. These applications are essential for modern societies, providing remote monitoring and control of smart devices.

In resource-constrained environments, sensor nodes connect using IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) [4,5], which allows each node to maintain its own IPv6 address. As a standard protocol for enabling IPv6 communication on wireless networks, 6LoWPAN is integral to realizing IoT.

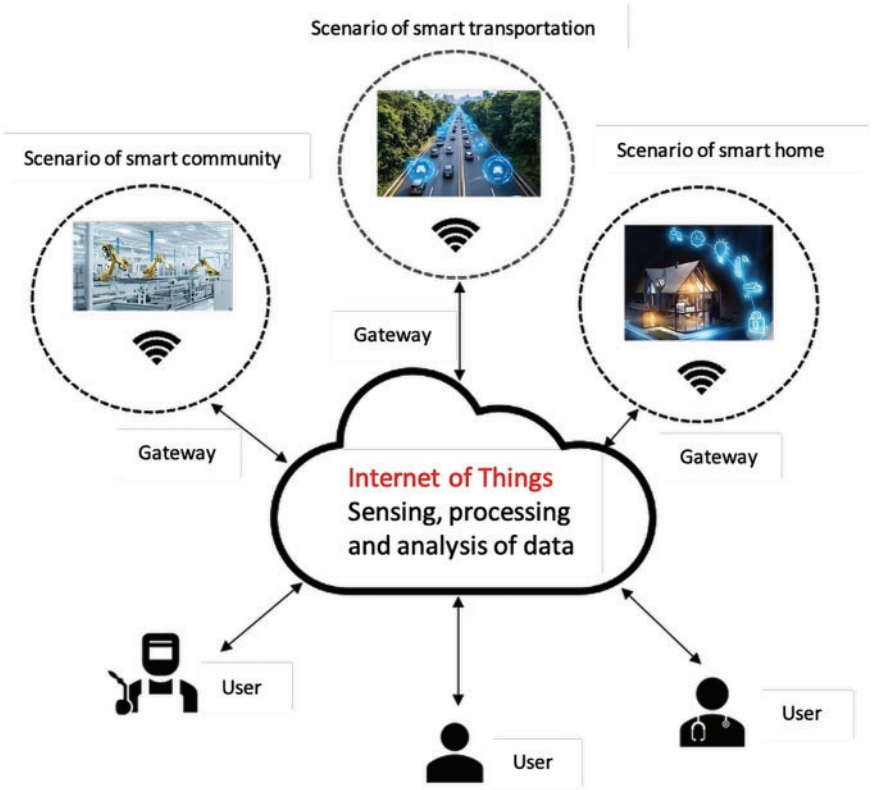

Figure 1: IoT Architecture [6]

The Routing Protocol for Low-Power and Lossy Networks (RPL) is the standardized IPv6 routing protocol designed for IoT networks. It builds a Destination-Oriented Directed Acyclic Graph (DODAG) rooted at a sink node, which is connected to the Internet via a gateway. RPL uses four ICMPv6 control messages—DODAG Information Object (DIO), Destination Advertisement Object (DAO), DAO Acknowledgment (DAO-ACK), and DODAG Information Solicitation (DIS)—to construct and maintain routing paths [6]. Fig. 2 illustrates an example of an RPL DODAG with three nodes and a gateway. These control messages enable nodes to select the optimal parent and establish efficient routes.

While IoT offers substantial benefits, it also introduces distinct security challenges compared to traditional networks [7–9]. Attacks such as blackhole, selective forwarding, wormhole, Sybil, and particularly sinkhole attacks pose significant threats to the integrity of IoT networks [1].

Among these, the sinkhole attack is especially dangerous in wireless networks. In this attack, a malicious node deceptively advertises a low hop count to attract network traffic, subsequently dropping, delaying, or modifying packets [10,11]. Fig. 3 illustrates a typical sinkhole attack scenario.

Detecting sinkhole attacks in highly lossy and sparse networks remains a significant research challenge, as most existing methods are optimized for high-density deployments and perform poorly under such adverse conditions. This study addresses this gap by proposing NADSA (Novel Approach for Detection of Sinkhole Attacks), a lightweight and efficient method based on the RPL protocol that effectively detects and isolates sinkhole nodes in sparse and lossy IoT environments.

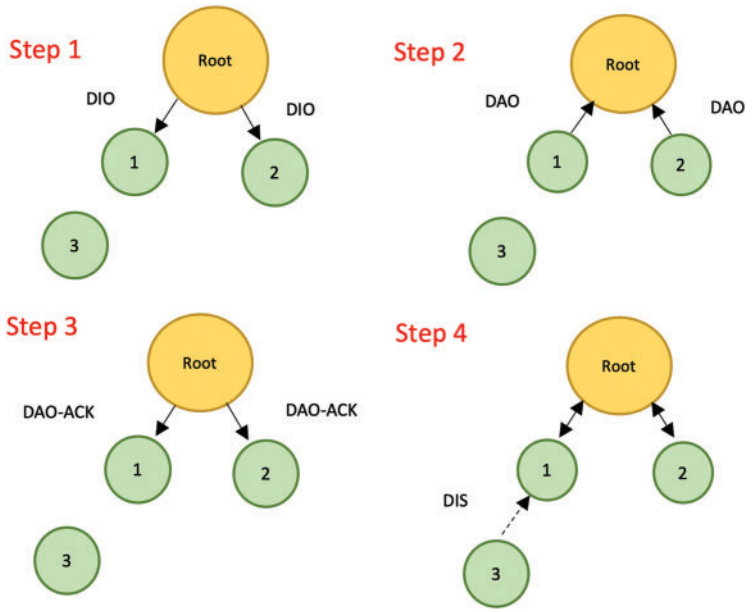

Figure 2: The formation of RPL DODAG [7]

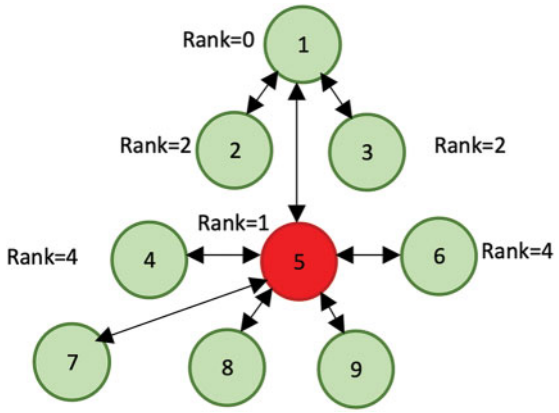

Figure 3: Sinkhole Attack Scenario [12]

To validate the proposed approach, NADSA is simulated under various scenarios using the Contiki-NG (Cooja) simulator and evaluated using key performance metrics. The main contributions of this paper are as follows:

- Introduction of NADSA, a security enhancement mechanism for IoT networks.
- Targeted detection of sinkhole attacks using a combination of DIO count, Expected Transmission Count (ETX), Received Signal Strength Indicator (RSSI), and distance measurement, specifically designed for sparse and lossy environments.
- Demonstration of effective sinkhole node isolation to enhance network resilience.
- Provision of mathematical analysis to support the proposed model.
- Implementation and evaluation of NADSA through real-world simulations.

- Comparative benchmarking of NADSA against existing approaches, demonstrating superior performance in terms of Packet Delivery Ratio (PDR), End-to-End Delay (E2ED), True Positive Rate (TPR), and False Positive Rate (FPR).

The remainder of this paper is organized as follows: Section 2 reviews related literature. Section 3 details the proposed approach. Section 4 presents the simulation results and discussion. Section 5 outlines future research directions. Finally, Section 6 concludes the paper.

2 Related Works

The authors in [13] proposed a strategy based on Received Signal Strength Indicator (RSSI). This method measures the signal strength of all nodes during network deployment and forwards key information to the sink node. Based on this data, the sink constructs a virtual geographical topology graph. Subsequent RSSI values are compared to the original measurements to detect signal changes; if significant deviations are found, the sinkhole attack is identified and mitigated. While this method effectively detects sinkhole and wormhole attacks, its implementation incurs considerable cost.

Raza et al. [14] developed SVELTE, an Intrusion Detection System (IDS) capable of detecting routing attacks such as sinkhole and selective forwarding. SVELTE comprises three components: 6LoWPAN Mapper (6Mapper), which collects IPv6 routing protocol data and reconstructs the network topology; a data analysis module that detects malicious behavior; and a mini-firewall to block unwanted traffic. While SVELTE detects sinkhole attacks with minimal false alarms and low overhead, it has been reported to suffer from a high false positive rate, resulting in inefficient resource usage [15].

The authors in [16] introduced INTI, a trust- and fame-based system combined with watchdog mechanisms. Nodes are arranged hierarchically into leaders, associates, or members, with roles subject to change. Each node monitors its superior by evaluating past and incoming traffic. Upon detecting an attack, it broadcasts an alert and disconnects the malicious node. Although INTI demonstrates a high detection rate and low false positives, it is primarily suited for mobile networks and lacks evaluation on low-capacity nodes [17].

InDReS [15] is another IDS designed to detect sinkhole attacks. It evaluates whether a node's current rank has increased by the minimum rank increment and whether its parent node's rank is within an acceptable range. If discrepancies are found, the suspicious node is isolated and the network is alerted. Compared to INTI, InDReS shows superior performance in Quality of Service (QoS) metrics like energy consumption, throughput, packet drop ratio, and overhead. However, it still suffers from high false positives and energy depletion.

In [11], the authors proposed a method to detect sinkhole attacks in hierarchical wireless sensor networks. The network is divided into clusters, each managed by a cluster head responsible for detection. The process involves two phases: identifying the attacker and classifying the type of sinkhole attack (e.g., message delay, drop, or modification). Malicious nodes are blacklisted, and their cluster members are notified. This method has low computation and communication overhead, making it suitable for resource-constrained environments. However, the computation cost remains high.

The Secure-RPL (SRPL) protocol proposed in [18] enhances the RPL protocol by scanning node behavior to detect suspicious rank values. It uses a threshold function and hash chain authentication to detect sinkhole attacks. Although SRPL effectively detects such attacks, applying the threshold function to all nodes results in increased overhead and a packet loss rate of approximately 22%–23%.

To reduce this overhead, SecTrust-RPL [19] was introduced. It protects against rank and Sybil attacks by evaluating the trustworthiness of neighboring nodes using both direct and recommended trust values. Nodes

with higher trust values are selected for routing, while those with lower values are classified as selfish or malicious. Although the protocol improves routing performance, it introduces latency and does not address uncertainty in trust recommendations.

Qureshi et al. [20] presented a two-phase framework for detecting sinkhole, version number, blackhole, and Hello-flood attacks in RPL networks. The first phase establishes detection thresholds for each attack type, and the second performs the actual detection. Performance evaluations show the framework is effective for IPv6-based routing in lossy IoT networks.

In [21], the authors proposed SoS-RPL, a model for sinkhole attack detection consisting of two main components. The first component calculates node rate and rank based on distance measurements. The second component analyzes node behavior using the Average Packet Transmission RREQ (APT-RREQ) to identify misbehavior and isolate malicious nodes. However, the model struggles with message overhead and does not accommodate mobility. Additionally, it only detects malicious parent nodes, leaving child node attacks unresolved.

Almusaylim et al. [22] proposed SRPL-RP, a secure RPL-based routing protocol for detecting and isolating rank and version number attacks. It identifies malicious behavior using rank comparisons and attack status tables, and mitigates threats through blacklisting and alert mechanisms. Simulation results show that SRPL-RP enhances packet delivery, detection accuracy, and reduces control message overhead.

RFTRUST [23] adopts a trust-based approach for RPL-based IoT networks using Random Forest (RF) and Subjective Logic (SL) to identify sinkhole attacks. This model improves detection accuracy and enhances network performance. Its main limitation lies in the assumption that nodes can monitor the forwarding behavior of their neighbors.

Kumar et al. [24] proposed TIDSRPL, a trust-based enhancement to RPL aimed at detecting sinkhole, selective forwarding, and Sybil attacks in Low-Power and Lossy Networks (LLNs). TIDSRPL outperforms the standard MRHOF-RPL protocol in terms of packet loss and network stability. However, its reliance on centralized trust evaluation introduces a single point of failure and scalability challenges.

The reviewed literature demonstrates various techniques for enhancing network security and detecting sinkhole attacks. While these approaches offer promising results under standard network conditions, many struggle under adverse scenarios, such as sparse or highly lossy environments.

In this study, we propose a novel two-phase method for detecting sinkhole attacks. The first phase analyzes hop counts, while the second uses a combination of DIO counts, RSSI, ETX, and distance metrics within a fuzzification process. This approach improves accuracy and robustness over existing methods by applying diverse detection techniques tailored for sparse and lossy networks. A summary of selected approaches is presented in Table 1.

Table 1: The summary of some solutions to secure IoT

S#	Reference	Year	Attack type	Limitation	Advantage of proposed method
1	[13]	2009	Sinkhole attack and Wormhole attack	Costly	Low complexity
2	[14]	2013	Spoofed or altered information, Selective forwarding attack, and Sinkhole attack	High false positives, and high false detection rate	High accuracy on lossy and sparse network
3	[16]	2015	Sinkhole attack	Low false positive rate	High false positive rate
4	[15]	2016	Sinkhole attack	High false positive alerts and high energy node depletion	High accuracy on lossy and sparse network
5	[11]	2016	Sinkhole attack	Costly	Low complexity

(Continued)

Table 1 (continued)

S#	Reference	Year	Attack type	Limitation	Advantage of proposed method
6	[18]	2016	Sybil attack and Rank attack	High packet loss rate on average between 22%–23%	Low packet loss rate
7	[19]	2019	Sinkhole attack	Delay occurs	Fast detection time
8	[20]	2020	HELLO Flood attack, Version number attack, Blackhole attack, and Sinkhole attack	Evaluated using a small number of network nodes.	Evaluated with various numbers of network nodes
9	[21]	2020	Sinkhole attack	Malicious parent node can only be detected by child node through comparing the rank value. Malicious child node cannot be detected.	Child nodes are detected and isolated.
10	[22]	2020	Rank and Version attacks	Requires nodes to monitor network traffic	No need for monitoring
11	[23]	2021	Sinkhole attack	Assumes nodes can monitor forwarding behavior of neighbors	No need for monitoring
12	[24]	2025	Sinkhole, Selective Forwarding, and Sybil attacks	Centralized	Distributed

3 Proposed Work

6LoWPAN network is a lossy and wireless network which involves resource constrained nodes with a unique IPv6 address, and often uses RPL as a routing protocol. In the present study, the authors have designed and implemented NADSA and sinkhole attacks to test this novel approach.

NADSA

NADSA is a method that primarily focuses on detecting sinkhole attacks. This novel approach consists of two phases. The first phase achieves the minimum possible hop count and compares it with the node's hop count. If any suspicious activity is detected, it identifies the sinkhole attack and disrupts the DIO. The second phase measures the number of DIO messages sent by both the sender and the receiver. In this study, the DIO count is important because, in the attack scenario, it is assumed that the malicious node increases the number of transmitted DIO messages. Therefore, the DIO count is also utilized for attack detection. Next, the method uses RSSI, ETX, and distance measurements for fuzzification. These parameters were chosen because, given the highly unstable and sparse network conditions considered in the scenario, it is necessary to use metrics related to channel state (such as ETX and RSSI). If the proposed method detects any suspicious behavior, it then identifies the sinkhole attacks. The phases are as follows:

1. The main idea of the first phase to detect the existence of sinkhole attacks is that: a malicious node advertises a tempting rank and hop count. In this phase, when a DIO message is delivered to a node, first the node calculates R which is the distance between a sender and a root node through Eq. (1):

$$\mathrm{R} = \mathrm{RSSI}(d_0) - 10n \log_{10}\left(\frac{d}{d_0}\right) + X_\sigma \tag{1}$$

where $\mathrm{RSSI}(d_0)$ is the reference signal strength at distance d_0, n is the path loss exponent, d is the transmitter-receiver distance, d_0 is the reference distance, and X_σ represents shadowing noise. Then it calculates M, that is the minimum possible hop count between the root and the receiver, through Eq. (2):

$$M = \lceil R/T \rceil \tag{2}$$

where T is the transmission range of each node in the network.

Finally, it compares the hop count of the sender (S) with M. If it is less than M, the sender is a malicious node.

2. The main idea of the second phase is that: every node has a counter (C_r) and a counter list of their neighbours (C_s). When a node sends a DIO message, its C_r will be increased by one. When a node receives a DIO, the counter of the sender on the node's counter list of its neighbours (C_s) will be increased by 1 as well. When a node receives a DIO message, it compares D and differentiates between C_r and C_s, using Eq. (3). If C_r is less than C_s, the sender is a malicious node.

$$D = C_r - C_s \tag{3}$$

Here, whenever the network is running and a node joins lately to the network, a false detection happens. Due to the lower C_r, a sinkhole attack is detected by mistake. To overcome this problem, the authors have defined a maximum number of sent DIO (C_{max}) for all the nodes. When the C_{max} reaches the maximum number, then C_r and C_s will be reset. At the end, there is a fuzzy system which consists of three parameters: RSSI, ETX, and D. According to the value of these parameters, the fuzzy logic system identifies the existence of a malicious node. Since the fuzzy system is used solely for decision-making purposes, the defuzzification step is not required. RSSI is a way to estimate the measure of the strength of a radio signal which a node receives from another node. For example, at a large distance, the signal is weak; therefore, the RSSI value is considered poor signal strength. Eq. (4) shows RSSI format:

$$P_r = P_t G_t G_r (\lambda / 4\pi R)^2 \tag{4}$$

P_t: transmit power. G_r: receiver antenna power gain G_t: transmitter antenna power gain. λ: transmit signal wavelength. R: distance between a sender and a receiver.
ETX is the expected transmission count for each packet that can be successfully delivered to the destination. ETX looks for links of higher quality per unit of time. The ETX is obtained in the Eq. (5) [25]:

$$ETX = 1/(DF * DR) \tag{5}$$

DF measures the probabilities of delivering a packet to the neighbour. DR measures probabilities of receiving an acknowledgment packet. D is explained in Section 2.

Figs. 4–6 show the membership functions of RSSI, ETX, and D. In fact, the fuzzy system receives the input variables of each parameter and uses membership functions to obtain fuzzy values. The fuzzy value range is defined as low, medium, and high. When the network is in good condition and the sender is close to the receiver, RSSI is high and ETX is low. D is low whenever C_r and C_s are equal.

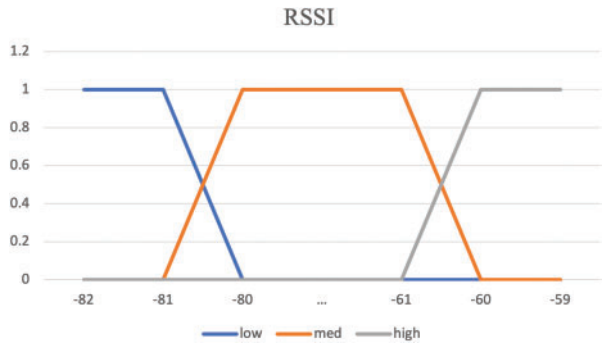

Figure 4: Membership functions of RSSI

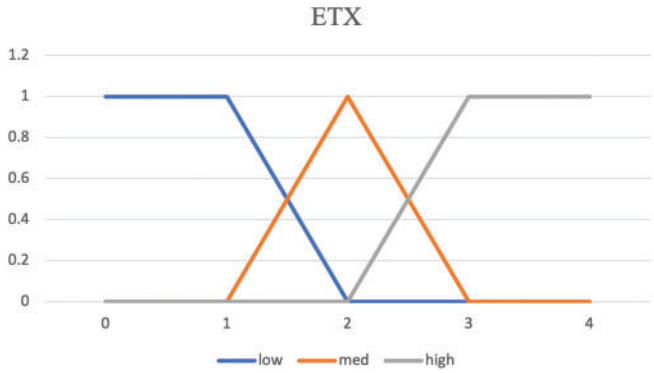

Figure 5: Membership functions of ETX

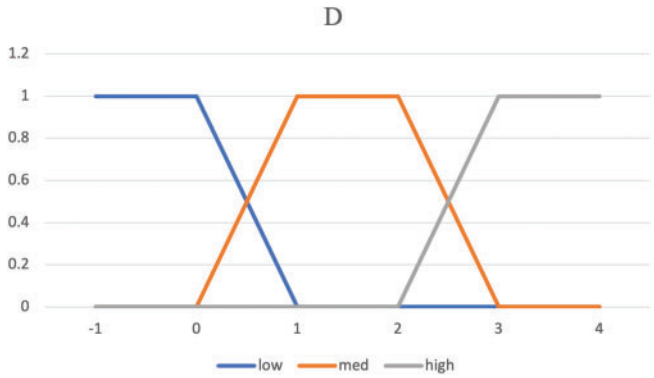

Figure 6: Membership functions of D

Table 2 represents the fuzzy rules that are the combination of the three input fuzzy sets, including RSSI, ETX, and D. The fuzzy system in the present study involves three inputs, a fuzzy controller, and one output. The last column in Table 2 shows the output fuzzy variable.

Table 2: The fuzzy rules base

ETX	RSSI	D	Attack
High	Low	Low	TRUE
Med	Low	Low	TRUE
Med	Med	Low	TRUE
Med	High	Low	TRUE
Low	Low	Low	FALSE
Low	Med	Low	FALSE
Low	High	Low	FALSE
Low	Low	Med	FALSE
Low	Med	Med	FALSE
Low	High	Med	FALSE
Low	Low	High	TRUE
Low	Med	High	TRUE

(Continued)

Table 2 (continued)

ETX	RSSI	D	Attack
Low	High	High	TRUE
Low	Med	Low	TRUE
Low	Low	Low	TRUE
Low	Low	Med	TRUE
Low	Med	Med	TRUE
Low	Low	High	TRUE
Med	Low	High	TRUE
High	Low	High	TRUE
Low	Med	High	TRUE
Med	Med	High	TRUE
High	Med	High	TRUE
Low	High	High	TRUE
Med	High	High	TRUE
High	High	High	TRUE

The pseudo code to detect the sinkhole attack is given in Algorithms 1 and 2. Figs. 7 and 8 show the process of each algorithm in receiving and sending DIO packets.

Algorithm 1: Pseudo code of every receiving DIO

Input: $C_s, C_r, S, RSSI, ETX, D$
Output: attack detection or normal DIO Process

for every received DIO **do**
  C_s++;
  $attack\_flag \leftarrow 0$;
  Calculate R;
  Calculate M;
  if $M > S$ **then**
    $attack\_flag \leftarrow 1$;
  else
    $D \leftarrow C_r - C_s$;
    if $D < 0$ **then**
      $attack\_flag \leftarrow 1$;
    else
      Fuzzificate RSSI;
      Fuzzificate ETX;
      Fuzzificate D;
      Set attack_flag according to fuzzy rules table;
    end if
  end if
  if attack_flag = 1

(Continued)

Algorithm 1 (continued)

```
20:       Raise alarm for attack;
21:       Drop the DIO;
22:    else
23:       Normal DIO process;
24:    end if
25: end for
```

Algorithm 2: Pseudo code of every sending DIO

Input: C_r, C_s, C_{max}
Output: Update C_r and C_s

```
1:  for each sent DIO do
2:        C_r + +;
3:        if C_r > C_max then
4:           C_r = 0;
5:         for the nodes in neighbors list do
6:              C_s = 0;
7:         end for
8:     end if
9: end for
```

Algorithms 1 and 2, along with Figs. 7 and 8, illustrate the core operational logic of the proposed NADSA method for detecting sinkhole attacks. Algorithm 1 and Fig. 7 describe how a node processes each received DIO message. The procedure begins with incrementing the sender's DIO count, followed by calculating the estimated distance (R) and the minimum possible hop count (M). This value is then compared with the sender's advertised hop count (S). If the advertised hop count is less than the calculated minimum, the node is flagged as suspicious. If not, the method computes the difference (D) between the receiver's own DIO counter (C_r) and the sender's counter (C_s). A negative D value suggests inconsistent behavior, indicating a potential attack. If this check is also passed, the node applies fuzzification based on RSSI, ETX, and D to assess the risk using fuzzy rules.

Algorithm 2 and Fig. 8 describe how a node handles the sending of DIO messages. Every time a DIO is sent, the node increments its own counter (C_r). To avoid false positives caused by newly joined nodes, the algorithm resets both the sender's counter (C_r) and its neighbors' counters (C_s) when a maximum threshold (C_{max}) is reached. This counter synchronization helps maintain the reliability of the detection logic across the network.

In summary, these algorithms and flowcharts ensure that NADSA captures abnormal behavior based on both hop count inconsistencies and DIO message frequency, while the fuzzy system adds an intelligent layer of decision-making in uncertain conditions. This dual-phase approach enhances detection accuracy and adaptability in dynamic, low-power wireless networks.

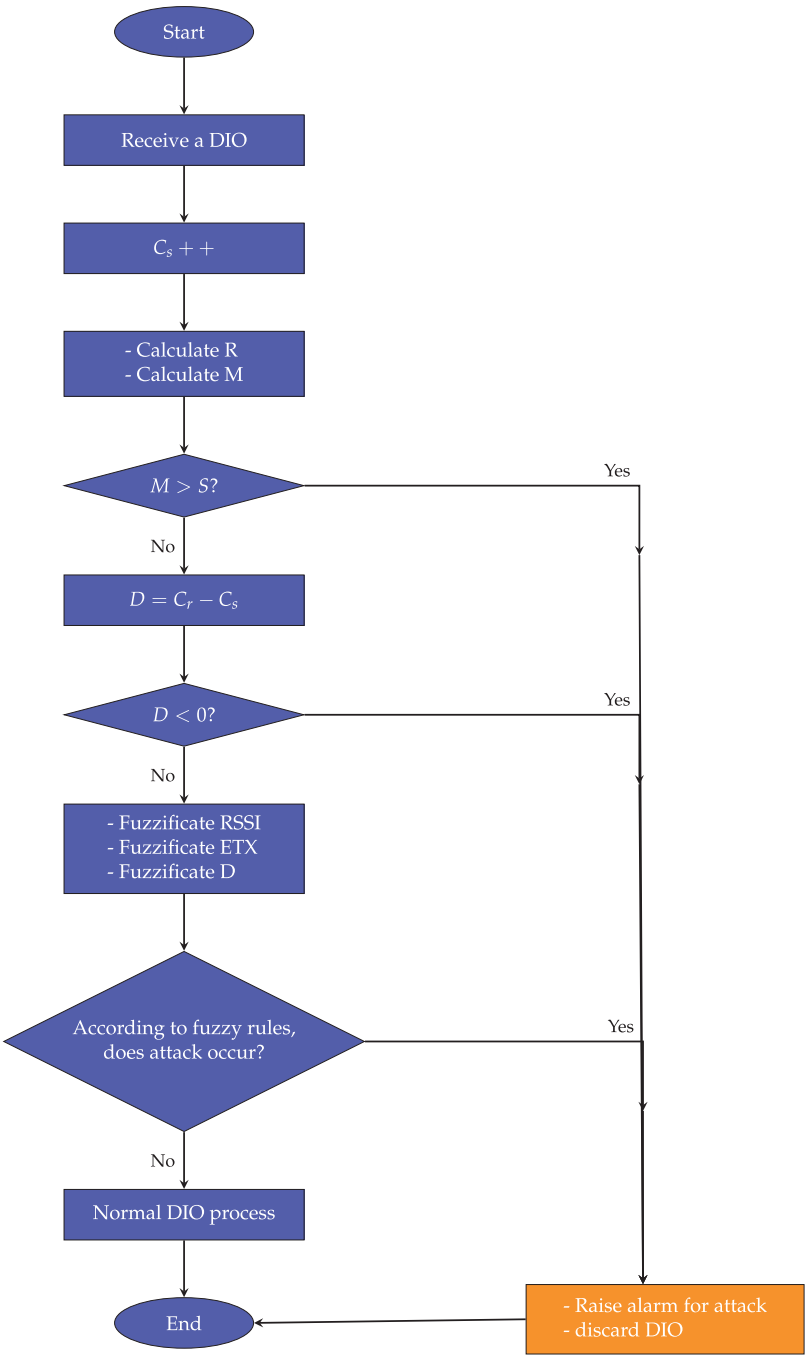

Figure 7: The process of receiver node

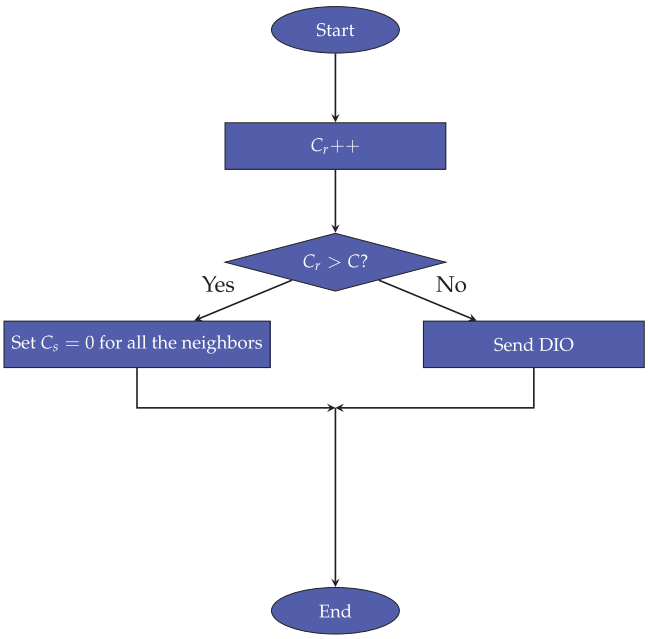

Figure 8: The process of sender node

4 Results and Discussion

The proposed NADSA method demonstrates effective detection of sinkhole attacks within RPL-based IoT environments. This section presents the simulation setup and evaluates the system's performance across various metrics.

The evaluation results validate NADSA's capability to accurately detect sinkhole attacks while preserving overall network performance. The system achieved high detection accuracy, successfully identifying most malicious nodes with a low false negative rate. This performance can be attributed to NADSA's multi-dimensional approach, which integrates various network indicators—such as DIO message frequency, Expected Transmission Count (ETX), Received Signal Strength Indicator (RSSI), and physical distance measurements—to assess anomalies and detect attacks.

From a network performance perspective, NADSA maintained stable operation even under challenging sparse and lossy conditions. It achieved high packet delivery rates, outperforming several existing approaches under similar scenarios. Additionally, the method maintained low end-to-end delays, making it suitable for latency-sensitive IoT applications. These attributes highlight NADSA's practical value for real-world deployments where both security and performance are essential.

However, the analysis also revealed some limitations that present opportunities for future enhancement. While NADSA accurately identified the majority of malicious nodes, it occasionally produced false positives by misclassifying legitimate nodes as suspicious. Such outcomes are not uncommon in sparse networks, where variations in signal and routing behavior can resemble attack patterns.

Despite this, NADSA's two-phase detection mechanism, initially analyzing routing behavior and subsequently applying a more detailed, metric-driven assessment, exhibited strong adaptability in environments where traditional methods often underperform. This layered architecture enhances robustness and enables NADSA to function effectively in realistic IoT deployments facing sophisticated security threats.

4.1 Simulation Setup

The authors designed and implemented NADSA[1] in Contiki-NG [26], which is a kind of operating system for IoT. Contiki OS provides an environment to implement and evaluate low power and lossy networks. RPL is widely used as a routing protocol for resource constraint devices in IoT. NADSA is essentially designed to detect sinkhole attacks; however, RPL implementation is used to develop NADSA in Contiki-NG. Cooja [27,28], a simulator working with Contiki, shows realistic results [29]. It is a different simulation software. Cooja works better with resource-constrained devices [30].

To evaluate NADSA, the authors implemented sinkhole attacks against RPL and ran a topology of 10, 30, and 60 normal nodes with 10% malicious nodes. Table 3 shows the simulation parameters and Fig. 9 illustrates the topology of 10 nodes.

In our experiments, the default TX and RX success ratio was set to 75%. To further investigate the robustness of the proposed method under varying channel conditions, additional tests were conducted using TX/RX success ratios of 70% and 80%.

Moreover, the ratio of malicious nodes was fixed at 10%. However, for a more detailed analysis of the system's sensitivity to different attack intensities, we also considered scenarios with 5% and 20% malicious node ratios. The experiments were conducted using networks consisting of 10, 30, and 60 nodes to assess scalability and performance across different network sizes.

Table 3: Simulation parameters

Network parameters	Values
Operating system	Contiki-NG
Simulator	Cooja
Routing protocol	RPL
Topology type	Random
Rank metric	OF
Mode of operation	1
Mote type	Z1
Area size	100×100 m^2, 300×300 m^2, 600×600 m^2
Number of packets sent	1500≈, 4500≈, 9000≈
Packet sent interval	7680 ms
TX success ratio	70%, 75%, 80%
RX success ratio	70%, 75%, 80%
T	50 m
Interference range	150 m
C_{max}	20
Type of attack	Sinkhole attack
Number of nodes	10, 30, 60
Malicious nodes ratio	5%, 10%, 20%
Simulation time	20 min

[1] https://github.com/SinkholeAttackDetection/contiki-ng-NADSA (accessed on 18 June 2025).

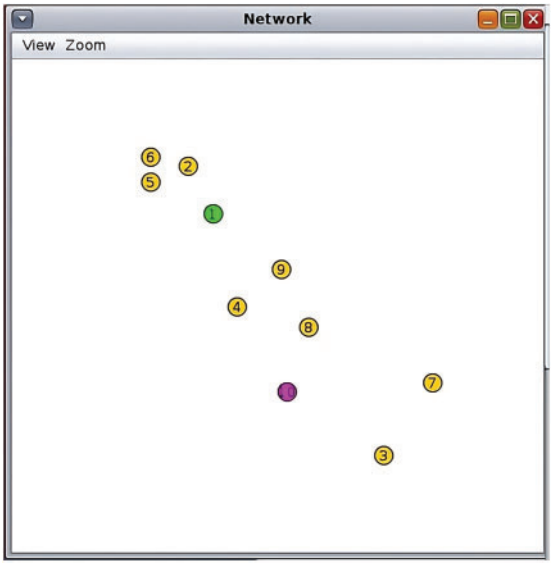

Figure 9: The topology of 10 nodes

4.2 Performance Evaluation

The following performance parameters were used to evaluate NADSA and the results were compared with INTI [16], SoS-RPL [21], and RFTRUST [23] models.

- TPR: This parameter shows the rate of the total number of malicious nodes successfully detected. TPR is obtained through Eq. (6).

$$TPR = \frac{TP}{TP + FN} * 100 \tag{6}$$

True Positive (TP) is the number of malicious nodes accurately detected. False Negative (FN) is the number of malicious nodes not identified in the system.

- FPR: This represents the rate of the total number of times that the normal nodes are considered as malicious nodes. FPR is calculated by Eq. (7).

$$FPR = \frac{FP}{FP + TN} * 100 \tag{7}$$

False Positive (FP) is the number of normal nodes detected as malicious nodes. True Negative (TN) is the number of normal nodes determined as the normal node.

- PDR: This determines the rate of the total number of data packets successfully received at the receiver to the total data packet sent by the source during simulation. Eq. (8) illustrates the PDR.

$$PDR = \frac{\sum_{i=1}^{N} receivedPacket}{\sum_{i=1}^{N} sentPacket} * 100 \tag{8}$$

In Eq. (8), N stands for the total number of nodes.

- E2ED: It is a required time to send a packet from the sender node to the receiver node. E2ED is calculated through Eq. (9).

$$E2ED = \sum_{i=1}^{n} \left(T_{proc}^{i} + T_{trans}^{i} + T_{prop}^{i} + T_{queue}^{i} \right) \tag{9}$$

where T_{proc}^{i} is processing delay at node i, T_{trans}^{i} is transmission delay at node i, T_{prop}^{i} is propagation delay at node i, T_{queue}^{i} is queuing delay at node i, n indicates total nodes in path.

To analyse the performance of NADSA in terms of PDR and E2ED, the proposed method was compared with the normal RPL scenario, the sinkhole attack scenario, INTI, SoS-RPL, and RFTRUST. In the normal RPL scenario, the network runs normal RPL routing protocol. In the sinkhole attack scenario, the network includes normal RPL as well as the sinkhole attack. In NADSA, the network consists of the authors' proposed method with the sinkhole attack.

In terms of TPR and FPR, the authors compared the proposed method with INTI, SoS-RPL, and RFTRUST.

4.2.1 Packet Delivery Rate Analysis

Packet Delivery Rate (PDR) represents the ratio of successfully delivered packets to the total number of packets sent. This metric was used to evaluate the performance of various methods under extremely lossy and sparse network conditions.

As shown in Fig. 10, the PDR was compared across the following scenarios: normal RPL, sinkhole attack, NADSA, RFTRUST, SoS-RPL, and INTI. In a baseline scenario without any attacks, normal RPL achieved the highest delivery rate of approximately 81% across varying node densities.

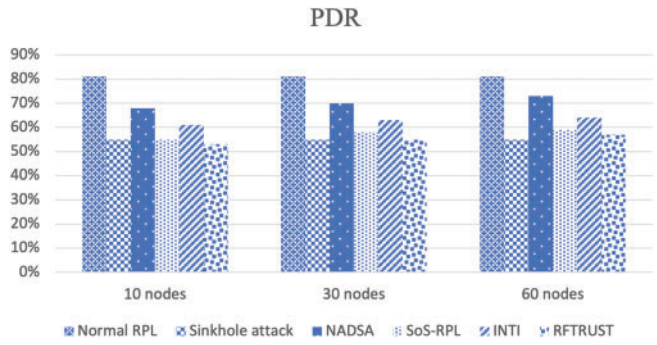

Figure 10: Comparisons of PDR for Normal RPL, Sinkhole Attack, NADSA, SoS-RPL, INTI, and RFTRUST

Under attack conditions, the proposed NADSA method outperformed other detection mechanisms by achieving PDRs of 68%, 70%, and 73% depending on the number of nodes. NADSA's superior performance can be attributed to its use of key parameters—such as RSSI and ETX—in a fuzzy logic-based decision process. These parameters allow the system to effectively differentiate between legitimate packet loss due to lossy and sparse conditions and malicious packet drops caused by sinkhole attacks.

In contrast, methods such as INTI, SoS-RPL, and RFTRUST showed reduced performance in sparse environments. These methods rely on denser node deployments to ensure stable communication and connectivity, which becomes less effective in highly sparse and lossy networks. As a result, their ability to maintain high PDR under adverse conditions is significantly limited compared to NADSA.

4.2.2 End-to-End Delay Analysis

End-to-End Delay (E2ED) measures the average time required for a packet to travel from the source node to the destination node. This metric typically increases in the presence of misbehaving nodes within the network. One major contributing factor to increased E2ED is packet retransmission, which is more prevalent in lossy and sparse environments, especially when sinkhole attacks are present.

As illustrated in Fig. 11, the delay performance is compared across the following scenarios: normal RPL, sinkhole attack, NADSA, INTI, SoS-RPL, and RFTRUST. In the normal RPL scenario, where the routing protocol operates without any attacks or intrusion detection mechanisms, the average E2ED values are 82, 72, and 60 ms, depending on the number of nodes.

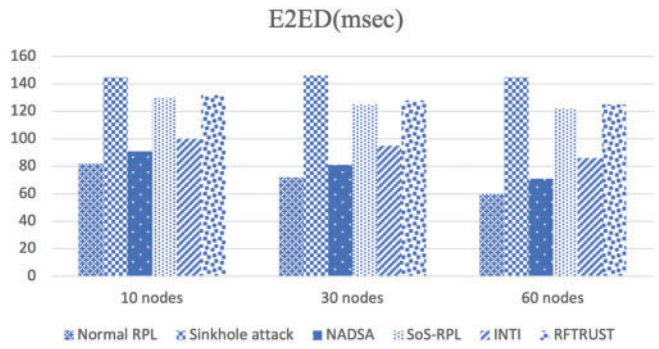

Figure 11: Comparisons of E2ED for Normal RPL, Sinkhole Attack, NADSA, SoS-RPL, INTI, and RFTRUST

In contrast, the sinkhole attack scenario—where malicious nodes are present and no countermeasures are applied—yields the highest E2ED due to frequent packet drops and retransmissions caused by malicious routing behavior.

The proposed NADSA method achieves improved delay performance with E2ED values of 91, 81, and 71 ms. While NADSA introduces a slight overhead due to its detection mechanisms, it significantly reduces unnecessary retransmissions by preventing malicious nodes from being selected as preferred parents in the routing path. This results in more stable and efficient data forwarding, especially under sparse and lossy network conditions.

Compared to other methods such as INTI, SoS-RPL, and RFTRUST, NADSA demonstrates better adaptability to challenging network environments, ensuring lower delays while maintaining secure routing operations.

4.2.3 True Positive Rate Analysis

True positive rate demonstrates the rate of the correct decisions to test the malicious nodes. Fig. 12 displays that the proposed method achieves 89%, 83%, and 80% true positive rates. Other methods don't pay attention to extra lossy and sparse networks and can't have a high TPR compared with NADSA. The proposed method uses ETX and RSSI parameters to identify the malicious node, separate attack situations, and extra lossy and sparse situations. Thus, the TPR of NADSA achieves higher than other methods.

4.2.4 False Positive Rate Analysis

False positive rate (FPR) refers to the proportion of legitimate nodes that are incorrectly identified as malicious. As illustrated in Fig. 13, the proposed NADSA method achieved FPR values of 24%, 28%, and

33% across different network sizes. Although these values are relatively high, NADSA still performs better compared to other approaches.

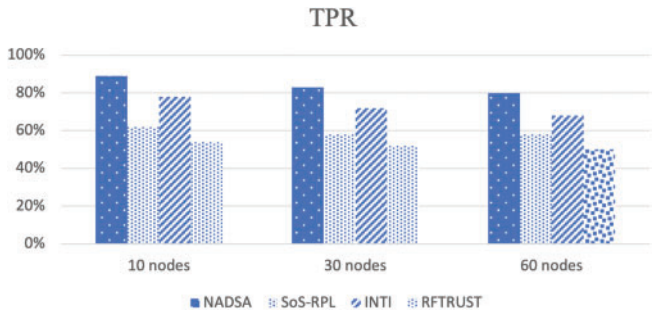

Figure 12: Comparisons of TPR for NADSA, SoS-RPL, INTI and RFTRUST

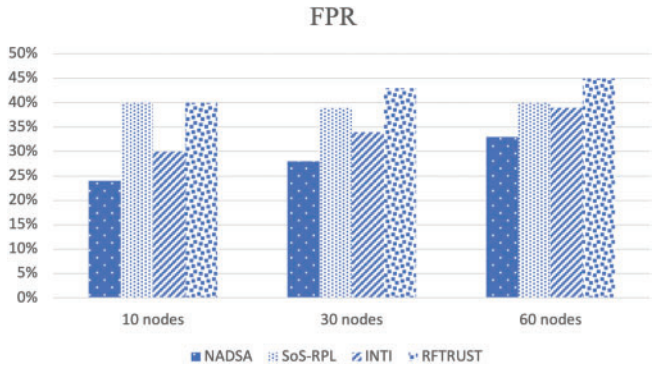

Figure 13: Comparisons of FPR for NADSA, SoS-RPL, INTI and RFTRUST

The increased FPR is primarily due to the challenging conditions of an extra lossy and sparse network. In such environments, normal nodes may occasionally exhibit irregular behavior due to unstable connectivity or signal degradation, leading to their misclassification as attackers. However, unlike other methods, the proposed approach includes a re-evaluation mechanism. When a node is suspected of being malicious, it is not immediately and permanently isolated. Instead, its DIO messages are temporarily discarded, and the node is given another opportunity to send DIO messages. This allows the system to reassess its behavior before making a final decision.

This adaptive strategy helps reduce the risk of mistakenly isolating legitimate nodes. As shown in Fig. 13, NADSA consistently demonstrates a lower FPR compared with INTI, SoS-RPL, and RFTRUST. These existing methods do not explicitly consider the possibility of extra lossy and sparse network conditions, which results in higher rates of false detection.

RFTRUST incorporates a metric that can detect sinkhole attacks even under severe network conditions. However, due to the difficulty in differentiating between actual attacks and normal losses in sparse networks, its FPR is higher than other models. In contrast, the proposed NADSA approach uses multiple detection parameters and a two-phase evaluation process, improving accuracy in complex network environments.

4.2.5 Sensitivity Analysis

In fuzzy logic systems, parameter selection is typically based on expert knowledge or engineering experience. In the proposed approach, a specific set of fuzzy membership functions was carefully designed

through detailed analysis. To evaluate the robustness of the method and its ability to detect more adaptive or stealthy attackers—who may gradually alter their behavior—a sensitivity analysis was conducted using three different sets of fuzzy membership functions: NADSA-M1 (the original configuration), NADSA-M2, and NADSA-M3. These alternative configurations represent realistic variations in parameter choices that an engineer might reasonably adopt.

The proposed NADSA method was tested using each of these configurations across networks consisting of 10, 30, and 60 nodes, to observe the effect of system scale on detection performance. The results, presented in Table 4, demonstrate that NADSA-M1 consistently achieves the highest detection accuracy in all scenarios. NADSA-M2 and NADSA-M3 exhibit only minor reductions in performance, indicating that while the selection of membership functions can impact accuracy, the overall system remains effective even when the parameters differ from the original design.

Table 4: Effect of membership function on NADSA

Membership Function	Number of Nodes	TPR
NADSA-M1	10	90%
NADSA-M1	30	84%
NADSA-M1	60	81%
NADSA-M2	10	86%
NADSA-M2	30	80%
NADSA-M2	60	76%
NADSA-M3	10	83%
NADSA-M3	30	78%
NADSA-M3	60	74%

These findings confirm that the proposed NADSA method is resilient to variations in fuzzy logic parameters. It maintains robust detection capabilities, even under dynamic conditions where attackers may attempt to evade detection by slowly modifying their network behavior.

In this section, we further assessed NADSA under varying TX/RX success ratios and different percentages of attackers to test its performance under more realistic and variable conditions.

The impact of the communication channel on the proposed method was evaluated by varying the TX/RX success ratio, as summarized in Table 5. The results indicate that, since the proposed approach is specifically designed for lossy and sparse network environments, its performance improves as the TX/RX success ratio decreases.

Table 5: Effect of TX/RX success ratio on NADSA TPR

TX/RX success ratio	TPR
70%	92%
75%	90%
80%	86%

We also analyzed NADSA's performance with different attacker densities in a network of 30 nodes: 5%, 10%, and 20% malicious nodes. Table 6 presents how the number of attackers affects NADSA's TPR.

Table 6: Effect of number of attackers on NADSA TPR

Number of attackers	TPR
5%	94%
10%	90%
20%	84%

4.2.6 Stealthy Attack Analysis

In fuzzy logic systems, the decision-making process is influenced by the combination of all input parameters. When one input remains constant or provides limited information, the system adjusts by placing greater emphasis on the remaining inputs, as defined in the rule table (Table 2). This characteristic allows fuzzy-based methods such as NADSA to retain robustness, even when attackers attempt to evade detection by subtly modifying only a portion of their behavior.

To evaluate the effectiveness of the proposed method against such stealthy attacks, a set of experiments was conducted in which attackers varied their behavior gradually rather than abruptly. The results, presented in Table 7, show that NADSA is capable of maintaining reliable detection accuracy under these conditions. This is attributed to the system's multi-metric approach, which leverages diverse inputs such as RSSI, ETX, hop count, and DIO frequency. When one or more of these parameters are manipulated by an attacker, the others can still provide sufficient evidence for anomaly detection.

Table 7: Comparison of normal attack and stealthy attack in TPR metric

Attack Type	Number of Nodes	TPR
Normal attack	10	90%
Normal attack	30	84%
Normal attack	60	81%
Stealthy attack	10	83%
Stealthy attack	30	78%
Stealthy attack	60	74%

These findings highlight the strength of the proposed method in dealing with more sophisticated and adaptive threats, making it suitable for real-world deployments where attackers may employ stealth techniques to avoid detection.

As shown in Table 7, NADSA is evaluated under both the original and stealthy attack scenarios across different network sizes comprising 10, 30, and 60 nodes. The results indicate that NADSA retains its effectiveness even under stealthy attack conditions. Specifically, in a network of 10 nodes, the true positive rate (TPR) was 90% under the original scenario and 83% under the stealthy scenario. For a 30-node network, the TPR decreased from 84% to 78%, and in the 60-node configuration, it dropped from 80% to 74%.

Although the TPR values are lower in the stealthy scenario, the observed performance degradation remains moderate. This demonstrates that while adaptive attacker behavior can reduce detection accuracy, NADSA is still capable of identifying such threats with reasonable reliability. These findings highlight the resilience of the proposed method in realistic and challenging network environments, where malicious behavior may be subtle and not immediately apparent.

4.2.7 Detection Latency Analysis

To assess the applicability of NADSA in time-sensitive intrusion detection scenarios, we measured its detection latency and compared it with other established methods. Detection latency is defined as the duration between the initiation of an attack and the system's successful identification and response to the malicious activity. This metric is critical in environments where delayed detection can lead to significant degradation of network performance and security.

As shown in Table 8, the detection times for each method are as follows: NADSA detects attacks in 50 s, while SoS RPL requires 60 s, INTI takes 110 s, and RFTRUST achieves the fastest detection at 40 s. These results indicate that NADSA performs competitively in terms of latency, offering faster detection than SoS RPL and INTI, and approaching the responsiveness of RFTRUST. Given its balanced performance in both detection accuracy and latency, NADSA is well suited for practical deployment in real time intrusion detection scenarios.

Table 8: Detection Time Comparison of Intrusion Detection Methods

Method	Detection time (seconds)
NADSA	50
SoS-RPL	60
INTI	110
RFTRUST	40

5 Future Directions and Improvements

Although NADSA demonstrates strong performance in detecting sinkhole attacks, future research could focus on enhancing its effectiveness and expanding its applicability. One potential direction is to extend NADSA to identify other prevalent Internet of Things threats, such as selective forwarding, wormhole, and Sybil attacks. Developing a more comprehensive detection framework would improve the overall security of resource constrained networks and increase the robustness of intrusion prevention systems in diverse operating conditions.

6 Conclusion

The Internet of Things consists of web enabled smart devices that facilitate communication between physical entities. However, this emerging paradigm introduces significant security challenges, one of which is the sinkhole attack. This type of attack can severely disrupt network operations, yet it has not been thoroughly addressed in existing research. Although several intrusion detection methods have been proposed to counter sinkhole attacks, their effectiveness tends to diminish in networks that are extra lossy and sparse.

In this study, a new method named NADSA was introduced, comprising two stages to identify and isolate malicious nodes. Simulation results demonstrate that NADSA outperforms existing methods such as INTI, SoS RPL, and RFTRUST, particularly in challenging network conditions. Specifically, NADSA achieves better performance in terms of packet delivery rate (68%, 70%, and 73%), end to end delay (81 ms, 72 ms, and 60 ms), true positive rate (89%, 83%, and 80%), and false positive rate (24%, 28%, and 33%). While the compared methods perform well in dense networks with low retransmission, their performance deteriorates in sparse and lossy conditions. NADSA maintains its reliability by leveraging multiple parameters such as distance measurement, DIO count, expected transmission count, and received signal strength indicator to detect sinkhole attacks effectively.

Despite its promising results, NADSA has certain limitations. Although it remains effective against stealthy or adaptive attackers, its detection accuracy shows a moderate decline in such scenarios. This suggests that more sophisticated evasion strategies may still pose challenges. Furthermore, as with other fuzzy based systems, NADSA's performance is influenced by the design of its membership functions and rule sets, which may require refinement for deployment in diverse network environments.

Acknowledgement: Not applicable.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Atena Shiranzaei, Emad Alizadeh; methodology and software development: Atena Shiranzaei, Emad Alizadeh; data collection and investigation: Atena Shiranzaei, Emad Alizadeh; analysis and interpretation of results: Atena Shiranzaei, Emad Alizadeh, Mahdi Rabbani; validation: Mahdi Rabbani; draft manuscript preparation: Atena Shiranzaei; writing—review and editing: Atena Shiranzaei, Emad Alizadeh, Mahdi Rabbani, Sajjad Bagheri Baba Ahmadi, Mohsen Tajgardan; supervision and project administration: Atena Shiranzaei. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the Corresponding Author, Atena Shiranzaei, upon reasonable request.

Ethics Approval: Not applicable.

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

References

1. Shiranzaei A, Khoshbakht F. A novel detection method for grey hole attack in RPL. Int J Recent Innov Trends Comput Commun. 2023;11(7s):492–7. doi:10.17762/ijritcc.v11i7s.7027.
2. Challa S, Wazid M, Das AK, Kumar N, Reddy AG, Yoon EJ, et al. Secure signature-based authenticated key establishment scheme for future IoT applications. IEEE Access. 2017;5:3028–43. doi:10.1109/access.2017.2676119.
3. Alsukayti IS, Alreshoodi M. RPL-based IoT networks under simple and complex routing security attacks: an experimental study. Appl Sci. 2023;13(8):4878. doi:10.3390/app13084878.
4. Ashrif FF, Sundararajan EA, Ahmad R, Hasan MK, Yadegaridehkordi E. Survey on the authentication and key agreement of 6LoWPAN: open issues and future direction. J Netw Comput Appl. 2023;221(1):103759. doi:10.1016/j.jnca.2023.103759.
5. Al-Amiedy TA, Anbar M, Belaton B, Kabla AHH, Hasbullah IH, Alashhab ZR. A systematic literature review on machine and deep learning approaches for detecting attacks in RPL-based 6LoWPAN of internet of things. Sensors. 2022;22(9):3400. doi:10.3390/s22093400.
6. Rudrakar S, Rughani P. IoT based agriculture (Ag-IoT): a detailed study on architecture, security and forensics. Inf Process Agric. 2024;11(4):524–41. doi:10.1016/j.inpa.2023.09.002.
7. Alfriehat N, Anbar M, Aladaileh M, Hasbullah I, Shurbaji TA, Karuppayah S, et al. RPL-based attack detection approaches in IoT networks: review and taxonomy. Artif Intell Rev. 2024;57(9):248. doi:10.1007/s10462-024-10907-y.
8. Wu D, Yang B, Wang H, Wang C, Wang R. Privacy-preserving multimedia big data aggregation in large-scale wireless sensor networks. ACM Trans Multimed Comput Commun Appl (TOMM). 2016;12(4s):1–19. doi:10.1145/2978570.
9. Al-Sarawi S, Anbar M, Alabsi BA, Aladaileh MA, Rihan SDA. Passive rule-based approach to detect sinkhole attack in RPL-based internet of things networks. IEEE Access. 2023;11:94081–93. doi:10.1109/access.2023.3310242.
10. Mishra N, Pandya S. Internet of things applications, security challenges, attacks, intrusion detection, and future visions: a systematic review. IEEE Access. 2021;9:59353–77. doi:10.1109/access.2021.3073408.

11. Wazid M, Das AK, Kumari S, Khan MK. Design of sinkhole node detection mechanism for hierarchical wireless sensor networks. Secur Commun Netw. 2016;9(17):4596–614. doi:10.1002/sec.1652.
12. Islam MS, Tasnim M, Kabir U, Jahan M. Securing smart home against sinkhole attack using weight-based IDS placement strategy. IET Wire Sens Syst. 2023;13(6):216–34. doi:10.1049/wss2.12069.
13. Tumrongwittayapak C, Varakulsiripunth R. Detecting sinkhole attacks in wireless sensor networks. In: 2009 ICCAS-SICE; 2009 Aug 18–21; Fukuoka, Japan. p. 1966–71.
14. Raza S, Wallgren L, Voigt T. SVELTE: real-time intrusion detection in the Internet of Things. Ad Hoc Netw. 2013;11(8):2661–74. doi:10.1016/j.adhoc.2013.04.014.
15. Surendar M, Umamakeswari A. InDReS: an intrusion detection and response system for internet of things with 6LoWPAN. In: 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET); 2016 Mar 23–25; Chennai, India. p. 1903–8.
16. Cervantes C, Poplade D, Nogueira M, Santos A. Detection of sinkhole attacks for supporting secure routing on 6LoWPAN for Internet of Things. In: 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM); 2015 May 11–15. Ottawa, ON, Canada. p. 606–11.
17. Zarpelão BB, Miani RS, Kawakani CT, de Alvarenga SC. A survey of intrusion detection in Internet of Things. J Netw Comput Appl. 2017;84(3):25–37. doi:10.1016/j.jnca.2017.02.009.
18. Glissa G, Rachedi A, Meddeb A. A secure routing protocol based on RPL for Internet of Things. In: 2016 IEEE Global Communications Conference (GLOBECOM); 2016 Dec 4–8; Washington, DC, USA. p. 1–7.
19. Airehrour D, Gutierrez JA, Ray SK. SecTrust-RPL: a secure trust-aware RPL routing protocol for Internet of Things. Fut Gen Comput Syst. 2019;93:860–76. doi:10.1016/j.future.2018.03.021.
20. Qureshi KN, Rana SS, Ahmed A, Jeon G. A novel and secure attacks detection framework for smart cities industrial internet of things. Sustain Cities Soc. 2020;61(1):102343. doi:10.1016/j.scs.2020.102343.
21. Zaminkar M, Fotohi R. SoS-RPL: securing internet of things against sinkhole attack using RPL protocol-based node rating and ranking mechanism. Wire Pers Commun. 2020;114(2):1287–312. doi:10.1007/s11277-020-07421-z.
22. Almusaylim ZA, Jhanjhi N, Alhumam A. Detection and mitigation of RPL rank and version number attacks in the internet of things: SRPL-RP. Sensors. 2020;20(21):5997. doi:10.3390/s20215997.
23. Prathapchandran K, Janani T. A trust aware security mechanism to detect sinkhole attack in RPL-based IoT environment using random forest-RFTRUST. Comput Netw. 2021;198(6):108413. doi:10.1016/j.comnet.2021.108413.
24. Kumar A, Budhiraja I, Garg D, Garg S, Choi BJ, Alrashoud M. Advanced network security with an integrated trust-based intrusion detection system for routing protocol. Alex Eng J. 2025;120(12):378–90. doi:10.1016/j.aej.2025.01.087.
25. Vasseur JP, Kim M, Pister K, Dejean N, Barthel D. Routing metrics used for path calculation in low-power and lossy networks. Fremont, CA, USA: Internet Engineering Task Force (IETF); 2012.
26. Kurniawan A. Practical Contiki-NG: programming for wireless sensor networks. 1st ed. New York City, NY, USA: Apress; 2018.
27. Osterlind F, Dunkels A, Eriksson J, Finne N, Voigt T. Cross-level sensor network simulation with cooja. In: 2006 31st IEEE Conference on Local Computer Networks; 2006 Nov 14–16; Tampa, FL, USA: IEEE. p. 641–8.
28. Thomson C, Romdhani I, Al-Dubai A, Qasem M, Ghaleb B, Wadhaj I. Cooja simulator manual. Edinburgh, UK: Edinburgh Napier University; 2016.
29. Oikonomou G, Duquennoy S, Elsts A, Eriksson J, Tanaka Y, Tsiftes N. The Contiki-NG open source operating system for next generation IoT devices. SoftwareX. 2022;18(4):101089. doi:10.1016/j.softx.2022.101089.
30. Makarem N, Diab WB, Mougharbel I, Malouch N. On the design of efficient congestion control for the Constrained Application Protocol in IoT. Comput Netw. 2022;207(1):108824. doi:10.1016/j.comnet.2022.108824.