Fuzzy-Logic Based IDS for Detecting Jamming Attacks in Wireless Mesh IoT Networks

Michael Savva∗†, Iacovos Ioannou∗†, and Vassos Vassiliou∗†
∗Department of Computer Science, University of Cyprus
†CYENS Centre of Excellence, Nicosia, Cyprus

Abstract—The investigation in this paper targets the design and the evaluation of jamming intrusion detection based on Fuzzy Logic in wireless mesh IoT Networks in a distributed manner. Our approach uses information collected at local nodes and from the sink as input to the fuzzy logic controller. In order to find the best set of inputs, distributed or centralized, we made a comparison between five different combinations of parameters. The investigation uses the values of the ETX, Retransmissions, Packets Drop per terminal (PDPT) and Packet Delivery Ratio (PDR) as inputs to a fuzzy inference system to get Jamming Index (JI) as the system’s output. The proposed method was evaluated based on the following metrics: Accuracy, Precision, Specificity, False positive rate (FPR), Recall, False negative rate (FNR) and ROC. In order to evaluate this approach, we implement simulations in various scenarios using the Contiki OS and the Cooja simulator tool.

Keywords—Jamming, Intrusion Detection, Fuzzy logic, Internet of Things, Wireless Sensor Networks

I. INTRODUCTION

Internet of Things (IoT) comprises sensors and actuators included in embedded devices with the use of IPv4 and IPv6 technologies, enabling these electronic devices to connect and exchange data that can inter-operate within the existing Internet infrastructure. The IoT devices collect valuable data with the help of various technologies and then, autonomously, transfer these data between other devices [1]. However, IoT devices and networks are often connected to critical infrastructure networks. As a result, security and protection are needed. IoT networks such as WSNs are vulnerable to several threats as the nature of such devices has many constraints, for example, limited storage, low computation power and limited power consumption [2]. Additionally, the coordination of diverse technologies, heterogeneity and the distributed nature of the network magnifies the threats to the IoT system [2]. Due to the constraints involved in the IoT, not all existing security mechanisms can be applied to protect such devices. As a consequence, IoT devices remain exposed to various vulnerabilities, which keep the associated infrastructure and applications in danger [3].

There are so many attacks against IoT networks that can downgrade their performance and functionalities [4]. However, one of the most common and dangerous attacks is a jamming attack [5]. IoT systems based on wireless mesh technologies are vulnerable to jamming attacks which could drain the battery of target devices, by disrupting their data transmission and making them repeatedly re-transmit [6]. Jamming in wireless networks has become a significant research problem due to the ease of its execution [7].

Furthermore, jamming threats can only be prevented at the physical layer (PHY) but not at the MAC or network layer. When a wireless network suffers from jamming attacks, its wireless signals are typically overwhelmed by irregular or sophisticated radio jamming signals, making it hard for legitimate wireless devices to decode data packets. Consequently, any approaches at the MAC layer or above are incapable of preventing jamming threats, and innovative anti-jamming approaches are required at the physical layer [8].

A practical solution for detecting malicious behaviour within the network is an Intrusion Detection System. In this paper, we provide an intelligent and adaptive intrusion detection technique using Fuzzy logic algorithms in a distributed1 and centralised2 manner, capable of performing detection locally (with the use of ETX, Retransmissions and PDPT as inputs) or centrally (with the help of PDR input) in a network. The main contributions of the paper relate to: i) method to perform local detection; ii) detection of Jamming attacks at the MAC and Network layer; iii) comparison of five different combinations of parameters as inputs to the Fuzzy controller, and iv) evaluation of different jammer settings (48 different locations of the jammer).

The rest of the paper is structured as follows. Section II provides background information on jamming attacks. It also provides related work on approaches addressing jamming attacks using fuzzy logic. The problem description is provided in Section III. Specifically, the examined approaches of Anomaly Detection using Fuzzy Logic are provided in Section III-A. The efficiency of the investigated approaches is examined, evaluated and compared in Section IV. Finally, Section V includes concluding remarks.

II. BACKGROUND KNOWLEDGE AND RELATED WORK

This section provides a brief description of Jamming attacks and related work on using Fuzzy Logic algorithms for detecting jamming attacks.

A. Jamming Attacks

This subsection provides background knowledge regarding the jamming attack.

Jamming is a type of attack that interferes with the radio frequencies the network nodes are using [9]. A jamming source may either be powerful enough to disrupt the entire network or less powerful and only disrupt a smaller portion of the network. The most common and dangerous attack which can be proved harmful for wireless mesh WSN or IoT networks is jamming attack [5]. Wireless networks are especially vulnerable to radio jamming attacks for the reason that the jamming attacks are straightforward to launch [8].

1The decision is executed at the Node.
2The decision is executed at the Sink.
An attacker can easily Generate a jamming attack without requiring any special hardware [10] and without requiring information about the internals of the control system [11].

1) Types of Jamming Attacks: According to the literature [12], [13] there are three types of jammers, which are the most common. The Proactive, the Reactive, and the Specific-Function jammer.

a) Proactive Jammers: A Proactive Jammer attacks the network regardless of any data communication. It randomly transmits bits on the network, making all the functional nodes non-responsive. Finally, it functions only on a single channel and operates until its energy is depleted [5], [7]. Proactive Jammers can be categorized into Constant Jammers, Deceptive Jammers and Random Jammers.

1) Constant Jammer: A Constant Jammer continuously emits radio signals that interfere with the transmission of the network [14]. Furthermore, the Constant Jammer emits random signals which do not follow any underlying MAC protocol [13]. This type of jammer aims at keeping the channel busy and damaging the nodes’ communication [13]. On the other hand, constant jammer attacks are energy inefficient and also can be easily detected [7]. Moreover, this type of attack can be easily implemented, easily identified and works on single-channel [5].

2) Deceptive Jammer: Compared to a Constant Jammer, the Deceptive Jammer is more challenging to detect because it transmits legitimate packets instead of random bits. Similarly to the Constant Jammer, the Deceptive Jammer is also energy inefficient due to the continuous transmission; however, it has very easy implementation [7] and difficult detection [5].

3) Random Jammer: On the other hand, the random jammer alternates from the sleeping mode to the Jamming mode [7], [15]. It can either behave like a Constant Jammer or a Deceptive Jammer during its jamming phase. In contrast to the previous two jammers, this one reduces power consumption [7]. However, it is less effective than the two abovementioned jammers and incapable of jamming during sleep mode [5].

b) Reactive Jammers: A Reactive Jammer listens to the channel activity. If it identifies an action, it immediately sends out a random signal to collide with the existing signal on the channel [13]. The amount of power required to listen to a channel is much less in comparison with the power needed for proactive jamming [14]. Unfortunately, reactive jammers are difficult to detect, challenging to design, energy inefficient and work on a single-channel [5].

1) Reactive RTS/CTS Jammer: In this attack, the jammer begins the offensive when it senses a request-to-send (RTS) message to transmit from the sender. As a result, the receiver cannot send back a clear-to-send (CTS) reply because the RTS packet sent from the sender was destroyed. Finally, the sender will not send data because it believes that the receiver is busy with another ongoing transmission. Consequently, the sender is not sending data, and the receiver always waiting for the data packet [7].

2) Reactive Data Acknowledgement Jammer: In the Data/ACK attack, the jammer destroys the transmissions of data or acknowledgement (ACK) packets. The attacker does not respond until a data transmission begins at the transmitter end. This type of jammer can corrupt data packets or ACK packets. As a result, we can view an increase in retransmissions. That is happening because the data packets are not received correctly at the receiver in case of data transmissions. In the case of ACK, since the sender does not receive the ACKs, it believes something is going wrong at the receiver side [7].

c) Specific-Function Jammer: These types of jammers are manufactured to achieve a specific function. For example, they can be used to interfere with a single channel, or they can cause the jamming of the whole network depending upon their purpose, which means they can minimize their energy consumption or can increase their maximum throughput [5], [7].

1) Follow-on Jammer: This category of the jammer hops over all available channels very frequently and jams each channel for a short period [13]. If a transmitter detects an attack at a specific frequency and hops to another frequency, then the follow-on jammer will scan the channel and hop in the spectrum where there is traffic, or they can randomly hop and jam in different frequencies. To conclude, the follow-on jammer is particularly effective against anti-jamming techniques such as the frequency hopping spread spectrum (FHSS), which uses a slow-hopping rate [5], [7].

2) Channel-hopping Jammer: In the Channel-hopping attack, the jammer interferes while hopping between different channels. Besides, a jammer has direct access to channels because it can override the CSMA algorithm of the MAC protocol. Furthermore, Channel-hopping jammers can jam multiple channels at the same time. Therefore, the jammer is quiet and invisible to its neighbours during the discovery phase, and it starts performing attacks on different channels at different times according to a predetermined pseudorandom sequence [5], [7].

3) Pulse Noise Jammer: This attack category can switch channels and jam on different bandwidths at different periods. Moreover, the pulse noise jammer saves energy by turning off and following the jammer’s programming. Pulse noise jammers can attack simultaneously in multiple channels [5], [7].

2) Parameters for Jamming Attack Detection: The jamming detection parameters that were applied in existing systems will be discussed in this section.

- **Packet Send Ratio (PSR):** Xu et al. [12] defined PSR as the ratio of packets that are successfully sent out by a legitimate traffic source compared to the number of packets it intends to send out at the MAC layer.
- **Packet Delivery Ratio (PDR):** Xu et al. [12] describe PDR as the ratio of packets that are successfully delivered to a destination compared to the number of packets that have been sent out by the sender. [16]. From the outcomes, it was recognised that the PSR and PDR were difficult to decide about jamming and its types [17].
- **Average number of required transmissions per packet (ATX):** Heo et al. [18] define the ATX metric as a total number of transmissions divided by the number of successfully received unique packets.
- **Number of hops for received packets:** A Hop count refers to the number of routers that a packet goes through before reaching its destination.
through from its source to its destination [19].

- **Throughput**: Agah and Das [19] define Throughput as the measure that characterises the total number of forwarded packets over the total number of received packets.

- **Bit Error Rate (BER)**: According to Strasser et al. [20], the BER is calculated as the ratio of the number of corrupted bits to the number of total bits received by a node during a transmission session. However, it is hard to measure the BER by a sensor node since it needs to collect a tremendous amount of data. Moreover, this method cannot classify different kinds of jamming attacks [17].

- **Packets Dropped per Terminal (PDPT)**: Misra et al. [21], Cakiroglu et al. [22] and Balarengadurai et al. [23] used PDPT for detection of Jamming attacks. PDPT refers to the ratio of the number of received packets that have not passed the Cyclic Redundancy Check (CRC) carried out by the node, to the total number of packets received by the node over a given period.

- **Signal-to-Noise Ratio (SNR)**: SNR measured as the ratio of the received signal power at a node to the received noise power at the node. Misra et al. [21], Balarengadurai et al. [23] and Sasikala and Rengarajan [24] used SNR as a metric to detect Jamming attacks. The SNR is a useful metric to identify the behaviour of jamming at the physical layer [17].

- **Energy Consumption**: Cakiroglu et al. [22] define Energy Consumption as the approximated energy amount consumed in a specified time for a sensor network [15], [16].

- **Delay**: Delay in [15] is calculated as a total time from the transmission of a packet from a node to the sink, to the time when the sink received the packet.

- **Received Signal Strength Indicator (RSSI)**: RSSI [16], [17], [25] is defined as the power content of the radio signal when received by the receiver.

- **Packet Loss Ratio (PLR)**: PLR is calculated as the number of packets lost divided by the number of packets sent [25], [26].

- **Routing Overhead**: Chen et al. [26] proposed routing overhead as the average number of routing packets (including DIS, DIO, and DAO packets) transmitted in the whole network every minute.

- **Expected Transmission Count (ETX)**: ETX [27] represents the expected number of transmissions required to successfully transmit and acknowledge a packet on a wireless link.

Our solution uses as parameters in the fuzzy logic the values of the ETX, Retransmissions, PDPT and PDR as inputs to a fuzzy inference system. The decision of choosing these values is based on selecting metrics where collected at the node or Sink node (PDR). Additionally, from our empirical experience and also from our experimental results, these values extremely increase when having a Jammer attack.

B. Related Work on Using Fuzzy Logic Algorithms for Detecting Jamming Attacks

Fuzzy logic dealing with vagueness and imprecision has a capacity to describe imprecise forms of reasoning in areas where firm decisions have to be made in indefinite conditions and is found to be proper for intrusion detection [28]. Following the literature, Detection frameworks based on fuzzy logic have the capability to calculate ambiguous information availability [29]. Fuzzy logic can be making actual-time decisions, even with incomplete knowledge. Conventional control systems rely on an accurate representation of the environment, which commonly does not exist in reality. Fuzzy logic systems, which can handle the linguistic rules naturally, are suitable in this respect. Moreover, it can be used for context by blending different parameters – rules combined to produce a suitable result [23]. Furthermore, Fuzzy rules leave us to efficiently and easily construct if-then rules that reflect general ways of describing security attacks [30]. Thus, fuzzy logic can be an adequate means of defining network attacks [30]. AI methods such as decision trees, neural networks and fuzzy logic are applied for detecting anomalies in a network, in which a fuzzy-based system presents important advantages over other AI techniques [31]. Our approach uses a combination of the following metrics ETX, Retransmissions, PDPT and PDR as inputs to a Fuzzy inference system (Mandani’s Fuzzy Inference System) [32] to get Jamming Index (JI) as the output of the system. The JI value is between 0 and 1, signifying No Jamming to Absolute Jamming, respectively. According to the literature, the following works take into account the jamming attack and use Fuzzy logic algorithms to detect malicious activity in wireless mesh networks.

Misra et al. [21] proposed a fuzzy inference system for Jamming attack detection. In this approach, network nodes receive input values, while the base station does the jamming detection following a centralized methodology. The nodes send three inputs to the base station: the number of total packets received during a specified period, the number of packets dropped during the period, and the received signal strength (RSS). With these metrics, the base station can calculate the PDPT and SNR. Afterwards, the central node uses the values of PDPT and SNR as inputs to the fuzzy inference system to extract the jamming index. Finally, they make a confirmation of a Jamming Attack on a node. The validation is done through a 2-Means Clustering algorithm that constructs a confirmatory check through the study of the neighborhood of a node, to ascertain the correctness of the JI grade allotted to that node compared to the JI distributed to its neighbour nodes. This work is done using NS2, MATLAB and Simulink simulator and examined four types of jammers constant, deceptive, random, and reactive in 720 different simulations setups. In simulations, there are four positions for the jammer, two inside and two outside the grid. Six sets of inter-nodal distances: 5, 10, 15, 20, 25, and 30 meters and three sets of nodes: 25, 50, and 100. Finally, researchers have done simulations using standard power at the jammer node and high energy at the jammer node.

Balarengadurai and Saraswathi [33] detect Jamming attacks at the PHY and MAC layers in IEEE 802.15.4 low rate wireless personal area network using Fuzzy logic systems. This system used three inputs being sent by the nodes to the base station. The inputs are a) the number of total packets received by it during a specified period, b) the number of packets dropped by it during the period, and c) the received signal strength (RSS). Afterwards, the base station calculates the PDPT and SNR from these values, and then the base station uses the metrics of PDPT and SNR as inputs to a
fuzzy logic system to get Jamming Index as the output of the system. Finally, the confirmation of the Jamming attack Detection is done through the Fuzzy K-Means Clustering. In these experiments, used S-MAC protocol as the MAC standard. In simulations, the nodes sent one packet every 5 seconds for light traffic and two packages every 1 second for heavy traffic. In addition, the authors examined four types of jammers constant, deceptive, random, and reactive—the evaluation of this approach simulated in a Network Simulation environment.

Vijayakumar et al. [17] proposed a fuzzy logic-based jamming detection algorithm (FLJDA) to detect the presence of jamming in downstream data communication for cluster-based wireless sensor networks. FLJDA monitors the behavior of nodes by computing the jamming probability using two inputs in a fuzzy logic system: the packet delivery ratio and the received signal strength indicator. The evaluation of this approach is simulated in MATLAB. The authors examined four types of jammers constant, deceptive, random, and reactive.

Meenalochni and Sudha [25] proposed a hybrid algorithm based on Fuzzy logic and Ant Colony Optimization for the detection of jamming attacks. The evaluation of this approach is simulated in MATLAB. The authors used fuzzy logic to detect the interference node and the ant colony to route the data even in the presence of jamming. The ant colony approach discards the congested node and identifies a path from source to destination for successful transmission. The fuzzy logic system used PDR, PLR, and RSSI as input to determine the node’s jamming percentage. The authors used 12 wireless Zigbee real nodes where node one was assigned as Base Station and connected to a Laptop to display the received data in real-time.

In contrast with existing solutions shown above, which also detect jamming attacks using a fuzzy logic algorithm, our approach uses only two metrics as input. It acts with a lightweight edition of an IDS to achieve a high-performance evaluation system. Our solution uses only one algorithm in comparison with Mirsa et al. [21] who used the 2-means clustering algorithm additionally in order to achieve the best results. Similarly, Balarengadurai and Saraswathi [33] used an additional K-means clustering algorithm. Finally, Meenalochani and Sudha [25] used three metrics as inputs, the PDR, PLR and RSSI and additionally used the Ant Colony Optimization algorithm. Overall, none of the researchers examined so comprehensively the position of jammers constant, deceptive, random, and reactive.

Meenalochani and Sudha [25] used three metrics as inputs, the PDR, PLR, and RSSI and additionally used the Ant Colony Optimization algorithm. Overall, none of the researchers examined so comprehensively the position of jammers. Our approach examines forty-eight different metrics. Firstly, we get the results where the inputs of the Fuzzy controller are ETX, Retransmissions, PDPT and PDR: LOW, MEDIUM, and HIGH. In addition, we define four fuzzy sets over the universe of discourse (output), JI: NO attack, LOW, MEDIUM, and HIGH. We use Mamdani’s model [32], where combinations of ETX, Retransmissions, PDPT and PDR are the crisp inputs to the system, and JI is the crisp output obtained from the system after defuzzification using the centroid method.

A trapezoid shape is preferred to define fuzzy membership functions because of two reasons: firstly, it can be mathematically manipulated to be very close to the most natural feature, the Gaussian or Bell function, and secondly, it can be easily manipulated to be an unsymmetrical function (as required in the instant case) where the same cannot be done so easily with the Gaussian or Bell functions [21]. In this study, we chose trapezoidal shapes as an appropriate membership function for our fuzzy logic controller. The decision of which of the methods is going to use depends completely on the problem size and problem type [34]. The choice of trapezoidal shapes depends on the distribution of our data. In comparison with Gaussian fuzzy sets, the Trapezoidal shapes are easy to implement and fast to calculate [34]. Following the literature [35], there are two strategies for constructing the fuzzy sets: a) model-driven and b) knowledge-driven. Gaussian fuzzy sets can only be created from the model-driven approach, whereas trapezoidal fuzzy sets can be constructed from both model-driven and knowledge-driven approaches. As a result, working with the trapezoidal fuzzy sets gives the user more freedom in membership function construction [35]. To conclude, Trapezoidal fuzzy logic controllers are simpler in analysis [35].
TABLE I
VALUES OF VARIABLES USED IN THE DEFINITION OF MEMBERSHIP FUNCTIONS

| Universe of discourse (uod) | Set | a   | b   | c   | d   |
|-----------------------------|-----|-----|-----|-----|-----|
| ETX                         | LOW | -4  | -2  | 2   | 3   |
|                             | MEDIUM | -2  | 3   | 10  | 11  |
|                             | HIGH | 10  | 11  | 18  | 22  |
| Retransmissions             | LOW | -150 | -100 | 500  | 660 |
|                             | MEDIUM | 500  | 600  | 1100 | 1400 |
|                             | HIGH | 1100 | 1200 | 4100 | 4500 |
| PDPT                        | LOW | -10 | -8  | 8   | 10  |
|                             | MEDIUM | 5  | 10  | 78  | 80  |
|                             | HIGH | 78  | 80  | 148 | 150 |
| PDR                         | LOW | -0.5 | 0   | 0.45 | 0.5 |
|                             | MEDIUM | 0.45 | 0.5 | 0.78 | 0.8 |
|                             | HIGH | 0.78 | 0.8 | 1   | 1.05 |
| Jamming Indicator (JI)      | LOW | -0.05 | 0   | 0.25 | 0.5 |
|                             | MEDIUM | 0.25 | 0.5 | 0.75 | 0.8 |
|                             | HIGH | 0.75 | 0.8 | 1   | 1.05 |

B. Membership Functions

We designate the membership functions below:

\[
\mu_{\text{set}}(\text{uod}) = \begin{cases} 
\frac{\text{uod} - \alpha}{b - \alpha}, & \alpha \leq \text{uod} \leq b \\
1, & b < \text{uod} < c \\
\frac{d - \text{uod}}{d - c}, & c \leq \text{uod} \leq d \\
0, & \text{otherwise} 
\end{cases}
\]  

(1)

Where the separate values of the variables are as provided in Table I. The values of the variables, as presented in Table I, have been fixed through by the improvement of these values through a feedback factor generated by contrasting the original result (the output, JI of the method) and the expected outcome (the JI value). A fuzzy logic system can be constructed using observed data [25]. In order to find the ranges of the variables, we observe the values with the minimum value, the maximum value, the average value and the standard deviation. Furthermore, we observe the distribution of the values to build the membership functions. The graphical illustrations of these trapezoidal functions in respect of ETX, Retransmissions, PDPT, PDR and JI are shown in Figures 1, 2, 3, 4 and 5, respectively. Additionally, in Figures 6, 7, 8, 9 and 10 we show the Input-output surface corresponding to the membership values of inputs.

C. Detection Algorithm

The algorithm for detecting the jamming node is shown below.

**Algorithm 1:** Algorithm for detection of jammed node using fuzzy inference system

**Input:**
- ETX and Retransmissions (Distributed), PDPT and Retransmissions (Distributed), PDR and Retransmissions (Centralized), ETX and PDR (Centralized)

**Output:**
- Jamming Indicator (JI)

**BEGIN**
- Get the values of ETX and Retransmissions, PDPT and Retransmissions, PDR and Retransmissions, ETX and PDR of the nodes
- Fuzzify the crisp input parameters: ETX and Retransmissions, PDPT and Retransmissions, PDR and Retransmissions, ETX and PDR
- Apply the fuzzy rule base and get the fuzzy output
- Defuzzify the fuzzy output to get the crisp output
- The defuzzified crisp output gives the percentage of jamming of a node

**END**

In the approach where the inputs are ETX and Retransmissions the fuzzy rule base is given below:

1) If ETX is LOW and Retransmissions is LOW then JI is No ATTACK.
2) If ETX is LOW and Retransmissions is MEDIUM then JI is LOW.
3) If ETX is LOW and Retransmissions is HIGH then JI is MEDIUM.
4) If ETX is MEDIUM and Retransmissions is LOW then JI is No ATTACK.
5) If ETX is MEDIUM and Retransmissions is MEDIUM then JI is LOW.
6) If ETX is MEDIUM and Retransmissions is HIGH then JI is MEDIUM.
7) If ETX is HIGH and Retransmissions is LOW then JI is LOW.
8) If ETX is HIGH and Retransmissions is MEDIUM then JI is MEDIUM.
9) If ETX is HIGH and Retransmissions is HIGH then JI is HIGH.

In the approach where the inputs are PDPT and Retransmissions, the fuzzy rule base is given below:

1) If PDPT is LOW and Retransmissions is LOW then JI is No ATTACK.
2) If PDPT is LOW and Retransmissions is MEDIUM then JI is LOW.
3) If PDPT is LOW and Retransmissions is HIGH then JI is MEDIUM.
4) If PDPT is MEDIUM and Retransmissions is LOW then JI is No ATTACK.
5) If PDPT is MEDIUM and Retransmissions is MEDIUM then JI is LOW.
6) If PDPT is MEDIUM and Retransmissions is HIGH then JI is MEDIUM.
7) If PDPT is HIGH and Retransmissions is LOW then JI is LOW.
8) If PDPT is HIGH and Retransmissions is MEDIUM then JI is MEDIUM.
9) If PDPT is HIGH and Retransmissions is HIGH then JI is HIGH.

In the approach where the inputs are PDR and Retransmissions, the fuzzy rule base is given below:

1) If PDR is LOW and Retransmissions is LOW then JI is LOW.
2) If PDR is LOW and Retransmissions is MEDIUM then JI is MEDIUM.
3) If PDR is LOW and Retransmissions is HIGH then JI is HIGH.
4) If PDR is MEDIUM and Retransmissions is LOW then JI is LOW.
5) If PDR is MEDIUM and Retransmissions is MEDIUM then JI is MEDIUM.
6) If PDR is MEDIUM and Retransmissions is HIGH then JI is HIGH.
7) If PDR is HIGH and Retransmissions is LOW then JI is NO ATTACK.
8) If PDR is HIGH and Retransmissions is MEDIUM then JI is LOW.
9) If PDR is HIGH and Retransmissions is HIGH then JI is MEDIUM.

In the approach where the inputs are ETX and PDR, the fuzzy rule base is given below:

1) If ETX is LOW and PDR is LOW then JI is LOW.
2) If ETX is LOW and PDR is MEDIUM then JI is MEDIUM.
3) If ETX is LOW and PDR is HIGH then JI is HIGH.
4) If ETX is MEDIUM and PDR is LOW then JI is NO ATTACK.
5) If ETX is MEDIUM and PDR is MEDIUM then JI is LOW.
6) If ETX is MEDIUM and PDR is HIGH then JI is MEDIUM.
7) If ETX is HIGH and PDR is LOW then JI is NO ATTACK.
8) If ETX is HIGH and PDR is MEDIUM then JI is LOW.
9) If ETX is HIGH and PDR is HIGH then JI is MEDIUM.

IV. PERFORMANCE EVALUATION

A. Methodology

In order to evaluate this approach, we implement experiments in various scenarios with the use of Contiki OS and the Cooja simulator tool. In our study, we implement a Deceptive Jamming attack using the JamLab [36] suite, a Contiki-based library that allows repeatable experiments with radio interference. Our jammer continuously emits signals which include legitimate packets that interfere with the communication of the network.
B. Simulation Environment

In our study, we made experiments with the jammer placed in predicted scenarios within a grid of 25 nodes. Our implementation was constructed and evaluated via Cooja O/S and Contiki simulator tool. We extracted the simulation data from the Cooja Simulator, and we processed the data in Python and Matlab.

We placed 25 nodes on a grid in an area of 160 * 160 meters including a central node as a Sink. All nodes have a distance of 40 meters from each other. All nodes are equivalent to TelosB nodes and have a 50-meter transmission range and 70-meter interference range. Each node transmits one data packet of 48 bytes every 10 seconds.

Additionally, we run three different scenarios regarding the position of the sink. The situations are a) sink in the middle of the grid b) sink on top middle of the grid and c) sink on top left edge of the network. Each scenario run 16 different jammer positions and a healthy (benign) scenario. In accordance with the literature, these three scenarios were used in published works [21], [37], [38]. The experimental parameters of the nodes are shown in Table II.

![Simulation Set-up and Configuration](image)

Table II

| Parameter       | Value                      |
|-----------------|----------------------------|
| No. of nodes    | 25                         |
| Area Size       | 160 * 160 meters           |
| Scenario nodes  | TelosB                     |
| Sink Position   | SINK at Grid Center        |
| Scenario duration| 15 minutes                 |
| Transmission rate| 1 packet per 10 second     |
| Propagation Model| Unit Disk Graph            |
| Interference range| 50 meters                  |
| Transmission range| 70 meters                  |
| Routing Protocol | RPL                        |
| MAC layer       | ContikiMAC                 |
| Channel Check Rate| 8 Hz                      |

We used the Routing Protocol for Low power and Lossy links (RPL). For Medium Access Control (MAC) layer we used the CSMA and for Radio Duty Cycle (RDC) we used ContikiMAC with the Channel Check Rate of the 8 Hz.

The Jamming node is using nullmac and nullrdr protocols with the Channel Check Rate of the 128 Hz. The experimental parameters of the jammer are shown in Table III.

![Sink in the Middle of the Grid, Jammer Position 6](image)

Table III

| Parameter       | Value                      |
|-----------------|----------------------------|
| No. of nodes    | 1                          |
| Scenario nodes  | JAMMER                     |
| Type of Jammer  | Nullmac                    |
| Transmission range| 50 meters                  |
| Interference range| 70 meters                  |
| MAX power       | 50 mW                      |
| JAMMER          | ContikiMAC                 |
| Channel Check Rate| 128 Hz                     |

Figure 11 shows the simulation set-up and configuration where the Sink is located in the middle of the grid. The green area shows the transmission range of a node. More specifically, each node can communicate with all other nodes within the green circle. For instance, the Sink node, which has number 1, can communicate with nodes 9, 13, 14 and 18. Additionally, the grey circle around the green area displays the interference range. For example, when node 1 sends packets, nodes 8, 17, 10 and 19 in the grey area cannot receive packets and they are not able to receive packets from other nodes when the node 1 communicates simultaneously [39].

For each topology, we defined two types of scenarios: a malicious situation in which a compromised node is placed in the network, and a no attack scenario in which all nodes are in healthy condition. We run in total 160 malicious scenarios for each topology and one in the reasonable condition where we repeat the normal state ten times to view the deviation of the results. The average costs were used to compare states between normal conditions and attack conditions. We choose the grid scenario to do our experiments because its simplicity helps us build the security framework and understand its weaknesses.

The network topology with the Sink in the middle shows the best case scenario of all three, in which there are four nodes that can be used to reach the Sink and the maximum number of hops from the Sink is four. Additionally, the network topology with the Sink in the top-middle of the grid has three nodes that can directly access the Sink and the maximum number of hops is six. The network topology with the Sink at the top edge indicates the worst-case scenario of all three, has only two nodes that can directly access the Sink and the maximum number of hops is eight [37]. We performed the 16 different scenarios and placed a jammer in each one.

Figure 12 shows a jammer in position 6 with coordinates (-20,-20) when the sink is located in the middle of the grid. From the Figure it can be observed that the attacker jams the nodes with number 8, 9, 13, and the sink. Additionally, the malicious node attacks the nodes with numbers 3, 4, 7, 10, 12, 14, 17, and 18.

C. Jamming attack strategy

In this study we used jammers from the JamLab [36] implementation. More specifically, we used a modified proactive deceptive jammer, with an ON-OFF pattern. The jammer is ON for one second and OFF for 333ms. When ON, the jammer continuously emits interference signals. The attacker
sends packets with data that are not recognised by the nodes. The length of the transmitted data sequence is 8 kilobytes. Therefore, the nodes can transfer packets to the other nodes when the intrusion signal is off. The selection of the specific jamming attack is based on the attributes of the jammer, because it is a simpler jammer without a lot of abequites therefore the usage is simple and straightforward. Finally, each malicious node is placed in the centre of four nodes of the network.

D. Results

In this study, we use the Confusion Matrix as shown in Table IV for the description of the performance.

| TABLE IV CONFUSION MATRIX FOR JAMMING ATTACK DETECTION |
| Actual | Detection |
| Attack | TP | FN |
| No Attack | TN |

where: TP = True Positive, FP = False Positive, TN = True Negative, FN = False Negative [40].

The confusion matrix is a matrix that represents true and false classification results [41]. From the Confusion Matrix, we calculate the indicators of Accuracy, Precision, Specificity, FPR, Recall and FNR.

The True Positive state is when the IDS identifies an activity as an attack, and the event is an attack. A real positive is a successful identification of an attack [40].

The True Negative state is similar. This state is when the IDS identifies an activity as acceptable behaviour, and the activity is acceptable. A true negative is successfully ignoring acceptable behaviour. Neither of these states is harmful as the IDS is performing as expected [40].

The False Positive state is when the IDS identifies an activity as an attack, but the action is acceptable behaviour. A false positive is a false alarm [40].

The False Negative state is the most severe and dangerous state. This situation is when the IDS identifies an activity as acceptable when the event is an attack [40].

Accuracy defined as the percentage of correctly classified records over the total number of records [42]. The equation for Accuracy is shown below.

\[
Accuracy = \frac{TP + TN}{TP + TN + FP + FN}
\] (2)

Additionally, Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. Equation 3 shows how the Precision is calculated.

\[
Precision = \frac{TP}{(TP + FP)}
\] (3)

Furthermore, the Specificity is the proportion of true negative points to negative elements, as calculated using the equation:

\[
Specificity = \frac{TN}{(TN + FP)}
\] (4)

The false positive rate (FPR), represents the ROC curve “X-axis”, as calculated using the equation [43]:

\[
FPR = 1 – Specificity = \frac{FP}{(TN + FP)}
\] (5)

In addition, Recall or true positive rate is the ratio of correctly predicted positive observations to all observations in the actual class. True positive rate represents the ROC curve’s ”Y-axis”.

The equation for Recall is shown below.

\[
Recall = \frac{TP}{(TP + FN)}
\] (6)

The false negative rate (FNR) is calculated using the equation [43]:

\[
FNR = 1 – Recall = \frac{FN}{(TP + FN)}
\] (7)

Finally, the area under the curve (AUC) - receiver operating characteristics (ROC) (AUC - ROC) plot is another indicator that is used to evaluate the performance of classification models. The Receiver Operating Characteristics (ROC) of a classifier shows its performance as a trade off between False Positive Rate and True Positive Rate.

In this study, we performed a total of 160 different simulations for each scenario. We performed simulations with changes to the ranges of membership functions of combination ETX and Retransmissions, combination of PDPT & Retransmissions, combinations of PDR and Retransmissions, combinations of ETX and PDR, combinations of ETX & PDPT and finally, we performed corrections into the fuzzy rules of the model. In the following paragraphs we demonstrate the positions that the Sink is placed, the attacks that are executed in the selected place and the resulting attack identification accuracy rate.

E. Results of the Different Approach

According to the performance measure of TP, TN, FP and FN we calculated the Accuracy rate, the Precision rate, the Specificity, the FPR rate, Recall rate and the FNR rate as shown in the table V.

F. Comparison of Results Between each Investigated Approach

In order to find the optimum combination of the input metrics, we evaluate each solution based on the accuracy rate, the Precision rate, the Specificity, the FPR rate, Recall rate and, the FNR rate. Based on the results, the three combinations 1) ETX & Retransmissions, 2) PDPT & Retransmissions and 3) PDR & Retransmissions are the optimum inputs for the fuzzy controller. Figure 13 shows the Detection Accuracy of the different scenarios of each combination of input metrics. The combinations of ETX & Retransmissions and PDPT & Retransmissions achieve the best accuracy of 95%. Furthermore, we generate the ROC curve of our approach.

Figure 14 shows the ROC Curve when the sink is placed in the middle of the grid. In figure 14, the combination of ETX & Retransmissions, PDPT & Retransmissions and PDR & Retransmissions are better than the combination of ETX & PDR and ETX & PDPT, because at all cut-offs the true positive rate is higher and the false positive rate is lower. The area under the curve for the combination of ETX & Retransmissions, PDPT & Retransmissions and PDR & Retransmissions is larger than the area under the curve for the combinations ETX & PDR and ETX & PDPT.

Figure 15 shows the ROC Curve when Sink is placed in the top middle of the grid. Also, in figure 15 the combination of ETX & Retransmissions, PDPT & Retransmissions and PDR
| Approach                          | Middle | Top middle | Top-left edge | Middle | Top middle | Top-left edge | Middle | Top middle | Top-left edge |
|----------------------------------|--------|------------|---------------|--------|------------|---------------|--------|------------|---------------|
| ETX and Retransmissions          | 95%    | 93.51%     | 93.49%        | 96.98% | 94.20%     | 93.37%        | 95.89% | 95.13%     | 95.13%        |
| PDPT and Retransmissions         | 93.00% | 93.64%     | 93.68%        | 94.68% | 94.68%     | 92.39%        | 92.39% | 92.39%     | 92.39%        |
| PDR and Retransmissions          | 86.60% | 91.68%     | 91.68%        | 86.60% | 91.68%     | 86.60%        | 86.60% | 91.68%     | 86.60%        |
| ETX and PDR                      | 86.50% | 84.64%     | 84.64%        | 80.05% | 78.59%     | 78.59%        | 84.64% | 78.59%     | 78.59%        |

**TABLE VI**

| THE BEST OPERATING POINT |
|---------------------------|
| Approach | Detection Rate | False Alarm Rate |
|---------|----------------|-----------------|
| Middle | Top middle | Top-left edge | Middle | Top middle | Top-left edge | Middle | Top middle | Top-left edge |
| ETX and Retransmissions | 93% | 92.8% | 93.4% | 5% | 8% | 5.6% |
| PDPT and Retransmissions | 93% | 92.9% | 92.9% | 5% | 7% | 7.9% |
| PDR and Retransmissions | 95% | 92.8% | 94.9% | 4% | 7.8% | 4.5% |
| ETX and PDR | 74% | 81.80% | 74.3% | 2.8% | 21% | 25.6% |
| ETX and PDPT | 66% | 79.68% | 76% | 3% | 17% | 26% |

In conclusion, we summarize the best operating point for each approach in table VI. According to the table VI the best approaches are the combination of ETX & Retransmissions, PDPT & Retransmissions and PDR & Retransmissions are still the best.

V. CONCLUSIONS

Overall, results show that the Retransmissions metric, locally collected at the nodes, is the most suitable value as an input for the fuzzy controller. From the results, we observe that the combination of ETX & Retransmissions and PDPT & Retransmissions have the best accuracy achieve at 95%. These set of parameters are having better performance because these parameters are heavily affected by the jamming attacks. Note that both approaches are based on distributed data and decision can be executed at node. Additionally, ROC curves show that the combination of ETX & Retransmissions, PDPT & Retransmissions and PDR & Retransmissions are more useful from the combinations ETX & PDR and ETX & PDPT. Based on the results of the ongoing work, we have identified several open issues that will be studied as part of this research for future work.

REFERENCES

[1] D. Hendricks, “The Trouble with the Internet of Things,” London Datastore. Greater London Authority. Retrieved, vol. 10, 2015.
