IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2022.DOI

Real-Time Jamming Detection in Wireless IoT Networks

FATIMA TU ZAHRA[1], YAVUZ S. BOSTANCI[1,2], AND MUJDAT SOYTURK.[1,2]
[1]Vehicular Networking and Intelligent Transportation Systems Research Lab, Marmara University, Istanbul, Turkey
[2]Department of Computer Engineering, Marmara University, Istanbul, Turkey

Corresponding author: Fatima tu Zahra (e-mail: fatima.zahra@venit.org), Mujdat Soyturk (email: mujdat.soyturk@marmara.edu.tr)

This work is supported in parts by the InSecTT and Beyond5 projects. InSecTT and Beyond5 have received funding from the Electronic Component Systems for European Leadership Joint Undertaking under agreement no. 876038 and 876124 respectively

ABSTRACT IoT-based networks are vulnerable to jamming attacks due to their large-scale deployment and shared communication environment. Resource constraints and the low computational power of IoT devices make it harder to implement high-performance ML-based architectures for jamming detection. In this work, the effects of jamming attacks on a Wi-Fi network are presented and a novel real-time jamming detection mechanism is devised which can identify attacks on multiple channels in 2.4 GHz bandwidth simultaneously. The experiments are conducted in the lab environment by generating the jamming attacks with a Software Defined Radio. Certain QoS parameters in an end-to-end wireless IoT system are collected during normal operating conditions and during jamming attacks. The detection mechanism is implemented on IoT devices by employing the effects of jamming on wireless communication. The proposed real-time jamming detection method has an accuracy of 99% with zero false alarms. It benefits from the communication profile of a wireless network to detect jamming and requires minimal computational resources regarding memory and CPU usage which makes it a low-cost and easily deployable solution for IoT devices.

INDEX TERMS IoT, jamming detection, wireless communication, WiFi, SDR, real-time

I. INTRODUCTION

INTERNET of Things (IoT) has brought significant changes in the structure and functioning of many systems with the advantages and benefits it provides. It has improved daily and industrial life with its contributions to digitization. Numerous application areas such as healthcare, manufacturing, industrial processes, transportation, home automation, environmental monitoring, and security benefit from IoT systems [1]. A wide variety of wireless networks are used to facilitate IoT communications, each with unique characteristics and suitable applications. There are various kinds of IoT networks, including Long Range (LoRa) and Long Range Wide Area Network (LoRaWAN), Narrowband IoT (NB-IoT), Long Term Evolution for Machines (LTEM), SigFox, Zigbee, Z-Wave, and Bluetooth Low Energy (BLE).

LoRa/LoRaWAN offers long-range communication capabilities with low power consumption, making it ideal for applications requiring devices to send small amounts of data over large distances [2]. Similarly, NB-IoT is a Low Power Wide Area Network (LPWAN) radio technology standard that focuses on enabling a wide range of cellular devices and services [3]. LTE-M is a type of LPWAN designed for IoT or machine-to-machine (M2M) communications. LTE-M has advantages in terms of power efficiency and provides a higher data rate compared to other LPWAN technologies, making it suitable for IoT applications that require real-time communication [3].

SigFox offers global IoT connectivity through an LP-WAN that is designed to provide robust, power-efficient, and scalable communications to connected devices [4]. On a smaller scale, Zigbee, Z-Wave, and Bluetooth Low Energy (BLE) are used for creating personal area networks with small, low-power digital radios [5]. Wi-Fi offers high data rates, easy installation, and seamless integration with a lot of industrial and home applications making it a popular choice for IoT networks. It is important to consider different IoT network technologies as they cater to varying requirements such as range, power consumption, scalability, and network coverage. This study focuses on Wi-Fi IoT networks due to their high throughput, low latency, reliable connectivity, and widespread availability and adoption [6]. Furthermore, Wi-Fi networks offer high data transmission speeds, which

TABLE 1: Comparison of IoT Networks

Technology	Throughput (Mbps)	Latency	Max. Range	Power Consumption	Frequency
LoRa/LoRaWAN	<0.05	Seconds	2-15 km	Very Low	Sub-GHz
NB-IoT	0.025-0.250	Seconds to minutes	>10 km	Low	Licensed LTE
LTE-M	1	<10 ms	>10 km	Moderate	Licensed LTE
SigFox	<0.1	Seconds to minutes	30-50 km	Very Low	Sub-GHz
Zigbee	0.25	<15 ms	10-100 m	Low	2.4 GHz
Z-Wave	0.1	<100 ms	30-100 m	Low	900 MHz
BLE	1-2	<3 ms	100-300 m	Very Low	2.4 GHz
Wi-Fi	Up to 600	<10 ms	15-100 m	High	2.4 GHz, 5 GHz

is essential for real-time applications IoT applications that need to transfer large amounts of data quickly, such as video surveillance or smart home applications. A comparison of the various wireless IoT network based on important Key Performance Indicators (KPIs) is provided in Table. 1.

Wi-Fi networks are vulnerable to a large number of cyber attacks, and IoT Systems are more prone to such attacks due to their inherently open and shared nature. Wireless IoT networks are exposed to several malicious attacks not only because they use shared transmission mediums but also because most IoT devices have minimal security features implemented on them due to limited power and computational resources [7].

Jamming is a renowned threat in wireless communication as it can cause severe problems in real-time and security-critical applications. Therefore, it is crucial to identify the jamming attacks to take countermeasures and prevent any harm to the system. Jamming is a subset of Denial of Service (DoS) attacks where an attacker can block or jam the legitimate transmission by injecting spurious packets in the wireless channel in which the devices are operating. Jammers interrupt wireless communication by producing high-power noise near the transmitting and/or receiving nodes across the entire bandwidth of the transmission channel. When IoT devices are exposed to undesirable wireless transmissions of nearby devices (mobile phones, other IoT devices) the interference may cause the communication to be fully or partially blocked. This phenomenon is also known as unintentional jamming. Whereas, in an intentional jamming attack, someone would deliberately try to obstruct the wireless operation. There are various intentional jamming methods namely the constant jammers, deceptive jammers, reactive jammers, intelligent jammers, and random jammers [8]. For some methods, the pattern or the effect of the jammer can be easily identified but for others, the probability of detecting a jammer could be low (e.g. random jammers, intelligent jammers) [9]. Jamming attacks can interrupt communication, cause connectivity problems, avoid the availability of services, and eventually, degrade the performance of IoT devices significantly both regarding energy consumption, as well as the network throughput [9]. Communication interruptions, connection problems, and unavailability of the service can not only negatively affect the performance of the system and/or the inefficient use of resources, but also cause problems with process or system safety. For example, a jamming attack on the IoT devices on the production line can cause damage to the product, other devices, or the production line. It can be harmful to the human operator working on the production line and can cause significant production delays or even halt production altogether. So, designing effective mechanisms to detect and avoid such attacks is of utmost priority.

In this article, an analysis study is presented to understand and detect jamming attacks in Wi-Fi based IoT networks. For this purpose, a jamming test setup is built and several experiments are conducted to extract the normal profile of the communication network and observe the effects of jamming attacks. Certain QoS parameters are collected such as packet density, network throughput, packet delay, connectivity, and wireless link quality from a wireless IoT system. The data is collected from both the IoT client device and the server in normal conditions where there is no external interference and during jamming attacks which are applied to the IoT system with a commercial Software Defined Radio (SDR). Based on the observed effects of the jamming attacks on communication, a novel method is proposed to identify such attacks targeting the system in real time at different communication channels.

The paper is organized as follows: The related work in literature for jamming detection and classification is discussed in Section II. The modeling of the proposed jamming detection method is presented in Section III. Section IV describes the methodology and experimental setup. The performance evaluation results under normal conditions and during the jamming attack are displayed in section V. The comparison with state-of-the-art jamming detection methods is provided in section VI. Section VII is dedicated to discussion. Finally, Section VIII concludes the paper.

II. RELATED WORK

Numerous techniques have been proposed in the literature for jamming classification and detection. An analytical model was presented in [10] to evaluate the wireless network behavior and its performance under jamming attacks. The simulation results of the proposed model indicated that the throughput of the network was directly proportional to the power of the jamming device. It was also shown that the low

transmission rates are more adversely affected by jamming as compared to the high transmission rates. A study was done by [11] to estimate the effect of jamming from a physical layer perspective on wireless networks. They employed tools from stochastic geometry to analyze the performance metrics of wireless networks as well as the error probability of the receiver from a theoretical perspective.

Various studies have been carried out to identify jamming attacks in wireless networks. The research work carried out in [12] and [13] examined and identified the vulnerabilities in the 802.11 MAC layer which could be exploited, and the service could be denied to a legitimate user. In [14] the throughput performance of IEEE 802.11 MAC was investigated against several jammers. They implemented different jamming attacks and compared the results of network throughput concerning each jamming attack to identify the most effective jammer. In [15], a jamming detection method for wireless networks was discovered by identifying the correlation between the signal strength variation, PDR (packet delivery ratio), and received signal pulse width. The study compared the profiles of normal communication and demonstrated the jamming effects on Wi-Fi networks experimentally.

Machine learning is a popular method to identify jamming attacks. In the research conducted in [16], the authors simulated different kinds of jamming attacks and differentiated them based on how they affected the wireless network. They also developed multiple machine learning algorithms to identify jamming techniques and compared the effectiveness of their proposed jamming detection schemes. They concluded that the random forest algorithm performed the best in identifying jamming in that study.

An intrusion detection system based on machine learning was introduced in [17] to identify not only jamming attacks but also Distributed Denial-of-Service (DDoS) attacks and other network intrusions. The authors in [18] designed an intelligent jammer using adversarial machine learning and proved their jamming method to be more effective than sensing-based or random jamming attacks. They also developed a jamming defense strategy using deep learning methods. A deep learning model was also used by [19] to detect and mitigate jamming. It was shown that a mobile device could obtain an optimal communication policy by using an RL-based frequency-space anti-jamming system. It manipulated and used the spread spectrum and user mobility to avoid strong interference and jamming. A random forest-based detection method for 802.11 MAC layer jamming was implemented in [20]. In [21], the researchers contributed to the literature with a jamming attack detection technique based on gambling games. A jamming detection model was built for time-critical wireless applications such as a smart grid. They used a gambling-based model and introduced a new metric named message invalidation ratio to compute the systems' performance during jamming attacks. The message invalidation ratio increased from 0 to 1 in case of jamming.

The above-mentioned solutions are effective and efficient but machine learning and deep learning-based approaches may require complex calculations depending on the model architecture and the resources of commercial IoT devices may be insufficient to deploy such algorithms. Alternatively, Liu et al. used packet loss rate and RSSI information [22] to develop a jamming detection application on an android smartphone. In the work presented in [23], jamming detection was built for wireless ad hoc networks. This technique utilized the correlation between the correct reception time of the packet and the error to detect the presence of a jammer in the surroundings. In [24] the researchers established the potential impact of employing mobility control to a primary network to enhance resilience to jamming. The study showed that any change in the network geometry attained through node and jammer mobility could significantly impact the jamming attack.

There are also several anti-jamming techniques proposed in the literature to avoid jamming attacks. In [25] the authors introduced an effective anti-jamming method based on feature extraction and deep reinforcement learning to enhance the performance of wireless communication. Anti-jamming techniques based on reconfigurable intelligent surface (RIS) is presented in [26] to improve the received signal power in a hybrid satellite-terrestrial relay network. A practical solution for secure and energy-efficient beamforming in multibeam satellite systems and avoiding jamming attacks based on the signal-to-leakage-plus-noise ratio (SLNR) metric is presented in [27], [28]. For secure communication in an IoT network using active and passive RISs to optimize power allocation and to solve a secrecy energy maximization problem is investigated in [29].

A. RESEARCH GAP

The research gap in the literature is highlighted in the following points.

1) Most of the jamming detection mechanisms presented in the literature are focused on detecting jamming attacks on a single communication channel.
2) Jamming detection methods generally use high computational resources or additional hardware to collect data and identify jamming. Such methods are not suitable and easily deployable in IoT devices given their implementation on a large scale, low cost, and limited computational and power resources.
3) It is also not feasible to implement complex machine learning and deep learning-based jamming detection methods on the end devices though they can be implemented in the cloud or at an edge server. The downside of remote deployment is that the jamming attacks usually interrupt the connection of the IoT devices hence the data from the device can't be transmitted for analysis, and the detected jamming alert may not be sent back to the IoT devices, making them still vulnerable to jamming attacks.

B. OUR CONTRIBUTIONS

The work presented in this article is novel as the jamming attacks can be detected on multiple channels in real time on the IoT device. It requires only an additional Wi-Fi USB adaptor to capture the network traffic in monitoring mode as the built-in network interface card of the IoT device will be used for communication purposes in the Wi-Fi network. The proposed jamming detection method can be integrated with IoT devices enabling them to be more smart and independent of each other. In this way, they will be able to detect jamming on their own and can be programmed to respond to such attacks accordingly to prevent harm to the system. The state-of-the-art jamming detection method was tested on a real IoT network in the lab to observe its effectiveness and is found to be suitable for application on low-cost IoT devices.

III. MODELING OF JAMMING DETECTION SYSTEM

This section explains the jamming detection criteria and the proposed jamming detection system model with pseudo code.

A. JAMMING DETECTION SYSTEM CRITERIA

The jamming attacks lead to abnormal behaviors by obstructing or blocking communication in the wireless network. The jammer's goal is to decrease the SNR of the system below the threshold value so that the wireless devices cannot communicate with each other. SNR can be computed as [43].

$$SNR(dB) = 20log(S/N) \tag{1}$$

Here, S is the power received by the Wi-Fi node and N is the noise level. If there is a jammer in the environment then SNR including the Jamming signal noise becomes as shown in Eq. (2).

$$JNR(dB) = 20log(S/(N + J_s)) \tag{2}$$

J_s is the power of the jammer. If J_s is high then the SNR decreases as SNR is inversely proportional to Jammer's power. From Shannon's Equation, the channel capacity is given as:

$$C = Blog_2(1 + S/N) \tag{3}$$

In Eq. 3, B is the bandwidth of the channel and S/N is the power ratio of the received signal and noise level. When a jammer is also present in the environment, the noise level increase as jammer's own noise is added in the noise that is already in the channel as shown in Eq.2 thus lowering the signal-to-noise ratio. Hence the capacity of the channel decreases.

Several network parameters such as network throughput, packet delivery ratio, increased collision rate, and the RSSI level along with the difficulty in medium access can be an indicator for intrusion/jamming attacks against the wireless network. The outliers in these parameter values can be evaluated with statistical methods and used to generate alerts. However, they can also occur in regular communication due to the nature of wireless networks. Hardware-specific problems, operating system, and application-related issues, congestion in the network, queuing, and many other factors in the environment may also cause such outliers which should be addressed correctly to separate them from the behavior of the device during jamming attacks. In this paper, an effective algorithm for jamming detection is implemented which analyzes these parameters to detect a jamming attack that causes disruption in the communication channel. A brief overview of the Key Performance Indicators (KPIs) used for jamming detection system modeling is given below.

1) Throughput

The throughput of a network is the number of successfully transmitted packets or bytes per unit of time. It is dependent on the available bandwidth, noise, delay, and hardware limitations of the medium. In practice, the actual throughput of whether the wired or the wireless network is consequently lower than the maximum theoretical throughput because it is adversely affected by latency, network congestion, packet loss errors, and protocol limitations.

2) Packet Delivery Ratio

Packet Delivery Ratio (PDR) is a key performance metric in network analysis that measures the success rate of packets being delivered across a network from a source to its destination. It is often expressed as a percentage and calculated as the ratio of the number of packets successfully received to the number of packets sent. A high PDR indicates a reliable network indicating that most packets sent have successfully reached their intended destination. Conversely, a low PDR indicates issues with the network, such as high congestion, poor link quality, or other types of interference.

3) End-to-End Delay

End-to-end delay is the total time it takes a packet to traverse a network from source to destination. This includes all delays in transmission, propagation, queuing, and processing, as well as any additional time spent waiting for system resources to become available or due to possible network congestion. So, while latency is the time it takes a packet to travel from one point to another, end-to-end delay is the total time it takes the packet to travel from the originating host to the destination host. This would include sum of all the latencies along the path, including delays at intermediate nodes or routers.

4) RSSI

The Received Signal Strength Indicator (RSSI) is the measurement of the strength of the received signal in dBm (decibels relative to milliwatt) from the sender. It's required to be high enough to establish a successful wireless connection.

B. PROPOSED JAMMING DETECTION SYSTEM MODEL

Wi-Fi, as defined by the IEEE 802.11 standard, is one of the most prevalent wireless communication protocols employed by IoT devices due to its ubiquitous availability, high data

transfer rates, and easy integration with existing network infrastructure [31]. Wi-Fi networks that operate in 2.4 GHz bandwidth have 14 channels [30]. As shown in Fig. 1, each channel is spaced 5 MHz apart from each other except for channel 14, and has a bandwidth of 20 MHz. Therefore, the 1st, 6th, and 11th channels are the most commonly used channels since they do not overlap and cause interference with each other.

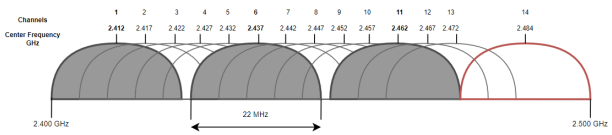

FIGURE 1: Channels in 2.4 GHz Wlan band

In the physical layer implementation, Wi-Fi communications employ Orthogonal Frequency Division Multiplexing (OFDM) and for the WLAN MAC layer, the signals are modulated using Quadrature Phase Shift Keying (QPSK) [32].

Due to the widespread use of Wi-Fi networks in today's world, the reliability of communication is of key importance. Radio jamming is one of the simplest ways to disrupt these networks. A data-driven jamming detection method is proposed in this paper utilizing parameters explained in Section 3.1. In this method, An ordinary profile of the IoT network is observed under normal conditions when there is no jamming attack and the device and server are communicating with each other. This ordinary profile reflects "the regular state" of the system and acts as a benchmark having no jamming attacks. The values of throughput, PDR, delay, and RSSI are sampled in the regular state profile and the statistical measures are obtained to generate the threshold values which would be used to threshold values for outlier and anomaly detection. This regular state is compared with the system's profile during the jamming experiments in real-time in which jamming signals are applied to the IoT device at certain time intervals.

The data collected from the regular state profile of the system were analyzed statistically and their distribution was examined. It is observed that the data do not comply with the normal distribution when the Shapiro-Wilk and Anderson-Darling tests are applied to the data set. The Chebyshev's Theorem, valid for all datasets, provides the necessary computation to identify outliers. Based on the Chebyshev's Theorem, at least 88.8% values would fall within 3-sigma values so the threshold values of the parameters were set using 3-sigma rule. Several experiments were conducted to verify the effectiveness of the threshold values as well calculated using (6) and (7).

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \mu)^2}{N}}. \tag{4}$$

Here N denotes the total number of observations in the dataset, μ is the mean of the dataset, and x_i represents the ith value of variable X. The threshold values for the observed data or measurements can be calculated by the arithmetic mean and standard deviation values using Eq. (5). Any x_i values exceeding these defined thresholds are considered outliers e.g. x_U and x_L are the outliers represented in (6) and (7).

$$threshold = \mu \pm 3\sigma \tag{5}$$

$$x_U > \mu + 3\sigma \tag{6}$$

$$x_L < \mu - 3\sigma \tag{7}$$

The hypothesis that if a jamming attack occurs, the throughput, PDR, delay and RSSI values will deviate from the regular state parameters has been confirmed in the experiments. Even if the jammer senses the effects of jamming in the IoT network and changes its setting, still the attack would deteriorate the performance of IoT system and the values of network delay, throughput, and RSSI would not lie in the normal range during the attack. The set threshold is sensitive to system changes and when the values would not lie in the threshold it would indicate that there is an anomaly in the network.

Eq.(8) is used to calculate the accuracy of the jamming detection system.

$$Accuracy = (TP + TN)/(TP + FP + TN + FN) \tag{8}$$

Here in (8) TP, FP, TN, and FN stand for True Positives, False Positive, True Negatives and False Negatives, respectively.

Jamming attacks are carried out in a controlled laboratory environment and the accuracy of the jamming detection system is measured in both real-time and offline data collected in the form of pcap and csv files during the experiments. Offline data was labeled and false alarms generated by the system were also calculated.

The pseudo-code of the basic jamming detection method is provided in Algorithm-1. The function explained in Algorithm-1 is called for every incoming packet captured by the Wi-Fi network interface. For every non-overlapped channel in 2.4 GHz bandwidth, the recorded and stored parameters threshold values are compared to the values of each incoming packet. If the incoming packets have continuously abnormal values of the recorded parameters for more than the specified timeout value on a specific channel, it is declared that a jamming attack is being done on that channel. The abnormalities are declared after a timeout to avoid the occasional outliers in the parameter values.

The function explained in Algorithm-1, the devised method captures the runs for every incoming packet captured by the Wi-Fi network interface. The measured values beyond the $\pm 3\sigma$ are considered as outliers. Outliers cannot be considered as jamming, but they might present patterns

Algorithm 1 Jamming Detection Mechanism

The threshold values of End-to-End delay, throughput, PDR and RSSI are represented as d_{th}, Th_{th},PDR_{th}, and $RSSI_{th}$, while the current values are represented by d, Th, PDR and $RSSI$. The current time is represented by t_c

Input: d,Th,$RSSI$,PDR
Output: $JammingAt[n]$
$t_0 = t_c$, $\Delta_t = 0$

procedure JAMMING DETECTION($Packet$)
  for `Channel=[1...n]` **do**
    for `each packet` **do**
      if $\Delta_t < timeout$ **then**
        if $d > d_{th}$ and $Th < Th_{th}$ and $PDR < PDR_{th}$ and $RSS1 >= RSSI_{th}$ **then**
          $\Delta_t = t_c - t_0$
        else
          $t_0 = t_c$
          $\Delta_t = 0$
          $JammingAt[Channel] = False$
        end if
      else
        $JammingAt[Channel] = True$
      end if
    end for
  end for
end procedure

similar to jamming. Therefore, the continuity of such outlier values should be monitored along with the change in other parameters, e.g. RSSI. In the algorithm, the throughput and PDR are calculated and updated every second. RSSI is calculated in dBm and provided by the WiFi card of the device whenever a new packet is captured. During a jamming attack, packets will be corrupted and will not be captured by the NIC. Since there is no packet received during jamming, it is not possible to calculate RSSI values for them. Therefore, a very low RSSI value is assigned for that time period since it is clear that the jammer doesn't have the incentive to increase the signal quality. On the other hand, corrupted packets cause a drop in the throughput and PDR of the device. Therefore, there is an upper bound for throughput and PDR which is only achievable in the regular state profile (when there is no jamming), and the throughput and PDR value will decrease during the jamming attack. Therefore, a lower threshold value of throughput and PDR is defined to detect outliers during the jamming attack. On the other hand, the network delay presents an opposite characteristic compared to the throughput. In the regular state profile, the inter-packet delay will be at the minimum, and its value will increase during the jamming due to the dropped/corrupted packets. Therefore, only an upper threshold value for the inter-packet delay is defined and used in Algorithm-1.

For all the channels in 2.4 GHz bandwidth, the incoming packets from each channel are captured in monitoring mode, and the recorded and stored parameters threshold values are compared to the values of each incoming packet. In Algorithm-1, t_c is the current time (the real point in time) and initially, t_0 which is a temporary value of time is set equal to t_c and Δ_t is zero. t_0 is updated whenever a new packet is captured. For each incoming packet, if the values are under the normal range, δ_t remains zero and t_0 is set to t_c. If the values are out of bound then Δ_t is set to the difference between the current time and t_0. When Δ_t becomes larger than the timeout value, it indicates an abnormality in the parameters. If this abnormality is continuous for more than the timeout, then it is stated as a jamming attack. If the values become normal before the timeout is reached, It is considered an outlier, and the values of t_0 and t_c are reset. The abnormalities are declared after a timeout to avoid the occasional outliers in the parameter values to minimize false alarms. The devised method can detect jamming in all the channels in the 2.4 GHz band simultaneously. The flowchart of the whole jamming detection process for a single channel is also presented in Fig. 2.

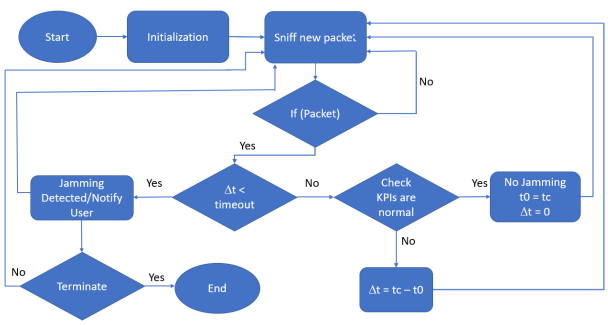

FIGURE 2: Flowchart of the Jamming Detection System

IV. METHODOLOGY AND EXPERIMENTAL SETUP

For the experiment, an IoT system is designed to create a Wireless Local Area Network (WLAN) operating in the 2.4 GHz frequency band. The designed system consists of a server, an IoT device (Raspberry Pi), and an access point. Both the client and the server are connected to the common access point and are communicating with each other over a specific channel (e.g. channel 11 at 2.462 GHz). The devised jamming detection mechanism can detect jamming in all the channels in 2.4 GHz bandwidth but for the ease of this study, the 3 most commonly used channels (channel 1, channel 6, and channel 11) are considered in the following experiments.

Software-Defined Radios (SDRs) are communication devices commonly used for receiving and/or transmitting radio signals. SDRs are also used previously [32], [33] for Wi-Fi jamming studies. A spot-jamming scenario is implemented using NI-USRP-2932 Software Defined Radio (SDR). Noise signals are generated with SDR by sending non-Wi-Fi packets with different text messages as the payload at 2.4 GHz bandwidth. SDR is tuned to concentrate all its power on a single frequency and send the packets in a continuous loop to

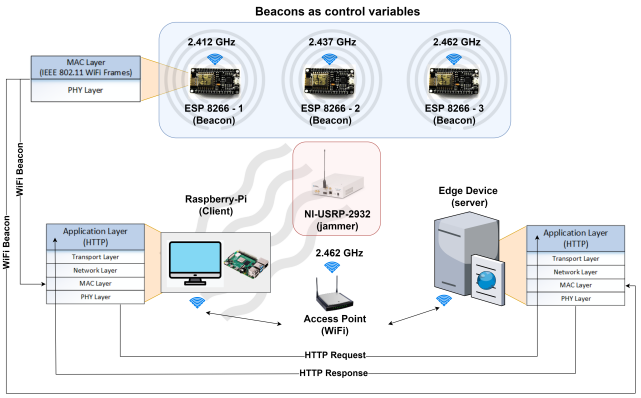

FIGURE 3: Experimental Setup

execute the jamming attack. The signal generated by the SDR disrupts wireless communication between the IoT device and the server.

Additionally, three NodeMCU ESP8266 devices are used in the experiment to generate IEEE 802.11 beacon frames in channels 1 (2.412 GHz), channel 2 (2.437 GHz) and channel 3 (2.462 GHz) respectively. Each device broadcasts one beacon frame of fixed size at regular intervals, e.g. every 10 ms. The reason for using the ESP8266 devices is to observe the effects of jamming on wireless communication where the network parameters are known and consistent. The experimental setup showing the client, jammer, server, access point (AP), and ESP8266 devices are presented in Fig. 3.

At first, the data in which the devices are communicating with each other under normal conditions for 300 seconds (5 minutes) is collected to grasp the regular state profile of the system. After that, the jamming attacks are applied on each channel. An omnidirectional antenna is used on the jammer which is placed next to the client to minimize the external factors affecting the signal. Different jamming attacks were performed on the system to check its behavior and the effectiveness of the jamming detection system. During the experiment different QoS parameters such as packet density, network throughput, PDR, network delay, device connectivity, and wireless link quality are collected and stored from both the IoT device under observation and the server. The data is collected and stored for both scenarios when the devices are operating under normal conditions and also during the jamming attack. On the client device, an additional USB Wi-Fi dongle is used to capture packets in the monitoring mode and the onboard WiFi interface is used in the managed mode to communicate with the server.

In the managed mode the wireless network interface controller (WNIC) can only capture the packets which have our client's MAC address as the destination MAC but the monitoring mode allows the WNIC to capture all the traffic on the wireless channel [34]. Frequency hopping is applied to the USB dongle every 100ms to capture the packets from the three channels (1, 6, and 11) in monitoring mode. The packets sent by the ESP 8266 devices are captured in the monitoring mode as a reference to calculate the network throughput, PDR, delay, and RSSI. The data is saved in *pcap* and *csv* files for further analysis and accuracy calculations. All the experiments are conducted multiple times and with different clients to verify the consistency and accuracy of the captured packets' parameters under the same conditions.

Two different sets of experiments were performed, details of which are given below.

A. EXPERIMENT-1

In this experiment, the jamming attack is applied on each channel one by one. The signal emitted by the SDR is configured to a bandwidth of 20 MHz and a gain of 30 dB. The jamming signal pattern is shown in Fig. 4.

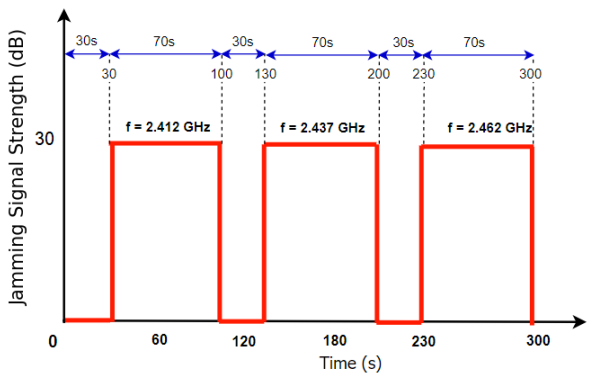

FIGURE 4: Jamming signal pattern for Experiment-1

At first, channel-1 is attacked, then channel-6, and at the end channel-11 was attacked. The total duration of the experiment was 300 seconds. Each channel was attacked for 70 seconds.

B. EXPERIMENT-2

The jamming signal pattern for these experiments is shown in Fig. 5

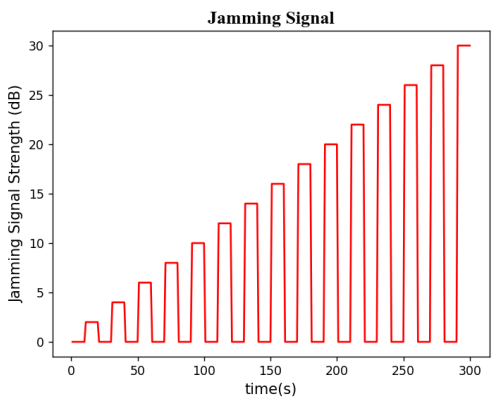

FIGURE 5: Jamming signal pattern for Experiment-2

In this set of experiments, the jamming signal with varying power levels was applied to all the channels one by one. The power of the jamming signal was varied from 2 dB to 30 dB with 20 MHz bandwidth and the effect of lower power on the system was observed. The jamming signal was applied such that for 10 seconds the jammer was off and for 10 seconds the jammer was on. For each cycle, the jammer power was increased by 2dB. In all the experiments, the jammer's distance from the client was kept constant.

V. EVALUATION OF THE RESULTS

A. THROUGHPUT IN THE MANAGED MODE

In this section, the throughput under normal conditions and during the jamming attack for both IoT device and the server is presented.

1) Results of Experiment-1

The throughput of the client and the server in the managed mode for the first set of experiments is plotted in Fig. 6.

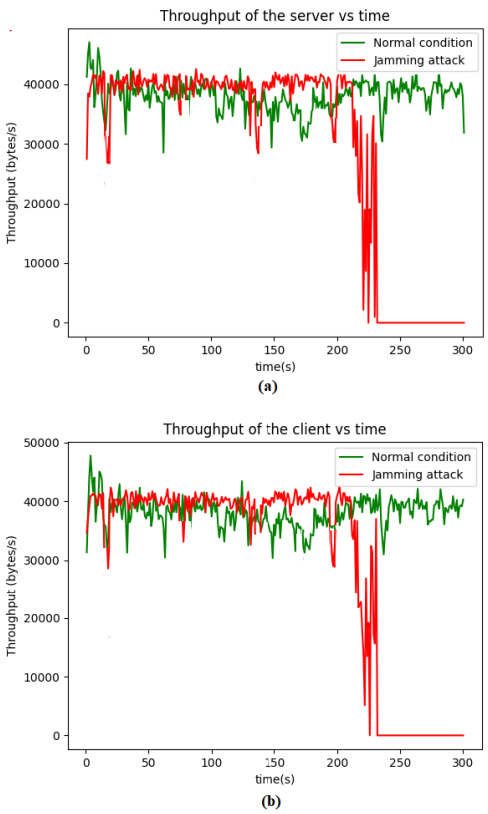

FIGURE 6: Results of Experiment 1: Throughput of channel-11 (in which the devices are operating) in managed mode **a**)Throughput of the server under normal conditions (green) and during the jamming attack (red) (**b**) Throughput of the client under normal conditions (green) and during, jamming attack (red)

Green indicates the throughput of both the server and client under normal conditions and red indicates the throughput when the jamming signal is being sent from the SDR. It is visible that as both devices are operating at f=2.462 GHz, the jammer only affects the throughput when it starts sending the signals on channel 11 at t=230 s. It is observed that the throughput of both the server and client decreased severely after the jamming attack started and eventually reached zero. It is also noticed that jamming attacks caused connectivity problems among the devices and as a result, the client disconnected from the access point during the jamming attack.

2) Results of Experiment-2

For the second experiment, As the channels are non-interfering when channel-1 and channel-6 were attacked with varying power levels of the jamming signal, no visible effect is noticed on the throughput and performance of the IoT system as it is operating in channel 11.

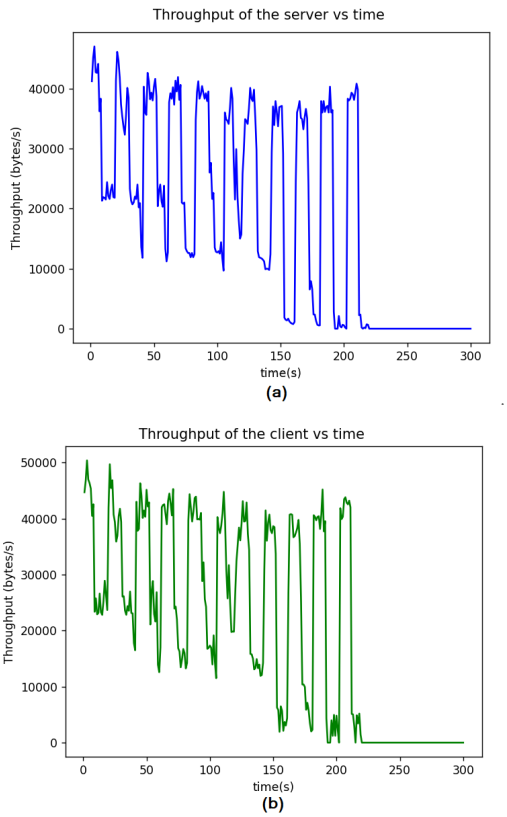

FIGURE 7: Results of Experiment-2: Throughput of channel-11 (in which the devices are operating) in managed mode **(a)**Throughput of the server **(b)** Throughput of the client

But when channel-11 is attacked, the jamming effects are observed. As can be seen from Fig. 7, As the power of the jamming signal increased the throughput declined more and more. When the jammer's power reached 22 dB, the client got disconnected from the access point due to severe jamming effects and the throughput reached zero. The jamming signal was applied after regular intervals, but the time in between attacks was so small that the client was unable to establish the

connection with the server again so the throughput remained zero until the end of the experiment.

B. PERFORMANCE RESULTS IN MONITORING MODE

1) Results of Experiment-1

The packets sent by the ESP8266 devices in three channels are captured for both normal communication and during the jamming attack in the monitoring mode using the Wi-Fi USB dongle. The network throughput and PDR is measured in each of the channels as shown in Fig. 8 It is visible from the figure that the throughput and PDR of the channels drop only when the jamming signal is sent on the respective frequency on which the channels are operating.

This is due to the fact that the bandwidth of the jamming signal is set to be 20 MHz and non-overlapping channels (1, 6, 11) are chosen in the experiments. The beacon packets sent by ESP 8266 devices are blocked during jamming resulting in a decrease in throughput and PDR. Fig.8 (a),(d), and (g) shows the throughput of the channels in normal conditions. Fig. 8 (b) and (c) show the throughput and PDR of channel 1 under the jamming attack. As the jamming signal of 2.412 GHz is sent from t=30 s to t=100 s, the throughput and PDR of channel 1 is decreased in that period. Similarly, a drop in the throughput and PDR of channel 6 is observed when the jamming signal is sent at the frequency 2.437 GHz between t=130 s to t=200 s (Fig. 8 (e, f)) and finally, Fig. 8(h, i) shows that the throughput and PDR of channel 11 declined between t=230s to t=300s.

Wi-Fi parameters experience large temporal variations which can also be observed in Fig. 8. The reason that the momentary dips in throughput like around t=100 in Fig. 8 (a,d,g) will not generate a false jamming detection alarm is that we have set a timeout value, and if the dip is smaller than the timeout value it will be considered an outlier but not a jamming attack. The timeout value has been selected after performing various experiments such that the system does not generate false alarms and also does not miss jamming attacks.

A similar pattern can be observed in Fig. 9 as well where the central frequency channel from which the beacon packets are captured is represented on the y-axis. Although it is seen as a straight line during normal conditions, it has discrete scattered points indicating the time that a packet is captured in a certain frequency. When there is jamming on a particular channel, there are blank spaces representing the time period when no packet is captured on that channel. As can be seen from Fig.9 (b) that the packets are missing/corrupted during the time when a jamming attack is applied on a channel(e.g. missing packets at f=2.412 GHz between 14:52 to 14:54).

Fig. 10 displays the inter-packet delay of the beacon packets sent by ESP8266 devices in the respective channels. During the normal condition as plotted in Fig. 10 (a), (c), and (e), the inter-arrival times of the incoming packets is usually around 10 ms but under the jamming attack, it increases due to the corruption of packets. Fig. 11 displays the RSSI values of the captured packet. During jamming attacks, some of the packets are not received by the device so the RSSI can't be calculated for those packets. This indicated that the packet is corrupted by the jammer and the client is unable to receive beacon packets from the NodeMCU devices when their respective channel is jammed. This can be seen from Fig. 11 where missing RSSI values are observed during jamming attacks. From t=30 to t=100, channel-1 was under attack. From t=130 to t=200, channel-6 was under attack and from t=230 to t=300, channel-11 was under attack. During these time periods, the channel which was under attack had missing RSSI values as shown in Fig. 11 (b,d, and e).

2) Results of Experiment-2

For the second set of experiments, The jamming attacks are applied separately with varying power levels on all three channels, and the beacon packets are captured. Similar effects are observed in all three channels so here in this article only the results of the jamming attacks from channel-11 are presented. The beacon packets are captured in the monitoring mode during the experiments and the throughput, PDR, inter-packet delay, and RSSI for this experiment are presented in Fig. 12. It is observed that though the jamming attack has low power still as the antenna was pointed toward the client so it had a severe effect on the system even when the jammer's power was very low. Fig. 12 (a) and (b) show how the throughput and PDR decline during the jamming attack. The normal interpacket delay of channel-11 is around 100ms as can be observed from Fig. 10 (e) but during these jamming experiments, it has increased upto 10 times as can be observed in Fig. 12(c). The effect on RSSI is not clearly visible from the figure as during the jamming attacks where power is less than 20dB, not enough packets are corrupted that can be observed in the figure but as the power increased from 20dB to 30dB missing RSSI values can be noticed in the Fig. 12 (d)

C. JAMMING DETECTION AND ACCURACY CALCULATIONS

Based on the results presented above performance results from the data collected in the managed mode as well as in the monitoring mode, it can be deduced that:

1) The throughput of a certain channel decreases significantly under the jamming attack as compared to the normal operation.
2) The jamming attack causes a substantial decrease in the channel's PDR compared to normal operation.
3) The received packet delay increases during jamming attacks due to corruption and re-transmission
4) The RSSI of the packet can't be calculated for the missing packets

Based on these observations, the proposed jamming detection mechanism can be applied to detect a jamming attack on any channel by the device itself and the necessary countermeasures can be implemented. For the experiment conducted above, the implemented jamming detection system can successfully detect jamming on all three channels as seen in Fig. 13.

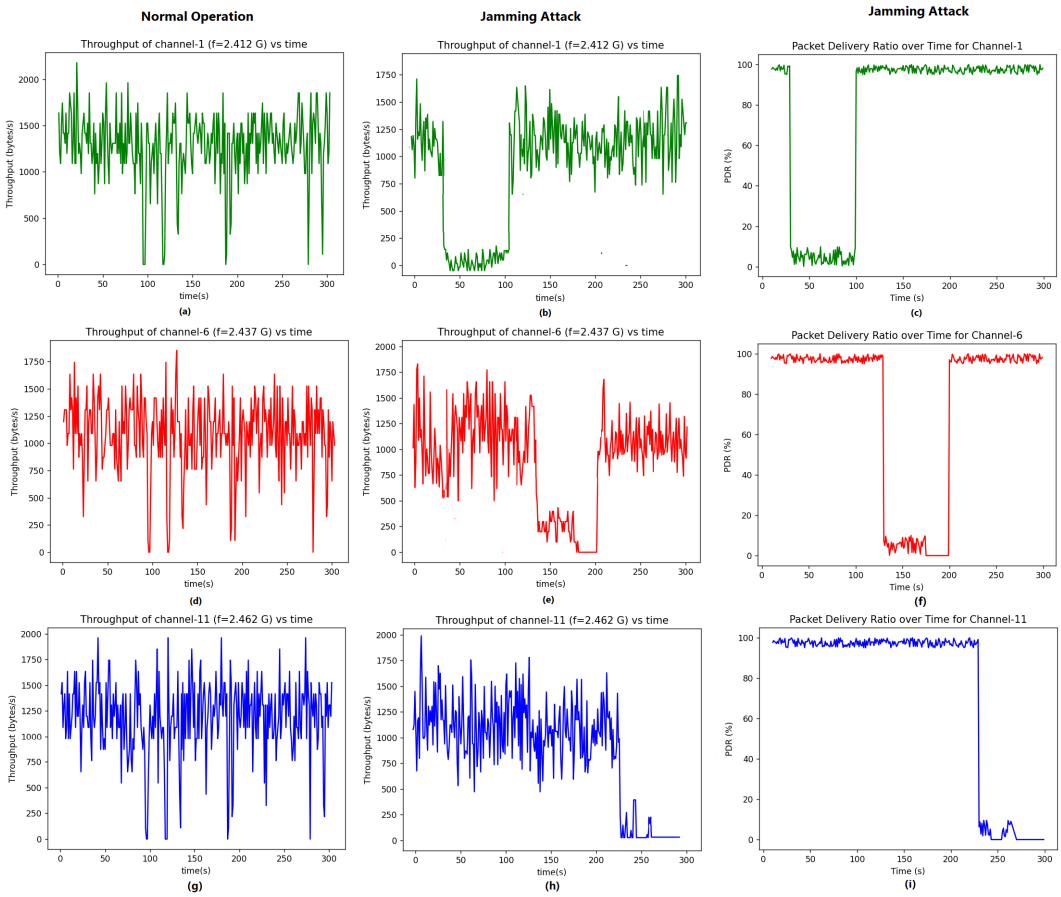

FIGURE 8: Throughput and PDR of channels in monitoring mode (**a**) Throughput of channel-1 under normal condition (**b**) Throughput of channel-1 under jamming attack (**c**) PDR of channel-1 under jamming attack (**d**) Throughput of channel 6 under normal condition (**e**) Throughput of channel 6 under jamming attack (**f**) PDR of the channel-6 under jamming attack (**g**) Throughput of channel 11 under normal condition (**h**) Throughput of channel 11 under jamming attack (**i**) PDR of channel-11 under jamming attack

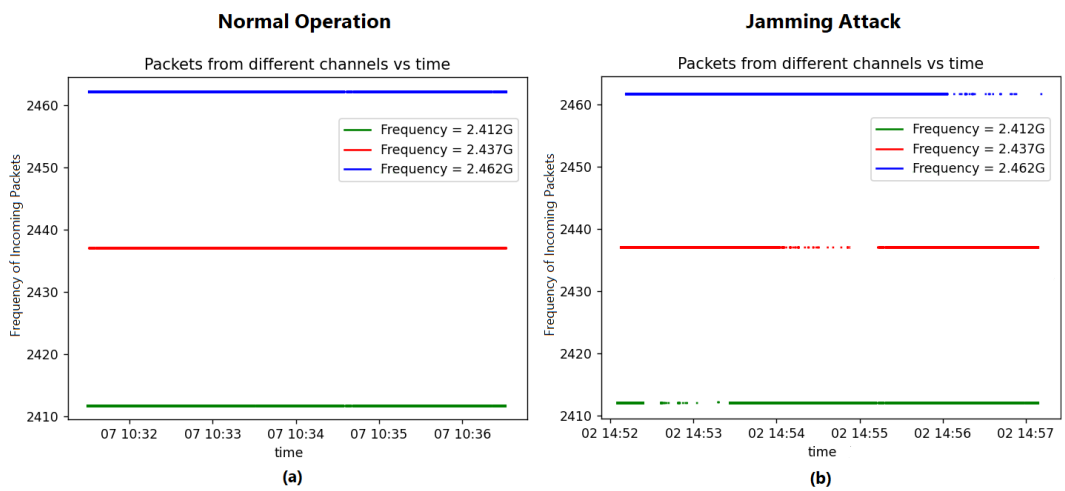

FIGURE 9: (**a**) Frequency of beacon packets captured from channels 1, 6, and 11 under normal conditions (**b**) Frequency of beacon packets captured from channels 1, 6, and 11 under jamming attack.

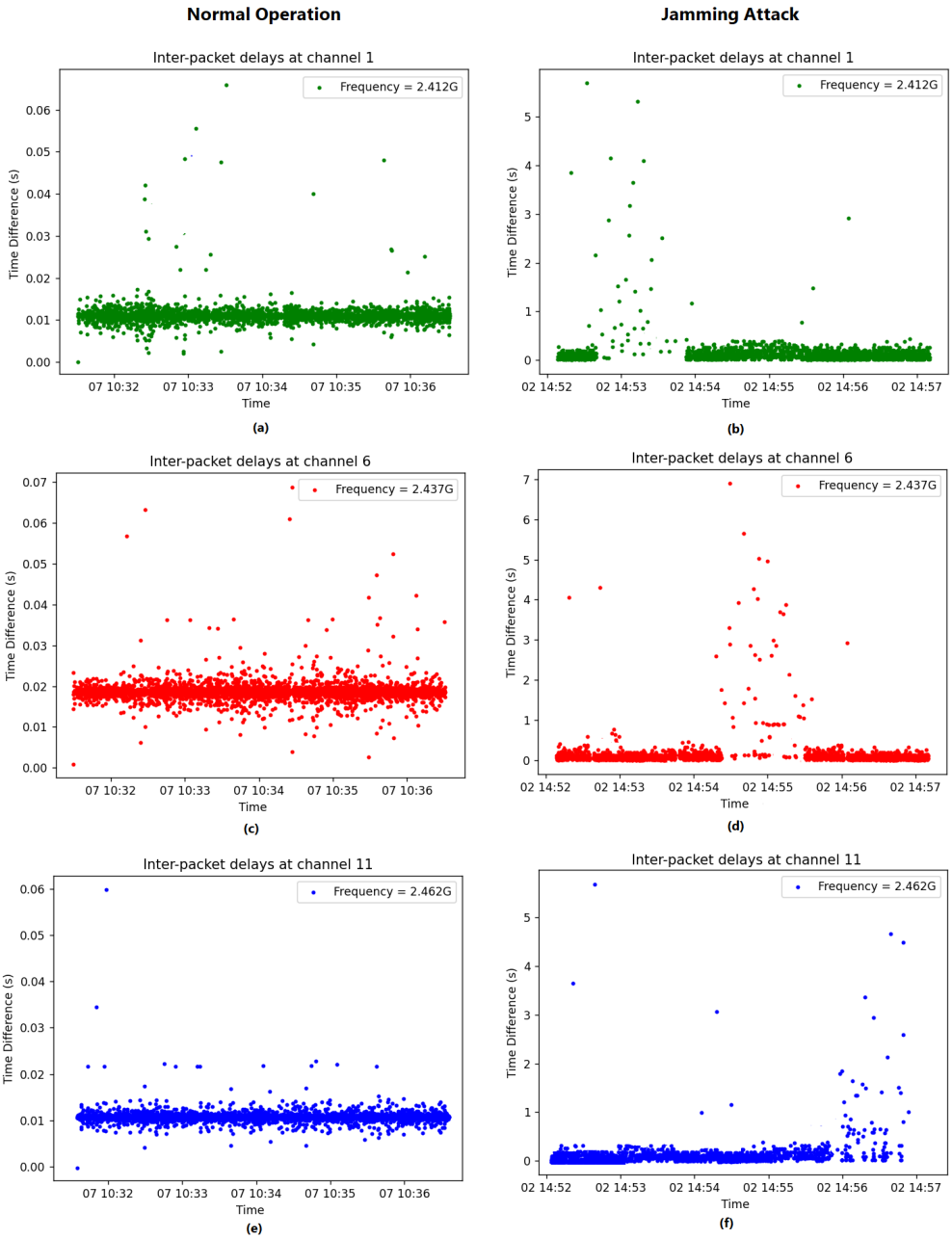

FIGURE 10: (**a**) Inter-packet delay of beacon packets in channel 1 under normal condition (**b**) Inter-packet delay of beacon packets in channel 1 under jamming attack (**c**) Inter-packet delay of beacon packets in channel 6 under normal (**d**) Inter-packet delay of beacon packets in channel 6 under jamming attack (**e**) Inter-packet delay of beacon packets in channel 11 under normal condition (**f**) Inter-packet delay of beacon packets in channel 11 under jamming attack.

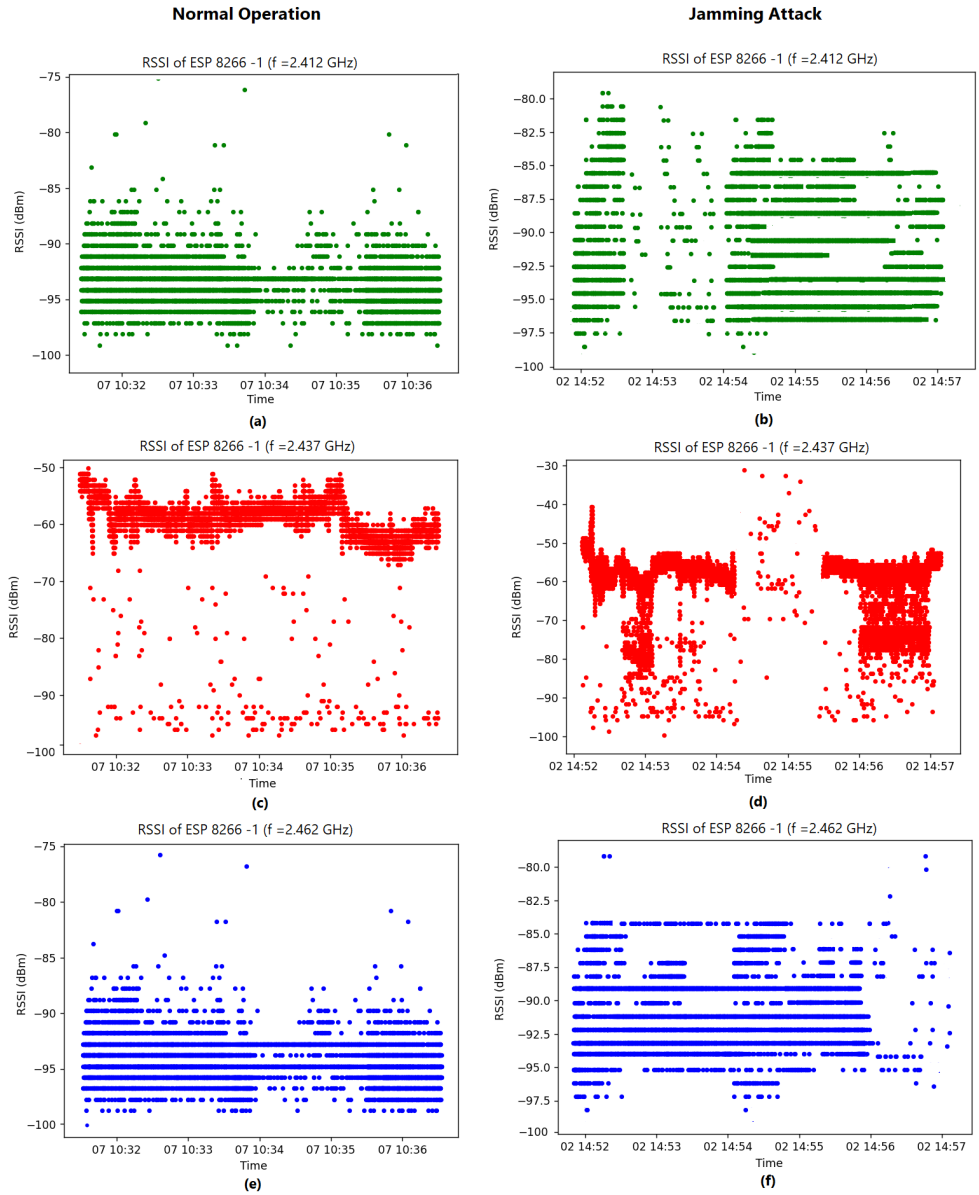

FIGURE 11: (**a**) RSSI of captured packets of ESP 8266- 1 (f=2.412 GHz) under normal condition (**b**) RSSI of captured packets of ESP 8266- 1 (f=2.412 GHz) under Jamming attack (**c**) RSSI of captured packets of ESP 8266- 2(f=2.437 GHz) under normal condition (**d**) RSSI of captured packets of ESP 8266- 2 (f=2.437 GHz) under Jamming attack (**e**) RSSI of captured packets of ESP 8266- 3 (f=2.462 GHz) under normal condition (**f**) RSSI of captured packets of ESP 8266- 3 (f=2.462 GHz) under Jamming attack

The accuracy is calculated both in real-time and using the offline data collected from the experiments. In real-time, various experiments are designed in order to further evaluate the performance of our jamming detection model with different jamming attacks incorporating random jamming signals as well. These tests aimed to assess the accuracy of the system under various jamming attack scenarios where the jamming signal was random in nature. As the random jamming signal exhibits unpredictable and random characteristics in its frequency, timing, or modulation. Unlike specific predefined patterns, a random jamming signal does not follow any predetermined structure or sequence, making it difficult to anticipate or counteract. To generate the random jamming signal gaussian noise was introduced in the system using SDR as shown in Fig. 14. During all the experiments the jammer's power and distance from the client were kept constant. All the data from experiments are collected and later analyzed to calculate accuracy offline as well. The accuracy was calculated using Eq. (8).

From the calculations and experiments, It is observed that the system provided zero false alarms (FP=0) in all of the experiments when the jamming signal was stronger than

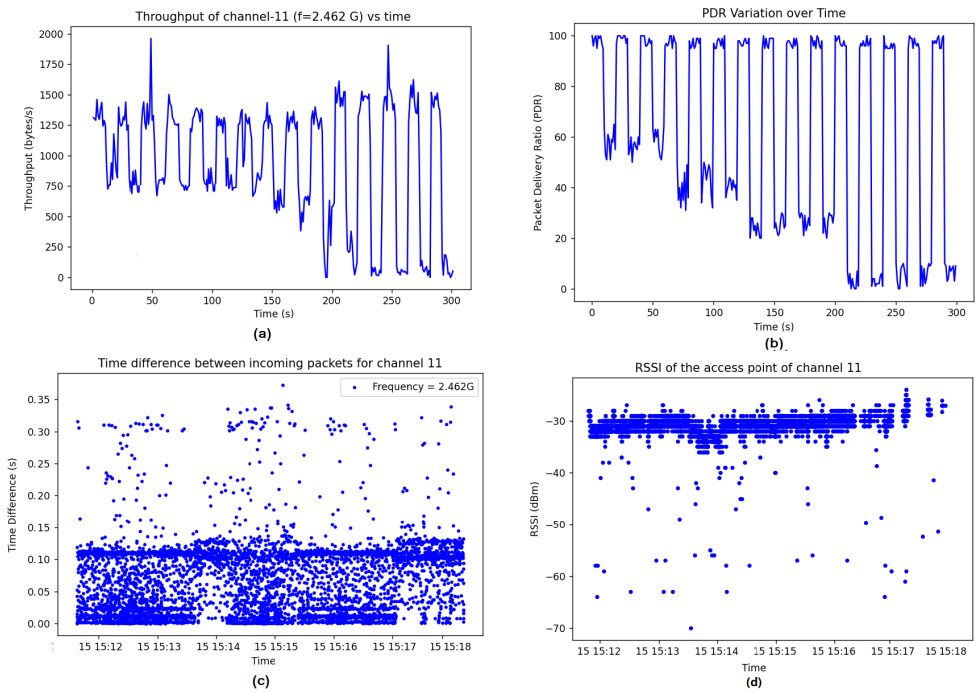

FIGURE 12: (**a**) Throughput of Channel-11 in monitoring mode (**b**) PDR of Channel -11 (**c**) Inter-packet delay of beacon packets in channel 11 (**d**) RSSI of captured packets of ESP 8266-3 (f=2.462 GHz)

10dB. It reflects that the system does not raise jamming attack alerts when there is no jamming attack but 1% of the time it was unable to detect the jamming (1% FN). This is due to the fact that a timeout value is used in Algorithm-1. When the values of the parameters under consideration exceed the threshold values and an abnormality is detected, the system doesn't immediately indicate the existence of a jamming attack, but it continues to measure and observe data to see whether these conditions persist until the timeout is reached. This timeout is set to remove the occasional outliers in the data. So if the jamming duration is extremely small to have a long-duration effect on the network (more than the timeout value), it would not recognize it as an attack but as an outlier.

Although the accuracy of the jamming detection system depends on the jamming signal strength, we observed high accuracy rates even at low-power jamming signals. When the jamming signal strength is greater than 12 dB the detection rate is around 99%. Lower power levels affect the accuracy slightly as presented in Fig. 15.

D. CPU USAGE, MEMORY CONSUMPTION, AND ENERGY EFFICIENCY

.

The Raspberry Pi 3 Model B was utilized to run the jamming detection application to test the performance of the application and its suitability. The application, when operational, exhibits a CPU usage of approximately 12.3% and uses a peak memory of 0.214 Mbytes.

The power consumption of the Raspberry Pi 3 Model B varies depending on the workload. It consumes around 1.2-1.4 W when idle and up to 3.7 W under maximum CPU load. We can approximate the power consumption at 12.3% CPU utilization using linear interpolation:

$$P = P_{\text{idle}} + (P_{\text{max}} - P_{\text{idle}}) \cdot \left(\frac{CPU}{100}\right) \tag{9}$$

Substituting the given values in (9):

$$P = 1.2 + (3.7 - 1.2) \cdot \left(\frac{12.3}{100}\right) = 1.5075 \;\; W \tag{10}$$

If the application runs for 1 hour, it would consume:

$$E = P \cdot T = 1.5075 \;\; Wh \tag{11}$$

The energy efficiency of the application can be defined as the amount of work performed per unit of energy consumed. The energy efficiency of such an application can't be easily calculated as it's not performing computations that can be easily measured. So, a custom measure of energy efficiency that is relevant to the application is defined. Energy efficiency is the number of distinct jamming incidents detected per unit of energy given that there is jamming attacks on the system. In this context, given that the application can detect a maximum of 120 distinct jamming attacks per hour, the energy efficiency of the application can be calculated as provided in (12):

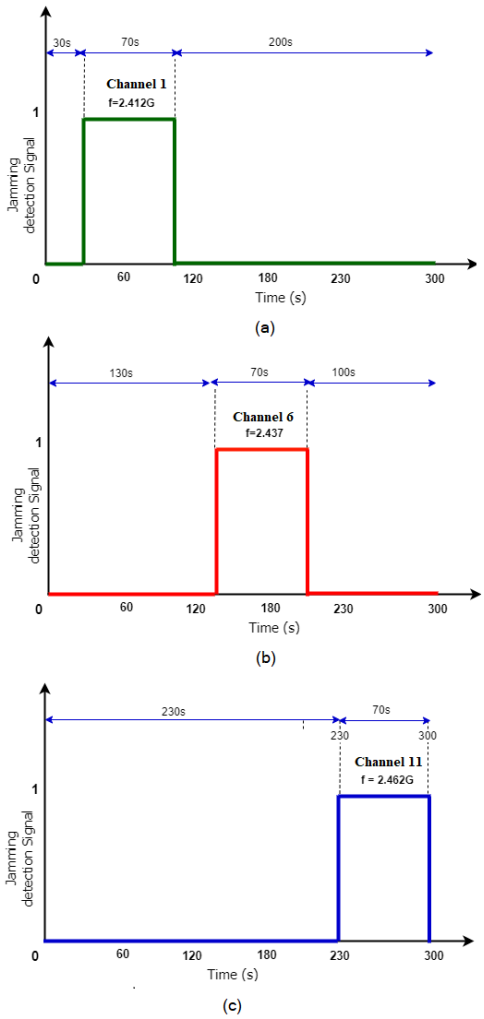

FIGURE 13: The output of jamming detection mechanism implementation (**a**) Jamming detection on channel-1 between 30s-100s (**b**) Jamming detection on channel-6 between 130s-200s (**c**) Jamming detection on channel-11 between 230s-300s.

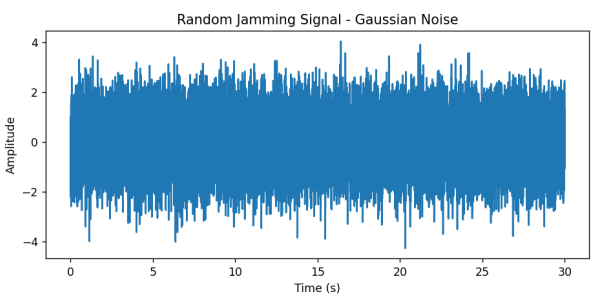

FIGURE 14: Random Jamming Signal

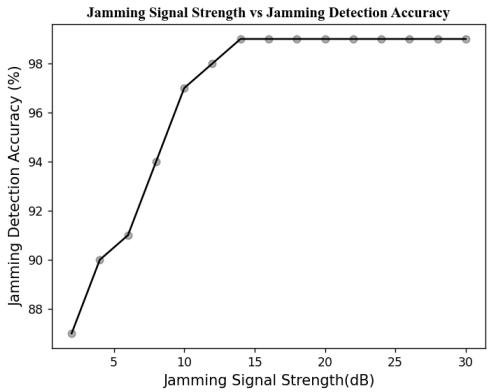

FIGURE 15: Jamming detection accuracy vs jamming signal strength

$$\text{EE} = \frac{N_{\text{attacks}}}{E} = \frac{120}{1.5075} = 79.6 \text{ attacks/Wh} \qquad (12)$$

Therefore, the energy efficiency of the jamming detection application is approximately 79.6 attacks detected per Wh of energy consumed.

VI. COMPARISON WITH STATE-OF-THE-ART JAMMING DETECTION SYSTEMS

A performance comparison of the existing state-of-the-art jamming detection systems is presented in Table 1. Some common detection matrices used in most of jamming detection systems are as follows:

- **PDR** - Packet Delivery Ratio: The ratio of the number of packets received to the number of packets sent.
- **PR** - Packet Rate: The rate at which packets are being sent or received in a network.
- **RSS** - Received Signal Strength: A measurement of the power present in a received radio signal.
- **RSSI** - Received Signal Strength Indicator: A measure used to approximate the received power of a signal in a wireless environment.
- **BPR** - Bad Packet Ratio: The ratio of the number of bad (corrupted) packets to the total number of packets received.
- **ECA** - Energy Consumption Amount: The total amount of energy consumed by a device, system, or process over a given period of time.
- **BER** - Bit Error Rate: The number of bit errors per unit time, or the ratio of the number of bit errors to the total number of bits sent.

In [35], the authors proposed a jamming detection method based on PDR, BPR and ECA of sensor nodes. They provided the simulation results where the detection rate varies between 97% to 100% based on the power and distance of jammer. The drawback of this approach is that the detection rate was slow and obviously the results in a real environment may vary significantly. In [36], the detection mechanism is fast because

TABLE 2: Comparison with State-of-the-Art Jamming Detection Systems

Jamming Detection Mechanism	**Dataset source**	**Detection Matrices**	**Application**	**Accuracy**
Query Based Jamming Detection Algorithm [35]	Simulations	PDR, BPR, ECA	WSN	97% - 99%
Fast Jamming Detection [36]	Experiments	PDR	WSN	97% - 100%
Consistency Checking Approach [37]	Experiments	PDR, RSSI	WSN	99%
Reactive Jamming Detection [38]	Experiments	RSSI, BER	WSN	90%
Artificial Bee Colony (ABC) [39]	Simulations	Energy, Packet Loss,RSS,PDR, Distance	WSN	High
ML Approach (RF, SVM,MLP) [40]	Simulations	PR,PDR,RSS	Wi-Fi Networks	97.5%
Edge AI-Based [41]	Experiments	RSS	Wi-Fi Networks	86.3%
ML Based (SVM, DT,RF) [42]	Simulations and Experiments	RSS	Wi-Fi IoT Networks	98% and 89.7%
Our Proposed Method	Experiments	Throughput, PDR, Latency and RSSI	Wi-Fi IoT Networks	99%

only one parameter, the PDR, is taken into account, however, it is not sufficient to declare jamming just on the basis of PDR alone as the PDR depends on many other factors such as network congestion and link quality. The authors in [37] took into account of RSSI with PDR, if PDR drops and RSSI is high, a jamming attack is declared whereas if both are low, it is considered a weak link or network congestion problem. The issue with this approach is that during jamming attacks, it gets difficult to get RSSI information from the sensor nodes preventing this method to be applicable in many scenarios. The jamming detection technique presented in [38] has a few hardware constraints and is better suited for radios which provide continuous RSS estimation and data demodulation. Several evolutionary algorithms are developed and tested to address jamming detection. In [39], the Artificial Bee Colony method is introduced to detect jamming. This algorithm takes a number of parameters and requires a lot of processing time and computational power which contradicts with efficiency for IoT devices. The machine learning and deep learning approaches presented in [40]–[42] require high memory requirements and processing power. It takes a huge amount of data to train an ML algorithm and as the network parameters can change due to the addition/removal of one or more IoT devices from the network, there is a need to apply incremental learning to the model frequently. These jamming detection methods can be integrated with the edge devices and cloud servers in the IoT network but are not feasible for the end devices.

There are a few key points that are very crucial when the current solutions are compared to the solution proposed in this article.

1) The jamming detection system provided in this paper can detect jamming on multiple channels simultaneously even on the channels the IoT network is not currently operating.
2) It is tested on a real testbed to measure its performance. Experimental results were found to be consistent with the expected results.
3) It is suitable for real-time applications.
4) It is a lightweight solution with no additional hardware requirements and can be integrated with the IoT devices easily. In this way, the end devices can detect jamming themselves and take countermeasures immediately based on their own inferences.

VII. DISCUSSION

The study presents the adverse effects of jamming attacks on Wi-Fi IoT networks. Multiple jamming attacks are demonstrated on the real wireless IoT network by utilizing a commercially available SDR. The jamming attacks lead to significant performance degradation of the wireless network. The beacon packets sent from ESP-8266 devices in different channels of 2.4 GHz bandwidth acts as the ground truth and are used to validate the increase in inter-arrival packet times and the packet drop rate. The communication packets collected from the client and the server are analyzed for real-time jamming detection. An additional Wi-Fi USB dongle is used to capture the network traffic in the monitoring mode and the NIC of the IoT device is used to collect data in the managed mode simultaneously. In the managed mode, the device connects to an access point and communicates with the server. The network interface card captures only packets

that have the device's destination MAC address; so if the link is down or there are fading and obstructions this will affect the throughput, PDR, network delay, and RSSI of the device.

When the device operates in monitoring mode, the device's interface captures all the packets within range, even if the destination MAC address does not belong to the device. Data collected in the managed mode is used to extract other connectivity issues (i.e. the link is down). The jamming detection system is mainly built on the data collected in the monitoring mode. Lack of connectivity, e.g. link unavailability or other issues does not affect data collected in monitoring mode. The Beacon packets from the ESP devices are used to set the ground truth values regarding network throughput, PDR, End-to-End network delay and RSSI. The channels under observation do not interfere with each other but there is interference from other channels which are overlapping them (e.g. Channel-1 is being overlapped by channel-2, channel-3, and channel 4). Also, there is interference due to other Wi-Fi devices in the environment, but the jamming signals and beacon packets generated in the experiments don't generate interference among channels.

It is also worth mentioning that if the jammer has high bandwidth then it will definitely affect the other consecutive channels as well. In our case, since the channels do not overlap and the jammer has a bandwidth of 20 MHz, the jammer does not affect other channels considered during the experiment. However, However, if a jamming attack is performed with a central frequency of 2.412 GHz and 20 MHz bandwidth and channel-1, channel-2 and channel-3 are examined. The jamming effect will be observed in all three channels, while the interference effects of jammer will not be significant on channel-2 and channel-3 as compared to channel-1. If the effects of the jamming attack are so significant that the values of the considered parameters exceed the threshold values in all three channels, jamming will also be reported in all of these channels. Similarly, if the jammer's bandwidth is increased to 30 MHz and channel-1 is attacked, the effect is also noticed in channel 6, but less severe than on channel-1. If the effects are minimal, the jamming detection system may treat it as interference from other channels and will not alarm in such cases.

While this study primarily focuses on evaluating the performance of our proposed jamming detection mechanism in a lab environment, noise and interference are accounted in the experiments to mimic real-world IoT systems. The lab. also has background noise sources from other operating systems, mobile phones, and other electronic devices. All the experiments are done in noisy environment conditions and multiple repetitions of the experiments were done to validate the consistency of the results. Interference from other wireless devices operating in the same 2.4 GHz frequency band, such as Zigbee and Bluetooth, is also accounted for as the experiments and evaluations considered the presence of interference from other devices by conducting tests in the presence of background noise. The proposed method maintained a high level of accuracy (99%) even in the presence of noise and interference. Further validation and performance evaluations in real-world IoT environments are crucial to fully establish its efficiency.

Link quality metrics in a wireless network show a lot of variation and outliers. Instantaneous changes in these parameters can cause false alarms. Therefore, it is essential to define a timeout value to identify jamming from instantaneous outliers as implemented in Algorithm-1. This timeout value is carefully determined after various experiments so that the system does not miss any jamming attacks. Moreover, if there is a change in the wireless environment (No. of IoT devices increases, the access point position is changed, etc.), variability of this parameter will change the threshold values In such cases, there would be a need to update the threshold values.

The main objective of this study is to design a cost-effective jamming detection solution that can be easily integrated with IoT devices, considering their limited processing capabilities. To ensure efficiency and resource optimization, various KPIs have been investigated and the most appropriate ones are carefully selected after thorough consideration. RSSI alone may not provide a complete picture of jamming detection and can be misleading as an attacker may create a noisy environment that results in a high RSSI but low Signal-to-Noise Ratio (SNR) and Signal-to-Interference plus-Noise Ratio (SINR). In this research, RSSI is selected as one of the indicators along with throughput, PDR and packet delay due to its simplicity and minimal computational requirements. RSSI is not measured directly from the devices themselves. Instead, the packets are captured in monitoring mode to obtain the RSSI values. This approach allows us to observe the impact of jamming on the received packets and identify potential indicators of jamming attacks. During the jamming attack, the decline in throughput, PDR, and the inability to receive packets properly or with high delay along with the missing RSSI values associated with these dropped packets serve as indicators of jamming activity.

The jamming detection technique provided in this article is better suited for critical applications where real-time jamming detection is required and faster countermeasure is needed by the end devices to avoid any damage to the system.

VIII. CONCLUSION

This paper demonstrates the effects of jamming on 802.11 Wi-Fi networks and presented a real-time jamming detection mechanism that detects jamming on multiple channels simultaneously. For this purpose, a real test bed is set up and jamming attacks are performed with a commercially available SDR on specific 2.4 GHz Wi-Fi channels. The spot jamming scenario implemented in this study caused connectivity problems and performance degradation of the network leading to poor quality of service. To set a base ground for throughput, PDR, RSSI, and End-to-End network delay, additional beacon packets with a fixed size are sent using ESP 8266 devices in three non-overlapped channels. Jamming attacks also affect these packets adversely, corrupting the

packets and increasing the inter-arrival times. The effect of jammer with varying power levels is also investigated. Based on these observations, a jamming detection technique is developed for IoT devices which enabled the detection of intelligent jammers that can change their frequency and can shift channels. The developed application requires minimal computational and space requirements which makes it feasible for commonly available IoT devices.

The investigations on the network parameters are ongoing to exploit the behavior of IoT systems under the influence of different types of jamming attacks. Subsequent studies and field trials to assess the practicality and adaptability of this solution will be conducted in future studies. Although further research and real-world deployment evaluations are needed, this study provides a foundation for an effective jamming detection application in real-time scenarios. The creation of a jamming attack dataset and the development of machine learning approaches with appropriate feature selection for jamming detection and identification solutions model construction will also be the main focus of future works.

REFERENCES

[1] D. Singh, G. Tripathi, and A. J. Jara, "A survey of internet-of-things: Future vision, architecture, challenges and services," in WF-IoT, 2014.

[2] Z. Zhou et al., "Pervasive LPWAN Connectivity Through LEO Satellites: Trading Off Reliability, Throughput, Latency, and Energy Efficiency". Springer, 2023

[3] R. Burczyk, A. Czapiewska, M. Gajewska, and S. Gajewski, "LTE and NB-IoT Performance Estimation Based on Indicators Measured by the Radio Module," Electronics, vol. 11, no. 18, p. 2892, Sep. 2022, doi: 10.3390/electronics11182892.

[4] M. Centenaro, L. Vangelista, A. Zanella and M. Zorzi, "Long-range communications in unlicensed bands: the rising stars in the IoT and smart city scenarios," in IEEE Wireless Communications, vol. 23, no. 5, pp. 60-67, October 2016, doi: 10.1109/MWC.2016.7721743.

[5] G. Kambourakis, C. Kolias, D. Geneiatakis, G. Karopoulos, G. M. Makrakis, and I. Kounelis, "A State-of-the-Art Review on the Security of Mainstream IoT Wireless PAN Protocol Stacks," Symmetry, vol. 12, no. 4, p. 579, Apr. 2020, doi: 10.3390/sym12040579.

[6] J. J. Nielsen, I. Leyva-Mayorga and P. Popovski, "Reliability and Error Burst Length Analysis of Wireless Multi-Connectivity," 2019 16th International Symposium on Wireless Communication Systems (ISWCS), Oulu, Finland, 2019, pp. 107-111, doi: 10.1109/ISWCS.2019.8877248.

[7] A. Mukherjee, "Physical-layer security in the internet of things: Sensing and communication confidentiality under resource constraints," Proceedings of the IEEE, vol. 103, pp. 1747-1761, 2015.

[8] W. Xu, K. Ma, W. Trappe, and Y. Zhang, "Jamming sensor networks: Attack and defense strategies," IEEE Network, vol. 20, pp. 41-47, 2006.

[9] S. D. Babar, N. R. Prasad, and R. Prasad, "Jamming attack: Behavioral Modelling and analysis," in Wireless VITAE, 2013.

[10] A. Benslimane, A. El yakoubi, and M. Bouhorma, "Analysis of jamming effects on IEEE 802.11 wireless networks," in IEEE ICC, 2011.

[11] S. Amuru, H. S. Dhillon, and R. M. Buehrer, "On Jamming Against Wireless Networks," IEEE Trans. Wirel. Commun, vol. 16, pp. 412-428, 2017.

[12] J. Bellardo and S. Savage, "802.11 denial-of-service attacks: Real vulnerabilities and practical solutions," in USENIX Security, 2003, pp. 15-28.

[13] D. J. Thuente, B. Newlin, and M. Acharya, "Jamming Vulnerabilities of IEEE 802.11e," in MILCOM, 2007, pp. 1-7.

[14] E. Bayraktaroglu, C. King, X. Liu, G. Noubir, R. Rajaraman, and B. Thapa, "On the Performance of IEEE 802.11 under Jamming," in IEEE INFOCOM, 2008.

[15] N. Sufyan, N. Saqib, and M. Zia, "Detection of jamming attacks in 802.11 b wireless networks," EURASIP J Wirel Commun Netw, vol. 2013, pp. 1-18, 2013.

[16] Z. Feng and C. Hua, "Machine Learning-based RF Jamming Detection in Wireless Networks," in Proc. ICC SSIC, 2018, pp. 1-6.

[17] Z. Yu and J. J. P. Tsai, "A framework of machine learning-based intrusion detection for wireless sensor networks," in Proc. IEEE SUTC, 2008, pp. 272-279.

[18] T. Erpek, Y. E. Sagduyu, Y. Shi, "Deep learning for launching and mitigating wireless jamming attacks," IEEE TCCN, vol. 5, pp. 2-14, 2019.

[19] L. Xiao, D. Jiang, D. Xu, H. Zhu, Y. Zhang, and V. H. Poor, "Two-dimensional antijamming mobile communication based on reinforcement learning," IEEE Trans. Veh. Technol., vol. 67, pp. 9499-9512, 2018.

[20] O. Pual, I. Akta, C. J. Schnelke, G. Abidin, K. Wehrle, and J. Gross, "Machine learning-based jamming detection for IEEE 802.11: Design and experimental evaluation," in Proc. IEEE WoWMoM, 2014, pp. 1-10.

[21] Z. Lu, W. Wang, and C. Wang, "From jammer to gambler: Modeling and detection of jamming attacks against time-critical traffic," in Proc. IEEE INFOCOM, 2011, pp. 1871-1879.

[22] G. Liu, J. Liu, Y. Li, L. Xiao, and Y. Tang, "Jamming Detection of Smartphones for WiFi Signals," in Proc. VTC Spring, 2015, pp. 1-3.

[23] A. Hamieh and J. Ben-Othman, "Detection of jamming attacks in wireless ad hoc networks using error distribution," in Proc. IEEE Int'l Conf. Commun, 2009, pp. 1-6.

[24] P. Tague, "Improving anti-jamming capability and increasing jamming impact with mobility control," in Proc. IEEE MASS, 2010, pp. 501-506.

[25] X. Chang, Y. Li, Y. Zhao, Y. Du, and D. Liu, "An improved anti-jamming method based on Deep Reinforcement Learning and feature engineering," IEEE Access, vol. 10, pp. 69992-70000, 2022.

[26] Z. Lin, H. Niu, K. An, Y. Wang, G. Zheng, S. Chatzinotas, and Y. Hu, "Refracting ris-aided hybrid satellite-terrestrial relay networks: Joint Beamforming Design and optimization," IEEE Transactions on Aerospace and Electronic Systems, vol. 58, pp. 3717-3724, 2022.

[27] Z. Lin, K. An, H. Niu, Y. Hu, S. Chatzinotas, G. Zheng, and J. Wang, "SLNR-based secure energy efficient beamforming in Multibeam Satellite Systems," IEEE Transactions on Aerospace and Electronic Systems, pp. 1-4, 2022.

[28] Z. Lin, M. Lin, J.-B. Wang, T. de Cola, and J. Wang, "Joint beamforming and power allocation for satellite-terrestrial integrated networks with Non-Orthogonal Multiple Access," IEEE Journal of Selected Topics in Signal Processing, vol. 13, pp. 657-670, 2019.

[29] H. Niu, Z. Lin, Z. Chu, Z. Zhu, P. Xiao, H. X. Nguyen, I. Lee, and N. Al-Dhahir, "Joint Beamforming Design for Secure Ris-assisted IOT Networks," IEEE Internet of Things Journal, vol. 10, pp. 1628-1641, 2023.

[30] ISO/IEC/IEEE, "Information Technology–Telecommunications and information exchange between systems–local and metropolitan area networks–specific requirements part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications amendment 1: Prioritization of Management Frames (adoption of IEEE Std 802-11AE-2012)," IEEE Std. 802.11ae-2012, pp. 1-61, 2012.

[31] K. Pahlavan and P. Krishnamurthy, "Evolution and Impact of Wi-Fi Technology and Applications: A Historical Perspective," Int. J. Wireless Inf. Networks, vol. 28, no. 1, pp. 3-19, Mar. 2021

[32] J. Duarte García, "Software Defined Radio for Wi-Fi Jamming," 10.13140/RG.2.2.23772.90240, 2020.

[33] A. Sârbu and D. Neagoie, "Wi-Fi Jamming Using Software Defined Radio," in International Conference on Knowledge Based Organization, vol. 26, pp. 162-166, 2020.

[34] P. Goyal and A. Goyal, "Comparative study of two most popular packet sniffing tools-Tcpdump and Wireshark," in International Conference on Computational Intelligence and Communication Networks (CICN), pp. 77-81, 2017.

[35] M. Çakiroglu and A. T. Özcerit, "Jamming detection mechanisms for wireless sensor networks," in Proceedings of the Third International ICST Conference on Scalable Information Systems, 2008.

[36] K. Siddhabathula, Q. Dong, D. Liu, and M. Wright, "Fast jamming detection in sensor networks," in 2012 IEEE International Conference on Communications (ICC), 2012.

[37] Y. Chen, W. Xu, W. Trappe, and Y. Zhang, "Detecting jamming attacks and radio interference," in Securing Emerging Wireless Systems, 2008, pp. 1-18.

[38] M. Strasser, B. Danev, and S. Čapkun, "Detection of reactive jamming in Sensor Networks," ACM Transactions on Sensor Networks, vol. 7, pp. 1-29, 2010.

[39] E. Sasikala and N. Rengarajan, "An intelligent technique to detect jamming attack in wireless sensor networks (wsns)," International Journal of Fuzzy Systems, vol. 17, pp. 76-83, 2015.

[40] Y. Arjoune, F. Salahdine, M. S. Islam, E. Ghribi, and N. Kaabouch, "A novel jamming attacks detection approach based on machine learning for wireless communication," in 2020 International Conference on Information Networking (ICOIN), 2020.
[41] A. Hussain, N. Abughanam, J. Qadir, and A. Mohamed, "Jamming detection in IOT wireless networks: An edge-AI based approach," in Proceedings of the 12th International Conference on the Internet of Things, 2022.
[42] B. Upadhyaya, S. Sun, and B. Sikdar, "Machine learning-based jamming detection in wireless IOT Networks," in 2019 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS), 2019.
[43] W. Aldosari and M. Zohdy, "Tracking a jammer in wireless sensor networks and selecting boundary nodes by estimating signal-to-noise ratios and using an extended Kalman filter," Journal of Sensor and Actuator Networks, vol. 7, no. 3, p. 48, 2018.

FATIMA TU ZAHRA received her B.S. degree in Electrical Engineering from National University of Sciences and Technology (NUST), Islamabad, Pakistan in 2015 and M.S. degree in Electrical and Electronics Engineering from Bilkent University, Ankara, Turkey in 2018. From 2015 to 2017, she was a Research Assistant at the National Magnetic Resonance Research Centre (UMRAM) Turkey. After that, between 2018-2020, she joined Cyber Reconnaissance and combat Lab., Bahria University, Islamabad as a researcher and visiting faculty member. Currently, she is working as a researcher at VeNIT Lab., Marmara University, Istanbul since 2021. Her research interests include embedded systems, wireless networks, and IoT.

YAVUZ SELIM BOSTANCI is a researcher working at VeNIT Lab in the Dept. of Computer Engineering, Faculty of Eng., Marmara University, Istanbul. His research interests and engineering expertise are in wireless networks and IoT. He works on analyzing the link quality, measuring the performance of wireless links, and developing tools to collect network performance parameters. He participates in national and international research projects.

MUJDAT SOYTURK received his Ph.D. and M.Sc. degrees in Computer Engineering in 2007 and 2002 respectively from Istanbul Technical University. He is an Associated Professor at the Department of Computer Engineering, Faculty of Engineering, Marmara University since 2012. His research interest and projects focus on V2X Communications, Connected Cars, Intelligent Transportation Systems, and IoT. In addition, statistical analysis and performance evaluation of the various network types are additional skills. He participated in several EU framework research projects. He is directing Vehicular Networks and Intelligent Transportation Systems (VeNIT) Research Lab which he founded in 2013.