Ad Hoc Networks 120 (2021) 102562

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Developing novel low complexity models using received in-phase and quadrature-phase samples for interference detection and classification in Wireless Sensor Network and GPS edge devices

George D. O'Mahony [a,*], Kevin G. McCarthy [a], Philip J. Harris [b], Colin C. Murphy [a]

[a] Department of Electrical and Electronic Engineering, University College Cork, Ireland
[b] Raytheon Technologies Research Center, Cork, Ireland

ARTICLE INFO

Keywords:
Detection
Edge devices
Interference
Jamming
Machine learning
Security
WSNs and ZigBee

ABSTRACT

Despite Wireless Sensor Networks (WSNs) significantly developing over the past decade, these networks, like most wireless networks, remain susceptible to malicious interference and spectrum coexistence. Other vulnerabilities arise as WSN applications adopt open standards and typically resource and energy-constrained commercial-off-the-shelf equipment. Deployments include safety-critical applications such as the internet of things, medical, aerospace and space and deep-sea exploration. To manage safety and privacy requirements across such a diverse wireless landscape, security on wireless edge devices needs improvement while maintaining low complexity. This paper improves wireless edge device security by developing a novel intelligent interference diagnostic framework. Received in-phase (I) and quadrature-phase (Q) samples are exclusively utilized to detect modern, subtle and traditional crude jamming attacks. This I/Q sample utilization inherently enables decentralized decision-making, where the low-order features were extracted in a previous study focused on classifying typical 2.4–2.5 GHz wireless signals. The associated optimal intelligent models are leveraged as the foundation for this paper's work. Initially, Matlab Monte Carlo simulations investigate the ideal case, which incorporates no hardware limitations, identifies the required data type of signal interactions and motivates a hardware investigation. Software-defined radios (SDRs) collect the required live over-the-air I/Q data and transmit matched signal (ZigBee) and continuous-wave interference in developed ZigBee wireless testbeds. Low complexity supervised machine learning models are developed based exclusively on the low-order features and achieve an average accuracy among the developed models above 98%. The designed methodology involves examining ZigBee over-the-air data for artificial jamming and SDR jamming of ZigBee signals transmitted from SDR and commercial (XBee) sources. This approach expands to a legitimate node classification technique and an overall algorithm for wireless edge device interference diagnostic tools. The investigation includes developing Support Vector Machine, XGBoost and Deep Neural Network (DNN) models, where XGBoost is optimal. Adapting the optimized models to global positioning system signals establishes the transferability of the designed methodology. Implementing the designed approaches on a Raspberry Pi embedded device examines a relatively resource-constrained deployment. The primary contribution is the real experimentally validated interference diagnostic framework that enables independent device operation, as no channel assumptions, network-level information or spectral images are required. Developed models exclusively use I/Q data low-order features and achieve high accuracy and generalization to unseen data.

1. Introduction

Successful wireless sensor network (WSN) deployments across a diverse range of safety-critical applications have developed WSNs into essential telecommunication infrastructure components. Both civil, for example, the Internet of Things (IoT) [1] and medical, for example, remote patient monitoring [2], deployments utilize WSNs as visualized in Figs. 1(a) and 1(b), respectively. Adopting WSNs permits easier design, installation and maintenance, while simultaneously providing new deployment options and cost benefits. However, as WSNs become integrated with critical use-cases, the incentive to attack/disrupt these networks intensifies. Other incentivizing deployments include

* Corresponding author.
E-mail addresses: george.omahony@umail.ucc.ie (G.D. O'Mahony), k.mccarthy@ucc.ie (K.G. McCarthy), harrispj@rtx.com (P.J. Harris), colinmurphy@ucc.ie (C.C. Murphy).

https://doi.org/10.1016/j.adhoc.2021.102562
Received 4 January 2021; Received in revised form 9 April 2021; Accepted 26 May 2021
Available online 2 June 2021

control systems for smart homes [3], space-based WSNs [4], missile defense [5], wireless body area networks (WBANs) [6], aerospace's fly-by-wireless, using Low Earth Orbit satellites as components [7] and typical surveillance and industrial sensing. These machine-to-machine and machine-to-people communications generally transmit private time-dependent data, resulting in critical privacy and reliability requirements. These requirements inherently create new security, spectral coexistence and threat identification challenges. The unprecedented growth rate of wireless devices (29 billion devices are forecast by 2022 [8]) magnifies these challenges. Typically, WSNs are long-lived deployments consisting of low-cost, compact resource-constrained devices that are coupled to their operating environment and do not have the capability to implement complex or computationally intensive security protocols. Securing WSNs and associated edge devices is essential as an attack on a WSN enabled application could have significant privacy and safety consequences, particularly in real-time sensitive medical and control systems.

This paper's central research question surrounds how to improve security on resource-constrained wireless edge devices by only using data consistently available to a functioning receiver. The aim is to extract high-level interference information from low-level in-phase (I) and quadrature-phase (Q) received samples. Despite technological advancements, wireless networks remain vulnerable to radio jamming attacks, due to the open nature of wireless channels and the lack of practical physical-layer wireless technologies that can efficiently decode data packets in the presence of jamming attacks. This paper focuses on improving security through interference detection diagnostic tools in wireless sensor networks and global positioning system (GPS) signals. Stealthy attackers transmit short jamming signals to become less detectable with less energy and yet powerful enough to ruin entire packet transmission [9]. As a result, both crude traditional and modern stealthy attacks are examined. Jamming attacks (denial, deception and/or destruction) have traditionally been the domain of Electronic Warfare [10]. However, these techniques are gradually being adopted for criminal activities as readily available hardware supports the development of effective systems that can circumvent jamming prevention techniques. WSN compromise, whether malicious or unintentional, is achievable and threat detection and analysis need to match advancing attack strategies [10], while not overly consuming device resources.

This study uses the IEEE 802.15.4 derived ZigBee [11] protocol and available GPS signals to develop a novel low complexity interference diagnostic framework for WSN and GPS resource-constrained edge devices. The framework exclusively utilizes consistently available received I/Q samples, permitting the analysis of signal interactions with the fluctuating wireless environment to develop decentralized decision-making device capabilities. Performing mitigation once an attack (or packet loss reason) is detected compels this development, as edge nodes can usually deliver packets to non-jammed neighbors [12]. Initially, I/Q data is acquired by applying Matlab Monte Carlo simulations across various jamming-to-signal ratios and interference types. Simulations evaluate the ideal case for using I/Q samples without hardware limitations, identify the required data for model development and, notably, operate on a reduced feature set. The positive simulation results motivated and guided the real over-the-air ZigBee/SDR testbed, where the focus was applied to matched signal (ZigBee) and continuous wave (CW) interference. Software-defined radios (SDRs) provided access to the live I/Q data from XBee ZigBee devices and implemented the required jamming signals and continuous ZigBee transmissions when required. The live I/Q data diagnostic tools use novel low-order features previously extracted across time, frequency and space in [13] and further elaborated on in [14]. In-depth analysis and validation of the low-order features for interference detection was achieved using machine learning-based classifiers, namely Support Vector Machine (SVM), XGBoost and deep neural networks (DNNs). The developed models are evaluated using available test data, including "unseen data", and K-fold cross-validation.

The developed live data diagnostic tools investigated legitimate node classification and over-the-air jamming using commercial and SDR ZigBee transmitters. An artificial jamming scenario was also examined, using individually collected signal data and adding the samples in software. The developed methodology's transferability was investigated by applying the strategy to GPS signals. The study illustrates the value of analyzing received I/Q samples and low-complexity solutions for interference detection on edge devices. The main contribution lies in the developed diagnostic framework that enables independent operation, as no channel assumptions, network-level information or spectral images are required. It differentiates itself by solely analyzing the I/Q data, which is consistently available to functioning receivers while achieving high accuracy and generalization to unseen data. To fully validate the designs, DNNs are developed and compared to the low-complexity solutions. A Raspberry Pi embedded device implementation study examines a relatively resource-constrained deployment. An overall diagnostic algorithm is formulated based on the developed models and the signal classification approach in [13].

The remainder of this paper is organized into three main sections, excluding the related work discussion and conclusion. Section 2 describes the related work and outlines what differentiates this study. The initial ZigBee simulation-based experimental approach and associated results are described in Section 3. Section 4 depicts the live wireless data strategy and collection process for both WSN and GPS signals. The live wireless signal machine learning evaluation results are outlined in Section 5, which includes an implementation on a Raspberry Pi embedded device. Section 6 concludes this paper and outlines future work.

2. Related work

This section reviews the main related topics and how this paper's work differentiates itself from comparative literature. Topics include interference detection in wireless communications networks, namely WSNs, and the exclusive use of raw I/Q samples. Regarding WSNs, detecting interference and classifying the signal type is not an original concept but requires continual improvements to keep up with current hardware/software enhancements. For WSNs and associated resource-constrained devices, jamming is a significant threat. Key design objectives here are low complexity and independent operation, as resource consumption can be reduced if devices make real-time decisions and algorithms have low computation costs.

Vital aspects of WSN intrusion detection systems (IDSs) are discussed in [15], where jamming is outlined as a very destructive attack and the need for comprehensive IDS analysis, in both simulation and real-world implementations, is identified. Developing a balance between accuracy, generalization to new data and consumption of resources is also highlighted. In terms of this paper, simulation and real-world experimental analysis are provided in Sections 3 and 5, respectively. The work focuses on anomaly detection in received WSN signals, but the potential exists for malicious node identification, as explained in Section 5. The authors in [16] describe the lack of traditional physical switches or gateways in WSNs as a vulnerability and emphasize the need for detection approaches. Finally, an IDS should not degrade WSN performance or introduce new weaknesses but should be reliable and transparent to the system [16].

Classic WSN techniques and IDSs typically analyze the received signal strength indicator (RSSI) and packet rates [17]. This method requires high volumes of transmitted packets to calculate representative packet rates, such as packet error rate (PER), and devices, even those operating at the edge, to obtain network-level knowledge. Other packet rate systems analyze collaborative approaches [18] that evaluate packet delivery rates (PDRs) in a given area. This permits faster detection than end-to-end PDR and achieves jamming detection accuracy of over 97% [18]. In contrast, chip sequence error patterns are used in [19] to identify the channel and, as a result, the emitting interference. This is a

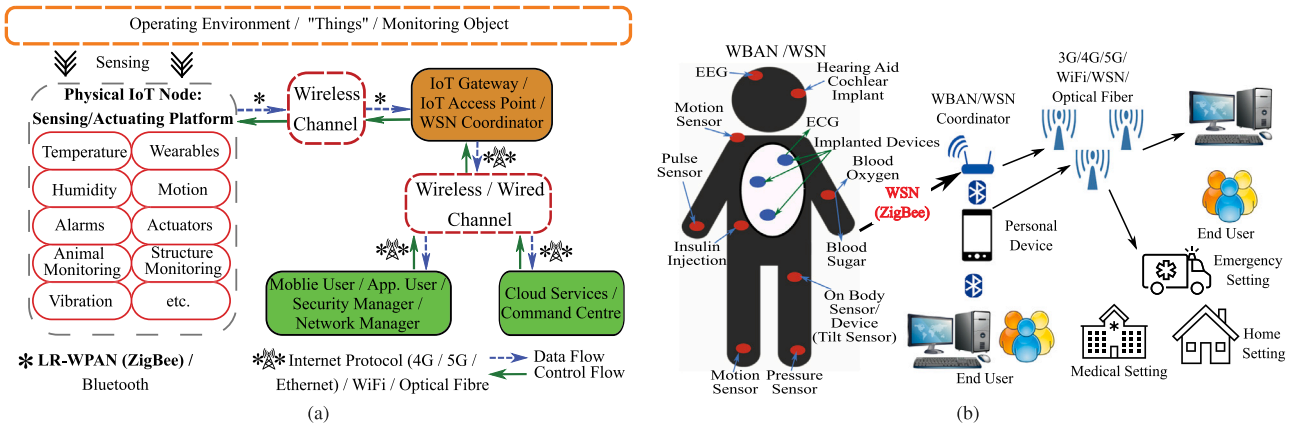

Fig. 1. Example critical use-cases of WSNs. (a) Civil WSN utilization in WSN and/or IoT applications by providing the communication link from the sensing/actuating platform to the IoT gateway or network access point. (b) Example WBAN (subset of WSNs) application depicting the potential critical use in health-care architectures and associated private data being transmitted. The end users include physicians, emergency services, medical/personal server etc.

step above raw I/Q sample analysis and four major chip error patterns were identified that allowed the distinction between inference from different sources, including IEEE 802.11 and 802.15.4. This approach requires edge devices to buffer known patterns and calculate a pattern recognition classifier. Also, if packets cannot be received, essentially all the chips are incorrect and the type of interference can be one of a variety of transmissions, given enough power. In contrast, SonIC [20] samples received RSSI values to extract features for a decision tree classifier for edge device applications. However, this process is limited as it requires a successful retransmission of the previously identified error packet, for comparison, and needs a buffer to store the most recent error packet. SVMs and RSSI samples are used in [21] to develop an accurate and fast interference detection process using four SVMs and a logic decision stage.

In [22], the potential uses of machine learning in WSNs is discussed, where security and anomaly detection are identified as viable use cases. In [23], network information, for example, packet received signal strength, packet drop rate and retransmission rate, among others, is used to detect intrusions. The use of received signal strengths and packet descriptive rates are sufficient for specific applications. However, more information can be discovered by expanding into other frontiers. In [24], the throughput, packet drop ratio and the packet average delay of sensor nodes are used in a Bayesian classifier to identify anomalous nodes. Different techniques are compared in their ability to identify WSN outliers in [25]. An example of using decision trees as an intrusion detection method is provided in [26], where the main advantages include having the best detection performance, ease of model construction and interpretation and scalability for large datasets. Notably, Random Forest was highlighted as outperforming other classifiers in terms of identifying whether data traffic is normal or under attack when using the NSL-KDD dataset in [27]. The Random Forest technique was also previously shown to be capable of detecting jamming in GPS signals in [28].

Other approaches aiming to prevent jamming attacks exist in the literature. Examples of these approaches include a hybrid approach that uses a combination of direct sequence frequency hopping/time-hopping spread spectrum to protect against jamming [29] and is only validated in simulations. In [30], another simulation-based detection approach is proposed that focuses on effective channel utilization and interference power. In [31], a technique based on clustering and node timestamps is proposed, which performs well under several metrics, including routing overhead, energy usage and packet delivery rate. However, it is only validated using simulations (Matlab). Finally, the authors in [32] illustrate a link quality-aware bypassing mechanism to negate the presence of jamming by bypassing the jammed zone. Results indicated that typical network performance metrics increased, yet it is limited by the lack of a real wireless deployment or analysis. Many of the proposed approaches for negating or detecting jamming lack the results from live wireless signals and require information from high up in the network stack.

Focusing solely on using received raw I/Q samples for decisions is a relatively new concept. In [33], the authors achieve uncooperative direction finding using a single uncalibrated directional receiver by utilizing deep learning techniques focused on the characteristics of a transmitter's I/Q components. The authors in [34] identify transmitters by using high dimensional features extracted from I/Q imbalance information. Rogue transmitters are detected by applying a generative adversarial network (GAN), while trusted transmitters are classified by convolutional and fully connected deep neural networks. Recurrent neural networks (RNN) and I/Q information are used in [35] to predict primary user (SDR transmitters) activity in dynamic spectrum access networks. Unchanging, hardware-based characteristics of individual transmitters were extracted from I/Q data in [36] and used in a deep learning approach for radio device identification. Raw I/Q samples are used as specific components in the design of a signal classification approach for LTE, WiFi and DVB-T signals in [37]. However, additional RSSI and spectrogram features were required, in addition to raw I/Q samples. The literature proves that using only raw I/Q samples can be useful while opening new areas for investigation.

This paper differentiates itself from the literature by exploring a novel investigation concentrated on exclusively using raw I/Q samples and low-cost open-source hardware and software for WSN interference detection and classification. The solution focuses on independent edge device decisions based entirely on the wireless channel's effects on received I/Q samples, makes no channel assumptions and requires no network-level data. The Matlab simulations in [38] and [39] provided the initial motivation for solely using I/Q samples. This paper expands on previous simulations and explores real over-the-air signals. The use of fundamental supervised algorithms based on effective and descriptive data analytics/signal processing highlights that heavily studied machine learning approaches are still fit for purpose. The main contribution is the real world validated interference diagnostic tool for independent compact WSN devices based on raw I/Q samples and a novel low-order statistical feature-set. This design highlights that low-level data can be used to make typical higher-level decisions. Finally, in contrast to previous work, only the designed, optimized machine learning model is required on the device, and both malicious and unintentional interference can be classified.

3. Experimentation: ZigBee simulations

This study establishes a simulation-based approach to interference detection as the "ideal" case without such hardware limitations as,

Table 1
IEEE 802.15.4 PHY frame layout.

Synchronization Header (SHR)		PHY Header (PHR)	PHY Service Data Unit (PSDU)	
Preamble	SFD	Length	Payload	FCS
4 Bytes	1 Byte	1 Byte	0–125 Bytes	2 Bytes

for example, reference voltage and analog to digital converter (ADC) resolution. The previous signal classification work in [13] produced 14 features from the analysis of time, frequency and the probability density function (PDF). The simulations use a reduced feature set of 9, due to the lack of hardware restrictions. This simulated work identified the type of data required for interference detection and contrasted the data needed for legitimate signal classification. ZigBee was chosen as it is the de-facto standard for WSNs, as almost all available commercial and research sensor nodes have a ZigBee transceiver chip [40].

3.1. Experimental setup

The Matlab simulations focused on node-to-node communications and applied ZigBee specifications [11] where possible. The ZigBee frame, visualized in Table 1, contained the required preamble of four zeros, start frame delimiter (SFD) of "7a" and a randomly populated payload. The cyclic redundancy codes, used as a frame check sequence (FCS), were fixed at "aa", as all packets (and associated elements) were available during simulations, whether error-free or erroneous. Monte Carlo simulations were implemented across a range of jamming to signal ratios (JSRs), packet overlaps and payload lengths. To ensure randomness in the simulated payload data, the seed of the Matlab random number generator "rand" was set using the current time. The generated random numbers were between 0 and 255 (inclusive), as the payload is formulated using 8-bit numbers. The random number distribution was investigated using five million iterations and the maximum allowable PHY payload length of 125 bytes. The distribution results revealed a satisfactory uniform distribution.

Additionally, every simulated transmission includes additive noise, which satisfies a zero-mean Gaussian distribution, to support a simplified authentic transmission model. The noise signal is in-band noise with the same number of samples as the simulated ZigBee signals, where corresponding samples add together. The noise is constructed as a complex (I/Q) signal with sufficient power not to cause errors in the transmission process. This simulation approach was explored by translating the Matlab code into Python3, where equivalent Matlab functions from the "SciPy" library were implemented. These custom ZigBee samples were transmitted using an Analog Devices ADALM-PLUTO Active Learning Module (Pluto SDR) and compared to a commercial ZigBee transmitter, the DIGI XBee. The results are visualized in Fig. 2, where the simulated process correlates well with the commercial ZigBee transmission and, so, the proposed Matlab simulated setup is validated. This real-world investigation identifies that the transmitted signals were received without error in the presence of noise. Python3 was chosen as this enabled the SDR to be controlled remotely by a Raspberry Pi embedded device and allowed for a side-by-side comparison to the Raspberry Pi-controlled XBee devices. This testbed setup is explained in greater detail in Section 4. This experimental approach validated the ZigBee signal construction approach in the simulations.

Different forms of interference were examined using varying power levels to understand the effects of these interference signals on ZigBee (IEEE 802.15.4) transmissions. This range of JSRs provided for several interference classes, including error-free, unintentional, subtle jamming and saturation. A random frequency offset from the ZigBee operating frequency was added to each interference signal to resemble real-world transceiver conditions. This center frequency offset is in the range of a few tens of kilohertz and based on a random number from the standard uniform distribution. The applied interference signals were

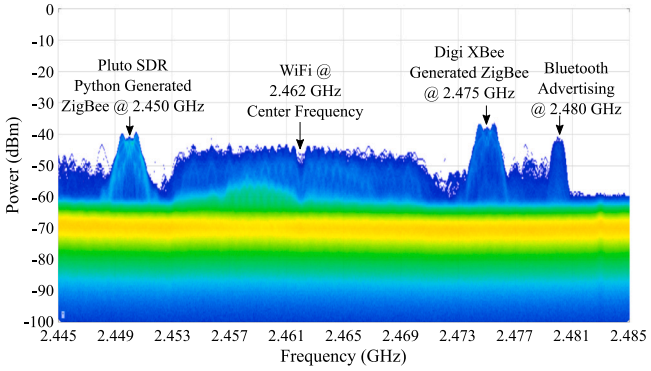

Fig. 2. A Tektronix DPX image [41], which is a digital signal processing software that rasterizes samples into pixel information, of the customized SDR and commercially transmitted ZigBee signal.

CW jamming, matched signal interference (same and adjacent channel), thermal noise (attenuation channel) and WiFi (802.11b) coexistence. Matched signal interference is an intelligent deceptive jammer attack style that transmits packets that match the spectral identity of legitimate signal (ZigBee) [42]. Each simulated interference signal is constructed with the same number of samples as the ZigBee signal and follows appropriate standards or signal formula such as, for example, the known CW I/Q signal model. The interference is in-band and sampled using the same sampling frequency as the simulated ZigBee signal. The corresponding samples from the legitimate ZigBee and interference signals combine with the noise signal to form the received signal at the simulated receiver. All interference signals are based on random data from a standard normal distribution. The samples are transmitted with an interference power related to the known power of the legitimate ZigBee transmission, providing the required JSR.

CW jamming corresponds to typical spurious jammers, including constant, random, deceptive and reactive approaches and does not need to know what protocol is in use. CW methods operate by emitting spurious RF signals into busy wireless channels without permission and breaking spectrum laws. Matched signal interference operates by monitoring the network and identifying the operating protocol (for example, ZigBee) before injecting protocol-specific interference, which is more difficult to detect than conventional jamming techniques [42]. The thermal noise approach relates to noisy and hostile environments, which cause higher transmission gains to achieve the same signal-to-noise ratio. WiFi signals, at the three possible frequency offsets (2, 3 and 7 MHz), were used to investigate the problem of system coexistence and whether misuse can lead to malicious interference. These interference signals and the described Monte Carlo approach were used to develop a database of received ZigBee error-free and erroneous I/Q samples.

A maximum likelihood decoder is deployed as the simulated receiver using the ZigBee sampling rate of 4 MHz. Each received 32-chip pseudo-random noise (PN) sequence "P" is compared with a lookup table of ZigBee's predefined sixteen direct-sequence spread spectrum PN codes (PN1, PN2, …, PN16). Specifically, received samples are compared to an ideal set of samples for each PN code. In either case, the comparison produces a set of results, (k1, k2, …, k16), which indicate the Hamming distance, H, of the received PN sequence to each sequence in the lookup table, where H indicates the number of misaligned samples. Minimizing H maximizes the correlation and k denotes the index producing the minimum value in (1), where $H(P, PN_k)$ is the Hamming distance between two sequences. Typically, during packet transmissions, samples/chips can be corrupted due to spurious intentional and/or unintentional interference, coexistence, fading, path losses, obstacles, etc. However, as long as the value of H (chip/samples errors

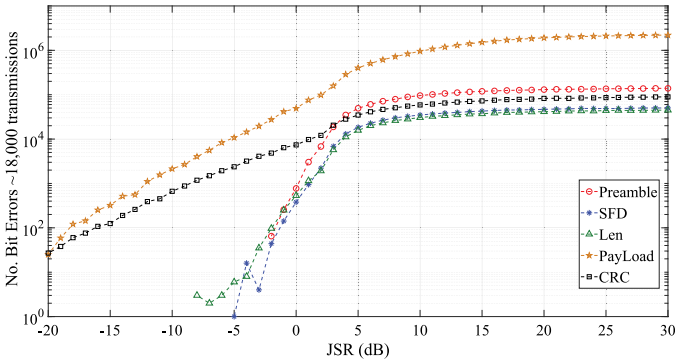

Fig. 3. Number of ZigBee frame bit errors in five categories under matched signal interference for a range of JSR values, which provided the initial motivation for investigating the use of received I/Q samples in designing an interference detection strategy.

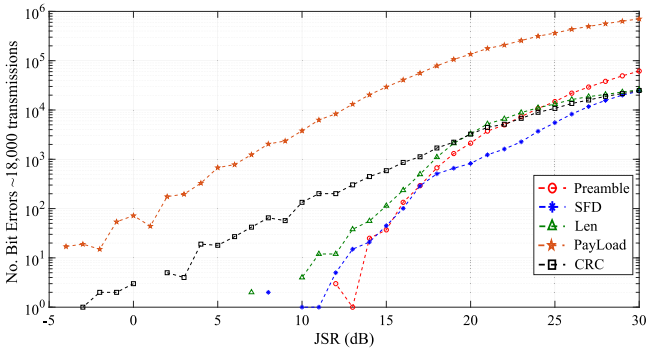

Fig. 4. Number of ZigBee frame bit errors in five categories under CW interference for a range of JSR values, which upheld the initial insight of using received I/Q samples in the design of an interference detection strategy.

per PN code) is below a certain correlator error threshold (identified as ten chip errors in [19]), the correct symbol will be selected.

$$arg_k \ min \ H(P, PN_k), \ \ for \ k = (1, 2, \ldots, 16) \tag{1}$$

For this study, a correlation failure, which is an incorrect symbol having the minimum Hamming distance, defines an error. In these simulations, the maximum likelihood receiver demodulates the received samples according to the minimum achieved Hamming distance for each set of 32 chips. As the transmitted signal is available in each simulated transmission, the received bits can be compared to the transmitted bits. As a result, the maximum likelihood receiver uses Eq. (1) to identify received errors analytically. A single-bit error causes a packet error since either the synchronization or FCS fails. The described simulation process provides both error-free and erroneous received I/Q samples. These samples are explored to detect if any statistical differences (features) exist between error-free and erroneous samples and between the interference signals. Particularly, subtle jamming attacks are explored as, typically, these attacks are more difficult to detect using traditional PER and RSSI methods.

3.2. Bit-error location analysis

The initial results focused on where the bit errors occur across the ZigBee frame for a range of JSR values. The matched signal interference attack was analyzed as it generates a PER of 18% at a JSR of 0 dB [42], thereby being the most effective of the studied attacks. The results are provided in Fig. 3, where the bit errors occur across the frame at high jamming powers and become more specific to the payload as the jamming power decreases. The CW attack was investigated and similar observations were obtained, as shown in Fig. 4. The bit-error location results indicate that the probability of bit errors occurring in the packet preambles decreases with decreasing jamming power, which increases the probability of synchronizing to packets under the presence of a jammer. Therefore, an interference detection framework needs to analyze packets with bit errors and when no packets can be received or before transmitting a packet. This finding was the first indication that investigating received I/Q samples had promise, as I/Q samples can always be received, by a functioning receiver, from the wireless channel.

3.3. Feature extraction

The simulation method depicted in Section 3.1 was used to develop a database of simulated I/Q samples consisting of six interference setups, fifteen packet overlap scenarios and error-free signals. The JSR range consisted of 1 dB increments from −15 dB to 30 dB, while the interference signal overlaps encompassed overlaps for both before and after legitimate transmissions, for percentage overlaps of 10, 20, 40, 50, 60, 80, 90 and 100%. The analysis signals included error-free ZigBee, matched signal interference, CW, WiFi (at the three possible center frequency offsets) and Thermal Noise. Extracted features aim to distinguish error-free ZigBee signals from ZigBee signals with errors caused by an interference signal. In practice, the I/Q samples are accessible using software-defined radios, as shown in Section 4, or, possibly, in the device's debug mode, if otherwise unavailable. The following features were originally extracted in [38] and analyzed in both [38] and [39], based on matched signal interference. This paper expands the previous work using the same features to analyze the full signal and overlap range.

Features were initially extracted from the measured PDF, as when interference is low and packets are error-free, the PDF is a relatively narrow bimodal shape with a low degree of variance. As the interference power increases in the channel, the PDF becomes more bimodal with larger variance. This result is evident in the PDF for matched signal interference, provided in Fig. 5, where the trend under increasing levels of JSR allows distinguishable features to be extracted. Notably, the error-free PDF in Fig. 5 closely resembles what is observed in the spectrum, visualized using the Tektronix RTSA in Fig. 2. The extracted PDF features include the area between bins −2 to +2, the averaged area of the bins −128 to −3 and +3 to 127, the number of non-zero bins and the maximum peak. Matlab's *trapz* function calculates the areas and is shown in (2), where the spacing is constant, due to PDF construction, $f(x)$ is the PDF function and N is the corresponding number of bins. As the JSR increases, both the total area in the center bins and the maximum peak decrease, while the averaged area in the outer bins and the number of non-zero bins increase. This process is possible here, as simulations incur no hardware related restrictions.

$$\int_a^b f(x)dx \approx \frac{b-a}{2N} \sum_{n=1}^{N} (f(x_n) + f(x_{n+1})) \tag{2}$$

The features are expanded by analyzing the received I/Q samples directly as a time series. Derived features include the sample variance (and standard deviation to investigate which feature is essential), the signal's entropy, the mean value and the absolute maximum value in the received sample set. As the jamming power increases, so too does the variance, standard deviation, mean and absolute maximum of the I/Q samples. The entropy is calculated using (3), where P_i contains the available samples' normalized histogram counts. The entropy decreases as the noise-like error-free signal becomes encompassed by a more dominant interferer. Entropy is described as "a statistical measure of randomness", which implies that noise signals typically have a higher entropy value than high powered dominant signals, like an applied interferer.

$$H = -\sum_i P_i \log_2 P_i \tag{3}$$

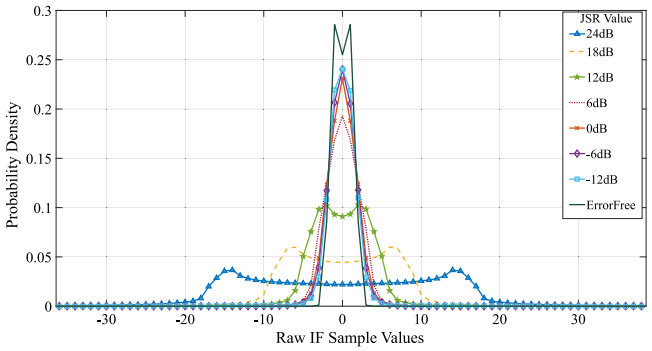

Fig. 5. Measured PDF of simulated I/Q samples under matched interference for various JSR values, specifying the increasingly bimodal shape as the JSR value is increased.

Table 2
Four interference types: random forest algorithm — sampling bias comparison.

	All data	Training data	Validation data	Testing data
Error Free	0.1219	0.1219	0.1219	0.1219
Matched	0.2416	0.2416	0.2417	0.2416
CW	0.0196	0.0196	0.0196	0.0196
WiFi	0.2416	0.2416	0.2416	0.2416
Noise	0.3753	0.3753	0.3753	0.3753

3.4. Interference detection results

For the simulated I/Q datasets, the well researched and implemented SVM [43] and Random Forest [44] machine learning techniques were applied based entirely on the feature set from Section 3.3. The procedure focused solely on matched signal interference and was validated using a SVM in [38] and expanded into a multi-class classifier by utilizing the Random Forest decision tree approach in [39]. The results here expand on those initial studies by including the full set of identified interference signals and overlaps. The data was split into training (70%), validation (20%) and testing (10%), resulting in an estimate of the error rate in new cases, known as the generalization error (or out-of-sample error), being achievable. In each dataset, the proportion of each interference signal and error-free samples was consistent to avoid sampling bias, which is visualized in Table 2.

The initial matched signal SVM results are provided in Table 3, where the radial basis function (RBF) kernel was selected using available validation data. In this analysis, the SVM was trained using data points from -10 dB $\rightarrow$ 20 dB and different binary detection thresholds to present a comprehensive study of the algorithm's performance. Table 3 identifies these thresholds, which are based on the bit-error analysis and feature trends outlined in [38]. Testing data included JSR points outside the training range to examine how the model generalizes to unseen data. The aim was to identify the lowest JSR value with which the algorithm could accurately identify the presence of interference. Table 3 indicates that sufficient differences exist between the non-interfered and interference corrupted data, even at low JSR levels. The main source of error is classifying between different interference regions, see Table 3. As a binary detection approach, the SVM achieves near-optimum performance, as per Table 4.

The Random Forest results for the matched signal interference case are shown in Fig. 6, where validation data determined the optimal metrics (hyper-parameters). These metrics were the number of decision trees and predictor depth and were chosen based on the lowest achieved generalization error, model training time and average prediction time. The four-class case is presented based on the packet and bit error analysis presented in [39] and outlines how a single Random Forest model can predict between different categories. A PER of $\leq 10\%$ and bit errors ≤ 15 defines a region where unintentional interference or high channel noise may exist, a PER from $11\% \rightarrow 32\%$ and bit errors from $16 \rightarrow 20$ defines a subtle jamming or signal collision region and above these resides a high impact jamming region. In terms of JSR, these zones correspond to < -2 dB, $-2\ dB \rightarrow 2$ dB and > 2 dB, respectively, which provided the four-class classification case when combined with error-free data. These thresholds differ from the SVM approach as the analysis was expanded to include the PER at the specific JSR values.

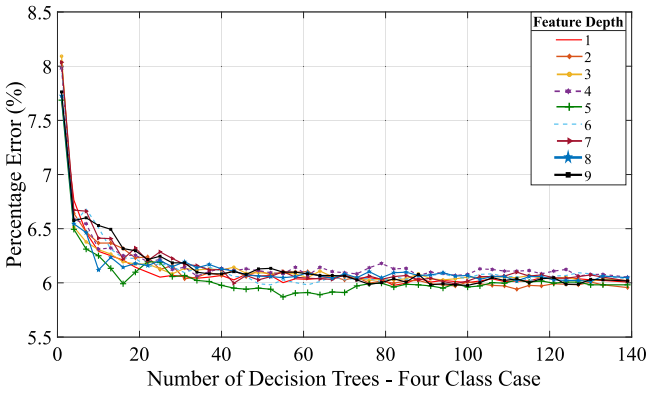

Fig. 6. Designed Random Forest algorithm for the matched interference Four-Class Case (Error free, unintentional, subtle jamming and high impact jamming) - Validation Generalization Error.

Table 3
Matched interference SVM results (training data): multiple detection thresholds and radial basis function kernel.

JSR detection threshold:	Selection reason	10-Fold cross validation error	Test data error
≥ 5 dB	Identified initial Threshold from Feature trends	6.9619%	4.4508%
≥ 0 dB	Expected spectral Power	3.9628%	2.6515%
≥ -5 dB	Below -5dB: likely Error-Free SHR (Preamble and SFD)	0.9741%	0.8380%
≥ -10 dB	Lowest training JSR point	2.35e−04%	0.0%

Table 4
SVM results (training data): binary classification and radial basis function kernel.

Training time	Number of test points	Percentage error	Avg. analysis time	10-Fold cross validation error
Matched Interference				
11.057 s	78,854	0%	1.15 ms	0.0011%
Four interference types — full packet overlap				
60.62 s	247,615	0.00723%	1.67 ms	0.0082%
Four interference types — varying packet overlaps				
426.69 s	1,611,311	0.001489%	1.3 ms	0.0016%

This design produced results as per Table 5, where most errors occurred during classification into interference operating zones. When this approach is applied as a binary classification, using the same metrics, the error was approximately 0% with an average prediction time of 43.1 ms. For the SVM, using the same data, the error was the same (0%), but the average prediction time was just 1.15 ms, see Table 4. These initial results motivated the expansion of the interference types. Furthermore, for binary interference detection at the edge, where real-time decisions are crucial as data can become obsolete in a matter of milliseconds, the SVM is the chosen approach.

The next phase examined using the designed features to classify the interference type, which included matched signal, CW, WiFi and

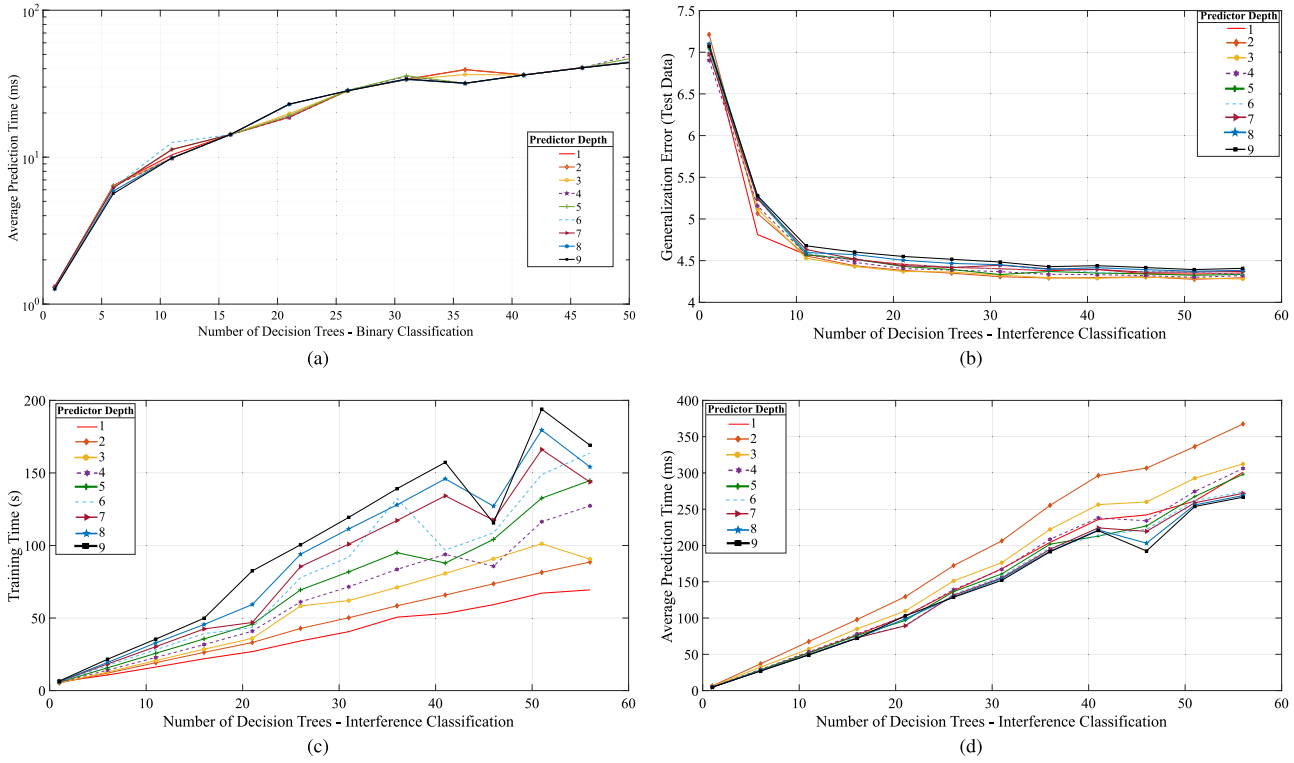

Fig. 7. The Interference Classification results for the Random Forest Algorithm using available validation and testing data. (a) Average prediction time results for binary classification between error free and interference signals. (b–d) The optimal metric detection results for the designed Random Forest interference classification algorithm, specifying (b) the generalization error, (c) the training time and (d) the average prediction time.

Table 5
Designed random forest algorithms: specifications.

Data	Predictor depth	Number of trees	Training time	Prediction time	Test error
4-Class matched interference					
Validation	5	55	31.23 s	133 ms	6.10%
Training	5	55	129.80 s	372 ms	6.14%
4 interference types (Matched, CW, WiFi & Noise Interference)					
Validation	4	46	85.72 s	234 ms	4.32%
Training	4	46	343.5 s	714.24 ms	4.227%
Inclusive of varying overlaps					
Training	4	46	6423 s	5.2525 s	4.027%

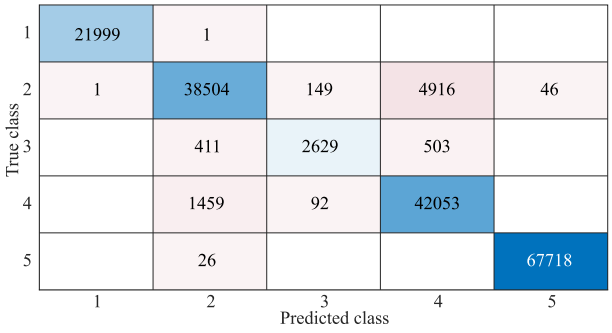

Fig. 8. Confusion Matrix for the designed multi-class Random Forest classifier, where the results are based on available testing data and classes are as follows: 1-No Interferer Present, 2-Matched, 3-CW, 4-WiFi, 5-Noise.

thermal noise. Based on the matched signal study, the examination applies a SVM as the binary interference detector, while the Random Forest algorithm predicts the interference type. This approach permits the design of individual SVM and Random Forest models, rather than multiple SVM models for different classification situations (thereby reducing computational requirements).

The initial results target full packet overlap, where the attack packet is the same length as the legitimate packet. The SVM results are supplied in Table 4, which correspond to classification between error-free I/Q samples and I/Q samples containing the presence of different interference signals. Analysis of the validation and testing data determined the RBF kernel to be optimal. Based on the available testing data, the generalization error for full packet overlap was established as 0.00723% with an average prediction time of 1.67 ms.

For the Random Forest approach, optimal metrics (predictor depth and the number of trees) were determined using the validation and testing data. The results provided in Fig. 7, which depicts the generalization error in Fig. 7(b), training time in Fig. 7(c) and average prediction time in Fig. 7(d), identified the optimal metrics. Fig. 7(b) shows that the error plateaus when approximately 40–50 trees are being used, regardless of the predictor depth. The lowest error occurs when using a predictor depth of two or three, but this produces the longest prediction time. A trade-off exists and the designed Random Forest model metrics were chosen to be 46 trees and a predictor depth of 4. The corresponding performance using training data is specified in Table 5, where the generalization error was 4.227% with an average prediction time of 714.24 ms. The associated confusion matrix is provided in Fig. 8, which provides an insight into the misclassification errors and suggests that the majority of cases can be correctly classified.

The binary classification results of the Random Forest method for full packet overlap achieves a similar degree of error (approximately 0.0011%), compared to the SVM. However, a higher average prediction time is required in all but the single tree case (as shown in Fig. 7(a)), thereby reinforcing the use of a SVM for initial interference detection.

For varying degrees of overlap, which is the most advanced SVM binary classifier designed, the error was 0.001489% with an average

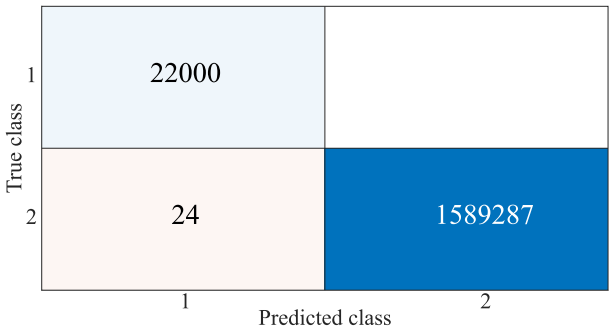

Fig. 9. Confusion Matrix for the designed SVM based on the data including varying overlaps, where the results are based on available testing data and the classes are as follows: 1-No Interference Present, 2-Interference Present.

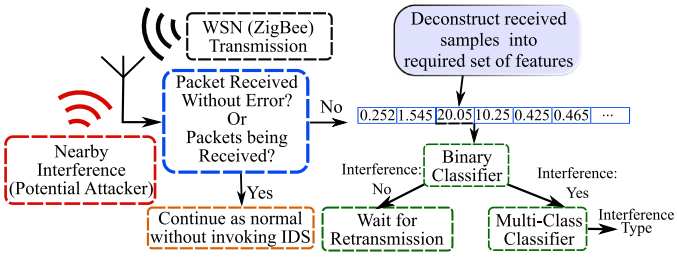

Fig. 10. Data flow diagram representing the developed two model approach which leverages binary and multi-class classifiers.

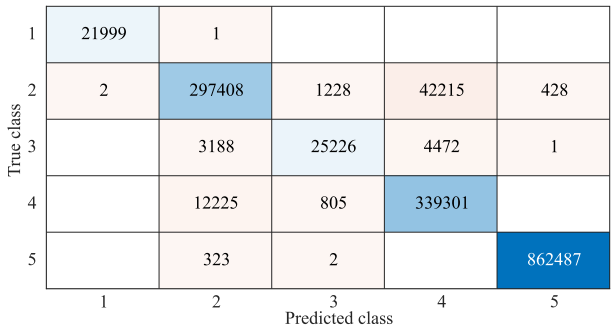

Fig. 11. Confusion Matrix for the designed multi-class Random Forest classifier including the varying packet overlaps, where the results are based on available testing data and classes are as follows: 1-No Interferer Present, 2-Matched, 3-CW, 4-WiFi, 5-Noise.

Table 6
SDR specifications.

	Analog pluto	RTL-SDR
Connectivity	USB 2.0	USB 2.0
Frequency range	325 MHz–3.8 GHz	25 MHz–1.75 GHz
Max. RF Bandwidth	20 MHz	2.4 MHz (3.2 MHz Max)
Sample rate	65.2 ksps - 61.44 Msps	2.048 Msps
Sample depth	12 bits	8 bits
TX | RX channels	1 | 1	0 | 1

prediction time of 1.3 ms. The confusion matrix for this SVM classifier was investigated and is supplied in Fig. 9, where the corresponding area under the curve (AUC) of the receiver operating characteristic (ROC) curve is approximately unity. These results motivated investigating a live over-the-air wireless signal approach and adopting the SVM classifier as the interference detection mechanism, visualized in Fig. 10. As energy is typically limited on edge devices, the process of interference detection should only occur once a packet error has occurred and results determined as quickly as possible to enable real-time responses.

When the varying overlaps data were analyzed using the optimal Random Forest metrics for full overlap, the generalization error was 4.027%. The confusion matrix is shown in Fig. 11, using all available testing data to understand where the errors were occurring. The results show that the designed multi-class classifier detects interference for each instance that encompasses a sufficient interference signal. The errors occur when determining the interferer type, as shown by examining the corresponding segment in the confusion matrix. Errors can be reduced by adding more instances of certain interference types, such as, for example, CW, or by utilizing a boosting algorithm like XGBoost. However, these are simulations and are computed only as an initial insight into the methodology development. The results indicate the optimal approach adopts a SVM for initial interference detection and, if interference is detected, a Random Forest interference classification model. The analysis of the live signals will investigate whether a similar approach is viable for wirelessly received signals in Section 5.

When these results are compared to fast jamming detection focused on collaborated packet rate information [18], the simulated performance achieves equal, if not better, results. The approach in this paper provides novelty over PDR based systems as individual nodes can make decisions based on I/Q samples received from the wireless channel. This method results in fast response times and high accuracy without the need for edge device collaboration or network parameter information. The simulated results have revealed a detection approach for individual edge devices that is both fast and accurate, which is advantageous compared to clustering or network parameter techniques [31].

However, these simulated results, which exhibit low levels of error and "ideal" classifier performance, do not have to consider hardware restrictions and cannot model live wireless signals (and associated environmental interactions) exactly. Wireless channel characteristics such as, for example, fading levels, obstacles, path losses, spurious interference, etc., are inadequately modeled. The absence of a real ADC means available resources do not limit the simulations. The simulation work provided insights for live data feature extraction and data analysis but lacked real environmental issues evident in wireless transmissions. The promising simulation results suggest that this framework is a feasible solution that warrants a hardware approach rooted in real over-the-air signals, focused on the simulation study's attributes. Notably, the simulation study has identified the type of data needed to train the jamming detection models, i.e., signal interactions of legitimate ZigBee signal and a jamming signal.

4. Live data collection & experimentation

The data strategy, accounting for data quality, quantity and source, incorporates SDRs and Raspberry Pi embedded devices [45]. SDRs are reconfigurable radio systems whose characteristics are partially or fully defined via software or firmware [46] and whose main components are the antenna, RF front-end and processing unit. The Raspberry Pi can control, configure and power the selected SDRs by utilizing available Python3 libraries. This data strategy targets collecting typical wireless I/Q data in real-time for off-line feature extraction to develop an interference detection method. These SDRs interact with the RF environment by utilizing a hardware peripheral, whose capabilities characterize transceiver operation, while the software component's performance depends on the proficiency of the RF front-end. It is critical to use a SDR with the appropriate hardware for analyzing the chosen RF WSN signals. Two separate hardware approaches were developed for ISM RF band and GPS signals, respectively.

4.1. Data collection: ZigBee

This investigation's main work focuses on WSN edge device security by implementing an interference detection and classification tool. The WSN data collection process adopted a previously developed ZigBee testbed that consists of DIGI XBee ZigBee nodes connected to Raspberry Pi devices that incorporate live sensor data by implementing the SenseHat environmental sensor. The setup included one receiver (coordinator) and several transmitters. The Raspberry Pi controlled all transceivers and sensors using available Python 3 libraries, including "sense-hat" and "digi-xbee". The SenseHat sensor provided access to live environmentally sensed data, including temperature, humidity and pressure, while all transmissions could be tracked using the Raspberry Pis. IoT capabilities are enabled by authorizing a WiFi-enabled coordinator to utilize the open-source Dropbox Uploader tool, exploiting a Dropbox API application, to upload received data to the internet for remote analysis. This functionality emphasizes how ZigBee provides sensor level communication in the overall IoT architecture (Fig. 1(a)).

To receive I/Q samples from and to implement penetration tests on this WSN testbed, the Analog Pluto [47] (≈ $149) SDR was utilized. This device has specifications as per Table 6 and is based on the Analog Devices AD9363 transceiver. This SDR receives I/Q samples in the range of [−248:2047]. The Pluto SDR can be controlled by Matlab/Simulink, using the Communications Systems Toolbox add on, and by python (pyadi-iio) or $C/C\#/C++$, using the "libiio" library. The python3 "pyadi-iio" and "libiio" libraries are implemented on a Raspberry Pi device for lightweight and remote operation. At the same time, Matlab is utilized to provide the jamming signals with sufficient frequency to interact with the WSN signals. The ZigBee Siretta stubby antenna [48], designed for use in the 2.4→2.5 GHz range, was selected to provide optimal operation in the ISM RF band.

The commercial ZigBee nodes transmit the required WSN commercial packets while the SDR receives WSN data on the required channel, acquiring I/Q samples in the process. The SDR is additionally utilized to provide constant ZigBee based signals while in the presence of jamming (also transmitted from an Analog Pluto SDR). A sampling rate of 4 MHz, twice the ZigBee baseband signal bandwidth of 2 MHz, is applied in all experimental approaches, as this study targets WSN interference detection for edge devices. As shown in Section 3.1, the customized ZigBee approach can be used to transmit ZigBee signals (Fig. 2) from a suitable SDR. The hardware approach depicted in Fig. 12(a) can receive WSN I/Q data for off-line analysis and model development and transmit matched signal interference, along with other suitable interference signals. For the interference signals, a 20 dB CN0417 2.4 GHz RF power amplifier permits the Pluto SDR to transmit higher-powered signals, as the typical transmit power is limited to a maximum of 7 dBm. WSN jamming signals are applied wirelessly over short time intervals on a specific WSN to avoid attacking commercial applications while still analyzing legitimate signals in a typical wireless environment. A data storage device is utilized, as received I/Q data requires post-processing to extract the required signals and features. Overall, this SDR utilization provides the necessary I/Q data access under normal and interference conditions for various interference types. For this study, CW and matched signal interference are the focus.

4.2. Data collection: GPS

GPS signal data was acquired to investigate whether the developed diagnostic approach could be adapted to other wireless spectrum areas. GPS signals are becoming increasingly important for civilians, services and industries due to the dependence on GPS-derived location and time measurements. This study uses GPS signals, on account of previous work in GPS interference detection [28], signal availability and due to unintentional and malicious in-band interference being the single most significant threat to GPS applications and users.

To expand on previous work [28] and to analyze as many satellites as possible, GPS data was collected across a full 24 h period, where all available satellites were visible at least once. This approach was validated by using GNSSRadar [49], a software tool used to show the current GPS constellation for a specific location and the satellites' orbital speeds. This methodology means that associated results are not dependent on a specific subset of satellites. The NESDR SMArTee RTL-SDR (≈ $32) [50] was the chosen receiver and receives samples in the range [−128:127]. This SDR has specifications as per Table 6 and utilized a magnetically mounted active GPS L1 patch antenna that provides approximately 20 dB of gain. This RTL-SDR (Realtek RTL2832U chipset) was chosen as it has associated software, "rtl-sdr", for I/Q data collection, functions on a Raspberry Pi and produces data outputs that are compatible with "fastgps" [51]. A 2.048 MHz sampling rate is applied, which is twice the GPS baseband signal bandwidth. This program is a GPS software receiver that performs the entire signal processing in software, allowing for smooth adjustments at all stages: correlation, acquisition, and navigation. This receiver uses the received I/Q data from the hardware approach shown in Fig. 12(b) to identify the received satellites. GPS data needs to contain data from at least four satellites to be useful. The "fastgps" program, running on a Raspberry Pi, validates if the required number of received satellites are present in the collected I/Q data. This procedure allows for collecting both good (4 satellites or more) and interfered data for off-line analysis. This results in the approach illustrated in Fig. 12(b) being an all-in-one GPS data collector and receiver. The receiver's antenna position was confirmed to be able to access each of the required satellites before collecting the required data.

As GPS signals are received at such low power levels (typically -125 dBm), relatively low powered jammers, which broadcast noise on GPS frequencies, will typically block the reception of GPS signals. This fact implies that a jammer can have a large effective range, even though it might be a relatively low powered signal. Here, the primary source of interference was CW interference, while offset quadrature-phase shift keying (O-QPSK) transmissions were also investigated. The selection of these signals is based on the hypothesis that different signal types can jam a GPS receiver. The most common is a CW signal, but other modulation schemes can be used. The ZigBee modulation scheme is leveraged as a GPS jamming signal. This process investigates if the developed approach can be used in a different area of the RF spectrum. Both signals were emitted into the GPS reception method using a wired approach to avoid jamming nearby GPS receivers. The approach specified in Fig. 12(b) depicts the GPS reception method with and without interference. No power amplifier is required in this process as the transmission power needs to be attenuated and the Pluto SDR supplies a maximum attenuation of 89.75 dB. A DC block and a signal adder are applied in the collection process to avoid damaging the device. However, when no interferer is connected, a 50Ω termination needs to be applied to the DC block to mimic the Pluto SDR and have comparable results. Both cases need to be under matched impedance conditions.

5. Results: ZigBee and GPS data

The simulation results in Section 3.4 enabled jamming/interference detection for a variety of regions, including an unintentional interference or high channel noise region, subtle jamming or signal collision region and a high impact jamming region. The features extracted from the simulation results in Section 3.3 are as a result of the ideal case where the PDF and samples have no numerical limitations. However, additional features are required due to the hardware restrictions (reference voltage and ADC resolution). This requirement was identified when developing an ISM band signal classification tool in [13], where additional features were required to classify signals when the receiver was saturated or when signals used similar modulation schemes. The developed signal classification models achieved high accuracy and

Fig. 12. Pluto SDR and RTL-SDR dongle approaches, in Theory and Practice, being controlled and configured by a Raspberry Pi 3 embedded device for (a) WSN analysis and interference addition and (b) GPS signal reception and interference addition.

generalized well to unseen data. The received data was collected using an Analog Pluto SDR and the samples were scaled to the range [−1:+1] to provide features with similar value ranges. This technique typically results in higher-performing machine learning models. In this study, the Analog Pluto's received samples are scaled to the range [−1:+1] for consistency, while the GPS signal data remain in the RTL-SDR range of [−128:+127] for an additional examination of the features. The work in [13] extracted 14 low-order features, from sample sizes of 1250 samples, that included: (1) Number of non-zeros entries in the calculated PDF, (2) The area in the center bins ([−0.1:0.1] or [126:132]), (3) The area in the left hand side bins ([< -0.1], [< -126]), (4) Hjorth parameters [52] - Activity (Sample Variance), (5) Sample Absolute mean value, (6) The sample root-mean-square (RMS) value, (7) Hjorth parameter — Mobility, (8) Hjorth parameter — Complexity, (9) Shannon Entropy — using a user-specific approach, (10) Matlab's "approximateEntropy" function, (11) Number of Fast Fourier Transform (FFT) points over a predefined threshold, (12) Number of zero Crossings, (13) Unique function that uses the FFT points to estimate signal bandwidth and (14) PDF center bin (0) value. The features are numbered as per the associated column in the feature vector $X = \{x_1, x_2, \ldots, x_{14}\}$. The slight differences in the WSN and GPS signals' calculated features examine how the developed low-order feature set reacts to different numerical ranges of data, which is an important characteristic to determine.

The PDF features are determined from the calculated PDF (examples are visualized in Figs. 5 and 13), where areas are calculated using trapezoidal numerical integration. Each time-domain feature is determined from the specific set of I/Q samples being analyzed, where the feature is determined for each channel and the average taken. The Hjorth parameters use the 1250 samples as $y[n]$ and are determined based on the equivalent discrete-time equations developed in [52]. The FFT functions are unique in this paper. A 2048 point FFT, as this is the next power of 2 above the length of samples being analyzed (1250 samples), is applied. The function related to the number of FFT points over a predefined threshold calculates the complex point's absolute value and compares it to a predefined threshold. The number of points that surpass this threshold is the final result. This paper's threshold corresponds to 30, which was experimentally determined. The function for estimating the bandwidth was developed as a user-defined function where the primary goal is to distinguish larger bandwidths from narrower ones. The function takes the calculated FFT's absolute value and then analyzes the raw data to determine a single sided spectrum of data. The raw data is analyzed to find the summation of the values from a predefined start point to the outermost point of the FFT. In this study, the start point is fixed at 175 when the FFT length of the single side representation is 1024.

The independent ensemble decision-tree Random Forest approach [44] produced effective results in the simulations in Section 3. However, the simulations were the ideal case without numerical or hardware limitations. As a result, more advanced approaches that are known to outperform independent ensemble models are legitimately required. As part of the work in [13] two optimized multi-class machine learning approaches were developed based on the above features, which were XGBoost [53,54] and a DNN. These approaches applied the concept that feature-based machine learning approaches are the preferred real-world deployment option for signal classification [55]. As single classifier predictions result in low generalization to unseen data, more modern machine learning approaches define the implementation strategy. XGBoost was chosen as dependent ensemble methods are known to outperform independent ones [44] and boosting with regularization achieves the desired performance in this study. DNNs are developed and compared to the low-complexity solutions to determine if there is any loss of performance by not implementing modern neural network approaches. Based on these details and achieved high accuracy and generalization when classifying ISM band wireless signals, XGBoost and DNNs are applied in this live data study. The optimal XGBoost hyperparameters are specified in Table 7, while the DNN structure is provided in Table 8. Due to these developed models' achieved accuracies and ability to generalize to unseen data, the two approaches are applied here as the base model to develop the WSN machine learning diagnostic tool for interference detection and classification. Later in the study, the previously designed signal classification approach will be combined with the models developed in this paper to establish a WSN edge device diagnostic framework. SVM models are investigated, based on the simulated study, for high speed and low complexity detection, enabling jamming classification when jamming is detected. The overall diagnostic methodology is visualized in Fig. 10, where the received I/Q samples are deconstructed into the set of 14 features and passed to the trained classification models.

This concept is based on the hypothesis that I/Q samples are always available to a functioning receiver at the edge, while analysis focuses on the presence of a legitimate signal (ZigBee for WSNs and receiving four or more satellites for GPS applications). If an erroneous packet or less than four satellites is received, the samples can be passed to the interference detection algorithm for jamming detection and/or classification. For this, a database of legitimate signal data and combinations of legitimate signals and jamming signals is required. This approach requires high accuracy and model generalization to unseen data while simultaneously achieving low complexity and low order features, fast optimization and prediction times and a computation fingerprint that is as small as possible. As a result, a machine learning diagnostic tool is developed for use on low-power resource-constrained edge devices. Each model is trained and tested using the available Python3 libraries running on an Intel i7-9700 3 GHz CPU. Additionally, the DNNs are developed using Keras and TensorFlow on an Nvidia GeForce RTX 2060 graphical processing unit (GPU) with 6 GB of RAM. Notably, this study demonstrates that less complex machine learning approaches can match the performance of DNNs, but for a small fraction of the required time and resources when sufficiently detailed features are applied. A vital characteristic to observe relates to the designed models. As the data

Table 7
Optimized XGBoost hyper-parameters.

Parameter:	Value:
No. decision trees	5
Learning	0.8
Max. tree depth	10
Min. child weight	2
Data used	95%
Booster	gbtree
Column sub-sampling	75%
Min. Loss reduction (gamma)	0.5

Table 8
Optimized DNN structure: TensorFlow/Keras.

Layer type	Layer size	Activation function
Input	14 neurons	relu
Fully connected	50 neurons	relu
Fully connected	34 neurons	relu
Fully connected	17 neurons	relu
Output	No. classes	softmax

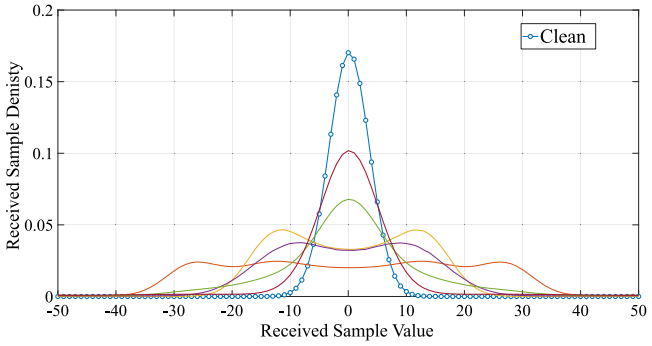

Fig. 13. The probability density of the raw received I/Q samples collected under a range of interference conditions and signals, with the clean case marked.

in this study are collected in a typical domestic operating environment under typical wireless channel fluctuations, the novelty is the designed methodology and diagnostic framework, rather than the trained and tested machine learning model.

5.1. GPS results

Initially, the focus is on GPS signals as their associated low received power levels, approximately −125 dBm, are comparable to signal classification as received signals resemble noise, where relatively low powered jammers can readily block satellite reception. A relativity simple feature set focused on the calculated PDF can determine the presence of unwanted signals. This approach was successful in a previous study [28] focusing on the Random Forest machine learning approach, a low complexity feature set and data from a subset of available satellites. The typical unimodal shape of the GPS signals (as shown in Fig. 13) becomes distorted as the jamming power increases. However, if a more advanced feature set is implemented, the jamming signal can be classified. For this process, the previously developed optimal machine learning models and associated features in [13] will be used to investigate whether a GPS jamming detection and classification approach can be designed. Data was collected for clean GPS signals and GPS signals in the presence of CW and O-QPSK modulated signals (ZigBee).

The designed GPS interference detection process is in two stages, a low-complexity linear binary classifier for detection and a multi-class classification model for signal classification. Low complexity and minimum response times (prediction and optimization) were the essential design requirements. To validate the developed approach, a DNN was designed and compared against less complex SVM and XGBoost approaches. This work expands on the previous study [28] by looking at satellite data across a full 24-hour period (31 satellites in total) and classifying the jamming signal.

The two-stage approach of detection and classification was the proposed design strategy, as jamming signal classification should only be implemented once interference is detected, to minimize computational use on resource-constrained devices. This investigation applies an ISM RF band wireless signal classification algorithm, based on received I/Q samples, to GPS signal jamming detection and classification. However, clean GPS data and associated data for jamming scenarios at 1.57542 GHz was required. The XGBoost hyperparameters as per Table 7, were applied and the DNN structure in Table 8 is applied with two output neurons for detection and three neurons for classification (Clean, CW and O-QPSK). GPS signals can leverage the full previously developed scheme as GPS signals resemble noise under typical operating conditions.

The binary jamming detection and multi-class classification results are provided in Table 9. The test data can be classified as unseen since the data was collected over 24 h and used several different power levels for the two jamming signals. The data were randomly allocated to training and testing in the conventional 80 : 20 split. Each received data grab was analyzed for at least forty different time segments, where both the GPS signals and jamming signals were time-varying. Hence, each data segment analyzed is a unique set of data points as no specific time instance is analyzed twice. Thus, the data instances are mutually distinct. This data analysis meant that the developed models were not dependent on any specific part of the received data outputted from the RTL-SDR dongle. Additionally, as the data instances were sufficiently randomized before allocation to either training or testing, the testing data was unseen during training.

The detection results clearly show that the designed approach can detect variations from the expected received signal even when Pluto SDR jamming signals incurred a "−70 dB" gain. At those low jamming levels, a few satellites can still be received. It was envisaged that the developed SVM would be the initial detector, where the radial basis function was determined to be the optimal kernel. However, the most accurate overall model was the developed XGBoost, which outperformed the SVM and DNN while simultaneously attaining the fastest training and average prediction times. Hence, a one-stage XGBoost approach could be implemented for this GPS interference detection problem. Notably, these models used the optimal parameters of the ISM band signal classification models. Due to the achieved accuracy and less computational resources compared to the DNN, it was concluded that this XGBoost model was optimal for this GPS investigation. The XGBoost models were investigated further by implementing 50-fold cross-validation for both the binary and multi-class models. The mean accuracy score of 50 different models was 99.9949% and 99.99489% for the binary and multi-class models, respectively. The corresponding standard deviation of the 50 different models' accuracy score was 0.02496% and 0.02499%, respectively. Both model's deviation is very low, meaning that it is unlikely that either model is overfitted.

These GPS results proved that the developed features in [13] are not confined for use in the 2.4–2.5 GHz RF band. Additionally, the ability for the developed, optimized models (Tables 7 and 8) to be transferred to new data and be useful for jamming detection and classification bodes well for use in WSN applications. Overall, this GPS signal investigation was a bridging investigation between the previously designed signal classification approach and the WSN jamming detection and classification. This GPS exploration proves that the concept of focusing on raw received I/Q samples for low complexity interference detection is a solution that has value, given a descriptive feature set and optimal model hyperparameters. Furthermore, this GPS examination proved that the 14 developed features also work for ranges larger than [−1:+1] as the GPS signal data was analyzed in the received range of [−127:128]. This result adds another layer of evaluation to the developed feature set and the usefulness of these features in received signal analysis.

Table 9
GPS jamming detection and classification results.

	Training time	Prediction time (ms)	Accuracy (%)	No. errors	No. test points	Model size (kB)
Binary detection						
SVM (RBF)	1067.82 ms	0.118	99.99	1	9808	22
XGBoost	45.85 ms	0.05745	100	0	9808	3
DNN	14719 s	22.4	99.98	2	9808	152
Multi-class classification						
XGBoost	103.72 ms	0.061	99.98	1	9808	11
DNN	15456.67 s	24.82	99.94	6	9808	153

5.2. WSN results

The designed GPS approach cannot be applied to WSN signals due to the received power levels, which are much higher in comparison. This observation is why the simulation study was critical, as it contributed to the development of the WSN interference detection strategy. The simulations were the ideal scenario, which operated without hardware restrictions and the data needed for jamming identification and classification was identified. The simulation results specified that the data needs to be a combination of legitimate and jamming signals to accurately train the interference detection model and identify the differences between error-free and jammed operation. The 14 features previously developed and extracted in [13] are explored to determine whether the new data can be leveraged with an existing ISM band signal classification scheme. Furthermore, by focusing on the same features for multiple models and classification, an overall diagnostic strategy can be formulated. In the subsequent model development investigations, low complexity and minimized response times (prediction and optimization) are the essential design requirements. As a result, where optimizations can be made over the parameters in Tables 7 and 8, the lower complexity XGBoost approach will be examined.

The simulation results in Section 3.4 established the detection methodology as a two-phase detection process. This process is visualized in Fig. 10 and implements a data pipeline approach. The first distinct model's output is used to decide whether the second multi-class model is applied to the input signal. This approach saves time and energy as the binary classification model is implemented initially when a packet is received with an error and used to activate the multi-class classifier, as required. The following results (and implementation study) demonstrate the developed diagnostic framework and feature set's usefulness. The collected data was split into training, validation and testing, which allows an estimate of the error rate on new cases, known as the generalization error (or out-of-sample error), to be found.

For the WSN jamming detection and classification study, several different situations were investigated using the 14 features and optimized models mentioned above. These different studies aimed to develop a diagnostic strategy for edge devices that can operate under several conditions. The work included the five steps mentions below, which will be investigated separately before being combined into an overall edge device interference diagnostic tool.

- Legitimate node classification (Radio Classification)
- Legitimate XBee node vs. non-legitimate SDR classification, where both signals have the same spectral image (Fig. 2)
- Artificial jamming of legitimate XBee node data
- SDR transmitted ZigBee live wireless jamming
- Live subtle CW jamming of commercial XBee nodes

The first concept investigated was whether the 14 features could identify individual commercial nodes (radio identification), which are DIGI XBee nodes in this study. This classification would allow a malicious commercial node to be identified if transmitting data in the network. Since the 14 features were designed to identify legitimate ZigBee transmissions, the results provided in Table 10 are as expected. The DNN and XGBoost models are compared here as this is a multi-class classification, where a single model was desired. The original model for the XGBoost outperforms the DNN and optimizing the DNN would increase the training, optimizing and prediction times, along with the overall model complexity. As a result, the optimum XGBoost model was determined to use 500 decision tree estimators, a learning rate of 0.4, a maximum tree depth of 20, a minimum child weight of 1, used 95% of available data per tree, used the "gbtree" booster, sub-sampled 90% of the feature columns and applied a minimum loss reduction of 0. This optimal model only achieved a 5.74% reduction in the model error. This error reduction is smaller when a 50 Fold cross-validation is applied, where the optimal approach achieves 71.4988% accuracy compared to 66.8211% when using the parameters as per Table 7. The confusion matrix for this approach is provided in Fig. 14(a) where the classifier fills every classification possibility. These features cannot develop a radio classification model, but the results suggest that legitimate radios may be classified, as the results indicate the 14 features do characterize the commercial nodes together.

As the individual XBee could not be individually identified with sufficient accuracy, the next phase aimed at classifying legitimate (XBee Node) and non-legitimate (SDR) nodes. This method involved using data from 5 individual XBee nodes, which transmitted SenseHat data and network operation signals, such as, for example, acknowledgments, and SDR ZigBee transmissions using SenseHat and random data. The ZigBee signals have the same spectral visualization, as shown in Fig. 2 and the SDR deploys the ZigBee frame (Table 1). The signals have similar packet structures and spectral shapes, but the SDR devices are not explicitly designed for WSN ZigBee operation. The 14 features extracted in [13] for WSN signal classification can be used to enhance diagnostics on edge devices by implementing this malicious node identification tool. The classification results are provided in Table 10 where the SVM, DNN and XGBoost models are compared. The XGBoost model, using the parameters in Table 7, outperformed the DNN and SVM, where the RBF kernel was determined to be the optimal approach. As the XGBoost achieved the best accuracy, it was further investigated to see if the error could be reduced. The optimum XGBoost model was determined to use 200 decision tree estimators, a learning rate of 0.6, a maximum tree depth of 5, a minimum child weight of 1, used 75% of available data per tree, used the "gbtree" booster, sub-sampled 75% of the feature columns and applied a minimum loss reduction of 0. These results show that the SDR can be identified, with relatively low error, from amongst the commercial XBee nodes as a malicious external node. The confusion matrix for this classifier is shown in Fig. 14(b). Using the positive result as an SDR transmission, the optimal classifier incurs two false positives and four false negatives. These results indicate a sensitivity of 0.993 and a specificity of 0.998, where sensitivity refers to the ability to correctly detect a positive result (non-legitimate signal) in the received I/Q samples and specificity characterizes the ability to correctly identify (reject) a legitimate ZigBee signal. As these classification concepts approach 1, the more optimal the developed classifier. However, the received samples do portray similar properties, as shown in Fig. 15, and spectral shape (Fig. 2). As a result, to overcome network failure under jamming, these SDR ZigBee signals can be used to examine the jamming and ZigBee signal interactions in live over-the-air transmissions to gain the required insights for commercial ZigBee signals. Additionally, it suggests that a matched signal attack could be detected and classified, as it is envisaged that matched interference attacks would be implemented using a type of SDR. Furthermore, as the small deviations between the SDR samples and XBee samples are identified by this feature set, subtle interference can be detected.

Before investigating the live signal jamming scenarios using the SDRs to transmit ZigBee signals continuously without the need to be

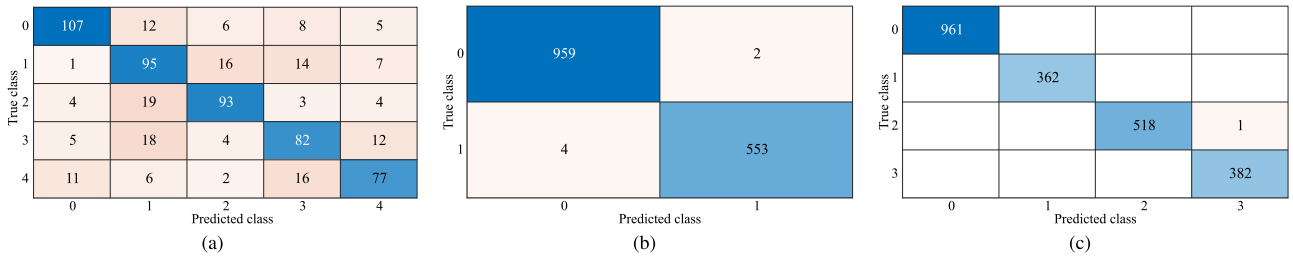

Fig. 14. The Confusion Matrices for the optimal XGBoost approach for (a) The XBee Node radio identification, where the predictors 0–4 correspond to a specific XBee node. (b) The Legitimate ZigBee Signal Identification where predictor 0 is the XBee Signals and 1 is the SDR ZigBee signals. (c) Artificial Jamming Signal Classification where the predictors are (0) ZigBee XBee (1) CW Jammed (2) WiFi Jammed (3) Matched Signal Interference.

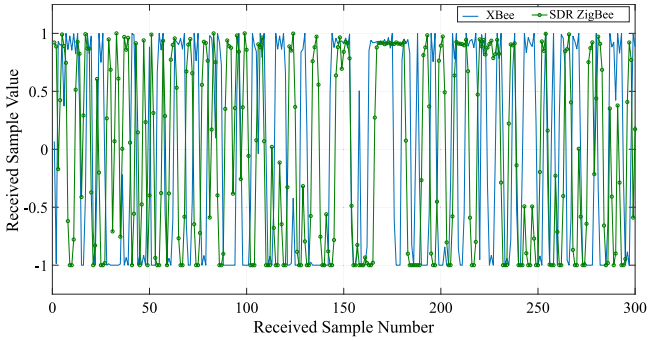

Fig. 15. A comparison between received samples from a ZigBee signal transmitted from a XBee device and a SDR.

Table 10
XBee node classification results.

	Training time	Prediction time (ms)	Accuracy (%)	K fold CV	K fold Acc. (%)	K fold Std. (%)
XBee node identification — 627 test points						
XGBoost	43.88 ms	0.064	66.67	50	66.82	7.91
DNN	909.27 s	23.17	55.66	5	53.05	1.765
XGBoost (Optimal)	1600.61 ms	0.216	72.4	50	71.49	5.53
Legitimate ZigBee signal — 1518 test points						
SVM (rbf)	1162.16 ms	0.143	96.9	50	97.8	1.15
XGBoost	477.38 ms	0.098	98.35	50	98.07	1.183
DNN	1778.24 s	27.54	95.52	5	95.45	0.77
XGBoost (Optimal)	210.44 ms	0.09	99.60	50	99.31	0.708

connected to a network, an "Artificial" jamming approach was developed. Previously collected over-the-air I/Q samples for XBee ZigBee, WiFi, CW and SDR ZigBee signals were utilized to jam the legitimate ZigBee data in software. This technique involved randomizing the previously collected XBee data and adding another previously collected signal to the received ZigBee I/Q samples segment. These signals included CW, WiFi and SDR transmitted ZigBee data acting as a matched interference signal. All of the data was collected through over-the-air live experiments in a typical operating ISM RF band environment where the number of connected devices, demand and services in operation can all change. The Pluto SDR receiver's maximum values were maintained by limiting all "jammed" samples to the region of [−2048:2047] or [−1:+1] when scaled. The investigation results are provided in Table 11, where each investigated model performs well for the jamming detection and classification. As this artificial jamming examination utilizes the original 14 features, the assessment of these features improves, as they have been adapted to WSN jamming detection.

In terms of jamming detection, as simulation results indicated, the SVM approach using the RBF kernel is the optimal model for both performance and complexity. However, the difference in performance between the SVM, XGBoost and DNN models is minimal. The XGBoost provides the quickest average prediction time, but overall the SVM is the optimal approach in terms of the trade-off between performance and implementation times. The DNN slightly outperforms the XGBoost model for jamming signal classification. However, the XGBoost model was investigated for a more optimal approach to determine if the lower complexity XGBoost could achieve the same results as the DNN for this classification problem. The examination produced the optimal XGBoost as using 25 decision tree estimators, a learning rate of 0.7, a maximum tree depth of 10, a minimum child weight of 1, used 95% of available data per tree, used the "gbtree" booster, sub-sampled 90% of the feature columns and applied a minimum loss reduction of 1. This optimal XGBoost outperforms the developed DNN and produces the results for reduced computation and time requirements, which is critical for edge device operation. These results indicate that the 14 features should be applicable to jamming detection in a live experimentation and the classifier's confusion matrix is specified in Fig. 14(c). This confusion matrix outlines that classifier performs perfect classification between non-jammed and jammed conditions, resulting in a sensitivity and a specificity of 1.

This artificial jamming scenario was crucial as it provided additional evidence that the designed feature set could be applied to WSN jamming detection and classification. Furthermore, it evaluated the required data outlined in the simulation study and provided a glimpse into what the signal interactions would resemble in live experiments. In contrast to the two-stage approach indicated by the simulations, this artificial jamming study suggests that a single multi-class model (Table 7) can be implemented without incurring much of a loss in accuracy. Simultaneously, the single XGBoost model achieves a faster prediction time than the binary SVM detection and any two-stage approach. These results indicate that the 14 features outperform the original simulation features even with hardware restrictions, potentially discovering the real-world differences that were a simulation limitation. Notably, these results motivate live over-the-air experimentation and outline expected types of signal interactions (Fig. 16(a)). Finally, this "artificial" jamming is justified as the signal data are not collected simultaneously, so the sample interactions are random. As a result, this approach imitates the random signal interactions in over-the-air transmissions, where different packet segments can interact, over time, with different jamming signals. This procedure is mainly investigating reactive high power jamming where the legitimate signal is sensed before transmitting a jamming signal. Hence, the signals get the opportunity to interact before being received. It also examined constant jammers where the legitimate signal has enough power to transmit through the interference but is received with errors.

As the artificial jamming allowed high-powered jamming to be examined and indicated successful model development, a live over-the-air approach was warranted. Enough power in the jamming signal disrupts legitimate ZigBee network operation. Reactive, which only jams once a legitimate signal is detected in the channel, and constant jamming approaches are investigated. As a result, an efficient method for the continuous transmission of signals that neglected the presence

Table 11
Artificial jamming results.

	Training time	Prediction time (ms)	Accuracy (%)	K fold CV	K fold Acc. (%)	K fold Std. (%)
Jamming detection — 2224 test points						
SVM (RBF)	359.04 ms	0.125	99.96	50	99.83	0.28
XGBoost	478.72 ms	0.0785	99.91	50	99.73	0.341
DNN	2269.19 s	22.14	99.87	5	99.76	0.1146
Jamming signal classification — 2224 test points						
XGBoost	253.3 ms	0.06	99.595	50	99.62	0.567
DNN	2320.94 s	22.22	99.73	5	99.6177	0.1566
XGBoost (Optimal)	689.3 ms	0.088	99.96	50	99.72	0.4093

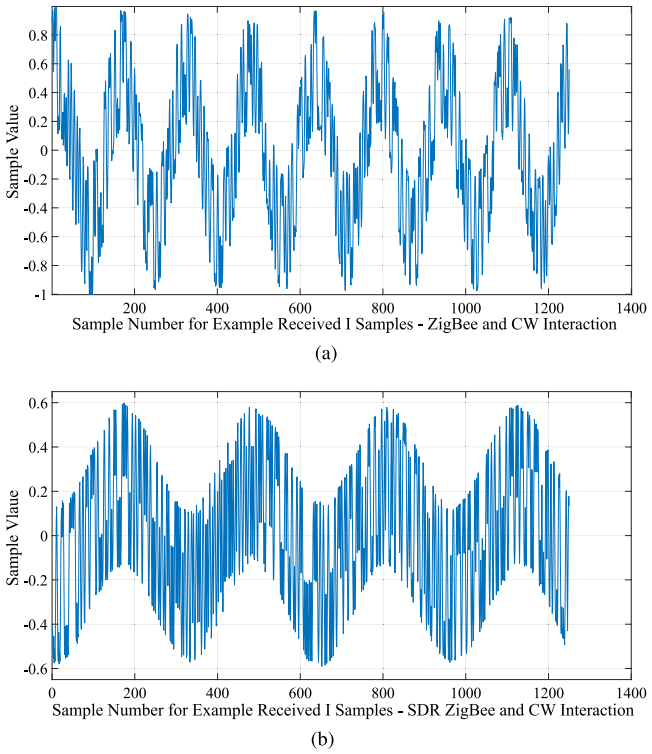

Fig. 16. An example of a ZigBee signal interacting with a CW Jamming signal for (a) The artificial jamming approach involving XBee signals and (b) The live SDR ZigBee and CW jamming investigation. Both approaches indicate a similar interaction of the ZigBee signal in each case includes a CW trend.

of jamming was required to gain access to the required data of O-QPSK (ZigBee) transmissions interacting with other signals. This method, as mentioned earlier, utilized SDRs to transmit the required ZigBee signal structure. Even though the SDR and XBee ZigBee signals can be classified from each other, this investigation allows for an initial insight into how the O-QPSK signals interact with jamming signals in a live operating environment. The SDR jamming results followed what was discovered during the "artificial" jamming investigation. The similarities are visualized by examining an example of the I data resulting from the ZigBee (O-QPSK) and CW interactions. This visualization is provided in Fig. 16, where Fig. 16(a) specifies the artificial situation and Fig. 16(b) visualizes the live SDR investigation. These results indicate that the XBee ZigBee/jamming signal interactions in the artificial scenario mirror the interactions observed in the live over-the-air SDR ZigBee/jamming signal interactions, but at higher jamming powers. This indication validates investigating SDR transmission and it can be hypothesized with sufficient confidence that the XBee over-the-air interactions would correlate with this SDR method.

The collected data included SDR transmitted signals with no jamming signal and in the presence of CW and matched signal interference. The SDR jamming signal power gains varied from −55 to −34 dB on the Pluto SDR, where the CN0417 power amplifier provided an additional 20 dB gain to the Pluto SDR output, approximately. The power amplifier was implemented to mimic typical scenarios where a power amplifier would be necessary to attack a sufficiently large network area. These signal powers were sufficient to cause signal interactions while not being too high so as to block all transmissions. As a result, this examination occupies the subtle and low-power jamming region, which is more difficult to detect than the high impact jamming that blocks all signals due to the power levels in operation. For this investigation, clean SDR ZigBee data is used as the "Good ZigBee" data since the results would be skewed if XBee data were used due to the previous SDR/XBee ZigBee classification results.

The need for a new model was validated by using the previously developed ISM band signal classification model on the collected data. This model was trained with SDR and XBee ZigBee instances. Nearly all of the SDR jammed data was classified as ZigBee due to subtle jamming being implemented and the signal classification being built upon distinct signals. As a result, a model based on signal interactions is needed and this SDR approach provides the most efficient solution. The collected SDR data is randomly assigned to training and testing in the ratio of 80:20, respectively, and is labeled as either clean, CW or matched interference data. As the data is collected as per Section 4.1, the SDR receives all data on the required channel during data collection. All of the data collected during the various jamming scenarios was labeled as jammed. However, some instances may be mislabeled as the Pluto SDR jammer stops transmitting briefly when changing gain or frequency values. As a result, some instances may not include jamming or only include some small jamming power that results in the received signal being error-free. However, this examination provides sufficient evidence that the developed features and models can detect jamming signals and generalize to new data. As the data is collected over time in a live wireless operating environment, each collected signal is unique. Each signal is divided into three segments for analysis and then randomly assigned to training or testing. The overall process results in the testing data being unseen to the training data due to each data segment's unique timestamps and interactions.

The results are specified in Table 12 for both jamming detection and classification. In contrast to the artificial jamming study, the XGBoost model achieved the highest accuracy at 97.57%. As a result, it was investigated to determine if a more optimal approach existed. The optimum XGBoost model for binary jamming detection uses 25 decision tree estimators, a learning rate of 0.2, a maximum tree depth of 20, a minimum child weight of 1, used 95% of available data per tree, uses the "gbtree" booster, sub-sampled 90% of the feature columns and applied a minimum loss reduction of 5. There was a trade-off for this optimal XGBoost model as a slightly more accurate (0.054%) model uses 250 decision tress, but the 25 decision tree approach was chosen for a less complex approach. A similar trend was discovered in the jamming classification since, of the two original models, the XGBoost outperformed the DNN and a more optimum approach existed. The optimal XGBoost setup for jamming classification (Clean, CW or matched interference) uses 25 decision tree estimators, a learning rate of 0.2, a maximum tree depth of 20, a minimum child weight of 1, used 95% of available data per tree, used the "gbtree" booster, sub-sampled 90% of the feature columns and applied a minimum loss reduction of 0. A slightly more accurate model (0.05%) uses 50 decision trees, but to keep the approach less complex, the 25 trees were chosen. The resulting confusion matrices for the binary and multi-class SDR classifiers are provided in Figs. 17(a) and 17(b), respectively. Both classifiers exhibit high performance in terms of identifying the presence of interference (positive case) and rejecting legitimate clean signals (negative case). The jamming detection model results in a sensitivity of 0.996 and a

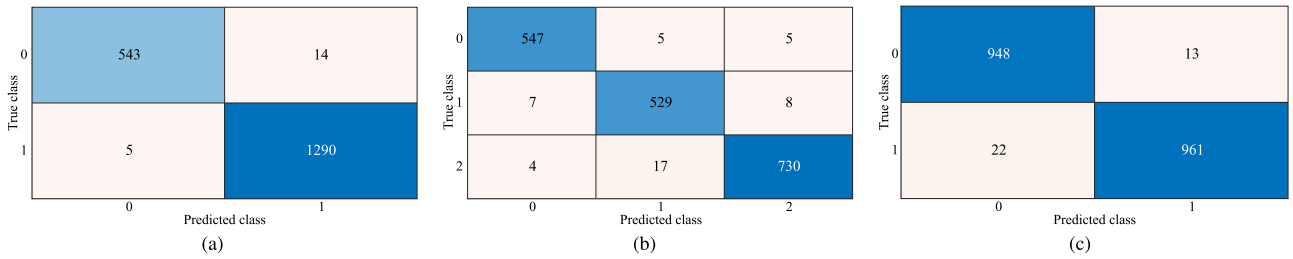

Fig. 17. The confusion matrices for the optimal XGBoost approach for (a) The SDR ZigBee Jamming detection where the predictor are (0) ZigBee Clean, (1) Jammed (b) The SDR ZigBee Jamming Classification where the predictors are (0) ZigBee Clean, (1) CW Jammed (2) Matched ZigBee Interference (c) XBee CW Jamming Detection where the predictors are (0) ZigBee XBee (1) CW Jammed.

Table 12
Over-the-air SDR jamming results.

	Training time	Prediction time (ms)	Accuracy (%)	K fold CV	K fold Acc. (%)	K fold Std. (%)
Jamming detection — 1852 test points						
SVM (RBF)	1817.24 ms	0.145	95.68	50	95.42	1.56
XGBoost	488.78 ms	0.062	97.57	50	96.52	1.406
DNN	2784.2 s	23.5	96.38	5	95.005	0.594
XGBoost (Optimal-250)	1046.22 ms	0.0743	99.03	50	97.65	1.42
XGBoost (Optimal-25)	93.749 ms	0.0705	98.97	50	97.46	1.32
Jamming signal classification — 1852 test points						
XGBoost	39.89 ms	0.0495	95.57	50	95.25	1.817
DNN	2619.56 s	22.22	95.03	5	92.97	0.917
XGBoost (Optimal)	481.71 ms	0.054	97.52	50	96.637	1.44

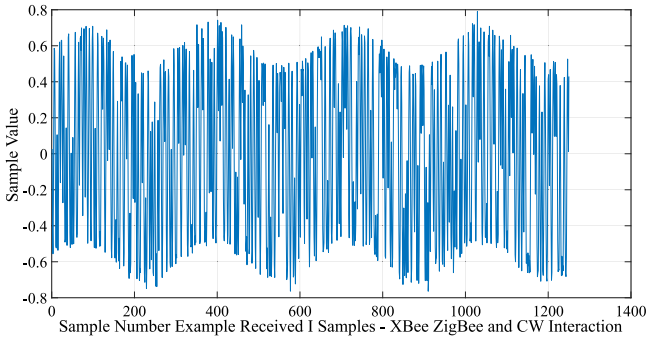

Fig. 18. An example of a ZigBee signal interacting with a CW Jamming signal for the live XBee ZigBee and CW jamming investigation, which incurs a similar interaction to that observed during the artificial and SDR jamming situations.

specificity of 0.975, which means that the classifier can correctly detect interference and reject legitimate ZigBee signals.

This study's final aspect is the achievable low power CW jamming in an active ZigBee network consisting of XBee nodes. The data collected in this approach is sparse and inefficient as the network reacts to the jamming signals. CW jamming was chosen as the simulations predicted this jamming approach to be less effective than matched interference, resulting in a higher chance of collecting the necessary data using the SDR. An example of the collected I data, labeled as a CW interference signal interaction, is shown in Fig. 18, which indicates a similar trend to the artificial and SDR jamming approaches in Fig. 16. This data strategy was implemented to identify the signals that incurred a CW interaction. As the number of packets received at the receiver was tracked (on the Raspberry Pi), it was known that some good signals were being transmitted and those I Data segments followed the previously developed signal classification data [13]. This is in contrast to the SDR approach as there was no packet monitoring during that investigation. The ZigBee packets were received as the SDR jammer briefly stops transmitting due to the Matlab functions being used. A set of CW jammed XBee data was collected and used to develop a model based entirely on XBee data and as testing data for the model designed using SDR ZigBee data. The previously developed XBee data was employed as the "good" ZigBee data and an additional test dataset was collected as "unseen" data. The results for the XBee models are specified in Table 13. The optimal determined XGBoost model uses 200 decision tree classifiers. However, an approach that uses only five decision trees produces a less complex design and only suffers from a 0.1% accuracy drop. As a result, the optimal hyperparameters use 5 decision tree estimators, a learning rate of 0.9, a maximum tree depth of 10, a minimum child weight of 5, 95% of available data per tree, the "gbtree" booster, sub-sample 90% of the feature columns and apply a minimum loss reduction of 0. The resulting confusion matrix that achieves 98.2% accuracy is specified in Fig. 17(c). Using the case that a positive result ("1") indicates a CW jammed reception, the classifier's sensitivity is 0.978 and the specificity is 0.9865. These model results prove that the developed classifier has high performance in detecting jamming and rejecting legitimate ZigBee signals.

The SDR based jamming and unjammed XBee ZigBee data were used to train a new XGBoost model, which was tested using only unjammed XBee and XBee CW jammed data. This model investigated whether the SDR jammed data could be used to detect interference in the XBee data. The optimal XGBoost hyperparameters used five decision trees and the achieved accuracy was 90.62%. This result indicates the usefulness of the SDR data, which can be classified amongst XBee nodes, and the accuracy can be improved with more subtle jamming data. All of the available XBee and SDR ZigBee data were used to train and test another XGBoost model. This method examined if the SDR data's presence reduced the detection accuracy and if the SDR data can be applied as additional data for continuous jamming signals that deny services and block transmission in ZigBee networks. This approach would enable acquiring data for reactive jamming scenarios. An XGBoost model consisting of 25 decision tree classifiers produced an accuracy of 98.34%. These results validate focusing on SDRs for a more efficient WSN jamming signal investigation. Additionally, the result indicates that the SDR approach, where higher jamming data and different jamming signals can be implemented, enhances WSN jamming detection and classification diagnostic tools once available commercial ZigBee (XBee) data is also employed. This insight means more efficient model development is achievable without the loss of accuracy.

This study's developed models provided sufficient evidence that the developed features and model structure are applicable for WSN ZigBee jamming detection and classification. The previously developed signal classification model can be combined with the jamming detection and classification models to create an overall low-complexity WSN edge device interference diagnostic framework based on data that is always available to a functioning receiver. The required implementation algorithm for the edge node is provided in Algorithm 1. This algorithm

Table 13
Over-the-air XBee CW jamming results.

	Training time	Prediction time (ms)	Accuracy (%)	K fold CV	K fold Acc. (%)	K fold Std. (%)
XBee jamming detection — 1237 test points						
SVM (linear)	829.88 ms	0.132	93.93	50	90.6413	3.00
XGBoost	199.43 ms	0.06	94.9	50	96.544	1.70
DNN	17788.88 s	22.275	93.16	5	91.77	1.45
XGBoost (Optimal)	205.42 ms	0.07	98.2	50	96.625	1.572

outlines when each of the designed models should be implemented and why using the same 14 features in each developed algorithm is crucial for the diagnostic framework's overall design. Each model can use the same 14 features as an input, and so it minimizes, to an extent, the computation required at the edge. This algorithm and the developed machine learning diagnostic tools achieve low complexity solutions to edge device diagnostic challenges in WSN applications. The individual machine learning models achieve similar, if not better, accuracy to the approaches introduced in Section 2. For example, the collaborative packet rate analysis system in [18] achieves jamming detection accuracy of over 97% and the chip sequence error pattern approach in [19] achieves accuracies of over 96% for all signals analyzed and an average of 98.29%. In [37], where raw I/Q samples are the main focus, the highest accuracy achieved for the designed signal classification approaches is 98%. As a result, the framework developed in this study uses a less complex methodology, has several applications, is transferable across the RF spectrum, and achieves accuracy comparable to the literature. The novelty surrounds using data continuously available to a functioning receiver and low order features to develop low complexity yet high performance machine learning models. The developed 14 features and associated low complexity models achieve equivalent accuracy and generalization error results as the developed fully connected deep neural networks, but for a small fraction of the computation requirements and in a vastly reduced timescale. This designed diagnostic framework for interference detection enhances WSN edge devices' security and provides multiple detection scenarios, critical for an overall secure wireless system.

Algorithm 1: Edge Node Diagnostic Routine

```
Data: Received I/Q Samples
if Packet Received without errors then
    Run Legitimate Node Classification;
    if Node is legitimate then
        Continue;
    else
        Malicious Signal;
    end
end
if Packet received with errors then
    Implement Jamming Detection Model;
    if Jamming Detected then
        Run Jamming Classification;
    else
        Continue;
    end
end
if No Packets Received then
    Run ISM band channel/Signal classification model;
else
    Other Fault;
end
```

5.3. Implementation

In this study, a subset of the developed supervised machine learning algorithms was implemented on a Raspberry Pi embedded device with 1 GB of RAM and a quad-core Broadcom Arm Cortex A53 processor (1.4 GHz). This Raspberry Pi utilization is the initial implementation step required to achieve edge device operation. The Raspberry Pi can run python code and all required machine learning libraries for XGBoost. The data for training and testing the models were stored on the Raspberry Pi device. The jamming detection and classification results for the artificial jamming and SDR over-the-air scenarios, along with the XBee CW jamming detection, were investigated. The optimal XGBoost models were applied in each case and the results are provided in Table 14, where, for each model, similar accuracies were achieved, due to differences in multi-threading. However, the training and average prediction times incurred substantial increases. These increases in training and prediction times would be much more considerable for a DNN, but, generally, training times are not a concern as it can be rectified by training and optimizing the model on a much more advanced machine. Only the optimally trained model must be uploaded and used on the lightweight embedded device, but the prediction time and required computational resources are important factors. A few ms differences can be significant factors for real-time decisions, and lightweight models are needed for resource-constrained devices, where energy usage needs to be optimum. The Raspberry Pi was chosen as it is an example of how low cost hardware has advanced over the past decade or so. As we look to the future, it is not unreasonable to suggest that edge devices will have similar specifications.

Table 14
Raspberry Pi XGBoost model results.

Training time (ms)	Prediction time (ms)	Accuracy (%)
Artificial jamming detection		
939.82	12.92	99.78
Artificial jamming classification		
16831.39	14.52	99.69
SDR jamming detection		
1569.59	0.789	98.596
SDR jamming classification		
5163.98	0.791	97.08
XBee CW jamming detection		
157.03	0.85	97.99

Furthermore, implementing machine learning at the edge on resource-constrained devices is feasible due to advances in hardware, like the Raspberry Pi implementation, and low numeric approaches for machine learning. In [56], customizable hardware architectures, such as field-programmable gate arrays (FPGAs) provide opportunities for data width specific computation by implementing unique logic configurations. As a result, highly optimized processing, that is unattainable by full precision networks, is achieved. The techniques gained from such low numeric experimentation will enable machine learning at the edge. Low precision networks, which suit computationally constrained devices, typically incur a classification accuracy penalty. However, this can be recovered through increased computation [56]. As a result, this study and the developed diagnostic framework is a feasible solution for WSN applications and wireless GPS edge devices.

Overall, the simulations, the GPS investigations and the several ZigBee over-the-air data examinations have provided sufficient evidence that the developed low order features, and associated models, can detect and classify interference in WSN and GPS applications. The low-order feature set based entirely on analyzing received I/Q samples is novel and several models can use the same feature set as an input to develop a diagnostic strategy. More data is required to realize the potential of the developed framework fully. However, for a typical domestic environment, this study's results have demonstrated the effectiveness of the designed methods, which differ from the literature by only requiring access to raw received I/Q samples, permitting independent device decisions and using low-order statistical features. Despite the trend to use deep learning approaches, this study employed potent data analysis and signal processing to prove that traditional techniques effectively facilitate parsimonious interference detection and classification.

6. Conclusion

This study employed low-order features extracted entirely from received I/Q samples to develop a WSN interference diagnostic framework for resource-constrained edge devices. Matlab Monte Carlo simulations provided the initial evidence that I/Q samples can be employed to detect interference in ZigBee signals. These simulations involved CW, matched, WiFi and thermal noise interference in the ideal case, where no hardware restrictions were in operation. The developed simulations concluded that both jamming detection and signal classification were achievable and the required data was ZigBee and jamming signal interactions. These simulations implemented a smaller feature set, due to the lack of hardware restrictions, resulting in the PDFs and I/Q samples having no numerical limits. SDRs, XBee commercial ZigBee nodes and Raspberry Pi embedded devices were employed to design hardware testbeds and the overall data strategy to investigate live over-the-air signals. Both WSN ZigBee and GPS signals were investigated using an SDR as the jamming device, transmitting both CW and matched ZigBee interference.

The live signal investigation used a previously designed low order feature set [13] that extracted features directly from the time, frequency and PDF analysis of received I/Q samples. These features used the simulations as a base but needed expansion to overcome the hardware limitations. The optimal XGBoost and DNN architectures from the previous study were leveraged in this jamming detection and classification study to increase efficiency and begin diagnostic framework design. The XGBoost algorithm outperformed or matched the DNN in each case for a small fraction of the required time and computational resources. This insight proved that traditional feature-based methods are still fit for purpose, particularly for low complexity solutions, and achieve high performance when potent data analysis and novel descriptive features are applied.

Accurate XGBoost models were developed for legitimate XBee node vs. non-legitimate SDR classification, artificial jamming of legitimate XBee node data, SDR transmitted ZigBee live wireless jamming and live subtle CW jamming of commercial XBee nodes. Additionally, GPS jamming detection and classification models proved that the framework is transferable across the RF spectrum. The designed models were combined to develop an edge device low complexity, low order WSN interference diagnostic framework, which is the main contribution of the study. The framework provides multiple detection scenarios, using the same features in each case, and enables independent decisions on edge devices as no network-level data is required. This novel framework has low complexity, high accuracy ($> 95\%$) and fast optimization and prediction times, critical for real-time edge device operation.

Future work aims to analyze a more extensive set of potential interference attack styles while collecting data on edge devices in different operating environments. This additional data will enable more accurate models to be developed and increase the probability of generalizing to new operating environments. Model and feature optimization can be further explored to see whether more refined approaches can be obtained for resource-constrained edge devices. However, as the same 14 features were used for various models and classifications in this study, trade-offs might exist when refining the feature set, where similar or better accuracy is achieved for one approach, while a performance degradation is seen in another. As a result, due to the multiple uses of the developed low-order feature set, this work's main future goals focus on applying the designed interference diagnostic framework to edge devices in different operating environments.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The Irish Research Council (IRC) and Raytheon Technologies Research Center, Ireland, support this work under the IRC Postgraduate Enterprise Partnership Scheme, EPSPG/2016/66.

References

[1] L. Atzori, A. Iera, G. Morabito, The Internet of Things: A survey, Comput. Netw. 54 (15) (2010) 2787–2805.

[2] S. Tennina, M. Santos, A. Mesodiakaki, P.-V. Mekikis, E. Kartsakli, A. Antonopoulos, M. Di Renzo, A. Stavridis, F. Graziosi, L. Alonso, C. Verikoukis, WSN4QoL: WSNs for remote patient monitoring in e-Health applications, in: 2016 IEEE International Conference on Communications, ICC, 2016, pp. 1–6.

[3] M. Li, H. Lin, Design and implementation of smart home control systems based on wireless sensor networks and power line communications, IEEE Trans. Ind. Electron. 62 (7) (2015) 4430–4442.

[4] T. Vladimirova, C.P. Bridges, J.R. Paul, S.A. Malik, M.N. Sweeting, Space-based wireless sensor networks: Design issues, in: 2010 IEEE Aerospace Conference, Mar. 2010, pp. 1–14.

[5] P. Katopodis, G. Katsis, O. Walker, M. Tummala, J.B. Michael, A hybrid, large-scale wireless sensor network for missile defense, in: 2007 IEEE International Conference on System of Systems Engineering, Apr. 2007, pp. 1–5.

[6] P.K.D. Pramanik, A. Nayyar, G. Pareek, Chapter 7 - WBAN: Driving e-healthcare beyond telemedicine to remote health monitoring: Architecture and protocols, in: H.D. Jude, V.E. Balas (Eds.), Telemedicine Technologies, Academic Press, 2019, pp. 89–119.

[7] A. Addaim, A. Kherras, Z. Guennoun, Design of WSN with relay nodes connected directly with a LEO nanosatellite, Int. J. Comput. Commun. Eng. 3 (5) (2014) 310–316.

[8] Cisco, Cisco Annual Internet Report (2018–2023), Tech. Rep., 2020.

[9] J. Heo, J. Kim, J. Paek, S. Bahk, Mitigating stealthy jamming attacks in low-power and lossy wireless networks, J. Commun. Netw. 20 (2) (2018) 219–230.

[10] Keysight Technologies, Electronic warfare: Vying for control of the electromagnetic spectrum, Whitepaper (2020) 1–5.

[11] ZigBee Alliance, ZigBee Specification. ZigBee Document 053474r20, Tech. Rep., 2012, [Online]. Available: https://zigbeealliance.org/wp-content/uploads/2019/11/docs-05-3474-21-0csg-zigbee-specification.pdf.

[12] A. Wood, J. Stankovic, S. Son, JAM: a jammed-area mapping service for sensor networks, in: 24th IEEE Real-Time Systems Symposium, Dec. 2003, pp. 286–297.

[13] G.D. O'Mahony, K.G. McCarthy, P.J. Harris, C.C. Murphy, Developing a low-order statistical feature set based on received samples for signal classification in wireless sensor networks and edge devices, MDPI IoT (2021) Currently Under Review.

[14] G.D. O'Mahony, Intelligent Low-Complexity Widely Deployable Diagnostic Tools for Wireless Edge Device Security using Machine Learning (Ph.D. thesis). University College Cork (UCC), Chapter 6, expected publication Autumn 2021 on UCC's CORA.

[15] A. Abduvaliyev, A.-S.K. Pathan, J. Zhou, R. Roman, L.W.-C. Wong, On the vital areas of intrusion detection systems in wireless sensor networks, IEEE Commun. Surv. Tutor. 15 (3) (2013) 1223–1237.

[16] I. Butun, S.D. Morgera, R. Sankar, A survey of intrusion detection systems in wireless sensor networks, IEEE Commun. Surv. Tutor. 16 (1) (2014) 266–282.

[17] D. Liu, J. Raymer, A. Fox, Efficient and timely jamming detection in wireless sensor networks, in: IEEE 9th International Conference on Mobile Ad-Hoc and Sensor Systems, MASS, 2012, pp. 335–343.

[18] K. Siddhabathula, Q. Dong, D. Liu, M. Wright, Fast jamming detection in sensor networks, in: 2012 IEEE International Conference on Communications, ICC, Jun. 2012, pp. 934–938.

[19] K. Wu, H. Tan, H.L. Ngan, Y. Liu, L.M. Ni, Chip error pattern analysis in IEEE 802.15.4, IEEE Trans. Mob. Comput. 11 (4) (2012) 543–552.

[20] F. Hermans, O. Rensfelt, T. Voigt, E. Ngai, L.-Å. Nordén, P. Gunningberg, SoNIC : Classifying interference in 802.15.4 sensor networks, in: ACM/IEEE International Conference on Information Processing in Sensor Networks, IPSN, 2013, pp. 55–66.

[21] S. Grimaldi, A. Mahmood, M. Gidlund, M. Alves, An SVM-based method for classification of external interference in industrial wireless sensor and actuator networks, J. Sensor Actuator Netw. 6 (2) (2017).

[22] M.A. Alsheikh, S. Lin, D. Niyato, H.-P. Tan, Machine learning in wireless sensor networks: Algorithms, strategies, and applications, IEEE Commun. Surv. Tutor. 16 (4) (2014) 1996–2018.

[23] Z. Yu, J.J.P. Tsai, A framework of machine learning based intrusion detection for wireless sensor networks, in: IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, 2008, pp. 272–279.
[24] Z. Xiao, C. Liu, C. Chen, An anomaly detection scheme based on machine learning for WSN, in: 1st International Conference on Information Science and Engineering, ICISE, 2009, pp. 3959–3962.
[25] H. Ayadi, A. Zouinkhi, B. Boussaid, M.N. Abdelkrim, A machine learning methods: Outlier detection in WSN, in: 16th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering, STA, 2015, pp. 722–727.
[26] K.A. Jalil, M.H. Kamarudin, M.N. Masrek, Comparison of machine learning algorithms performance in detecting network intrusion, in: International Conference on Networking and Information Technology, 2010, pp. 221–226.
[27] M.C. Belavagi, B. Muniyal, Performance evaluation of supervised machine learning algorithms for intrusion detection, Procedia Comput. Sci. 89 (2016) 117–123.
[28] G.D. O'Mahony, S. O'Mahony, J.T. Curran, C.C. Murphy, Developing a low-cost platform for GNSS interference detection, in: European Navigation Conference, 2015, pp. 1–8.
[29] N. Rouissi, H. Gharsellaoui, S. Bouamama, A hybrid DS-FH-THSS based approach anti-jamming in wireless sensor networks, in: 2016 World Symposium on Computer Applications & Research, WSCAR, Mar. 2016, pp. 93–97.
[30] J. Ng, Z. Cai, M. Yu, A new model-based method to detect radio jamming attack to wireless networks, in: 2015 IEEE Globecom Workshops, GC Wkshps, Dec. 2015, pp. 1–6.
[31] S.G. Hymlin Rose, T. Jayasree, Detection of jamming attack using timestamp for WSN, Ad Hoc Netw. 91 (2019).
[32] P. Bhavathankar, S. Chatterjee, S. Misra, Link-quality aware path selection in the presence of proactive jamming in fallible wireless sensor networks, IEEE Trans. Commun. 66 (4) (2018) 1689–1704.
[33] V.A. Shanthakumar, C. Banerjee, T. Mukherjee, E. Pasiliao, Uncooperative RF direction finding with I/Q data, in: Proceedings of the 2020 4th International Conference on Information System and Data Mining, 2020, pp. 6–13.
[34] D. Roy, T. Mukherjee, M. Chatterjee, E. Pasiliao, Detection of rogue RF transmitters using generative adversarial nets, in: 2019 IEEE Wireless Communications and Networking Conference, WCNC, Apr. 2019, pp. 1–7.
[35] D. Roy, T. Mukherjee, M. Chatterjee, E. Pasiliao, Primary user activity prediction in dsa networks using recurrent structures, in: 2019 IEEE International Symposium on Dynamic Spectrum Access Networks, DySPAN, Nov. 2019, pp. 1–10.
[36] T. Jian, B.C. Rendon, E. Ojuba, N. Soltani, Z. Wang, K. Sankhe, A. Gritsenko, J. Dy, K. Chowdhury, S. Ioannidis, Deep learning for RF fingerprinting: A massive experimental study, IEEE Internet Things Mag. 3 (1) (2020) 50–57.
[37] J. Fontaine, E. Fonseca, A. Shahid, M. Kist, L.A. DaSilva, I. Moerman, E. De Poorter, Towards low-complexity wireless technology classification across multiple environments, Ad Hoc Netw. 91 (2019) 101881.
[38] G.D. O'Mahony, P.J. Harris, C.C. Murphy, Identifying distinct features based on received samples for interference detection in wireless sensor network edge devices, in: 2020 Wireless Telecommunications Symposium, WTS, Apr. 2020, pp. 1–7.
[39] G.D. O'Mahony, P.J. Harris, C.C. Murphy, Detecting interference in wireless sensor network received samples : A machine learning approach, in: 2020 IEEE 6th World Forum on Internet of Things, Jun. 2020, WF-IoT, pp. 1–6.
[40] B. Stelte, G.D. Rodosek, Thwarting attacks on ZigBee - Removal of the KillerBee stinger, in: Proceedings of the 9th International Conference on Network and Service Management, Oct. 2013, pp. 219–226.
[41] Tektronix, DPX overview, 2019, [Online]. Available: https://www.tek.com/dpx-overview.
[42] G.D. O'Mahony, P.J. Harris, C.C. Murphy, Analyzing the vulnerability of wireless sensor networks to a malicious matched protocol attack, in: 2018 International Carnahan Conference on Security Technology, ICCST, Oct. 2018, pp. 1–5.
[43] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297.
[44] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[45] Raspberry Pi, Raspberry Pi 3 B+. [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/?resellerType=home.
[46] J.T. Curran, C. Fernandez-Prades, A. Morrison, M. Bavaro, The continued evolution of software-defined radio for GNSS, GPS World 29 (2018) 43–49.
[47] Analog Devices, ADALM-Pluto, Software-Defined Radio Active Learning Module. [Online]. Available: https://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/adalm-pluto.html#eb-overview.
[48] Siretta, Delta 15 - Right Angled 2.4 GHz Stubby WiFi/WLAN & Bluetooth Antenna. [Online]. Available: https://www.siretta.com/products/antennas/delta-15.
[49] T. Suzuki, GNSS-Radar, 2014, [Online]. Available: http://www.taroz.net/GNSS-Radar.html.
[50] Nooelec, NESDR SMArTee v2 SDR. [Online]. Available: https://www.nooelec.com/store/nesdr-smartee-sdr.html.
[51] M. Quigley, S. Gleason, P. Abbeel, Fastgps, 2013, [Online]. Available: https://sourceforge.net/projects/fastgps.
[52] B. Hjorth, EEG Analysis based on time domain properties, Electroencephalogr. Clin. Neurophysiol. 29 (3) (1970) 306–310.
[53] T. Chen, C. Guestrin, XGBoost: A scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Aug. 2016, pp. 785–794.
[54] XGBoost Developers, XGBoost Documentation. [Online]. Available: https://xgboost.readthedocs.io/en/latest/index.html.
[55] A.H. Wahla, L. Chen, Y. Wang, R. Chen, F. Wu, Automatic wireless signal classification in multimedia internet of things: An adaptive boosting enabled approach, IEEE Access 7 (2019) 160334.
[56] P. Colangelo, N. Nasiri, E. Nurvitadhi, A. Mishra, M. Margala, K. Nealis, Exploration of low numeric precision deep learning inference using Intel® FPGAs, in: 2018 IEEE 26th Annual International Symposium on Field-Programmable Custom Computing Machines, FCCM, Boulder, CO, Apr. 2018, pp. 73–80.

George D. O'Mahony graduated top of his class with a first-class honors Bachelor's degree in Electrical and Electronic Engineering from University College Cork (UCC) in 2015. He graduated as a UCC Quercus University Scholar and received the Cork Electronics Industry Association and Engineers Ireland Awards for Top Graduate in Electrical and Electronic Engineering 2015. Subsequently, he was chosen the UCC College of Science, Engineering and Food Science (SEFS) Graduate of the year for 2015. After graduation, he worked as a Project Manager in the pharmaceutical industry with PM Group/MSD. Before commencing his Ph.D. in 2017, he worked as a research intern with United Technologies Research Center Ireland. He is currently pursuing his Ph.D. in Electrical and Electronic Engineering at University College Cork. His Ph.D. research focuses on enhancing security on resource constrained edge devices in wireless communication networks. Machine learning techniques are applied to develop low complexity interference diagnostic tools for wireless edge devices. His focus is wireless sensor networks and GPS applications, especially their utilization in Internet of Things applications. He applies support vector machines and decision tree-based machine learning approaches to achieve lowcomplexity solutions, while deep neural networks are also under investigation. This research is in collaboration with Raytheon Technologies Research Center Ireland and the Irish Research Council. He is a student member of the Institute of Electrical and Electronics Engineers (IEEE).

Kevin G. McCarthy obtained the B.E., M.Eng.Sc., and Ph.D. degrees from University College Cork (UCC), Cork, Ireland, in 1982, 1986, and 1992, respectively, with his M.Eng.Sc. and Ph.D. degrees being awarded for research carried out at the National Microelectronics Research Centre (NMRC) at UCC which subsequently became the Tyndall National Institute. He worked at Analog Devices, Limerick, Ireland, in both product engineering and computer-aided design (CAD) engineering roles before returning to the NMRC in 1993 as a research scientist. In 2000 he joined the academic staff of the Department of Electrical Engineering at University College Cork which now forms part of the School of Engineering at UCC. Kevin's research interests include semiconductor device characterization from DC to high-frequency (RF), modeling of semiconductor devices and circuits, and analogue, mixed-signal and RF circuit design and applications. Dr. McCarthy is a member of the Institute of Electrical and Electronics Engineers (IEEE) and Engineers Ireland and has served on the technical program committees of a range of national and international conferences such as the IEEE International Conference on Microelectronic Test Structures (ICMTS), the IEEE International Symposium on Radio Frequency Integrated Circuits (RFIC) and the European Solid-State Device Research Conference (ESSDERC).

Philip J. Harris (Male) is Group Leader, Networks & Embedded Systems, at Raytheon Technologies Research Center Ireland where he leads several research teams in the areas of wireless technology, embedded systems, and software engineering. Dr Phil Harris (BE (Hons), DPhil, CEng, MIET) has over 20 years of experience in scientific & research environments, including 6 years of service at Raytheon Technologies, and 11 years of service at Thales UK, Research & Technology working in systems engineering, software engineering, and algorithm development. Much of this activity has been as technical lead in the area of GNSS signal generation & scenario simulation for the verification of GNSS receivers.

Colin C. Murphy graduated top of his class with a first class honours degree in 1989, an M.Eng.Sc. degree in 1991, and a Ph.D. degree in 2006. After graduation, he worked as a research engineer in the Irish National Microelectronics Research Centre. In 1997, he joined the Department of Electrical and Electronic Engineering, University College Cork, Ireland, as a lecturer. His research interests include co-operative and noncooperative spectrum sensing algorithms for cognitive radio, weak signal acquisition, tracking algorithms for GNSSs and the application of machine learning algorithms to contemporary wireless digital communications challenges.