Title	Intelligent low-complexity widely deployable diagnostic tools for wireless edge device security using machine learning
Authors	O'Mahony, George D.
Publication date	2021-12
Original Citation	O'Mahony, G. D. 2021. Intelligent low-complexity widely deployable diagnostic tools for wireless edge device security using machine learning. PhD Thesis, University College Cork.
Type of publication	Doctoral thesis
Rights	© 2021, George D. O'Mahony. - https://creativecommons.org/licenses/by-nc-nd/4.0/
Download date	2025-07-04 09:02:30
Item downloaded from	https://hdl.handle.net/10468/12505

INTELLIGENT LOW-COMPLEXITY WIDELY DEPLOYABLE DIAGNOSTIC TOOLS FOR WIRELESS EDGE DEVICE SECURITY USING MACHINE LEARNING

George D. O'Mahony

Thesis submitted for the degree of
Doctor of Philosophy

NATIONAL UNIVERSITY OF IRELAND, CORK

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

December 2021

Head of School: Prof Jorge Oliveira

Supervisors: Dr Colin C. Murphy
Dr Philip J. Harris

Research supported by Irish Research Council in collaboration with Raytheon Technologies Research Center, Ireland

Table of Contents

List of Figures

List of Tables

This is to certify that the work I, George O'Mahony, am submitting is my own and has not been submitted for another degree, either at University College Cork or elsewhere. All external references and sources are clearly acknowledged and identified within the contents. I have read and understood the regulations of University College Cork concerning plagiarism.

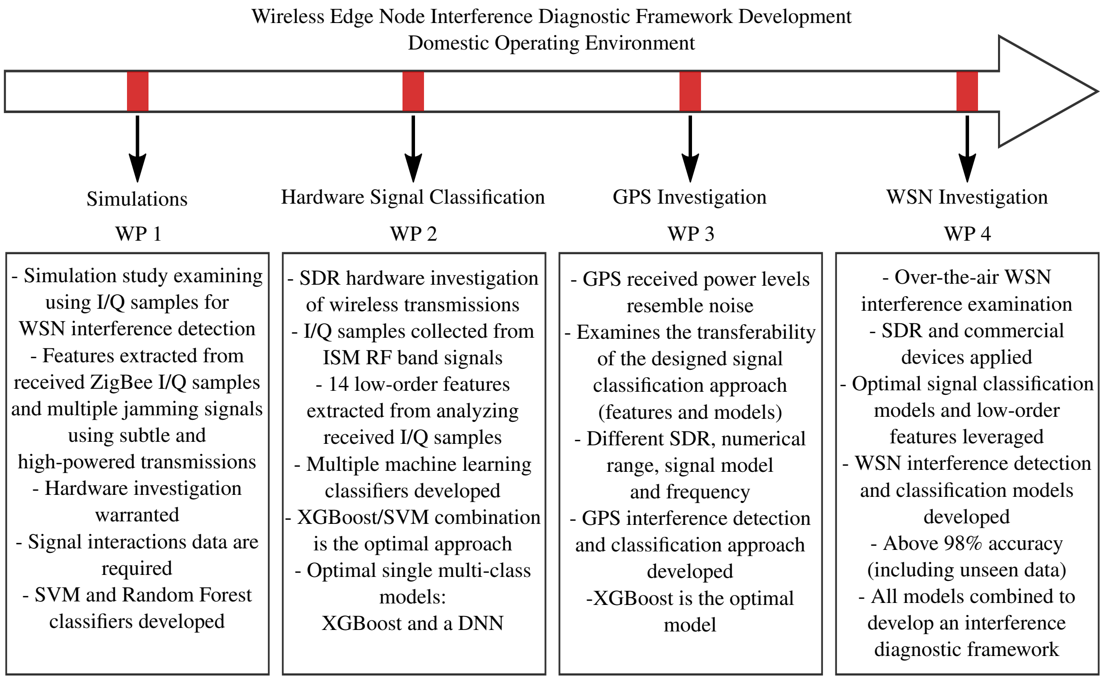
George O'Mahony

Abstract

Classifying the fluctuating operating wireless environment and detecting interference is crucial for wireless networks to reliably deliver authentic and confidential packets. Wireless systems have penetrated many aspects of modern lives and are an essential component of the telecommunications infrastructure. Despite significant technological advancements, most wireless networks remain susceptible to radio jamming. Modern stealthy attacks transmit short jamming signals that use little energy (to be more challenging to detect), yet cause entire packet transmission procedures to fail. This thesis concentrates primarily on wireless sensor networks (WSNs) due to their utilization in the Internet of Things and other applications, the rapidly changing wireless landscape, and susceptibility to radio jamming and spectrum coexistence. WSNs have developed significantly over the past decade and typically consist of resource and energy-constrained devices employing open standards and commercial-off-the-shelf equipment. The deployments are diverse and often form essential safety and privacy-related systems, including remote patient monitoring (health-care), space exploration, smart homes and cities and missile defense. The persistent threat of jamming and the abundance of coexisting signals in the industrial, scientific and medical (ISM) radio frequency (RF) band results in WSNs being readily compromised by malicious and/or unintentional signals. As a result, edge devices require increased capabilities to react to the wireless operating environment and make decentralized decisions, especially in remote deployments where it is difficult to have continued surveillance.

To detect state-of-the-art subtle jamming attacks and enable decentralized decisions, this thesis exclusively utilizes correlated in-phase (I) and quadrature-phase (Q) samples. This procedure focuses on the interactions of the wireless environment with the received signal. The I/Q data is consistently available to a functioning receiver and novel low-order statistical features are extracted from the time-domain, frequency-domain, and spatial analysis. The extracted features enable low complexity machine learning diagnostic tools that achieve similar accuracy to resource-intensive deep neural networks (DNNs). Diagnostics involve legitimate signal/channel classification for identifying the operating environment and interference detection and classification. The developed framework enables decentralized edge device decision-making, needed to enable appropriate independent security and transmitting mechanisms and to reduce retransmissions and energy usage.

Four main work packages (WPs), schematically represented in Fig. 1, are required, each addressing a specific aspect in developing the interference diagnostic framework. To summarize, the initial WP focuses on an extensive Matlab simulation study that


Wireless Edge Node Interference Diagnostic Framework Development
Domestic Operating Environment

Simulations WP 1	Hardware Signal Classification WP 2	GPS Investigation WP 3	WSN Investigation WP 4
- Simulation study examining using I/Q samples for WSN interference detection - Features extracted from received ZigBee I/Q samples and multiple jamming signals using subtle and high-powered transmissions - Hardware investigation warranted - Signal interactions data are required - SVM and Random Forest classifiers developed	- SDR hardware investigation of wireless transmissions - I/Q samples collected from ISM RF band signals - 14 low-order features extracted from analyzing received I/Q samples - Multiple machine learning classifiers developed - XGBoost/SVM combination is the optimal approach - Optimal single multi-class models: XGBoost and a DNN	- GPS received power levels resemble noise - Examines the transferability of the designed signal classification approach (features and models) - Different SDR, numerical range, signal model and frequency - GPS interference detection and classification approach developed -XGBoost is the optimal model	- Over-the-air WSN interference examination - SDR and commercial devices applied - Optimal signal classification models and low-order features leveraged - WSN interference detection and classification models developed - Above 98% accuracy (including unseen data) - All models combined to develop an interference diagnostic framework

Figure 1: Schematic representation of the thesis's work packages and the associated main findings and contributions.

neglects hardware restrictions. WP 2 extracts low-order features from wirelessly received ISM band signals and develops optimal machine learning signal classifiers. WP 3 analyzes Global Positioning System (GPS) signals, due to the benefit from accurate time and location measurements, similarities to signal classification due to the low received power levels and GPS signals being vulnerable to interference. This GPS work examines the developed methodology's transferability to a new RF spectrum area and applies different hardware and numerical ranges for feature extraction. WP 4 focuses on WSN (ZigBee) interference detection and classification using the developed low-order features. The optimized machine learning models in WP 2 form the foundation for interference diagnostic tool development.

Software-defined radios (SDRs) and Raspberry Pi devices implement low-cost yet high-performance WSN and GPS testbeds, penetration testers and data acquisition tools. Feature analysis uses I/Q samples received from commercial and SDR sources in a domestic wireless operating environment. Noise, ZigBee, continuous wave (CW), WiFi and Bluetooth data are examined for signal classification. Over-the-air ZigBee interference data is collected from SDR jamming of ZigBee signals transmitted from SDR and commercial (XBee) sources. A developed ZigBee wireless testbed is utilized in each approach, where the interference signals are the matched signal (ZigBee) intelligent deceptive jamming attack and conventional CW jamming.

Supervised machine learning algorithms, including support vector machines, Random Forest, XGBoost, K Nearest Neighbors and DNNs, evaluate the developed feature set in each application. The designed data analytics and features enable more fundamental approaches to achieve similar accuracy and generalization results, on unseen data, to DNNs, but for a small fraction of the time and computation requirements. Compared to existing schemes, the low-order features that neglect prior network knowledge are novel and prove supervised fundamental approaches can generalize to new data, given powerful data analytics. The principal contribution is the real-world validated intelligent, novel, low complexity, widely deployable interference diagnostic tools. These tools utilize novel low-order features and achieve an average accuracy above 98%, which matches or outperforms the related literature. Adapting the optimized models to GPS signals establishes the transferability of the designed methodology. A Raspberry Pi embedded device implementation exemplifies a relatively resource-constrained deployment.

Acknowledgements

I was fortunate to work with and learn from several exceptional people who helped me complete my Ph.D. journey. Firstly, I would like to express my sincere gratitude to my academic supervisor Dr Colin Murphy for making this Ph.D. opportunity possible and for his continuous guidance and support throughout my Ph.D. journey. I have greatly benefited from his research insights and council at many stages over the past few years. His endless encouragement, attention to detail and unique perspective on this work have been invaluable. It was an honour to work with him and I hope to collaborate in the future and remain in contact with him for many years to come.

I want to thank Dr Philip Harris for being my enterprise mentor and for his specific industry guidance throughout the Ph.D. His support and unique research perspective were always appreciated. He was always willing to help, offer advice and his different perspective made this thesis into a more well-rounded piece of research. He helped me understand the benefits of an industry mindset, which played a vital role in my research. I would also like to thank Dr Kevin McCarthy for being my academic advisor. His advice, particularly over the past year or so, has been invaluable. To Dr James Curran, a special thanks for taking the time to collaborate during my first year of study as I learned so many valuable skills during that time. This collaboration with multiple peers, and the associated insights gained, played a vital role in developing my research skills and mindset. This thesis would not have been possible without their support and guidance. A special thanks goes to Prof Ahmed Abdelgawad and Prof Liam Marnane for examining my thesis. Their thorough evaluation led to a refined dissertation which was highly appreciated.

I want to thank the Irish Research Council and Raytheon Technologies Research Center for funding this research as part of the postgraduate enterprise partnership scheme. Without their combined support, this project would never have gotten off the ground. I want to thank all the School of Engineering staff at University College Cork (UCC) for their support and help whenever needed. This involves Ralph O'Flaherty, Niamh O'Sullivan, Mary O'Leary, Claudia Cashman and others. It made the last four years as part of the Department an absolute pleasure.

I was surrounded by extraordinary colleagues throughout my Ph.D. studies, including all the postgraduate students in the Electrical Engineering building and those from my time spent as part of the SEFs postgraduate committee. They all have shared this long journey with me and filled it with countless laughs, memorable moments and I will cherish the many, many memories for years to come. For their highly appreciated company, I want to thank Robbie, Brian, Oksana, Alison, Oliwier, Adrian, Brendan, Yeny,

Mark, Alex, Shiyao, Tim, Marco, Phil, Sergi, Charu, Haitong, Dan and Kev. I will miss the 5'o clock tea and cards the most as it was always there to get you through even the most difficult of days, provided constant comedic relief and, surprisingly, some in-depth discussions. A special thanks go to Robbie, Brian, Oksana and Alison, who welcomed me into the office and made me feel right at home from my first day. I would also like to thank Arno, Niamh, Katie, Hannah and Ger for making the Ph.D. even more unique by working together as part of the SEFS postgraduate committee. To my friends outside the Ph.D. world who have been along this journey with me, James, Aidan, Shane B, Shane G, Rob, Cian, Sean and Joanne, to name just a few, thanks for all the support, great chats, nights out and for taking my mind off the Ph.D. when needed.

To my sister, Anne, brothers John, Pat and Rob, their partners and my nieces Caoimhe and Niamh, thanks for everything and always providing a supporting hand and distraction when needed. Being the youngest, I always felt I could tackle the world by following the examples you all set and knowing I would never be alone in my ventures. Finally, and most importantly, I want to thank my parents, Geraldine and Patrick, for all their love and support, hot dinners, cups of coffee, endless advice and council. Their love and support throughout these few years, and all of my life, is a debt I will never be able to repay. Someday, I hope to repay you all in some form for all the help and support that you have given me throughout my life. The truth is that without the daily support, love and encouragement from my family and friends, I would not be able to reach this point in my life. I feel blessed to have such a caring family and close friends.

George O'Mahony

List of Abbreviations

ACK	Acknowledgement
ADC	Analog to Digital Converter
AES	Advanced Encryption Standard
AGC	Automatic Gain Control
ANN	Artificial Neural Network
AUC	Area Under the Curve
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BLE	Bluetooth Low Energy
CAP	Contention Access Period
CCA	Clear Channel Assessment
CDMA	Code Division Multiple Access
CNN	Convolutional Neural Network
COTS	Commercial-off-the-shelf
CPU	Central Processing Unit
CR	Cognitive Radio
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CTS	Clear to Send
CW	Continuous Wave
DL	Deep Learning
DNN	Deep Neural Network
DoS	Denial of Service
DSSS	Direct Sequence Spread Spectrum
DVB-T	Digital Video Broadcasting Terrestrial
ECU	Effective Channel Utilization
EW	Electronic Warefare

FCS	Frame Check Sequence
FFD	Full-Function Device
FFT	Fast Fourier Transform
FHSS	Frequency Hopping Spread Spectrum
FPGA	Field-Programmable Gate Array
Gamma	Minimum loss reduction (XGBoost parameter)
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GPU	Graphical Processing Unit
GTS	Guaranteed Time Slot
HOCC	Higher-order Cyclic Cumulant
HOM	Higher-order Moments
I	In-phase
IDS	Intrusion Detection System
IoT	Internet of Things
ISM	Industrial, Scientific and Medical
JSR	Jamming-to-Signal Ratio
k-NNs	K Nearest Neighbors
LEO	Low Earth Orbit
LLN	Low Power Lossy Network
LR-WPAN	Low-Rate Wireless Personal Area Network
LSB	Least Significant Bit
LTE	Long Term Evolution
MAC	Medium Access Control layer
MHR	MAC Header
MIC	Message Integrity Code
MIMO	Multiple-Input Multiple-Output
ML	Machine Learning

MLD	Maximum Likelihood Decoder
MPDU	MAC Protocol Data Unit
MSB	Most Significant Bit
MSDU	MAC Service Data Unit
NN	Neural Network
O-QPSK	Offset Quadrature Phase-shift Keying
OFDM	Orthogonal Frequency-Division Multiplexing
OOB	Out-of-Bag
PAN	Personal Area Network
PDF	Probability Density Function
PDR	Packet Delivery Rate
PER	Packet Error Rate
PHY	Physical Layer
PN	Pseudo-Noise
PPDU	PHY Protocol Data Unit
PRN	Pseudo-Random Noise
PU	Primary User
Q	Quadrature-phase
QAM	Quadrature Amplitude Modulation
RBF	Radial Basis Function
ReLU	Rectified Linear Unit
RF	Radio Frequency
RFD	Reduced Function Device
RMS	Root-Mean Square
RN	Relay Node
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
RSSI	Received Signal Strength Indicator

RTS	Request to Send
RTSA	Real-Time Spectrum Analyzer
SDR	Software-Defined Radio
SFD	Start Frame Delimiter
SNR	Signal-to-Noise Ratio
SU	Secondary User
SVM	Support Vector Machine
USRP	Universal Software Radio Peripheral
WBAN	Wireless Body Area Networks
WP	Work Packages
WPAN	Wireless Personal Area Network
WSN	Wireless Sensor Network

Chapter 1

Introduction

This thesis'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 on the edge node. The diagnostics involve legitimate signal/channel classification and interference detection that enables decentralized edge device decision-making, needed to enable appropriate security and transmitting mechanisms and to reduce retransmissions and energy usage. The aim is to extract high-level interference information from low-level in-phase and quadrature-phase received samples. Wireless networks are 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 thesis focuses on wireless sensor networks and global positioning system signals.

1.1 Motivation

Wireless networks are an essential component of the telecommunications infrastructure. The devices in use vary from resource-constrained wireless sensor edge nodes to resource-abundant laptops and hand-held devices. The applications of wireless devices have penetrated many aspects of modern lives, resulting in the increased importance of, and need for, wireless services [1]. This phenomenon results from the rapid generation of wireless devices and protocols and the explosion of Internet-based mobile applications. Despite wireless technologies significantly advancing over the past several decades, most wireless networks are still vulnerable to radio jamming attacks. This circumstance is due to wireless channels' open nature and the increased availability of hardware capable of jamming multiple networks using various techniques. Additionally, there has been a lack of progress in the design of jamming-resistant wireless networking systems, culminating in legitimate wireless devices being generally unable

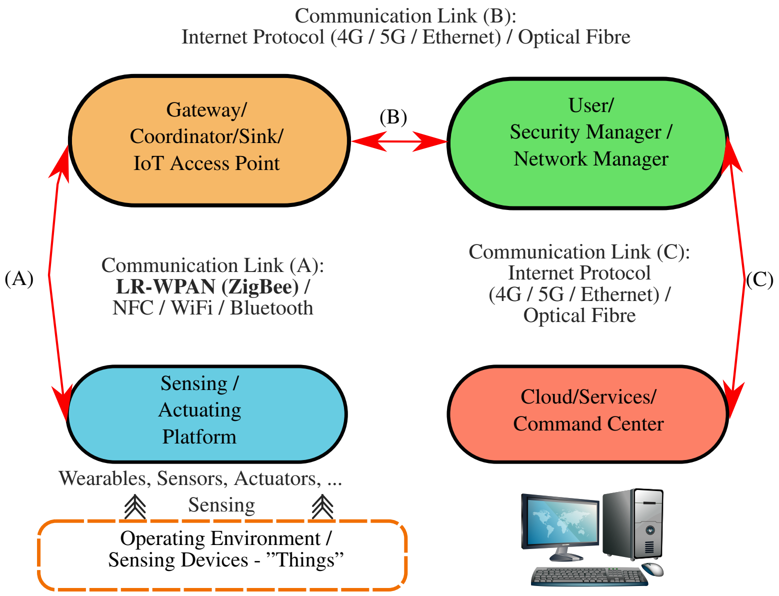


Figure 1.1: An example visualization of the technical architecture of wireless communications in modern society. Example communication model, highlighting the potential use of WSN protocols as the communication link between sensing devices (“Things”) and the internet access point, for use in applications such as the Internet of Things.

to decode data packets in the presence of jamming attacks. This thesis focuses on this jamming attack problem by examining resource-constrained applications, where typical security algorithms cannot be applied. Nevertheless, the associated devices are utilized in safety-critical applications.

Wireless sensor networks (WSNs) have developed significantly over the past decade. These networks are being deployed across a diverse application space, including health care (wireless body area networks) [2], remote patient monitoring [3], missile defense [4], space exploration [5, 6], aerospace [7] and even using Low Earth Orbit satellites as components [8]. Notably, the emerging area of the Internet of Things (IoT) [9] leverages WSNs to provide the communication link between the sensing/actuating devices and the internet access point. This utilization is demonstrated in Fig. 1.1, where the Low-Rate Wireless Personal Area Network (LR-WPAN) protocol is an example communication link. This WSN use case, coupled with the other potential applications, indicates the contribution of WSNs to the technical wireless infrastructure of modern society, while providing an insight into potential vulnerabilities.

These IoT developments are rapidly changing the wireless landscape, as advances are directly affecting (or creating) broadly accepted models such as smart cities, smart homes [10], edge/cloud computing and big data analytics, amongst others. In modern society, IoT research is leading to the truly connected world, smart homes and smart industries. A visualization of the “truly connected world” is provided in Fig. 1.2, which specifies example application areas where anything that can be profitably connected should be connected. These critical applications will, most likely, continue to embrace

WSNs in the modern cost-centered age due to enabling easier design, installation and maintenance, while simultaneously providing new deployment opportunities. However, these deployments persistently encounter malicious interference and spectrum coexistence challenges. To manage safety and privacy requirements, security on edge devices needs improvement, while also maintaining low complexity, to overcome jamming issues without causing device degradation. The preferred solution would require no device redesign or additional hardware expense. Any loss of service (or transmissions) from these deployed edge devices can significantly affect privacy, safety and system performance. Here, ZigBee is the investigated LR-WPAN wireless communication technology as it has been widely used to provide low-bandwidth wireless services for IoT applications such as building automation, medical data collection and industrial equipment control [11]. As WSNs continue to develop into an indispensable component of modern technology and, consequently, the radio frequency (RF) spectrum becomes increasingly congested, the communication link's enhanced security evolves into a necessity. Furthermore, as these applications can benefit from precise location and time data, a similar analysis can be extended to Global Positioning System (GPS) signals.

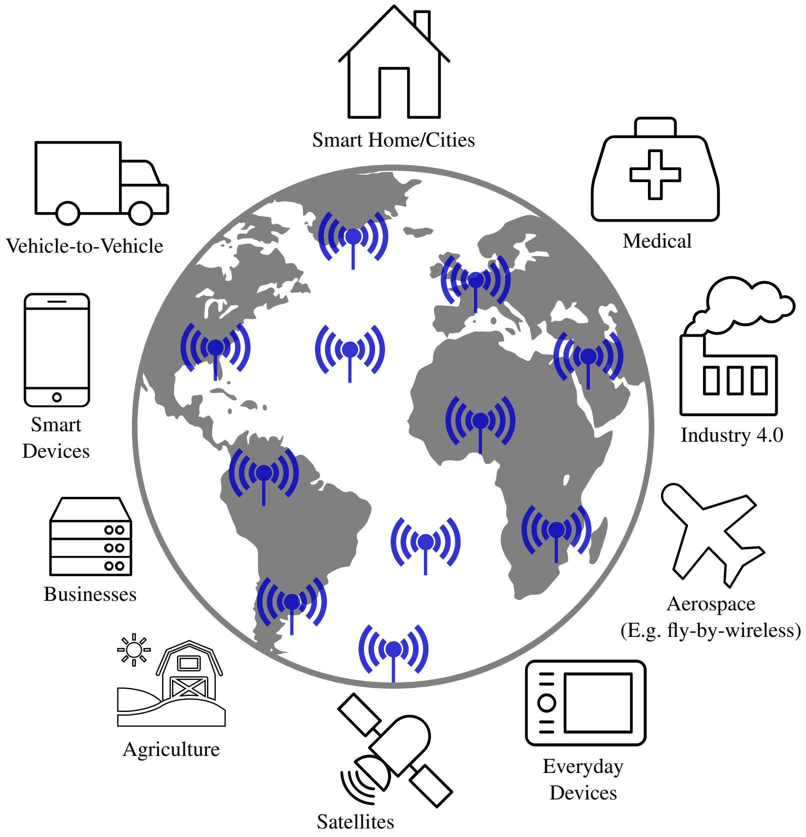


Figure 1.2: A visualization of the importance of wireless communications in modern society. An example of the applications involved in achieving the "truly connected world".

Since WSN utilization has expanded, new security challenges materialize due to the rise of stricter operational and availability requirements. Generally, ZigBee operates using resource-constrained battery-operated devices in the industrial, scientific and medical (ISM) RF band. This operation potentially results in unintentional interference from coexisting devices generating other legitimate ISM RF signals (e.g., WiFi and Bluetooth). As WSNs (ZigBee) are deployed in various real-world critical applications, it is essential to secure ZigBee communications to ensure reliable wireless connections. However, securing WSNs is challenging due to the burden of protecting the transmitted sensitive information across various applications while operating under unique security vulnerabilities and a fluctuating RF spectrum and physical environment. Although ZigBee communications use spectrum spreading at the physical layer for jamming resilience, the spectrum-spreading code sequence's length is 32 chips for every four bits. The resulting jamming mitigation capability that this spreading approach can offer is approximately $10log_{10}(32/4) \approx 9\ dB$, which is limited [11]. As a result, WSNs (ZigBee) are susceptible to both unintentional and malicious forms of interference where, if a device emits sufficiently powerful signals, all the ZigBee devices in its proximity will be unable to communicate. Consequently, WSN compromise, whether malicious or unintentional, is achievable and can have significant privacy and safety consequences. Therefore, the communication link's security and availability are essential for delivering authentic and confidential packets. Also, classifying fluctuating operating wireless environments can be crucial for identifying the dominant signals in the environment, which may allude to the cause of interference being malicious or otherwise. Couple this aspect with establishing a level of trust among network nodes while providing resilience to interference, and it becomes clear that maintaining security is challenging. Hence, as modern applications are increasingly reliant on wireless services, security threats require attention and problematic interference, whether unintentional or malicious, needs to be detected. The optimal detection approach does not degrade device performance or require physical device changes.

Jamming attacks are active attacks that, typically, aim to overpower the legitimate signal with spurious radio-frequency transmissions. When an attacker transmits a jamming signal to disrupt communication, the target node under attack may repeatedly fail in its transmission attempts and retransmit the packets. This process will severely degrade network performance and quickly exhaust the battery of the device [12]. Several jamming techniques exist, where the crude, higher jamming power attacks, are more effective but boost detectability. As such, the attacker is typically driven to optimize signal interference to maximize packet loss while minimizing attack detectability. As a result, jamming in wireless networks has advanced to be more stealthy and long-lasting with limited energy. Stealthy attackers transmit short jamming signals, to become less detectable with less energy, and yet powerful enough to ruin entire packet transmission procedures [12]. In terms of this thesis, as ZigBee, typically, implements no forward error correction, a single bit-error results in a packet error, resulting in this form of subtle jamming being a significant threat. Hence, jamming is a very effective denial-of-

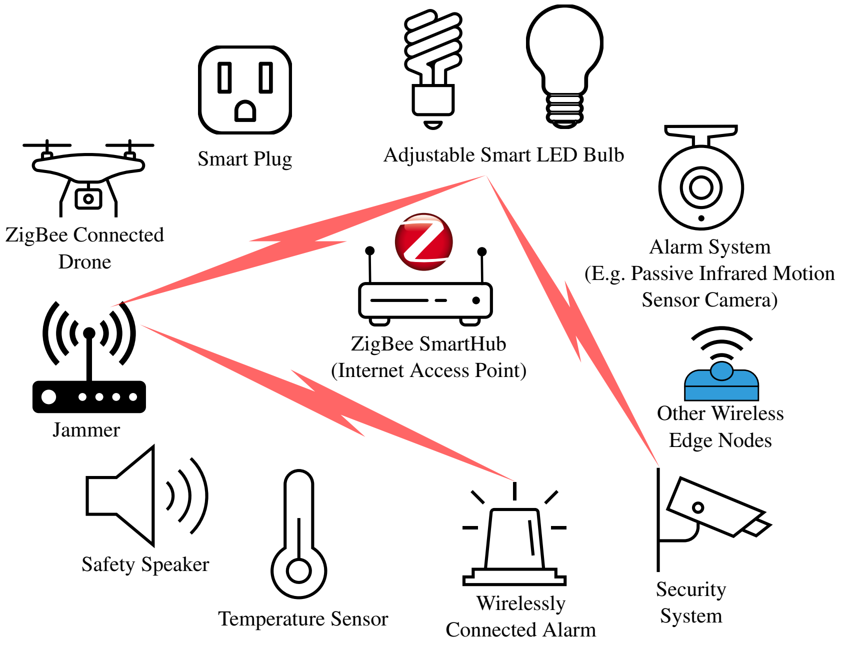

Smart Plug
Adjustable Smart LED Bulb
ZigBee Connected Drone
Alarm System (E.g. Passive Infrared Motion Sensor Camera)
ZigBee SmartHub (Internet Access Point)
Jammer
Other Wireless Edge Nodes
Safety Speaker
Temperature Sensor
Wirelessly Connected Alarm
Security System

Figure 1.3: A visualization of an example WSN application, where the communications strategy utilizes a central coordinator. An illustration of an applied jamming attack on the network is provided, where the jamming signal is illustrated by the lightning bolt symbol.

service attack that renders most higher-layer security mechanisms moot, yet it is often ignored in WSN design [13]. Jamming resistant techniques are applied and are crucial for applications where reliable wireless communication is required. Spread spectrum techniques, such as Frequency Hopping Spread Spectrum (FHSS) and Direct Sequence Spread Spectrum (DSSS), have been used as countermeasures against jamming attacks [14]. To further motivate examining jamming attacks in this thesis, examples of the consequences of jamming applications that leverage WSNs and other low power lossy networks (LLNs) are identified. Visualization of such attacks and typical "victim" applications is provided in Fig. 1.3. Jamming can be devastating to applications that are sensitive to delay or loss. For example, suppose a medical network that monitors patients' physiological data over an LLN [15] is jammed. In that case, serious problems may arise because the transmitted sensitive data is directly linked to patients' lives. Home security systems using WSNs [16, 17] incur similar problems, as intruders can implement jamming attacks to break into a house by disturbing the transmissions of alarm messages to the home control center. Smart grids and factories are also vulnerable to jamming attacks, where an attack can cause misjudgments and erroneous operations [18].

A coherent requirement to improve security methods for WSNs in the presence of interference attacks has thus been established. This thesis focuses on resolving this requirement by developing an interference detection security strategy applicable to resource-constrained wireless edge devices. The primary aim is to exclusively use data consistently available to a functioning receiver at the edge, namely received in-phase

(I) and quadrature-phase (Q) samples. The hypothesis surrounds the concept that once an attack is detected, it can typically be mitigated. This thesis's work develops a low-complexity intelligent interference diagnostics framework that is widely deployable, detects malicious and unintentional interference and applies to different receivers. Consequently, as edge nodes can usually deliver packets to non-jammed neighbors [19], the communication link's security is enhanced by detecting interference. The diagnostics involve extracting high-level information from low-level data for legitimate ISM RF band signal/channel classification and interference detection and classification for ZigBee and GPS signals. This approach enables decentralized edge device decision-making, needed to enable appropriate security and transmitting mechanisms and to reduce retransmissions and energy usage. Data is collected in time-distinct sessions from an active domestic wireless environment that is typically changeable. This I/Q data approach is used to create a concise novel low-order feature set based on time-domain, frequency-domain and spatial analysis. The classifiers embrace machine learning algorithms to exploit the temporal and spatial characteristics of received I/Q data to identity received signals and the presence of subtle and crude interference signals. The diagnostic concept refers to the node's overall ability to detect and classify interference under several scenarios, such as when packets are received and when no packets can be received. An overall algorithm is required as different data are required in each case, resulting in an interference diagnostics solution.

1.2 Aims and Scopes of the Thesis: Intelligent Diagnostic Tools for Wireless Edge Nodes

This thesis aims to enable resource-constrained edge devices to detect and classify interference by developing a novel method of solely using data consistently available to devices at the edge with functioning receivers. The optimal approach would detect interference when packets are received with errors and when no packets are received, while neglecting network-level data and making independent decisions. An approach that could achieve these requirements would have many benefits in decentralized edge device decision-making, such as, for example, supporting the implementation of appropriate security and transmitting mechanisms and reducing retransmissions and energy usage. As a result, this thesis aims to develop low complexity, widely deployable decision support systems for intelligent interference detection in WSN and GPS edge devices. Four main objectives are defined in this thesis to develop the proposed interference diagnostic framework that exclusively utilizes features extracted from the raw received in-phase and quadrature-phase samples. These objectives are as follows:

Objective 1 was to investigate the potential exclusive use of received raw I/Q samples through an extensive Matlab simulation study. This approach incurs no hardware-related restrictions, resulting in no limitations in the maximum or minimum numerical values. This approach is an efficient method to analyze high volumes of transmissions and extract different possible features. This simulation study's main aim was to determine if features extracted from the received I/Q samples had potential for interference detection using machine learning classifiers and if hardware based wireless experimentation was warranted. In the process, these simulations helped to identify the type of data required for interference detection. ZigBee simulations were applied as it is an essential technology for low-power, low-data rate and short-range wireless communication services such as home automation, medical data collection and industrial equipment control [1]. Interference detection was the singular diagnostic tool considered as it is the primary implementation aspect of the proposed diagnostic framework. The simulations were expanded to include interference classification, which suggests using received I/Q samples for signal classification.

Objective 2 was to investigate the practicality of applying the features extracted from the simulation study to real over-the-air wireless signals transmitted and received in hardware testbeds. This initial stage analyzed legitimate signal classification for signals operating in a domestic environment in the ISM RF band. Raw I/Q samples were collected using software-defined radios (SDRs) in a typical wireless operating environment that contained different signal sources, devices, obstacles and service usage. Received samples were visualized and analyzed across time, frequency and space (probability density function), which expanded the feature set extracted in simulations. As the simulations supported interference classification, this manifests as received signal classification in this research objective. Signal classification is required for two reasons: before interference can be detected, the legitimate wireless signals being transmitted need to be identified and the wireless channel (signals in transit) needs to be identified when no packets are received. This results in the first stage of developing an interference diagnostic tool for wireless edge devices since, when packets cannot be received, signal model data is required. A low-order feature set is the critical requirement and needs to differentiate signals using similar modulation schemes and when the receiver becomes saturated. Classification decisions are based entirely on low-order features extracted from the raw I/Q samples. Utilizing the raw I/Q samples was validated by focusing on fundamental machine learning algorithms, which need to be verified as fit for purpose through a comparison with deep learning. The developed features were verified by analyzing the achieved accuracy and ability to generalize to unseen data.

Objective 3 investigated the transferability of the developed low-order feature set through a GPS signal implementation. As GPS signals are received at such low-power levels (e.g. -125 dBm), interference classification is comparable to signal classification. This concept is applicable as received GPS signals resemble noise and relatively low-powered jammers can readily block satellite reception. This phenomenon results in GPS interference signals being classifiable in the presence of noise-like signals, which is consistent with the second research objective. This GPS signal research objective aims to examine the transferability of the developed low-order feature set to a different area of the RF spectrum and to determine if specific hardware or numerical data ranges are necessary. The GPS examination using the developed low-order features, calculated on a different numerical range, was compared to Objectives 2 and 4 by implementing a lower-cost SDR, which subsequently examines a new hardware reception platform. The successful transition of the developed features results in the features providing an underlying description of the received signal being analyzed and being compatible across platforms, signal models and data normalization (numerical range). Additionally, the developed, optimized models from the signal classification objective are being transferred to new data, the received GPS signals. Consequently, the useful transition of the optimal machine learning models to GPS jamming detection and classification signifies a successful example of transfer learning.

Objective 4 was to investigate adapting the low-order feature set to WSN interference detection and classification. This approach forms the interference diagnostic framework's primary concept, as WSNs are highly susceptible to jamming attacks and these networks are becoming increasingly important to modern applications. The research surrounds the physical layer and real over-the-air wireless data was collected by implementing a SDR experimental approach using ZigBee and SDR-based testbeds, where the focus was applied to matched signal and CW interference. The low-order feature set developed in Objective 2 was applied in this interference detection research objective. In-depth analysis and validation of the low-order features for interference detection are achieved using machine learning-based classifiers, namely support vector machine, the dependent ensemble XGBoost approach and a deep neural network (DNN). The methodology development 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 was expanded to a legitimate node classification technique and an overall algorithm for an edge device interference diagnostic framework.

1.2.1 Contribution

This thesis's primary contribution encompasses improving security on resource constrained wireless edge devices by only using data consistently available to a functioning receiver on the edge node. In the process of developing this security improvement, this thesis contributes to three primary areas of wireless communications research: wireless signal classification methods, interference detection in wireless communications networks, namely WSNs, and the exclusive use of raw I/Q samples. The principal contribution is the real-world validated intelligent, novel, low-complexity and widely deployable interference diagnostic tools. These tools form a framework designed for independent compact wireless devices and are based exclusively on raw received I/Q samples and a novel low-order statistical feature-set. The framework highlights that low-level data can be used to make typical higher-level decisions surrounding malicious and unintentional interference scenarios. In contrast to previous work in jamming detection for WSNs, only the designed, optimized machine learning model is required on the device, and both malicious and unintentional interference can be classified. The developed diagnostic framework enables independent operation, as no channel assumptions, network-level information or spectral images are required. It differentiates itself by solely analyzing the raw I/Q data, which is consistently available to functioning receivers, while achieving high accuracy and generalization to unseen data.

In terms of the developed feature set, the contribution and novelty surround the use of received raw I/Q samples and application of the feature-set across WSN and GPS interference detection and classification and ISM RF band wireless signal classification. The concise novel feature set is based on time-domain, frequency-domain and spatial analysis of the received I/Q samples collected from an active wireless environment that is typically changeable. To the best of the author's knowledge, the application of Hjorth parameters [20] in this thesis is novel and the use of the fast Fourier transform (FFT) dynamics produces novel features. The developed features differ from the literature by only requiring access to raw received I/Q samples, permitting independent device decisions and using low-order statistics. This differentiation is thoroughly explained in Chapter 3, where the lack of spectral images, received signal strength indicator samples, high-order cumulants and transforms other than the FFT are the differentiating factors. These low-order features provide a novel input to ISM RF band signal classification and wireless interference detection by providing an underlying description of the received signal. The low-order features enable single device diagnostics regarding the received signal and operating environment interactions to detect subtle deviations (stealthy jamming) from the expected reception.

The contribution is validated and enhanced by successfully implementing the developed intelligent classification methodology across different, relatively low-cost, open-source SDR receivers, numerical ranges, signal models, frequencies and implementation platforms. The developed methodology includes the low-order features, the data requirements, the identified machine learning approaches, the experimentation across platforms and the exclusive use of I/Q samples, along with the developed framework using real-world signals. The designs are fully validated by implementing deep neural networks and lower complexity fundamental machine learning solutions using open-source software programs. The developed supervised fundamental machine learning approaches detect and classify malicious and unintentional interference on wireless edge nodes while achieving deep learning performance for a small fraction of the time and resource requirements. Additionally, a minor contribution uses the low-order features as a simple commercial ZigBee node/software-defined radio classifier. Multiple intelligent models are designed as a combination to develop an edge device low complexity, low-order wireless interference diagnostic framework, which is the thesis's main contribution.

1.3 Thesis Layout

The development of intelligent low complexity diagnostic tools for the accurate detection and classification of interference and improving security on wireless resource-constrained edge devices is a multidisciplinary field of research that encapsulates knowledge of engineering, computer science and practical hardware deployment. Exclusively utilizing correlated in-phase and quadrature-phase samples to implement the tools and provide independent edge device operation incorporates an additional layer of complexity to the work packages. Therefore, the work conducted in this thesis has a broad focus and the thesis layout is represented as follows:

Chapter 2 provides the necessary background on the primary signals of interest and the associated proposed application space within which to deploy the developed interference diagnostic framework. The primary application area is WSNs, which are systems implemented across various deployments, including safety-critical, space and internet of things applications. These applications can benefit from precise location and time data, resulting in the analysis of GPS signals. The chosen WSN protocol is ZigBee as it is the de-facto standard since almost all available commercial and research sensor nodes are equipped with ZigBee transceiver chips [21]. This chapter also discusses security in terms of four interlinked but distinct components; requirements, vulnerabilities, attacks and defenses. This analysis develops a 3-D functional model for security that provides a simplified visual representation of some available security setups for wireless networks. This thesis's hop-by-hop event-driven (bit errors) detection on every transmission is one of the possible established security setups. The reason why jamming attacks were the chosen attack style for this thesis is also explained. This chapter also establishes the typical attack development strategy and describes the intelligent deceptive jammer attack style of matched signal interference. For readers who have no security background, this information may be relatively new and, therefore, it is crucial to clarify the vulnerability of wireless networks to various interference attacks.

Chapter 3 describes how this work is distinct from the literature. A comprehensive background examination of wireless signal classification is provided, where the primary use case is automatic modulation classification. The literature review is then expanded to include interference detection and prevention techniques. Throughout this state-of-the-art discussion, the attributes that indicate new solutions will be required at the edge, meaning that the resource-constrained devices will require an interference diagnostic framework, are established. The primary source of novelty is exclusively using the I/Q samples and the related research on this topic is discussed. Regarding wireless networks, detecting interference and classifying the signal type is not an original concept but requires continual improvements to keep up with current hardware/software enhancements, which enable more stealthy and advanced interference attacks. Focusing on using the received raw I/Q samples as the basis for interference detection and classification decisions is a relatively novel concept.

Chapter 4 explores the initial examination of exclusively using received I/Q samples to detect and classify interference in WSNs. The necessary I/Q data is collected by applying Matlab Monte Carlo simulations, across a range of jamming-to-signal (JSR) ratios and interference types, including matched signal interference, CW, WiFi and thermal noise. These simulations analyze the full packet overlap case, where the legitimate and interference packets fully interact, and different variations of legitimate and interference packet overlap. This approach evaluates the ideal case for using I/Q samples, as no hardware limitations exist in these Matlab simulations. The simulation approach is validated using a software-defined radio to transmit the simulated packet for a spectral comparison to a commercial device. This chapter uses the simulation study to identify the required data for developing machine learning classifiers and the developed features form the foundation for hardware experimentation in later chapters. This chapter's work determined if the raw I/Q samples had value and if there was motivation to investigate a live over-the-air study. The initially adopted machine learning algorithms of support vector machine (SVM) [22] and Random Forest [23] are also discussed and general classifier metrics established.

Chapter 5 discusses the hardware used in this thesis and explains why each device was selected. Each device's associated applications are specified, along with the chosen wireless devices' utilization in developing data strategies for each of the wireless over-the-air studies. These designed testbeds transmit environmentally sensed data using the ZigBee protocol and are the primary source of legitimate WSN data in this thesis. This chapter also demonstrates the advantages of utilizing SDRs, and available software packages, as wireless signal analysis and penetration testing tools. Extracting received I/Q samples in coexistence with both unintentional and malicious interference is essential for successfully developing the proposed methodology. As a result, this chapter discusses how Raspberry Pi embedded devices and SDRs were employed to produce low-cost, high-performance testbeds and data collection approaches to obtain the necessary I/Q samples. SDRs enable access to the required data and provide the necessary jamming signals when needed, as the SDR can produce several signal models using signal processing blocks or Python3 software. The work in this chapter is essential for the wireless studies in Chapters 6 and 7.

Chapter 6 leverages the results from Chapter 4 to develop a low-order feature set based on wireless over-the-air data received using SDR hardware. The features aim to differentiate signals in the ISM RF band, even when signals use similar modulation schemes and when the receiver becomes saturated. The analyzed signals are transmitted from both commercial and SDR sources and include channel noise, ZigBee, continuous wave, WiFi and Bluetooth signal data. Classification decisions are based entirely on the developed low-order features extracted from the raw I/Q samples. This approach develops a legitimate signal classification approach for the interference scenario when no packets can be received. This chapter validates the developed features by analyzing the achieved accuracy and ability to generalize to unseen data. The machine learning algorithms used are the previously introduced SVM and Random Forest models, along with the newly adopted k nearest neighbors (k-NNs), XGBoost, Naive Bayes and neural network (NN) approaches. These machine learning concepts are all discussed in detail, particularly the XGBoost and neural network approaches. The primary outcome of this chapter is the developed optimal features and associated optimal machine learning approaches.

Chapter 7 discuss this thesis' main contribution. The work in this chapter develops the overall interference diagnostic framework for wireless resource-constrained edge devices. The low-order features and optimal machine learning approaches from Chapter 6 are adopted in this chapter as a form of transfer learning to develop the interference detection and classification models efficiently. The focus is on the ZigBee WSN signal model and the transferability of the features and developed methodology is examined by studying GPS signals. The real over-the-air data are collected by implementing SDR testbeds, as described in Chapter 5. In-depth analysis and validation of the low-order features for interference detection are achieved using the SVM, dependent ensemble XGBoost approach and a deep neural network. The developed models are evaluated using available test data and K-fold cross-validation. The low complexity interference framework achieves an average accuracy among the developed models above 98%. This chapter's work involves examining ZigBee over-the-air data for artificial jamming and SDR jamming of ZigBee signals transmitted from SDR and commercial (XBee) sources, where matched signal (as per Chapter 2) and continuous wave interference are analyzed. A Raspberry Pi embedded device implementation study examines a relatively resource-constrained deployment. The overall diagnostic algorithm is formulated based on the developed interference detection models and the previous signal classification approach in Chapter 6.

Chapter 8 is the concluding chapter, which summaries the main findings and contributions of the thesis and discusses potential future research directions.

1.4 List of publications arising from this thesis

Journals:

1. G. D. O'Mahony, K. G. McCarthy, P. J. Harris and C. C. Murphy, ”Developing a Low-Order Statistical Feature Set Based on Received Samples for Signal Classification in Wireless Sensor Networks and Edge Devices”, IoT, vol. 2, no. 3, pp. 449-475, 2021, doi: 10.3390/iot2030023

2. G. D. O'Mahony, K. G. McCarthy, P. J. Harris and C. C. Murphy, ”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”, Ad Hoc Networks, vol. 120, p. 102562, 2021, doi: 10.1016/j.adhoc.2021.102562

Magazine Articles:

1. G. D. O'Mahony, J. T. Curran, P. J. Harris and C. C. Murphy, ”Interference and Intrusion in Wireless Sensor Networks,” in IEEE Aerospace and Electronic Systems Magazine, vol. 35, no. 2, pp. 4-16, Feb. 2020, doi: 10.1109/MAES.2020.2970262.

Conference Papers:

1. G. D. O'Mahony, P. J. Harris and C. C. Murphy, ”Analyzing the Vulnerability of Wireless Sensor Networks to a Malicious Matched Protocol Attack,” 2018 International Carnahan Conference on Security Technology (ICCST), Montreal, QC, 2018, pp. 1-5, doi:10.1109/CCST.2018.8585681.

2. G. D. O'Mahony, P. J. Harris and C. C. Murphy, ”Investigating the Prevalent Security Techniques in Wireless Sensor Network Protocols,” 2019 30th Irish Signals and Systems Conference (ISSC), Maynooth, Ireland, 2019, pp. 1-6, doi: 10.1109/ISSC.2019.8904934.

3. G. D. O'Mahony, P. J. Harris and C. C. Murphy, ”Developing Low-Cost Testbeds for Enhancing Security Techniques in Wireless Sensor Network Protocols,” 2019 30th Irish Signals and Systems Conference (ISSC), Maynooth, Ireland, 2019, pp. 1-6, doi: 10.1109/ISSC.2019.8904967.

4. G. D. O'Mahony, P. J. Harris and C. C. Murphy, "Analyzing using Software Defined Radios as Wireless Sensor Network Inspection and Testing Devices: An Internet of Things Penetration Testing Perspective," 2020 Global Internet of Things Summit (GIoTS), Dublin, Ireland, 2020, pp. 1-6, doi: 10.1109/GIOTS49054.2020 .9119606.

5. G. D. O'Mahony, P. J. Harris and C. C. Murphy, "Identifying Distinct Features based on Received Samples for Interference Detection in Wireless Sensor Network Edge Devices", 2020 Wireless Telecommunications Symposium (WTS), Washington, DC, USA, 2020, pp. 1-7, doi: 10.1109/WTS48268.2020.9198724.

6. G. D. O'Mahony, P. J. Harris and C. C. Murphy, "Investigating Supervised Machine Learning Techniques for Channel Identification in Wireless Sensor Networks," 2020 31st Irish Signals and Systems Conference (ISSC), Letterkenny, Ireland, 2020, pp. 1-6, doi: 10.1109/ISSC49989.2020.9180209.

7. G. D. O'Mahony, P. J. Harris and C. C. Murphy, "Detecting Interference in Wireless Sensor Network Received Samples: A Machine Learning Approach", 2020 IEEE 6th World Forum on Internet of Things (WF-IoT), New Orleans, LA, USA, 2020, pp. 1-6, doi: 10.1109/WF-IoT48130.2020.9221332.

Chapter 2

Wireless Sensor Networks, Signal Models and Security

This information is important for the simulation work in Chapter 4, the hardware testbed development in Chapter 5 and the experimental over-the-air work presented in Chapters 6 and 7. This chapter outlines the application space of this thesis, the associated signal models, the typical security requirements and the attacks that can be implemented on the wireless channel. The work in this chapter has been published in part in the following:

- *G. D. O'Mahony, J. T. Curran, P. J. Harris and C. C. Murphy, "Interference and Intrusion in Wireless Sensor Networks," in IEEE Aerospace and Electronic Systems Magazine, vol. 35, no. 2, pp. 4-16, 1 Feb. 2020, doi: 10.1109/MAES.2020.2970262.*
- *G. D. O'Mahony, P. J. Harris and C. C. Murphy, "Investigating the Prevalent Security Techniques in Wireless Sensor Network Protocols," 2019 30th Irish Signals and Systems Conference (ISSC), Maynooth, Ireland, 2019, pp. 1-6, doi: 10.1109/ISSC.2019.8904934.*

2.1 Introduction

This chapter examines the primary signals that will be analyzed as part of this thesis and the associated proposed application space for deployment of the developed interference diagnostic framework. The primary application area is wireless sensor networks (WSNs), which are systems utilized across a diverse array of deployments, including safety-critical, space and Internet of Things (IoT) applications. These applications can benefit from precise location and time data, resulting in the analysis of Global Positioning System (GPS) signals. The radio architectures and protocols used by WSNs are, typically, very similar and are based on IEEE 802.15.4. The possibility of designing a transferable security enhancement across protocols and services becomes a reality by concentrating on this standard and its associated security techniques. Security requirements are vital as WSNs and associated resource-constrained devices develop into indispensable components of modern technology. Fundamentally, transmitted data needs to be free from unauthorized intrusion and all services need to operate when requested. These applications have expanded significantly over the past decade or so and adopt commercial off-the-shelf devices and publicly available standards in diverse safety-critical applications, which inherently creates intruder incentives and security challenges.

Securing WSNs is a critical requirement due to the challenging burden of protecting the transmitted sensitive information across various applications while operating under unique security vulnerabilities and a fluctuating radio frequency (RF) spectrum and physical environment. Couple this aspect with establishing a level of trust among network nodes while providing resilience to interference, and it becomes clear that maintaining security is challenging. This chapter identifies unique vulnerabilities in WSNs and wireless edge devices, which directly impact privacy and safety. The prevalent security techniques used in the standard physical (PHY) and medium access control (MAC) layers of various WSN protocols are discussed to establish the essential security requirements. Fundamental attack styles and spectrum sharing/coexistence based intrusions are presented. Experimental visualization of the coexistence issues in the industrial, scientific and medical (ISM) RF band, which is integral for WSNs and IoT operations, is provided as an introduction to a new perspective on attacking WSNs. Typical methods, which use commercial-off-the-shelf (COTS) devices and open-source software to exploit WSN security holes, are also discussed. The need for expanding intrusion detection via a more holistic approach, while simultaneously improving WSN security, is illustrated. The overall security concept of wireless networks is depicted in terms of four primary pillars and visualized in a developed simplified 3D model.

Currently, the sensed data can, potentially, lose its value in a matter of milliseconds and, so, data or central coordinator's decisions can have expired by the time it arrives at the intended wireless edge node, data center or network coordinator. This aspect implies a potential need for edge devices to provide services and not wait for the data center's control response, including when cooperation between devices is required in a cluster topology. As more emphasis is applied to implementing a more decentralized approach, security solutions will be required to run on the wireless edge devices and decisions will be required in real-time. Thus, interference detection systems or diagnostic tools deployed on edge devices can enhance the communication link security at the edge. This concept utilizes the hypothesis that, typically, once an attack (or packet loss reason) is detected, it can be mitigated. As a result, a detection system that focuses solely on available data from the surrounding wireless environment at each wireless edge device would be advantageous. As the wireless operating environments become more diverse, as attacks become more sophisticated and as services become more decentralized, it becomes clear that interference detection mechanisms are necessary. This chapter discusses this topic and motivates the selection of WSNs and GPS applications and explains why interference detection on edge devices is the focus of this thesis.

2.2 WSN Description & Applications

As the radio spectrum changes frequently due to varying numbers of connected devices, demand, packet size or services in operation and the physical environment fluctuates due to varying fading levels, obstacles, path losses, and spurious interference, wireless channels are unique. Beyond these non-malicious factors, critical WSN applications and their associated transmitted sensitive data may incentivize malicious attackers to intentionally disrupt or compromise network operation by emitting malicious signals into the channel. For a wireless communications system, a channel refers to a logical connection over a multiplexed medium, such as a radio channel, used to convey all information signals, typically, digital bit streams, from one or several transmitters to one or several receivers. Each channel has a specific capacity (C) for transmitting information and Shannon's capacity theory defines the tight upper bound (R) on the rate at which information can be reliably transmitted. If $R \leq C$, a coding technique exists, allowing the probability of error at the receiver to be made arbitrarily small and the entire message to be decoded without error. Therefore, being able to classify the type of channel, or the dominant signal in transit, aides in the transmission procedure and can provide insights for packet rates. This concept expands to include whether the transmissions in the channel are malicious. In this thesis, the ZigBee WSN protocol (see Section 2.3) is the chosen legitimate signal model. ZigBee operates on sixteen 2 MHz wide wireless channels in the unlicensed 2.4 GHz ISM RF band.

WSNs consist of multiple lightweight resource-constrained devices (nodes) used to sense the physical world and, typically, incorporate a radio transceiver, a microcontroller, sensors and a limited energy source. These nodes gather data from their environment and, often, collaborate to transmit the sensed data to a centralized sink or cluster head. Typically, these devices are located at the edge without any fixed infrastructure in hostile or remote environments, where it is difficult to have continued surveillance. These networks' main characteristics include energy usage, handling of node failures, nodes joining and heterogeneity of nodes, operating in harsh conditions, ability to scale, and incorporating mobile nodes. A general network communication approach is provided in Fig. 2.1, where a WSN protocol is used for communications between the sensing devices (Endpoint) and coordinator, which acts as an access point

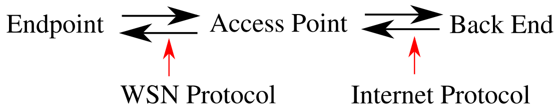


Figure 2.1: A visualization of the typical communications strategy in a WSN application, where the main communication points are identified.

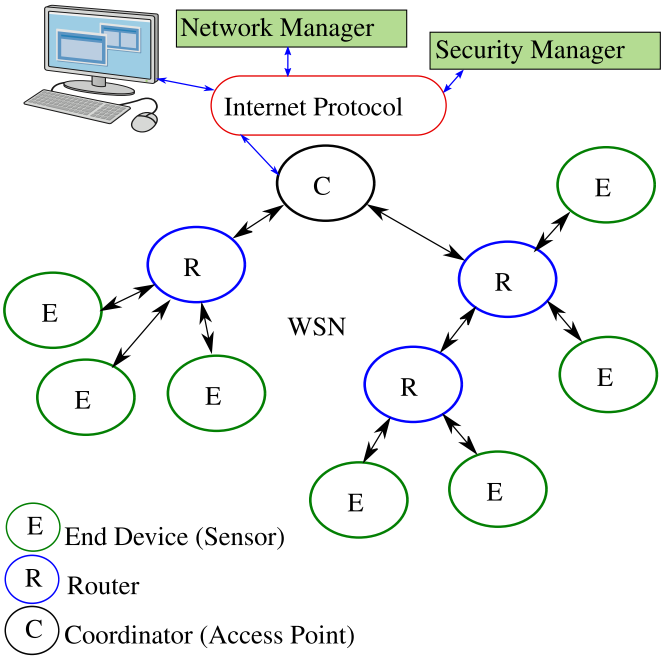


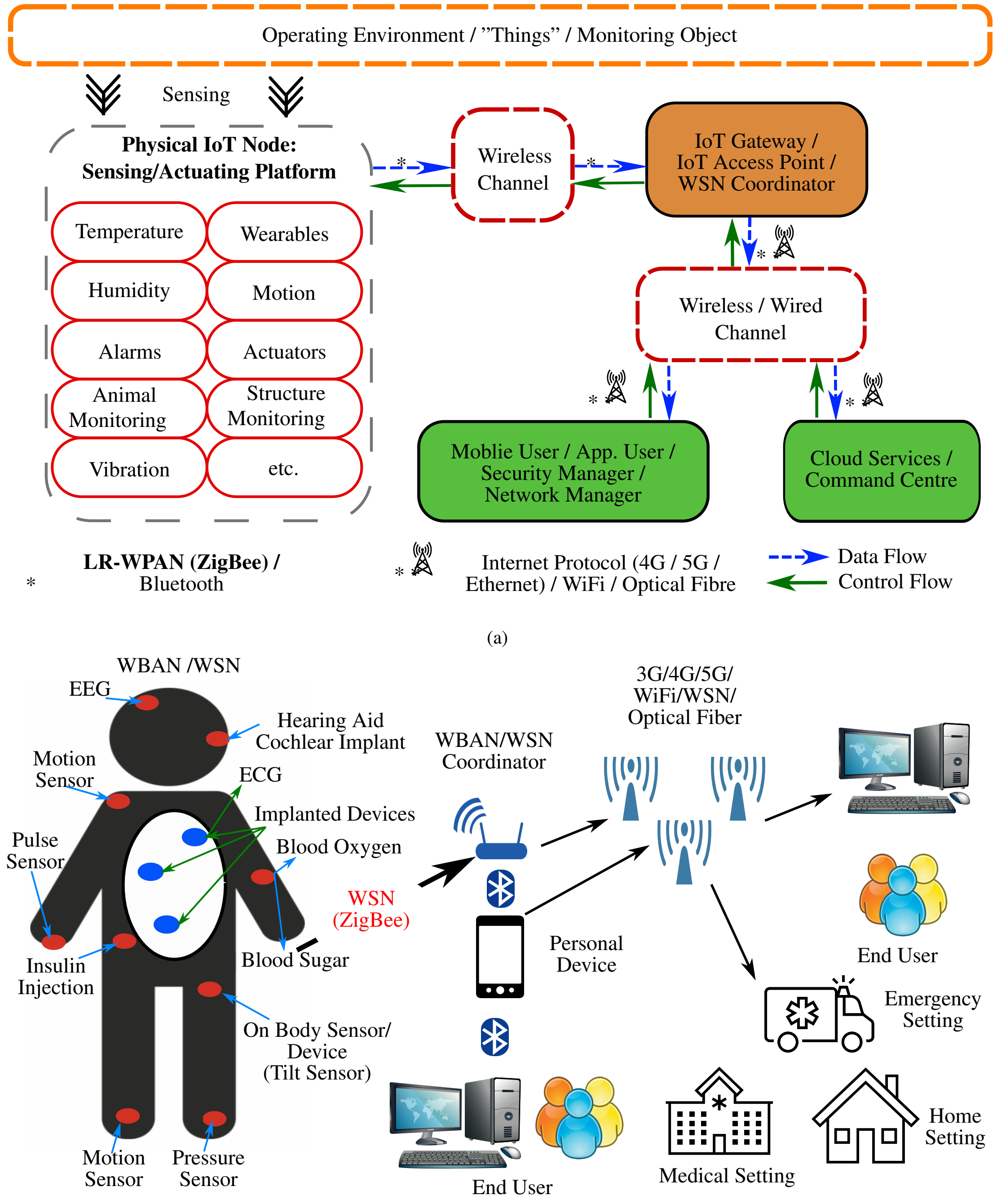
Figure 2.2: Example WSN mesh topology utilizing a cluster approach, showing how the communications strategy is implemented in practice.

for network users to capture and analyze data and permits the use of a network/security manager. Transmission between the access point and other components (Back End) is usually achieved by utilizing communication procedures such as, for example, internet access, cellular, wired connections and more computationally powerful relay nodes such as, for example, Low Earth Orbit (LEO) satellites as a WSN component [8]. As a result, these communications are either machine-to-machine or machine-to-people transmissions, which can include private data, resulting in critical privacy requirements. Typically, WSNs involve long-lived deployments consisting of resource-constrained devices coupled to their operating environment and available resources, which prohibit using complex or computationally intensive security protocols. The spectrum of these operating environments changes rapidly due to a large (often changeable) number of connected devices potentially running different protocols at the same frequency, location and time. Beyond these non-malicious factors, malicious attackers can cause additional operating environment changes to disrupt or compromise network operation.

These WSNs operate either a star, mesh or peer-to-peer topology and, in each case, are self-organizing, self-repairing, dynamic and can exploit the cluster head approach. Individual devices can be either a full-function device (FFD), which act as a Personal Area Network (PAN) coordinator, router or end device, or a reduced function device

(RFD), which acts as a simple sensing end device. A FFD can communicate with other FFDs or RFDs, while a RFD can only communicate with one FFD. Two or more of these devices (minimum of one FFD as a coordinator) operating on the same channel and within a personal operation space form a wireless PAN. Each PAN selects a unique identifier (PANId) and all devices have a unique 64-bit address. An example mesh topology, which exploits clustering, is provided in Fig. 2.2. This typical WSN approach can use data aggregation techniques at routers or relay nodes to minimize communication overhead and maximize energy efficiency. This clustering approach is generally achieved by unifying several data items into a single packet and then applying compression techniques or processing data at the relay nodes. From a security perspective, the clustering approach results in a jamming attack affecting several nodes by attacking one. A network manager is responsible for the configuration between nodes and a security manager is responsible for key management. This WSN topology can incur latency in implementing decisions that originate at the "Back End" before being transmitted to the edge node. Also, the sensed data can potentially lose its value in a matter of milliseconds and, so, can have expired by the time it arrives at the data center if transmissions are impeded. As a result, the induced latency in decision-making or retransmissions could have significant consequences due to data or decisions being obsolete on arrival at the edge or coordinator. Consequently, independent decision making at the edge is advantageous, specifically, if only data originating at the edge is required. This approach can identify intrusions to minimize the impact of retransmissions and negate the need for certain decisions to be transmitted by the coordinator and ensure timely data transmissions.

Successful WSN deployments in both civil (Fig. 2.3 (a)) and medical (Fig. 2.3 (b)) safety-critical applications has developed WSNs into essential components of modern technology. Applications such as the IoT [9] and remote patient monitoring in health care [3] utilize WSNs, as visualized in Fig. 2.3. These critical applications, and other innovative solutions, continue to adopt WSNs to permit easier design, installation and maintenance, while simultaneously providing new deployment options and cost benefits. These advantages are a consequence of the resource-constrained devices in use, applicable operating topologies and protocols in operation. However, as WSNs become integrated with critical use-cases, the incentive to attack/disrupt these networks intensifies. This is emphasized by deployments in control systems for smart homes [24], space-based WSNs [6], missile defense [4], wireless body area networks (WBANs) [25], aerospace, surveillance and industrial sensing. Other space applications include, amongst others, an in-orbit demonstration of an IEEE 802.15.4 protocol based WSN on the International Space Station [5] and space wireless local area networks [26]. Also, due to advances in the development of WSN architectures [27], LEO satellites can be used as WSN components [8] to receive aggregated packets from WSN relay nodes,



(a)

(b)

Figure 2.3: 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.

particularly in remote locations. Additionally, WSNs are being utilized in aerospace applications for aircraft control and health management systems [7] as a first step towards fly-by-wireless and increased monitoring capabilities. Notably, these WSNs are deployed as underwater-WSNs (UWSNs) as a method to remotely explore and monitor the vast underwater world with ease and less risk [28]. Uniquely, arrays of nanosatellites are used in a WSN approach to enhance mobile communications through lower-cost, space-based mobile phone services [29].

Notably, in modern society, the emerging IoT [30], which leverages WSNs, is leading to the truly connected world and smart homes/businesses. WSN signals are interlinked with IoT applications, as WSNs regularly provide the sensing/actuating device to internet access point communication link, as specified in Fig.2.3 (a). This utilization enables a larger number of connected devices and a longer range compared to wireless local area networks (WLANs). WSN signal models, for example, ZigBee, can provide a range in the order of a few kilometers compared to the tens of meters provided by WiFi networks. The mesh networking topology allows easy extension of the network's physical range by routing data smartly between different nodes compared to the star topology of WiFi. IoT solutions are directly affecting (or creating) broadly accepted models such as smart cities [10], smart buildings (District 555 Building Washington DC), edge/cloud computing, big data analytics and the use of LEO satellite constellations [31], amongst others. Operating areas are growing in importance and diversity, as usage encompasses biomedical, which typically relates to remote patient monitoring [3], transportation, environmental, industrial and such critical asset tracking as airplanes, oil tankers, medical equipment and moving vehicles, for example. Notably, Industry 4.0 aims to increase productivity, efficiency and flexibility by developing the smart factory. As a result, continuous improvement of security in WSNs or the IoT is an issue of concern to the scientific community due to the considerable growth in demand and use of these technologies, which, inherently, incentivize malicious actors to disrupt, compromise or deny network operation. WSNs and IoT applications' maturing environments results in long-lived deployments, where resource-constrained low-power embedded devices are tightly coupled to the operating surroundings and must execute received instructions and necessary data transmissions. Fundamentally, transmitted data needs to be free from unauthorized intrusion and all services need to operate when requested.

In addition, Fig. 2.3 (a) depicts the importance of the wireless channel. It specifies where interference, both malicious and unintentional, can be emitted to cause WSN disruption or compromise. The low processing power, memory and speed of WSN devices, coupled with a modest energy source, impedes using conventional security protocols, while WSN attack types are various and can occur across the entire communication protocol stack. Attacks (see Section 2.6) can vary from specific denial of service (DoS) attacks, which can corrupt all packets, to privacy attacks, which can seize sensitive data. However, techniques can be employed to protect important data and provide resilience against malicious attacks, such as, for example, cryptography, direct sequence spread spectrum (DSSS) modulation, frame check sequences and intrusion detection systems. In this thesis, security is coupled with classification in terms of distinguishing legitimate signals from interference. The hypothesis is based on the fact that once an attack is detected, it can be mitigated. Each of the discussed WSN infrastructures and applications requires protection and attack detection, as an attack could have significant consequences for privacy and safety. This chapter uses these critical WSN applications as the basis to provide a review of WSN vulnerabilities, security and attacks, including co-existence intrusions. This discussion can be expanded to include the GPS signals. GPS signals are becoming increasingly crucial for civilians, services and industries due to the dependence on GPS-derived location and time measurements. Typically, WSNs can incorporate received GPS data and a correlation exists between specific applications, for example, using LEO satellites as WSN components. Applying LEO satellites in a WSN requires orbital position and exact time measurements to efficiently transmit packets when the LEO satellites are visible in a specific location.

Furthermore, the concept of applying COTS devices and standardized protocols is becoming the standard due to the general trends towards the use of COTS components in commercial IoT networks and space applications. Both areas, typically, favor high redundancy and high replenishment rates over custom-built components. Examples include the international space station [5], inter-satellite communication modules [6], nanosatellite swarms [32] and the various commercially available IoT devices. As a result of this utilization and the general uptake in connected devices, the number of wireless devices is growing at an unprecedented rate. Approximately 29 billion devices are forecast by 2022 [33]. 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. WSNs require security across a wide range of physical environments, deployments scenarios and structures, in which privacy and safety are pivotal. Furthermore, these applications can be valuable and, typically, use sensitive data, which incentivize malicious actors to intentionally disrupt or compromise network operation. This phenomenon coupled with unique WSN vulnerabilities, provided in Section 2.5.1, demonstrates the difficulty in guaranteeing security. As a result, the use of WSNs in safety-critical applications creates new challenges in terms of security, spectral coexistence and threat identification, as discussed in Section 2.5. Any security solution or tool for wireless edge devices needs to be aware of these operating conditions.

2.3 ZigBee Signal Model

To understand WSN operation and the security techniques that help to overcome the vulnerabilities outlined in Section 2.5.1, the protocols that govern these networks' operation must be analyzed. WSNs can employ multiple protocols and the leading available technologies include ZigBee, WirelessHART, 6LoWPAN, ISA100.11a, Thread and MiWi. Presently, these protocols are used in applications, as mentioned in Section 2.2, and are the most prominent participants of the expansion of the IoT. The common aspect across these protocols is using the IEEE 802.15.4 [34] standard as the fundamental network infrastructure, on which the more complex networks are formed. This standard originated in 2003 and has undergone various amendments over the years and, here, the focus is on both the 2006 revision and IEEE 802.15.4e. Typically, the standard defines the PHY and MAC layers of Low-Rate Wireless Personal Area Networks (LR-WPANs), which describes WSNs. Three possible operating RF bands (868/915/2450 MHz) are provided, which use different modulation schemes, support various data rates and offer different topologies and security suites.

Here, the security analysis concentrates on the 2450 MHz band, which operates in the unlicensed ISM RF band, and the IEEE 802.15.4 PHY and MAC layers. These layers construct the signal and operate the channel access technique, which means these layers are linked to the wireless channel. The interference detection methodology and overall framework developed in this thesis focus on the channel interactions, resulting in an emphasis on the lower layers of the WSN protocol stack. This 2.4 GHz area of the RF spectrum is highly congested due to varying high numbers of connected devices, potentially utilizing different protocols at the same frequency, location and time. This level of congestion requires the efficient use of the spectrum and this aspect is experimentally visualized using Tektronix's digital phosphor technology (DPX) [35] (see Chapter 5 for full description) and a real-time spectrum analyzer (RTSA) in Fig. 2.4. Securing ISM band signals becomes increasingly complex due to these coexistence levels and congestion combined with the rapid ability of the spectrum to change due to the number of connected devices, demand, packet size or services in operation. This fact, coupled with the varying nature of the networks' physical environment, makes it clear that WSN security must contend with malicious and unintentional interference and intrusions.

In this thesis, the IEEE 802.15.4 based wireless protocol for LR-WPANs, ZigBee, is the chosen signal model. Currently, ZigBee is the de-facto standard for WSNs, as almost all available commercial and research sensor nodes are equipped with ZigBee transceiver chips [21]. The operating topology is either star, mesh or peer-to-peer and, in each case, is self-organizing, self-repairing, dynamic and can exploit clustering approaches [36]. Cluster heads are typically used as relay nodes that aggregate and for-

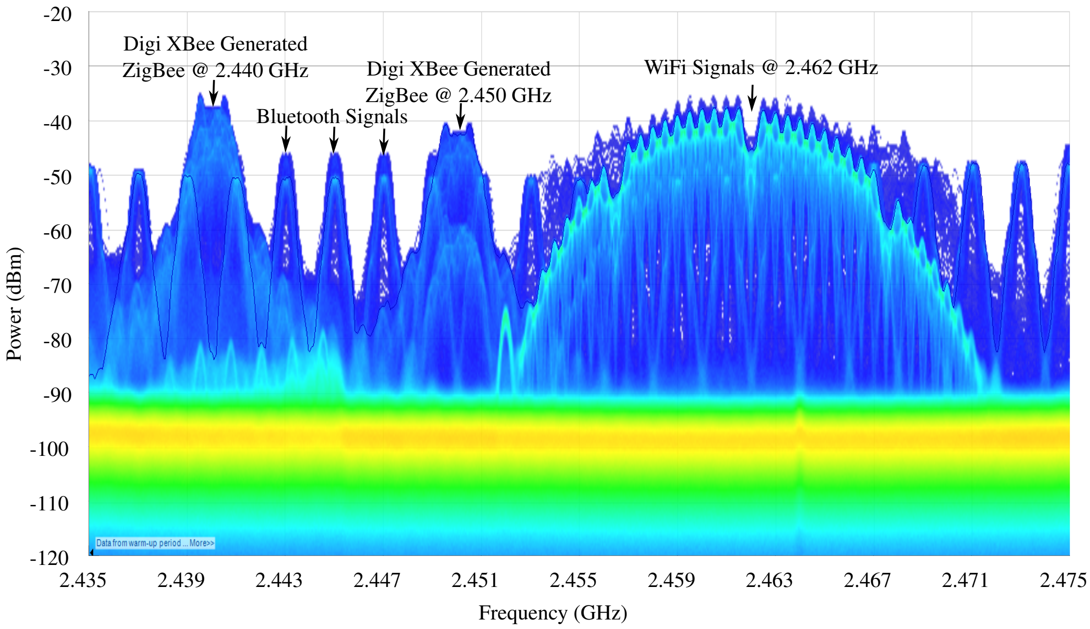


Figure 2.4: An IEEE 802.15.4 signal (ZigBee) coexisting with WiFi, Bluetooth & another separate ZigBee signal in 2.4 GHz ISM RF band, where visualization results from Tektronix's DPX software.

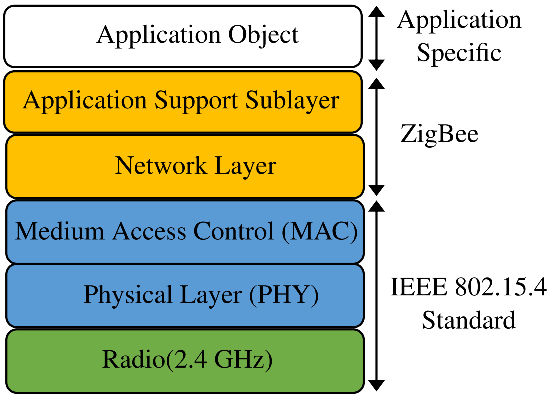


Figure 2.5: A simplified visualization of ZigBee's protocol stack, highlighting the use of the IEEE 802.15.4 protocol, which governs the signal construction and channel access.

ward data to a centralized sink. An example is using nanosatellites as relay nodes (cluster head), allowing access to remote areas by using the nanosatellites as links between each cluster and centralized sink [8]. However, the use of clustering results in the risk of data from multiple devices being affected by attacking the cluster head, which is identified through network reconnaissance. ZigBee is constructed using the PHY and MAC from IEEE 802.15.4 and uses a protocol-specific network layer, application support sublayer and application object layer [37], where this protocol stack is visualized in Fig. 2.5.

2.3.1 IEEE 802.15.4 PHY

The IEEE 802.15.4 PHY layer supports three different frequency bands: a 2.4 GHz band (16 channels), a 915 MHz band (10 channels) and an 868 MHz band (1 channel).

Table 2.1: IEEE 802.15.4 PHY Parameters

Parameter:	**2.4 GHz PHY Value:**	
Number of Channels	16	
Channel Spacing / Width	5 MHz	2 MHz
Data — Symbol Rate	250 kb/s	62.5 $ksymbols/s$
Chip Rate	2 $Mchips/s$	
Modulation	O-QPSK	
Pulse Shaping	Half Sine/Normal Raised Cosine	
Spreading	DSSS	
Maximum Packet Length	133 bytes	

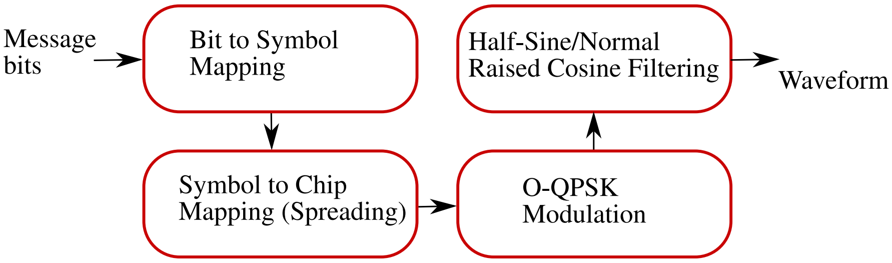

Message bits
Bit to Symbol Mapping
Half-Sine/Normal Raised Cosine Filtering
Waveform
Symbol to Chip Mapping (Spreading)
O-QPSK Modulation

Figure 2.6: A simplified four stage flow graph defining the IEEE802.15.4 2.4 GHz PHY layer, which specifies how the message bits are converted into the transmitted waveform.

Here, the 2.4 GHz band is selected and the 16 available 2 MHz wide channels, which range from 2400→2483.5 MHz and have an inter-channel gap of 3 MHz, have center frequencies as per (2.1), where F_c and i are the center frequency and channel number, respectively. The IEEE802.15.4 PHY layer can be visualized as a four-stage process, as specified in Fig. 2.6, where the associated PHY layer specifications are provided in Table 2.1. This PHY layer uses DSSS to split each outgoing byte into two 4-bit symbols, four most significant bits (MSB) and four least significant bits (LSB). Each symbol is spread to a 32-bit pseudo-noise (PN) sequence from a predefined mapping table of sixteen PN codes and this process is visualized in Fig. 2.7. The chip sequences are modulated using offset quadrature phase-shift keying (O-QPSK) with either half-sine or normal raised cosine pulse shaping. These sixteen PN codes are used at the receiver in an autocorrelation approach to identify the correct received chip sequence. Matlab simulations, using random payload bits, produced the example in-phase and quadrature-phase (I/Q) data in Fig. 2.8 and associated I/Q diagram, which illustrates the constant envelope nature of the signal, in Fig. 2.9. These simulated ZigBee samples identify the offset between the I and Q channels. The samples were investigated further through a comparison with received over-the-air ZigBee samples on the I-channel from a commercial transceiver, as illustrated in Fig. 2.10.

$$F_c = 2405 + 5(i - 11)MHz, \ for \ i = 11, 12, \ldots 26 \tag{2.1}$$

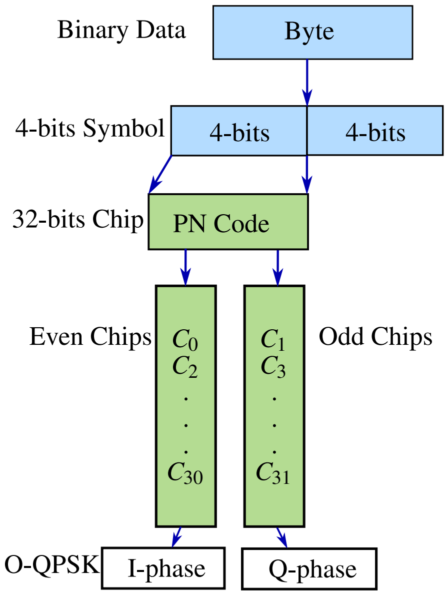


Figure 2.7: A flowchart visualizing the spreading of each byte, where each 4-bits is spread according to the same process.

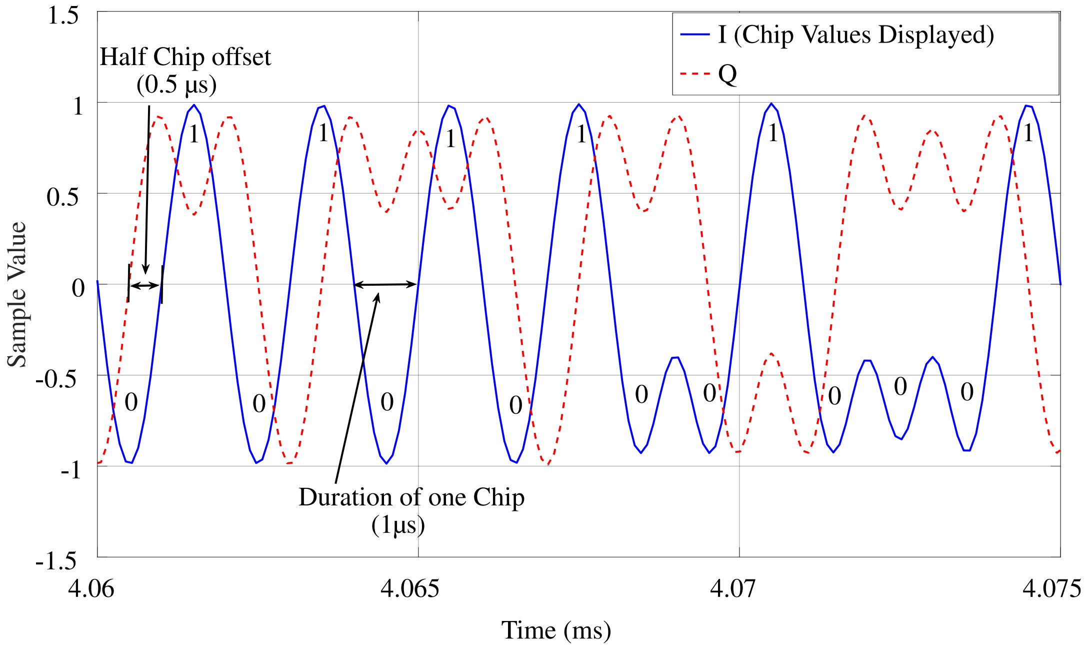


Figure 2.8: Visual representation of transmitted ZigBee signal for simulated ZigBee O-QPSK modulated I/Q data, with the I/Q chip offset and chip duration labeled.

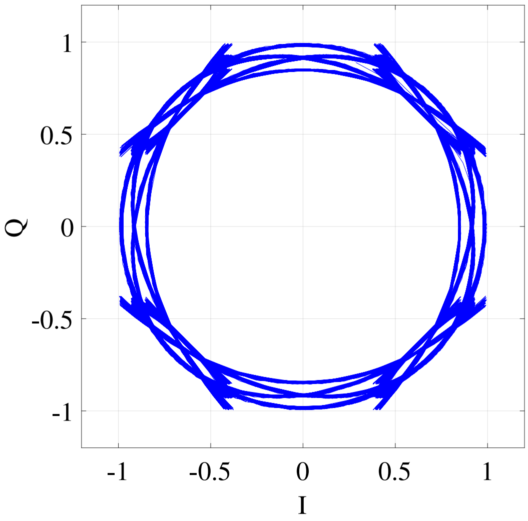


Figure 2.9: Simulated I/Q diagram for the transmitted ZigBee signal, specifying the constant envelope operation.

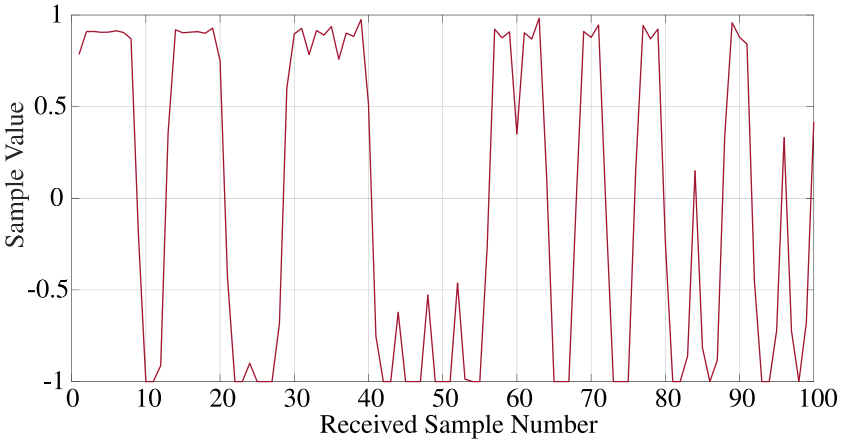


Figure 2.10: Visual representation of transmitted ZigBee signal for received over-the-air ZigBee signal samples on the I-channel, which corresponds well to the simulated samples but includes the effects of real environmental conditions.

The real samples in Fig. 2.10 include the effects of real environmental and hardware conditions. However, the simulated approach matched the sample/signal construction sufficiently, allowing these simulations to be investigated further to provide a deeper understanding of the IEEE802.15.4 PHY layer. This examination enabled the equivalent energy-per-bit (E_b) to be calculated using the period over which one byte is broadcast (T_{Byte}) and (2.2), where C is the signal power in Watts.

$$E_b = \frac{T_{Byte} * C}{8} \; J/bit \tag{2.2}$$

This simulated ZigBee signal was examined by exploiting the ZigBee frame (Table 2.2), under normal operation by simulating the bit error rate (BER) over a zero-mean additive white Gaussian noise (AWGN) channel for a range of energy-per-bit-to-noise (E_b/N_0) ratios. The BER was converted into the associated packet error rate (PER) by implementing the concept that a single bit-error corresponds to a packet error. This approach is valid as no forward error correction technique is applied in ZigBee signals. Thus, if a bit-error occurs, the packet will require retransmission. For example, if an error occurs in the payload, the frame check sequence will fail, while an error in the preamble may prevent synchronization to the packet. The predicted PER was incorporated using a predictive approach calculated using the probability of receiving an incorrect symbol (P_e), given sixteen different DSSS PN codes transmitted in an AWGN channel. Assuming a matched filter receiver, the symbol error probability can be expressed as (2.3) and the PER estimated using (2.4), where σ (2.5) is the channel variance that ensures the required E_b/N_0 is achieved, erf() is the error function, L is the number of codes and N_{Bytes} is the total number of bytes per packet. A matched filter receiver incorporates a correlation function between the received 32-chip sequence and each of the predefined sequences. The PN code with the highest correlation is chosen as the received sequence. The results are shown in Fig. 2.11, where the transmitted ZigBee signal's general performance can be inferred. The energy per bit to noise ratio is a normalized signal-to-noise ratio (SNR) measure known as the “SNR per bit.” As the description implies, E_b is the signal energy associated with each user data bit. As shown in (2.2), it is equal to the signal power divided by the user bit rate. If signal power is in watts and the corresponding bit-rate is in bits per second, E_b is in joules. The associated N_0 is the noise power spectral density, which is the noise power in a 1 Hz bandwidth, measured in watts per hertz or joules. The results express the PER for received packets across an AWGN channel for normal operating conditions. The PER curve indicates the ZigBee signals operate with an arbitrarily small error when the $\frac{E_b}{N_0}$ is above 0dB. However, as will be discussed later, other considerations, including miss-routing of packets, erroneous transmissions or attacks, may occur. The predicted and simulated results begin to differ as the PER reduces because the mathematical model assumes the pseudo-noise codes are orthogonal. In reality, there is a non-zero cross-correlation. From these results, it can be deduced that interference, which effectively alters the SNR, affects WSN signals.

$$P_e = 1 - \int_{-\infty}^{\infty} \frac{e^{-\frac{(-1+y)^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma} \left(\frac{1}{2} + \frac{1}{2}\operatorname{erf}\left[\frac{y}{\sqrt{2}\sigma}\right]\right)^{L-1} dy \tag{2.3}$$

$$PER = 1 - (1 - P_e)^{2 * N_{Bytes}} \tag{2.4}$$

$$\sigma = \sqrt{\frac{1}{2E_b N_o}} \tag{2.5}$$

The general PHY packet structure, known as a PHY protocol data unit (PPDU), is provided in Table 2.2, where the MAC frames passed to the PHY are enclosed

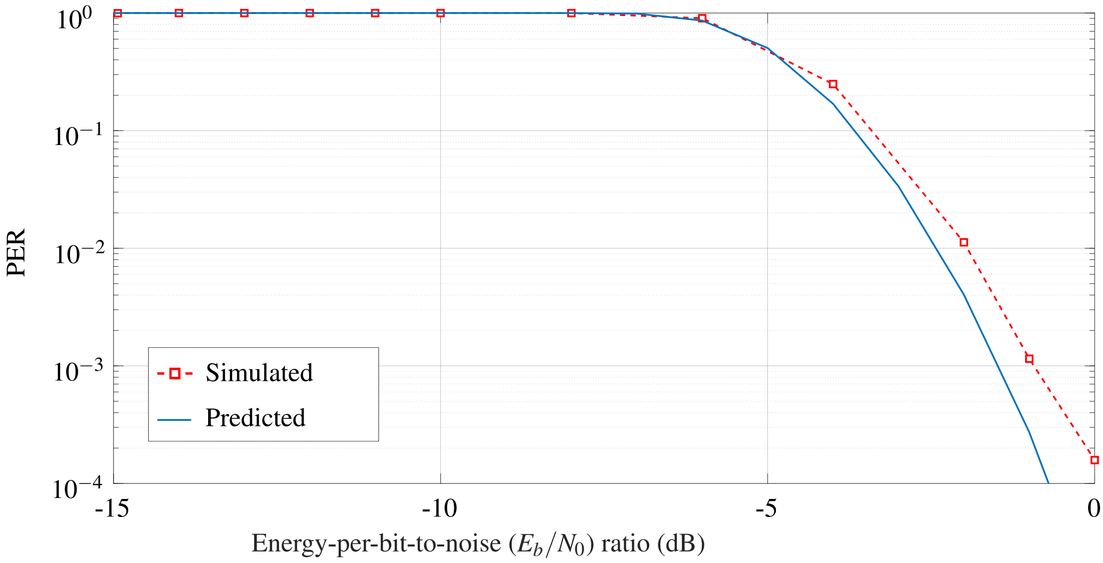


Figure 2.11: Predicted and simulated PER for a ZigBee signal over a range of energy-per-bit to noise ratios.

in the PSDU. This process is visualized in Fig. 2.12, where a simplified expanded ZigBee packet structure is specified. The payload of each frame contains the packet from the preceding level in the protocol stack. The PHY frame is of most interest as it is employed throughout the simulation study in Chapter 4 and the live experimentation in Chapters 6 and 7. In each case, the PHY frame is applied to mimic ZigBee's operation. The SHR contains a preamble, which allows receivers to synchronize and lock onto the packet bitstream, and the start frame delimiter (SFD), which marks the end of the preamble and the start of data. These values are predefined, for example, in ZigBee, the preamble sets all 4 bytes to $0x00$ and the SFD is $0x7A$. The PHR contains the number of bytes in the payload, including the 2-byte frame check sequence (FCS). The maximum IEEE 802.15.4 packet size is 133 bytes, including all headers, but some radios, like the CC2420, allow the preamble to be increased to 17 bytes [38]. From a security perspective, the FCS, and any encryption, are applied to the PSDU payload, meaning the headers are minimally protected.

Table 2.2: IEEE 802.15.4 PHY Frame Layout

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

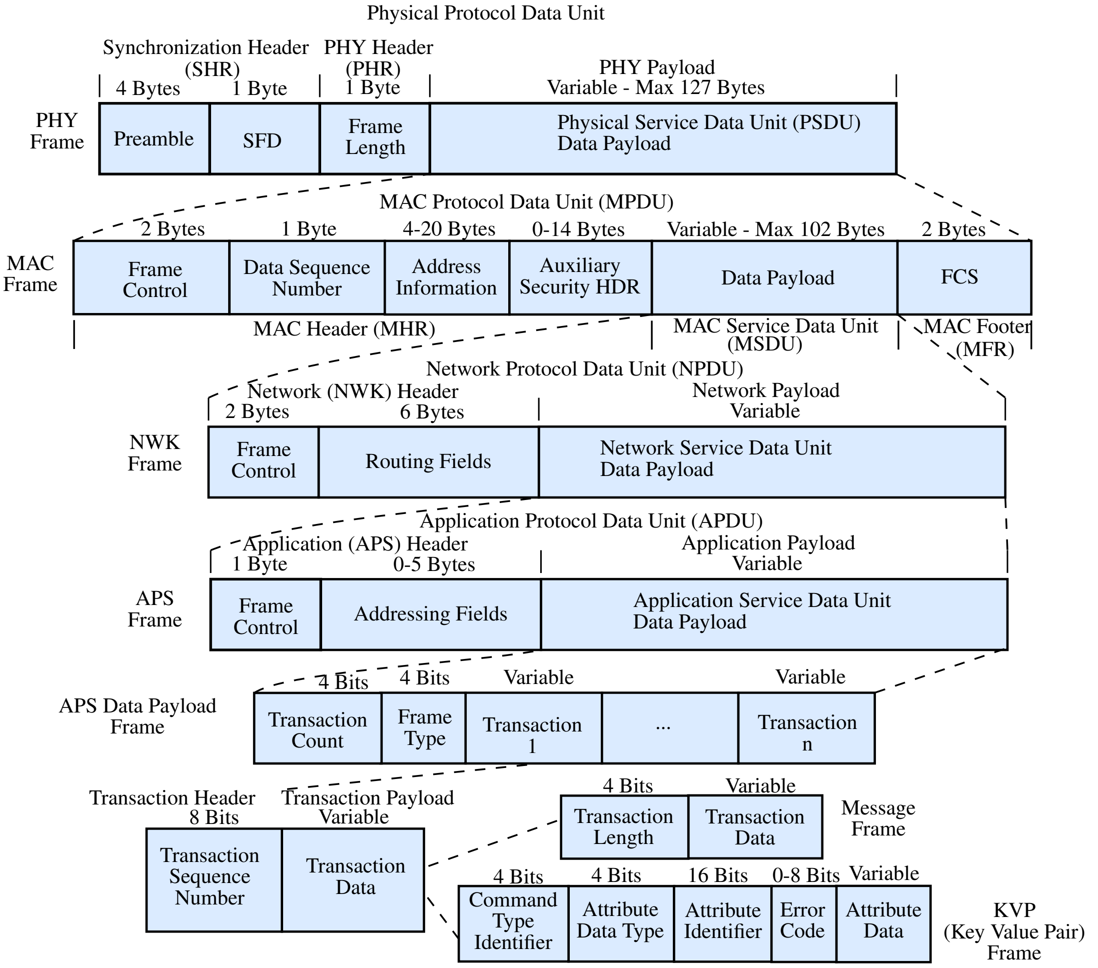


Figure 2.12: A simplified ZigBee frame structure visualizing the typical data output of a traditional packet sniffer by specifying frame specific bytes.

2.3.2 IEEE 802.15.4 MAC

These ZigBee signals are transmitted at center frequencies in the unlicensed ISM frequency band. The WSN signals must coexist with various signals, including Bluetooth, numerous LR-WPAN, WLANs and wireless metropolitan area networks. Due to the unlicensed operation, global availability and relatively long-range, the ISM frequency band is the first choice for wireless LAN solutions, including WSNs. Primarily, the IEEE802.15.4 MAC layer allows multiple devices to use the same physical radio channel by employing carrier sense multiple access with collision avoidance (CSMA/CA) [38]. Prior to transmitting a packet, devices perform a clear channel assessment (CCA) to ensure the channel is available. This decision is based on either energy detection, which uses the received channel energy to compare against a predetermined maximum threshold, or carrier sense, which identifies the occupying signal and, if an IEEE 802.15.4 signal is sensed, then the channel may be busy, even if the energy threshold is not exceeded. If the channel is busy, devices back off for a random period and try again, up to a user-defined maximum number of retries. As a result, this technique is particularly vulnerable to interference attacks and spectrum-sharing difficulties.

Two modes of operation are used: beacon-enabled and beaconless. In beacon-enabled mode, the coordinator transmits regular beacons used for synchronization and communication control. A superframe, divided into equal slots, is used to synchronize data transfer between devices and the coordinator by identifying active and inactive periods. Communications occur during the active period, which may consist of a contention access period (CAP) and a contention-free period (CFP). Nodes enter low-power mode during the inactive period. In a CAP, all devices use slotted CSMA/CA and the first device that identifies channel availability starts transmitting. A CFP uses guaranteed time slots (GTS) and occurs at the end of the active period, immediately after a CAP. In the beaconless approach, communications use unslotted CSMA/CA and the PAN coordinator does not transmit beacons, which means devices cannot be synchronized with one another and no GTS exist. Acknowledgement frames are sent without using CSMA/CA and are not encrypted [36].

A simplified IEEE 802.15.4 MAC data packet is provided as part of the overall ZigBee packet structure in Fig. 2.12. It specifies how the PSDU encloses the MAC protocol data unit (MPDU). The MAC header (MHR) contains information such as addressing and security, the payload includes data or commands and the FCS is an error-detecting code used as a security technique for data verification. The auxiliary security header is optional and incorporates information required for security processing [39]. Similarly, the MAC service data unit (MSDU) encompasses the network frame and the network payload encases the application frame, which incorporates a message integrity code. Other MAC frames that exist are the beacon, acknowledgement and command frames, which are out of scope here but are explained in detail in [40]. For this thesis, the main takeaway from the MAC layer operation is the susceptibility to interference due to the use of CSMA/CA. As a result, edge nodes that can detect interference in received packets or the channel's dominant signal have benefits for channel access and security.

2.4 Methodology Development: Global Positioning System

A second frequency and signal model were required to validate the interference diagnostic framework developed in this thesis. Successful operation across multiple frequency bands and signal models makes a developed solution more amendable to practical application. As a result, the Global navigation satellite system (GNSS), particularly GPS, signals were examined. GPS is defined as a space-based satellite navigation system that provides positioning, navigation and timing information and services in all weather conditions, anywhere on or near the earth where there is an unobstructed line of sight to four or more GPS satellites. These signals are becoming increasingly crucial for civilians, services and industries. This need is due to the dependence on the provision of GNSS-derived location and time measurements and, so, it is becoming increasingly important to protect this vulnerable wireless service. This study uses GPS signals, on account of previous work in detecting interference in GPS signals [41] and to demonstrate the designed interference detection methodology's transferability. Both unintentional and malicious in-band interference are the single most significant threats to GPS (GNSS) applications and users. The signals are received at extremely low power levels (typically -125 dBm) and, despite the protection offered by their code-division modulation (similar to IEEE 802.15.4), the signals are readily impeded by spurious emissions from terrestrial sources.

Apart from the inherent susceptibility to unintentional interference, the widespread availability of low-cost jamming hardware, through any one of a growing number of international internet-based retailers, poses a significant problem. To tackle this threat, a methodology for detecting and reporting on GNSS jamming events, which can be deployed as frequently and as widely as the jamming threats themselves, is warranted, which is complementary to WSN resource-constrained edge devices. Similar to WSNs, GPS and IoT can be linked together as it can be beneficial to monitor both the condition and location of edge nodes. Previous work on GPS signals [41] focused on the development of a low-cost hardware and software platform designed for the automatic detection and reporting of radio-frequency interference affecting users of GPS. The prototype was built satisfying the requirements of being available as unmodified, off-the-shelf components, within a single build budget of €100 and implemented a machine-learning based detection scheme, based on Random Forest [23]. These requirements still hold for this thesis, as the envisaged implementation platforms are resource-constrained embedded wireless edge nodes. This thesis expands on the previous study through data collection from all available satellites, classifying the interference signal and upgrading the machine learning classifier. Therefore, the problem space and vulnerabilities in GPS are comparable to WSNs, GPS is applicable to IoT scenarios and the signal structure will be shown to utilize DSSS spreading techniques. Consequently, the choice of GPS is complementary to the WSN use case, which motivates the transfer of the developed WSN diagnostic tool to GPS applications.

The GPS network consists of a constellation of 31 potentially heterogeneous satellites in orbit, which is maintained to have at least 24 satellites available at all times. This constellation effectively has 27 satellites available due to the recent addition of an extra three to improve coverage. These satellites are arranged into six equally spaced orbital planes, with each plane containing at least four satellites, as visualized in Fig. 2.13. The USA's NAVSTAR GPS satellites incur coexistence issues due to the several

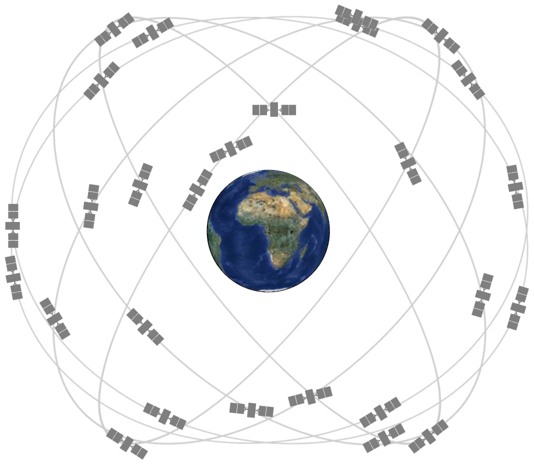


Figure 2.13: A visualization of the GPS satellite constellation diagram, showing the six orbital planes [42].

GNSSs in operation, including Europe's Galileo, Russia's GLONASS and China's BeiDou. The GPS constellation and satellite maintenance ensure at least four satellites are continually available without obstruction anywhere on or near the Earth. The satellites fly in a medium earth orbit at an altitude of approximately 20,200 km, meaning each satellite circles the Earth twice a day. To compute position, a receiver must observe at least four satellites, as four unknowns (time and position vectors x, y and z) need to be determined. Thus, for an error-free reception, at least four distinct satellite signals need to be processed. From these four pieces of information, longitude and latitude can be determined and, therefore, the user's position. The heterogeneous nature of the system is a result of using different generations of satellites simultaneously. As old satellites become degraded and obsolete, newer satellites are developed and put into orbit to replace these degraded decommissioned satellites. A few of these decommissioned satellites are held in orbit as a back-up if they are ever needed to be re-commissioned due to unexpected failures. This use of the heterogeneity of operating devices is similar to WSNs, adding another complementary factor between the applications. The entire network is maintained by the control segment, consisting of a global network of ground facilities. These facilities track the satellites, monitor each satellite's health and performance, monitor transmissions, perform analysis, and send commands and data to the entire constellation. However, individual GPS receivers do not have such a monitoring process, suggesting the need for edge devices to detect and report interference events.

For this study, the GPS signal model of interest is the L1 civilian signal derived from the fundamental clock frequency of $f_0 = 10.23MHz$, operates in the UHF band and has a center frequency as per (2.6). The signal is composed of three parts, the carrier wave (2.6), the navigation data, which has a bit rate of 50 b/s, and the spreading sequence, where each satellite has two unique spreading sequences or codes, the coarse acquisition code (C/A) and the encrypted precision code (P(Y)). Here, the C/A codes, which belong to the family of Gold codes, are analyzed and contain a PRN (Pseudo Random Noise) sequence of 1023 chips, which is repeated every milli-second, producing a 1.023 MHz chip rate.

$$f_{L1} = 154 f_0 = 1575.42\,MHz \tag{2.6}$$

$$\begin{aligned} s^k(t) = {} & \sqrt{2P_C}\Big(C^k(t) \oplus D^k(t)\Big)cos(2\pi f_{L1} t) + \\ & \sqrt{2P_P}\Big(P^k(t) \oplus D^k(t)\Big)sin(2\pi f_{L1} t) \end{aligned} \tag{2.7}$$

Each transmission is phase modulated using binary phase-shift keying (BPSK). However, unlike typical quadrature amplitude modulation (QAM) systems, where a single bitstream is split into two half-symbol-rate bitstreams, in GPS signals, the in-phase and quadrature-phase components are modulated on separate (but functionally related) bitstreams. This concept is specified in (2.7), where P_C and P_P are the powers of signals with C/A or P(Y) code, respectively, C^k is the C/A code sequence assigned to satellite k, P^k is the P(Y) code sequence assigned to satellite k and D^k is the navigation data. This method produces a spectral shape described by the sinc function with a channel width proportional to the chip rate of 2.046 MHz. All the signals in the system use the same center frequency and, to acquire an individual signal, the code of that signal must be used to correlate with the received signal. Thus, a code division multiple access (CDMA) approach is used with DSSS and bi-phase modulation of the carrier frequency. As a result, the GPS signal structure employs DSSS signals for ranging and CDMA for multiple access.

The DSSS technique spreads the bandwidth of the data signal uniformly for the same transmitted power, as the locally generated PRN code runs at a much higher rate (chip rate) than the data to be transmitted (data rate). The data for transmission is spread across the larger bandwidth using the PRN code. The data signal, with an “on” (+1) pulse duration of T_b, is XOR added with the code signal that has a pulse duration of T_c. The codes’ finite field is based on the binary field and results in modulo two operations, where addition corresponds to XOR. The bandwidth of the data and spread spectrum code signals are $\frac{1}{T_b}$ and $\frac{1}{T_c}$, respectively. Since T_c is much smaller than T_b, the bandwidth of the spread-spectrum signal is much larger than the bandwidth of the original signal and hence the spread spectrum method, where the ratio $\frac{T_b}{T_c}$ is called the spreading factor of the designed DSSS system.

The CDMA method is used so that the receiver can access multiple satellites simultaneously. The CDMA method’s optimal performance occurs when there is good separation between the different signals being received by the user or the different PRN codes being used. The separation of the signals is made by correlating the received signal with the user's locally generated code. The user's GPS receiver knows the PRN of each satellite and, therefore, can receive a signal from each satellite in orbit. If the locally generated PRN code matches the received signal, then the correlation function will be high and the signal can be extracted. If the locally generated code has nothing in common with the received signal, then the correlation function will be close to zero. This approach is how the GPS receiver works and can extract signals from every satellite by receiving the available signals and correlating the known PRN codes to identify the satellite and extract the signal. If the locally generated code is correlated with the received signal at any time offset other than zero, the correlation will be close to zero. This process is known as "auto-correlation" and is also applied in WSNs as sixteen codes are used to determine the received bits. A practical application of this receiving method is used in the simulated receiver in the WSN experimentation in Chapter 4.

The GPS signals of interest in this thesis can be summarized as the C/A code L1 band signals that have specifications as follows:

- L1 = 1.57542 GHz
- Data Rate = 50 bps
- C/A code length = 1023 bits
- Chip Rate = 1.023 Mcps

These figures result in a bandwidth of 2.046 MHz, which corresponds to 1575.42 MHz ± 1.023 MHz. As the data is spread across the I- and Q-channels, a sampling rate corresponding to the GPS bandwidth is sufficient to receive GPS data. This operation results from being twice the rate of the individual channels. As a result, a sampling rate of 2.048 MHz is applied in Chapters 5 and 7.

The received GPS data signal frame (or navigation message) comprises of three primary sub-frames. The entire message contains five sub-frames, where each sub-frame is 300 bits long. The received signal frame structure, and a description of each navigation message's primary sub-frames, are provided in Table 2.3. Since each sub-frame is 300 bits and the data rate is 50 bps, it takes six seconds to transmit each sub-frame, which implies that an initial connection to the satellite takes a considerable amount of time. As these GPS signals' data rate is much slower than the WSN data, the received packets are received over a longer period. As a result, the data analysis of received GPS data should occur over a larger number of I/Q samples, compared to the WSN approach. This approach ensures that GPS data is analyzed and not instances of noise. This use of a larger number of samples will be eluded to in the interference detection and classification results in Chapter 7.

Table 2.3: GPS Received Navigation Message Breakdown

Sub-Frame	Description
1	GPS date and time Satellites' State and Health
2 & 3	Ephemeris Data: Precise Orbital Information Receiver can calculate Position
4 & 5	Almanac Data Information and Status concerning all satellites Satellite Network Synopsis Error Correction

2.5 Security Requirements

Security defines various characteristics, which protect a network from attacks, especially when sensitive information is transmitted, ensures privacy and permits safety and reliability. WSN applications require security, particularly when the networks are designed for use in hostile environments, military, aerospace, commercial or IoT applications [43]. This concept can be extended to any wireless device which transmits sensitive data. As a result, the communication link's security and availability and the successful delivery of authentic and confidential packets are essential for any safety-critical wireless system (see Section 2.2). Fundamentally, transmitted data needs to be free from unauthorized intrusion and all services need to operate when requested. As this thesis focuses on interference, certain operating assumptions are made, which focus on how wireless networks react to an intrusion. Traditional wireless network operation, typically transmitted packets, and attack methods were examined and the assumptions adopted herein are as follows:

1. A reliable routing protocol is used and a packet can always reach the base station and other nodes when no attacks are present [44].
2. Basic jamming hardware in use may be similar to network nodes [44] but, notably, does not have to adhere to any standards, guidelines or rules.
3. Attackers can use advanced hardware (e.g., software defined radios, computers) without adhering to standards, guidelines or rules. [45]
4. The attacker can place/seize one or more basic sensing nodes in the network [44]. These basic sensing nodes have limited resources and energy supplies, which hinder the use of complex security algorithms. Control nodes contain more advanced hardware and, as a result, are more difficult to seize.
5. Nodes at the edge of a jammed region can receive messages from "jammed" nodes and relay an alarm to the controller and/or base station [19].

6. Intelligent jammers can monitor the network and determine the protocols being used [46].

7. Nodes can be deployed in environments where the possibility of being captured exists [47]. Captured (malicious) nodes can be used to implement attacks on the network and access sensitive data.

Application-specific assumptions also exist, for example, encryption and/or a key management system for data privacy may be of high importance in some applications. In contrast, other systems might implement origin authentication and data integrity but not encryption. Specific applications may not use any mitigation strategies, while critical applications may use DSSS, frequency hopping spread spectrum or a frame check sequence to fortify against external interference. Consequently, an application's environment and the prevailing external factors will typically govern operating conditions. This thesis focuses on developing an interference diagnostic framework for wireless edge devices, which results in focusing this security discussion on edge device's vulnerabilities and security requirements.

2.5.1 Vulnerabilities

When WSNs, specifically the edge nodes, are analyzed in terms of their construction, deployment and usage, certain unique security vulnerabilities become apparent. As a result, when compared to other wireless networks, securing WSNs to an appropriate level is more challenging as, typically, WSNs have certain unavoidable challenges. Furthermore, WSNs are susceptible to various attack styles, including jamming, eavesdropping and tampering, as discussed in Section 2.6. Therefore, when meeting the necessary security requirements, it is clear that WSNs will not be 100% secure. As WSNs become more integrated with modern applications, possibilities exist that security enhancements are required. These vulnerabilities, partially identified in [48], are expanded and summarized below:

- Open Interface: Generally, WSN protocols are unavoidably known publicly due to the requirement for interoperability between devices in operation and the use of standardized open access protocols. Additionally, the wireless channel is open to anyone with suitable equipment, resulting in increased susceptibility to specific WSN attacks and access to transmitted signals.

- Device Resources: Typically, WSN devices are deployed, left unattended, operate on a constrained energy supply and, for reasons of cost, have low processing power, memory, physical storage, and speed. Generally, these constraints hinder the use of conventional computationally expensive protocols/techniques. COTS devices are seen as relevant given the general trends in using these devices in applications where high redundancy and high replenishment rates are favored over custom-built components.

- Hostile Environments: WSNs are regularly deployed, typically remotely, and left unattended in harsh environmental conditions without any fixed infrastructure, where it is difficult to have continued surveillance [49]. This strategy means deployed legitimate nodes are potentially physically available to being captured and/or tampered with by attackers, leading to a high probability of node secrets being discovered and specific nodes becoming malicious. Tamper proofing nodes is possible, but, for reasons of cost, may not be appropriate and, so, encryption keys may be obtained from device memory. Typically, the physical environment contains varying fading levels, obstacles, path losses and spurious interference, while the RF spectrum changes rapidly as it adjusts to the number of connected devices, demand, packet size and services in operation. These non-malicious factors, coupled with nodes' availability, increase the probability of network compromise, thereby motivating countermeasures at the edge.
- Topology [49]: WSN topologies can be dynamic and change due to variations in the wireless channel/environment (fading levels, obstacles, path losses, spurious interference etc.), the natural dynamic nature of WSNs or damage/"death" of network nodes. This WSN feature can be exploited by potential attackers who wish to gain access or cause spurious harm to the network.
- Hardware Availability: As hardware becomes increasingly available at more cost-effective prices, potential attackers can prepare and develop attacks using real-world WSN hardware, which provides an increased chance of attacker success. Additionally, the computational ability of available devices is expanding, leading to advanced attack styles. Reconfigurable hardware, such as, for example, software-defined radios, is becoming increasingly available/accessible to a wider set of users/potential malicious actors, who can readily design and deploy more computationally expensive attacks.
- Deployment Diversification: As the application space of WSNs continues to expand into new frontiers, a more diverse range of deployments becomes the norm. WSNs were traditionally involved in monitoring applications but have now extended into space operations, WBANs and unmanned aerial vehicles (UAVs) etc. The potential uses and critical data of these innovative applications create security and spectral coexistence challenges.

2.5.2 Requirements

For deployed security techniques to provide robustness against interference, ensure data privacy and permit safety and reliability, certain security features are required [50], [48]. These essential elements of security are as follows:

- Confidentiality: The secrecy of sensitive transmitted data in the wireless channel must be maintained by keeping the contents from all but those authorized to have it. Classical cryptography can be adopted to encrypt critical parts of the transmitted packets before transmission such that only authorized nodes can decipher them. A strict key management system is essential as privacy attacks can degrade confidentiality and include eavesdropping and tampering. However, implementing a strict key management system may prove difficult, given WSN device resources.

- Authenticity: Verifying packet authenticity is essential as the receiving node should be able to autonomously assert that the received packet has not been modified in transit (data integrity) and from which node the packet originated (origin authenticity). Cryptographic schemes, such as digital signatures, can simultaneously provide both functionalities. Without this security aspect, attackers could spoof node identities and spread false information throughout a network.

- Availability: WSNs need to provide services whenever they are required, resulting in a need to exhibit qualities of robustness against a variety of impairments, both benign and malicious. Some degree of resilience (i.e., the ability to recover from faults), diagnostics (i.e., if services become unavailable, it should be possible to identify why), or mitigation (packet re-routing, channel switching) is necessary. The appropriate use of an interference diagnostic tool at the edge may help ameliorate the network's availability.

- Energy: Unique to wireless edge nodes, the constrained energy levels impact all security plans. Any security protocol or detection mechanism needs to consider this energy constraint since optimizing energy usage is vital for network longevity. This requirement affects the ability to ensure each basic security requirement and the use of computationally expensive algorithms.

- Data Freshness: Critical data circulating in a WSN must be the most recent update and, as such, outdated data should not circulate in a network.

- Node Ability: Wireless edge nodes must be self-organizing, react to node/link failures and only authorized nodes should be allowed to operate and share information in a WSN. This security concept motivates edge nodes to make real-time decisions to understand link failures and detect and report the presence of interference.

2.6 Wireless Attacks

Attacking a wireless network involves either unauthorized access to data, data manipulation or denial of system services. These attacks can be categorized into either passive or active attacks [50]. Passive attack styles do not modify information or messages but, instead, aim to learn the transmitted confidential data. Initially, this does not appear to have severe consequences, especially if data is encrypted. However, over time and given enough captured data, reverse engineering can provide the protocol in use and potentially grant network access or packet decryption, resulting in multiple network security consequences. In contrast, active attacks aim to modify/remove streams of data, cause a denial of service, disturb functionality or disguise an attacker as a legitimate node. For convenience, a selection of known attacks on WSNs are categorized and described, where the focus is placed on PHY and MAC layer attacks, including jamming and congestion style intrusions. Generally, it is envisaged that external attacks, for example, jamming, can be implemented using a software-defined radio approach. This use of software-defined radios as interference transmitters is discussed and implemented in Chapters 5 and 7. This type of hardware provides the necessary ability to receive, analyze and transmit wireless data. In contrast, internal attacks, for example, sinkhole (see Section 2.6.5), will typically use a network device that has been captured or identified. Attack effectiveness and/or affected area typically depend on the strength of the transmitting power or how “transparent” the approach needs to be.

At present, WSNs and associated IoT systems are susceptible to attacks, especially Denial of Service attacks, for example, jamming and eavesdropping attacks (e.g. replay attacks). These wireless attacks (denial, deception and/or destruction) have traditionally been the domain of Electronic Warfare (EW) [51]. However, these techniques are gradually being adopted for criminal activities as readily available hardware supports the development of effective systems that can match jamming prevention techniques. Additionally, network compromise, whether malicious or unintentional, is achievable due to these EW techniques and the existing unavoidable security vulnerabilities. Threat detection and analysis need to match advancing attack strategies [51], while not overly consuming device resources. This section will introduce the general attack strategy, describe the conventional wireless attacks and the typical defense strategies implemented by wireless nodes and protocols.

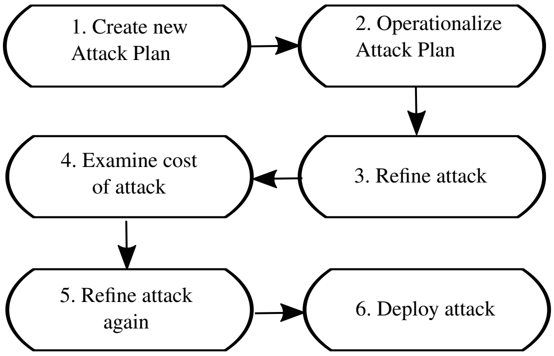

1. Create new Attack Plan
2. Operationalize Attack Plan
3. Refine attack
4. Examine cost of attack
5. Refine attack again
6. Deploy attack

Figure 2.14: Flowchart depicting a general attack creation strategy.

2.6.1 General Attack Approach

Initially, the general concept of developing a strategy to attack a wireless network is established to understand what is involved and what differentiates attack design from the industry that develops the wireless protocols. In terms of design and application, attackers are generally more agile, timely and less constrained (in terms of obeying protocols and laws) than their industrial counterparts. This approach allows attackers to focus on creating only what they truly need and disregard all laws and rules relating to the use of hardware, RF spectrum etc. For wireless networks, attackers can take advantage of the security issues described in Section 2.5.1 to develop new or modified attack plans. A general attack strategy is provided in Fig. 2.14 [52], where the time-line is only a few weeks and the cost of the attack relates to (2.8). Scalability refers to how deployable a specific attack is and the takeaway quantifies the gain achieved by using the attack. This approach, the availability of hardware and the extensive set of potential techniques means attack styles are hard to predict and have relatively short development timelines.

$$Attack\ Value = Scalability * Takeaway \tag{2.8}$$

However, an ever-present wireless attack is external interference as the wireless channel is available to any attacker with suitable hardware. This fact results in this thesis's work focusing on interference, for example, the matched protocol attack. This intelligent deceptive jammer attack style uses the attack plan shown in Fig. 2.15, where the identifier may not necessarily need to identify the exact protocol in use but just recognize enough to match the spectral identity and cause packet collisions. This learning-based attack determines the signal and frame structures based on monitoring the spectrum and frequencies in use and eavesdropping on transmitted packets. For example, a 2 MHz bandwidth signal at specific frequencies in the 2.4 GHz ISM RF band with a spectral visualization as seen in Fig. 2.4, indicates the use of the IEEE802.15.4 protocol. The associated operating channel is determined using (2.1), which is available from

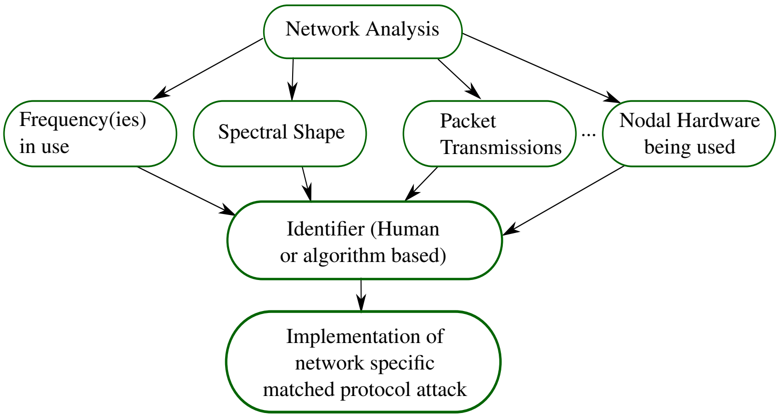

Network Analysis
Frequency(ies) in use
Spectral Shape
Packet Transmissions
...
Nodal Hardware being used
Identifier (Human or algorithm based)
Implementation of network specific matched protocol attack

Figure 2.15: A visualization of the general attack plan associated with the matched protocol attack, where the aim is to maximize packet loss while only transmitting expected spectral signals.

the standardized protocol literature. Packet eavesdropping, received power levels, and moving around a specific area where nodes are suspected to be located (higher power levels are expected in areas enclosing network nodes) are additional techniques to identify network operation. All of the acquired information is used in the attack plan, which results in the implementation of a network-specific matched protocol attack. Similar to other jamming attacks (see Sections 2.6.2 and 2.6.3), this attack causes collisions by introducing interference and blocks the CCA from determining a free channel. However, this matched protocol attack mimics legitimate network signals, resulting in the spectrum only exhibiting legitimate signal models. This network performance results in either packet loss or retransmitting of packets, causing an avalanche of problems that affect all levels of the communication stack [2]. For example, extra traffic introduced by retransmissions and attack packets can lead to link prediction fluctuations, path changes in routing protocols, applications buffering too many packets and certain nodes becoming unreachable, leading to, in extreme circumstances, a complete collapse of the network [2]. As a result, interference is a practical threat to wireless edge nodes, which requires detection, since if the attack is detected, it can, typically, be mitigated. This observation motivated this thesis's work to concentrate on interference diagnostics, to detect both subtle and crude interference forms.

2.6.2 Conventional Jamming Attacks

These active attacks, typically, aim to overpower the legitimate signal with spurious radio-frequency transmissions. While higher jamming power increases attack effectiveness, it also boosts detectability. The adversary is typically driven to optimize signal interference to maximize packet loss while minimizing total broadcast power. Such attacks include:

- The constant jammer continuously emits RF signals containing random data into the wireless medium without following any MAC protocol, can be readily detected and is energy inefficient. However, this jammer can be easily implemented and causes severe damage to a WSN, as congestion or destruction of packets can be achieved and the channel can appear permanently busy.

- The deceptive jammer regularly transmits protocol-specific packets into the network without pausing between successive packets, thereby preventing normal sources from transmitting successfully. Due to the transmission of legitimate packets, it is more difficult to detect than a constant jammer. It can cause considerable damage in WSNs adhering to the MAC protocols sensing for channel access or the presence/absence of a signal.

- Random jammers sporadically transmit random packets of data and conserve energy by switching between the jamming state, when jamming signals are emitted, and the sleeping state, when all transmissions are ceased. This unpredictable behavior makes this jammer challenging to mitigate and can cause similar damage as the constant and deceptive jammers.

- A reactive jammer [53] operates in idle mode until some legitimate activity is detected on the wireless channel. A RTS/CTS jammer detects request to send (RTS) messages and interferes with the channel to block any clear to send (CTS) messages, thereby denying further communications. Data acknowledgement jammers corrupt acknowledgement packets after a transmission has been sensed in the network and misleads nodes to decide that packets were undelivered, thereby invoking a retransmission and, potentially, resulting in the exhaustion of the power supply. This attack is particularly effective in protocols, such as ZigBee, which use CSMA/CA.

- Specific function jammers perform explicit functions, depending on their calibration, and cause jamming on either a specific channel or across an entire network while minimizing their energy consumption or maximizing their attack effect. For example, follow-on jammers jam one specific frequency at a time and maximize packet loss by continuously hopping between the channel frequencies. These jammers can be detected but are very effective, particularly in networks that use frequency hopping spread spectrum or when identified spectrum holes [54] are used to improve performance through spectrum sharing. Another example is the channel-hopping jammer, which follows a predefined pseudo-random sequence of channels and starts jamming at different time slots according to this sequence. By overwriting the sequence, multiple channels can be jammed at the same time. Finally, pulse noise jammers can be programmed to switch between different channels/bandwidths and conserve energy by temporarily halting transmissions.

2.6.3 Intelligent Jamming Attacks

Intelligent jammers are a combination of a passive and an active attack, as the jammer initially targets network privacy before inevitably targeting data packets. These devices are more likely to cause jamming but are more challenging to implement than conventional jammers. Protocol aware and statistical jammers aim to determine the MAC protocol being used by the victim's network to launch energy-efficient attacks [46]. Protocol aware jammers know the MAC layer operating rules and can deprive legitimate nodes of access to the channel and can, potentially, affect services identifying free channels or spectrum holes, used to, potentially, enhance spectrum coexistence [54]. Statistical jammers observe the packet inter-arrival time distribution and, based on its estimation, emit pulses of jamming signals to disrupt communications (DoS attack). Once the estimation is achieved, energy efficiency can be increased through pulse jamming. Collision makers target the identified acknowledgement packets by inhibiting transmissions. Specific intelligent jammers identify the cluster head/sinks by monitoring the network traffic and focus attacks on that specific node in an "Intelligent Cluster Head Attack." Learning-based jammers, like LearJam [55], have been produced to attack low duty cycle networks where nodes sleep most of the time (a typical WSN characteristic) and consist of a learning phase, wherein the node transmission pattern is observed, and an attacking phase, where these transmissions are compromised. Therefore, attackers can now learn the MAC and/or protocols in use by eavesdropping (privacy attack) on the channel for some period of time. This attack style could, for example, be launched on techniques for sensing the presence or absence of a signal (CSMA/CA or spectrum sharing) by learning when service should be idle and producing "dummy" packets to avert potential transmissions.

2.6.4 MAC Layer Jamming Attacks

Initially, these are passive attacks that react to the network protocol in use by eavesdropping on, or "sniffing", transmitted packets to gain access to network information. The analyzed results are utilized to implement active attacks, including replay attacks, spoofed packets or forcing a device to remain in listening mode, exploiting CSMA/CA. These are not jamming attacks but, instead, try to mislead wireless devices. A replay attack (also known as a playback attack) is a form of network attack in which valid data transmission is maliciously or fraudulently repeated or delayed. The use of message integrity codes should negate replay attacks. However, due to hostile deployment scenarios, secrets may be accessible as legitimate nodes may be physically available and, if no key management system is in use (e.g. in a typical home personal COTS network), devices may be available commercially and the keys extracted from device memory. Spoofing is a specific type of cyber-attack in which someone attempts to use a computer, device, or network to trick other computer networks or devices in a network by masquerading as a legitimate entity. As a result, these MAC layer attacks aim to achieve similar results to jamming attacks and apply different strategies that mislead rather than block device transmissions.

2.6.5 Network-Layer Attacks

Generally, these are active attacks that interfere with network operations, causing either a DoS, privacy or an impersonation attack.

- Sinkhole/Blackhole Attacks: In this congestion based DoS attack, a malicious network connected node acts like a black hole [56] and "pulls in" all of the traffic in the network. The malicious node listens to available route requests and replies that it has the shortest path, maximizing packet flow through the malicious node, resulting in maximum packet loss or unauthorized access to data.
- Selective Forwarding: Networks that rely on multi-hop transmissions require all nodes to faithfully forward any received packets to the base station. In this packet dropping DoS attack, a malicious network connected node in the routing path selectively drops sensitive packets. This attack could be particularly effective in wireless network control systems, where real-time responses to sensor data or control system inputs are critical.
- Node Replication Attacks: In WSNs, nodes are often deployed in unattended public environments where continued surveillance is unrealistic. In this impersonation attack, an attacker may replicate a legitimate node and introduce it to the network, thereby gaining access to the flow of packets throughout the network. This approach may involve capturing and analyzing a legitimate node when some cryptographic security level is applied. As a result, additional malicious node attacks can be implemented.
- Sybil Attacks: Many applications require node collaboration to accomplish a specific task. Applications can then implement management policies to distribute sub-tasks to different nodes. In this impersonation attack, a malicious node will pretend to be more than one node simultaneously, using the identities of other legitimate nodes to effectively cause collaboration processes to fail and can target data aggregation, routing mechanisms, etc.
- HELLO Flood Attacks: Often, routing protocols need to broadcast "HELLO" packets to discover one-hop neighbors. The attacker exploits this concept to attract and persuade nodes that an attacker is their neighbor. This method is especially effective if the attacking node has an extensive radio range and enough

processing power to flood an entire area of a network, affecting a large number of nodes and persuading these nodes to use the attacker as a relay node in the process. Packets are lost in this energy consumption DoS attack due to, for example, distances being too large for transmission as a node will try to transmit to a non-neighbor (attacker).

- Wormhole Attacks: An attacker records the packets at one location in the network and tunnels those packets to another area in the network using a long-range wireless channel or optical link. Attackers offer fewer hops and less delay, which entices nodes to use the attacker to forward packets, causing collisions and packet loss in this DoS congestion attack.

- Spoofing: Network nodes can become malicious and provide an attacker network access when nodes are physically available in environments without continued surveillance. Each node is not tamper proof due, generally, to cost reasons. Spoofing is the method of disguising a communication from an unknown source as being from a known, trusted source. It can severely harm any wireless or wired network as it is both difficult to detect and effective. A spoofing situation can involve either an attacker successfully identifying as a network node by falsifying data or transmitting falsified data with real credentials from a malicious node. This type of attack is difficult to detect and requires a specific dedicated intrusion detection system to identify node anomalies.

It is clear from analyzing the above attacks that a detection algorithm which has both centralized and distributed operation is optimal as the attacks in Sections 2.6.2, 2.6.3 and 2.6.4 could be detected in a distributed structure, while certain attacks in Section 2.6.5 will need to be detected in a centralized structure and others in a distributed manner. For example, a blackhole may fail to generate application-level acknowledgements that can imply network failure, even though the attacker is sending protocol level acknowledgements. Another fascinating point was highlighted in [57], which stated that, in future attacks, more than one style would likely be used simultaneously and multiple layers will be attacked in a cross-layer approach. For example, using a sinkhole attack to guide packets to a specific region so a jammer could jam a larger area. This concept further motivates providing interference diagnostic capabilities on edge devices to detect security breaches and help monitor remote systems, where continued surveillance is challenging.

2.6.6 System Coexistence

This section identifies intrusions from the spectrum co-existence and spectrum sharing fields. The discussion focuses on intrusions from the co-existence of systems in the same frequency range and when protocols misuse sharing capabilities. These attacks occur when legitimate protocols and devices adhere to laws but do not follow protocol rules (the non-compliant operator).

- A secondary user (SU) occupying a primary user's (PU) spectrum and causing interference. The SU operates for too long or when the PU is operating and interferes with the PU's performance. The intention was to maximize spectrum use, but the SU became an intruder by not complying with the spectrum sharing approach's protocols.
- An attacker, or a certain spectrum user, consumes all resources and deliberately denies spectrum sharing, causing other equal users to suffer performance loss or denial of service.
- Specific users being saturated by coexisting legitimate signals, leading to a DoS attack.

In these examples, network performance is affected and, so, the operation can be classified as intrusions. A SU occupying a PU's channel for too long and affecting the PU's performance becomes an attacker. Resources can be denied by, for example, blocking CTS packets, and so any device operating as such inherently becomes an intruder. In spectrum sharing, a cognitive radio (CR) senses for the absence of a PU (spectrum holes) [58] and a user could block the discovery of these spectrum holes, becoming an attacker in the process. This co-existence issue will be discussed further in Section 2.7.

2.6.7 Typical Defenses

The primary conclusion from analyzing the different potential wireless attacks is that no wireless network is 100% secure. It is challenging to design a wireless network where attackers cannot find some way in [43]. Timely mitigation strategies are required to combat attacks, especially attacks that exploit the WSN vulnerabilities. Attack incentives are created by the diverse application areas, which simultaneously increase the dependency on transmitted information in these applications. As a result, the potential risk of privacy and safety being compromised due to an attack rises. In WSNs, these intrusions are the largest contributor to link and path problems. The resulting packet losses can lead to avalanche effects and potential network collapse [2]. This outcome provides a need for security measures which are either preventive, reactive or detective solutions [36]. An intrusion/interference detection system (IDS) identifies the presence of intruders, so mitigation (or reactive) strategies can be implemented. The fundamentals of intrusions and intrusion detection were defined by James Anderson in 1980 and are; risk, threat, attack, vulnerability and penetration [59]. Additionally, an IDS includes the delicate balance between detection and false-alarm rates, which can be particularly challenging in environments where many different PHY layers (wireless protocols) occupy the same spectrum. Intrusion (or interference) detection can be achieved using different methods [19, 60]:

- Misuse Detection compares the action or behavior of transmitting/receiving nodes to well-known attack patterns. These attack signatures form the knowledge base of the IDS.
- Anomaly Detection defines the characteristics of normal operation and activities. Device transmissions are compared against this normal operation. The IDS classifies outliers, which are activities different from normal, as intruders.
- Hybrid or Specification-based detection includes IDSs which do not conform solely to anomaly or misuse. The expected behavior is manually defined by human perception. The focus is to determine deviations from this expected behavior when training data or machine learning algorithms do not define it. Specific hybrid approaches can combine both anomaly and misuse detection.

However, detection is not the only strategy applied to increase security in wireless networks, as preventive measures exist to provide a network with robustness to attacks. These techniques are employed in wireless protocols to protect critical information and add resilience to attacks. However, improvements are likely required because of the evolving nature of both attack styles and WSN deployments. This thesis focuses on improving detection techniques at the edge, as the resulting diagnostic can improve the mitigation and preventive approaches. Below, the leading security techniques used in IEEE 802.15.4 are discussed.

- DSSS: This spreading technique provides resilience to interference. Every four bit segment is spread to one of sixteen predefined 32 chip PN codes, which increases system redundancy because the codes are chosen to make the resulting signal noise-like. Therefore, there should be an approximately equal number of ones and zeros in the spreading code and few to no repeated patterns. This method provides some immunity from various kinds of noise, multi-path distortions and jamming and grants some security as only recipients who know the spreading code can recover the encoded information. Essentially, certain chip errors can occur while maintaining correct reception at the receiver by utilizing maximum correlation through, for example, a maximum likelihood decoder.
- Frequency Hopping was added to the IEEE 802.15.4e amendment to increase robustness against external interference and persistent multi-path fading. Multiple

available channels are used and only network nodes know the pattern. If interference is detected, this preventive measure could become a reactive mitigation measure by altering the frequency schedule. This approach is feasible as nodes at the edge of a jammed region can receive messages from “jammed” nodes and relay alarms to the controller and/or base station [19] or update frequency hopping schedules.

- A frame check sequence is an error-detecting code used to detect changes in the received raw data. The blocks of data being transmitted have a *checkvalue* attached based on the remainder of a polynomial division on their contents. In IEEE 802.15.4, the contents refer to the PSDU, exposing the preamble. On reception of the raw bits, the calculation is repeated and if the check values do not match, the packet is corrupt.

- Cryptography: To stop intruders from accessing sensitive information by simply listening to transmitted messages, data is encrypted before transmission. This method provides data confidentiality as the message is modified using a string of bits known as the security key. Theoretically, only the intended user can recover the original message without prohibitive effort. IEEE 802.15.4 only encrypts the MAC payload [36] and supports the advanced encryption standard (AES). Security depends on the pre-distribution, initialization, use and storage of the keys.

- Message Integrity Code (MIC): This approach protects against intruders modifying and resending messages, even if the packets are encrypted. By including a MIC with each transmitted message, data authentication is achieved because a confirmation of who transmitted the message is achievable.

2.7 Security Discussion

The discussions on security in wireless networks, specifically WSNs, highlight that security plays a significant role in WSNs. Security is integral for any successful WSN based critical application and, typically, four pillars of WSN security exist; **vulnerabilities, requirements, attacks and defenses** [61]. Generally, networks will have defined operating security requirements, e.g., confidentiality, and employ specific defense strategies (e.g.encryption) to ensure each requirement is met. Networks, especially WSNs, have vulnerabilities and attacks can use these vulnerabilities to increase attack efficiency. A notable example is the finite energy supply on wireless edge nodes. Thus, attackers can focus on this vulnerable point by causing devices to increase energy consumption, leading to devices losing power earlier than designed. Therefore, this

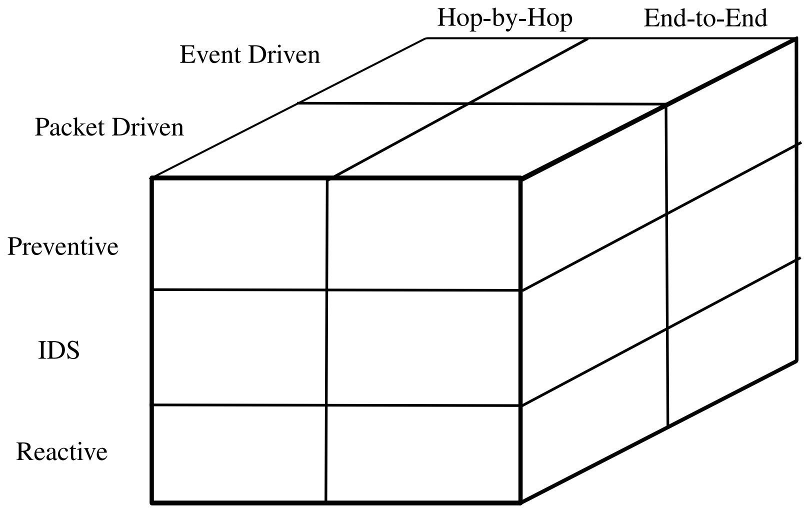
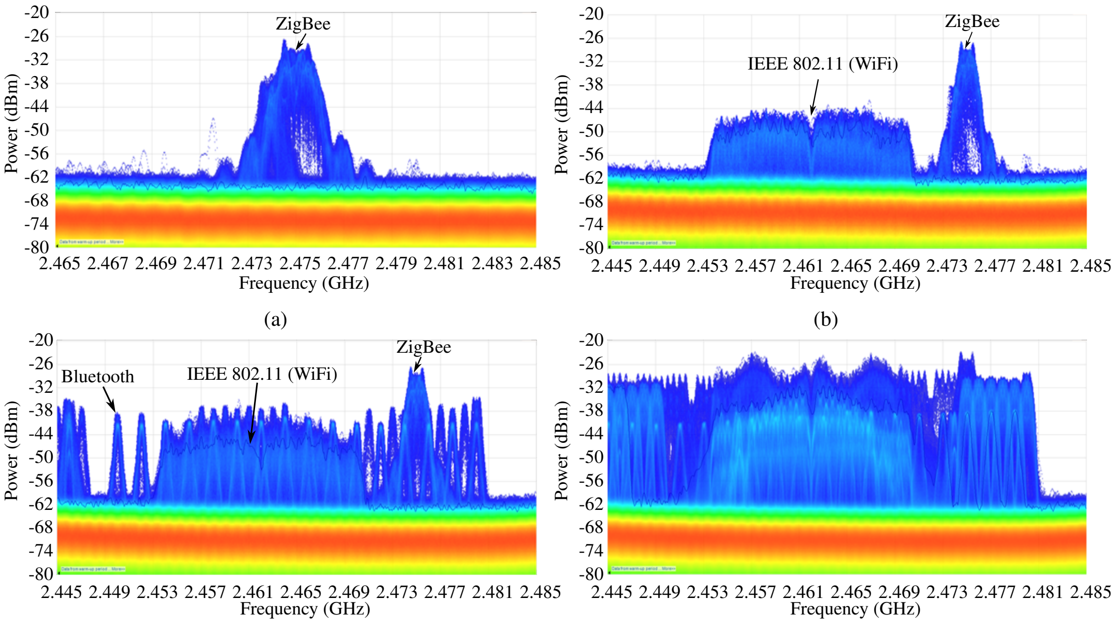


Figure 2.16: A functional simplified model for visualizing different security options in WSNs.

chapter's security analysis implies that the identified four pillars suit WSNs and other wireless networks consisting of edge nodes. The authors produced a 3D model for reliability in [62], which analyzes network reliability in terms of adopting a packet- or event-based scheme, applying retransmissions or redundancy techniques and whether reliability should be applied either on a hop-by-hop or end-to-end basis. Through the security analysis in this chapter, a similar approach can be determined for security. This approach, developed as part of this thesis, is provided in Fig. 2.16 and establishes a functional model for security using specific parameters. This model provides a simplified visual representation of some available security setups for wireless networks. The specified model analyzes whether a preventive, reactive or detection approach is used as the security mechanism, has a Hop-by-Hop or End-to-End basis applied and is security event-triggered or on each packet. Here, hop-by-hop refers to maintaining security across each and every link and end-to-end refers to only the source and destination maintaining security. Furthermore, typically, reliability provides bit loss recovery while security specifies bit loss prevention. Therefore, the topics can be linked in terms of packet loss and the model in [62] readily adapts to security.

Due to the identified unique vulnerabilities and security holes, the attack approaches identified in Section 2.6 can be applied to the IEEE 802.15.4 (ZigBee) protocol by using COTS hardware and open-source software. Currently, COTS devices are becoming cheaper and more computationally powerful, allowing various attacks to be implemented on a single device. Software-defined radios, combined with open source software, such as Python3 or GNU Radio, can apply various interference and privacy attacks by applying signal processing libraries or software blocks (GNU Radio) to mimic legitimate signals and packet structures. As a result, these devices exploit known WSN protocol designs and bitstreams by implementing intelligent deception jamming scenarios. This approach is described in detail in Chapter 5 and applied to over-the-air



(a) (b)

(c) (d)

Figure 2.17: DPX visualization of the ISM RF Spectrum during the benchtop experiments showing (a) DPX image of the transmitted ZigBee signal at 2475 MHz. (b) DPX image showing coexistence of ZigBee with 802.11 WiFi. (c) DPX image showing coexistence of ZigBee with 802.11 WiFi and Bluetooth. (d) DPX image showing the spectral environment when ZigBee packets were lost.

ZigBee signals in Chapter 7. These COTS devices' receiving capabilities support attacks on specific frame sections, where certain security measures are not implemented. For example, listen for the start of the SHR and focus the wireless attack on both the SHR and PHR, thereby potentially affecting packet synchronization or the received frame length, which, likely, causes inaccurate packet reception. SDRs or sniffers (TI's CC2531EMK [63]) can monitor a node joining process where, potentially, encryption keys are disclosed or can replay modified/unmodified packets. Notably, for WSN/IoT deployments, devices can be both physically and/or commercially available. This circumstance leads to the possibility of gaining network access through key extraction from memory, especially when no key management system is in use, which can be typical of domestic IoT deployments. Without the use of forward error correction, packets are more unreliable over noisy or hostile communication channels, which aides the effectiveness of jamming attacks and intrusions by coexistence. Jamming attacks need only cause a small number of chip errors to be effective, as the receiver cannot implement error correction. This value was identified as ten chip errors in [64], which corresponds to an incorrect autocorrelation on reception. Therefore, using both WSN knowledge and available COTS devices, specific aspects of protected networks can, potentially, be readily compromised. Exploiting wireless networks is possible, even when security techniques are used. Due to the ever-present threat of jamming attacks and its effectiveness not being limited to WSNs, this thesis focuses on improving security by enabling edge devices to implement interference diagnostic tools.

The IEEE 802.15.4 based protocols (ZigBee) coexist with various signals in the ISM RF band, including WiFi (IEEE 802.11b) and Bluetooth. This coexistence issue is experimentally tested using a ZigBee peer-to-peer network, multiple PCs and a Tektronix RTSA 306B using a Siretta ZigBee stubby antenna. WiFi signals (campus WiFi) and Bluetooth signals (local devices) were provided by enabling laptops, phones and speakers in the vicinity around one ZigBee transceiver. The associated hardware utilized in this coexistence examination is described in detail in Chapter 5. Spectrum graphs are developed using Tektronix's DPX [35] software, as previously shown in Fig. 2.4, which runs on the SignalVu-PC software package and acquires signals in real-time. DPX performs hardware digital signal processing and rasterizing of samples into pixel information, which can be plotted in real-time and as a bitmap image (instead of a conventional line trace). This software approach allows signals to be distinguished at the same frequency and a color scheme is used to identify signals which are more frequent than others. Here, to identify how spectrum coexistence can become an interferer, the spectrum was visualized for four separate scenarios. All transmitted and received packets were monitored using the XCTU software package [65] and transmissions require an acknowledgement packet, stating either "Delivery Status: Success" for a successful transmission or "Delivery Status: Address not found" for an unsuccessful transmission. The transmitted ZigBee signal on the 2475 MHz channel is shown in Fig. 2.17 (a) and real-time coexistence issues are visualized in Fig. 2.17 (b) and Fig. 2.17 (c), where the ZigBee signal coexisted with WiFi only and both WiFi and Bluetooth, respectively. No packets were dropped in these coexistence examples. However, packet loss is achievable using only commercial devices and legitimate protocols. Fig. 2.17 (d) provides the spectral analysis for when packets were dropped in the network, where unsuccessful transmissions were due to the interference caused by multiple devices using WiFi and Bluetooth in the vicinity of the intended ZigBee receiver. This situation differs from Fig. 2.17 (c) due to both the higher power interference signals at 2.475 GHz and the recurring number of transmissions, given by the fuller nature and more intense color of the DPX image. Compared to Fig. 2.17 (c), Fig. 2.17 (d) has approximately 6 dBm higher coexisting signals and, due to more connected devices, more frequent ISM band transmissions. Essentially, these tests provide visual proof of the spectrum coexistence issues, the noisy environments and legitimate signal intrusions that exist in WSNs. The undelivered packets, which occurred under extreme coexistence circumstances, provided evidence that environments and coexisting signals can be seen as unintentional and, in malicious cases, intentional interference. Therefore, detecting both intentional and unintentional interference is important for providing a holistic interference diagnostic approach on edge devices. This result is evident in this thesis's work and the final design outlined in Chapter 7. As a result, this security analysis of WSNs and the associated available attacks has identified wireless jamming as a real practical threat. Based on the WSN security options developed in Fig. 2.16, this thesis applies the hop-by-hop event-driven (bit errors) detection security approach.

2.8 Conclusion

This chapter introduced the application area for this thesis as wireless sensor networks, which have become an integral part of modern technology, as they are applied across diverse critical applications. WSN protocols are based on the IEEE802.15.4 protocol and the associated PHY and MAC layers were described in detail, as these are the most critical aspects in terms of the work in this thesis. ZigBee is the chosen protocol as it is the de-facto standard for WSNs, as almost all available commercial and research sensor nodes are equipped with ZigBee transceiver chips [21]. GPS signals and the associated signal model were introduced as an application area to analyze if this thesis's work can be transferred to a different application area of the RF spectrum and received packets. Interference and intrusions in wireless networks were discussed in terms of the four pillars of security; vulnerabilities, requirements, attacks and defenses. An extensive overview of both security issues and attacks defined the two main types of adversaries; the outlaw, who breaks the spectrum laws, and the non-compliant operator, who adheres to laws but does not follow protocol rules. A 3D security model was developed, which provides a simplified visual representation of some available security setups for wireless networks. This thesis focuses on the identified hop-by-hop event driven detection security approach. This security analysis concluded that, due to the ever-present threat of jamming attacks and its effectiveness not being limited to WSNs, this thesis focuses on improving security by enabling edge devices to implement interference diagnostic tools. The availability of COTS devices that can transmit jamming attacks and exploit edge device and wireless protocol vulnerabilities further supports focusing on interference attacks.

Furthermore, the diversity of innovative solutions requiring a WSN approach expands the array of applications and deployment areas but simultaneously incentivizes attackers. The more critical the application, the higher the probability of sensitive data being transmitted and attacks incurring a higher takeaway. The existence of unique WSN vulnerabilities has repercussions for providing adequate security levels and opportunities to any potential malicious actor. These vulnerabilities and essential security operating goals were analyzed by focusing on the fundamental use of the IEEE 802.15.4 standard in WSN protocols and its associated PHY and MAC layer security techniques. Experimental visualization of the ISM RF band coexistence issues demonstrated additional complexity when providing security. This chapter confirmed the existence of security holes in the standardized IEEE 802.15.4 protocol, notably in packet preambles and headers, open access to the wireless channel, WSN deployments and spectral coexistence. COTS devices and open source software can effectively capitalize on these security holes, providing evidence for establishing WSN security enhancements. As the interference can cause retransmissions and increased latency in wireless networks, sensed data or coordinator decisions could potentially be obsolete on arrival as data can lose its value in a matter of milliseconds, resulting in potential significant security consequences. This result motivates providing interference diagnostic capabilities on edge devices to detect security breaches and help monitor remote systems, where continued surveillance is challenging.

By utilizing ZigBee transceivers and commercial devices, real-time analysis of co-existing signals causing interference and denial of service was achieved. Packet loss occurred due to multiple commercial devices operating legitimately in close proximity to the ZigBee receiver, which indicates that this approach could be implemented maliciously or unintentionally. Therefore, this chapter's work implies that WSNs and other wireless systems can be vulnerable to interference and intrusions, but techniques can be used to add resilience and detectability. If WSNs are to become integrated into modern society and be used frequently and fully utilized in critical applications, like the IoT and aerospace, enhancements to security and the detection of intentional and unintentional intrusions are necessary. Detection strategies need to advance and look at aspects outside the norm, for example, received raw bits, while maintaining the optimization of device resources. The interference diagnostic tool should characterize the intrusion and be able to distinguish between intentional and unintentional interference, where independent edge device operation is optimal.

Chapter 3

Interference Detection and Classification: A Comprehensive Background

Continuously classifying fluctuating operating wireless environments in real-time can be crucial for successfully delivering authentic and confidential packets. The transmitted signals provide an insight into the operational wireless environment in terms of legitimate signals, background noise and potential interference. This chapter describes how this thesis' work is distinct from the literature, which is essential for the simulation work in Chapter 4 and the experimental over-the-air work presented in Chapters 6 and 7. This chapter aims to provide a sufficient literature review of the state of the art, enabling this thesis's work to be differentiated from the associated prior literature.

3.1 Introduction

Chapter 2 concluded that due to existing threats, the potential use in safety-critical applications and increasing congestion levels in the radio spectrum, new WSN security and signal identification challenges will likely emerge. This concept extends to the requirement for edge devices to implement services in a decentralized manner with a focus on autonomously reacting to channel variations and altering transmission metrics accordingly. If network services require cooperation between devices, then the secure communication links increases. Once edge devices can make independent decisions, a form of swarm intelligence [66] can be implemented among nodes to better understand the operating environment and reduce retransmissions, increase energy efficiency and improve channel access mechanisms. These different, yet connected, attributes indicate that new solutions will be required at the edge, resulting in the need for specific solutions designed for resource-constrained devices. This thesis's main objective is to understand the wireless environment so as to develop an interference diagnostic framework to enhance edge device operation and network transmissions. Real-time decision-making based on knowledge of the wireless channel produces optimal performance, where identifying the wireless operating environment conforms to the time series class of data science problems, as the primary data points are measured over a period. Securing wireless sensor networks (WSNs) and understanding the operating environment at the edge is challenging due to the burden of protecting the transmitted sensitive information across various applications while operating under unique security vulnerabilities and a fluctuating operating environment. Coupling this aspect with establishing a level of trust among network nodes while providing resilience to interference, it becomes clear that maintaining security is challenging.

To differentiate from other approaches, this thesis focuses on utilizing raw in-phase (I) and quadrature-phase (Q) samples, exclusively, to develop a novel low-order statistical feature set for wireless signal classification and interference diagnostics. As mentioned in Chapter 1, this diagnostic concept refers to the overall ability of the node to detect and classify interference under several different scenarios, such as, when packets are received and when no packets can be received. An overall algorithm is then required, as different data is required in each case, resulting in an interference diagnostic framework. The developed low-order feature set aims to enable wireless edge nodes to make decentralized decisions based exclusively on the analysis of the consistently available I/Q samples. This concept, as shown in Chapter 7, enables multiple decisions that utilize the same inputs of the developed low-order feature set. This approach is beneficial as it facilitates implementing appropriate security and transmitting mechanisms, reducing retransmissions and increasing energy efficiency. As discussed in Chapter 2, WSNs and their IoT utilization emphasize the significance of this time series interference signal classification problem. In this thesis, I/Q samples of typical WSN and industrial, scientific and medical (ISM) band transmissions are collected in a live domestic operating environment. As a result, this chapter reviews the main related topics and how this thesis's work differentiates itself from comparative literature. The three primary topics where this thesis makes a contribution are wireless signal classification (or modulation scheme identification) methods, 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. As identified in Chapter 2, for WSNs and associated resource-constrained devices, jamming is a significant ever-present (due to the wireless channel's natural open access) threat. This thesis's key design objectives are low complexity and independent operation, as resource consumption can be reduced if devices make real-time decisions and algorithms have low computation costs. It is worth noting that focusing on using received raw I/Q samples as the basis for interference detection and classification decisions is a relatively novel concept.

3.2 Related Work: Wireless Signal Classification

This section summarizes comparable previous work that focused on enhancing clear channel assessment (CCA) techniques and utilizing machine learning to identify received wireless signals, where spectrum sensing and adapting to the wireless environment are crucial applications. For wireless signal (technology) classification, the previous studies cover traditional approaches and more modern deep learning methods. Additionally, for WSNs, improving the CCA for IEEE802.15.4 protocols is of interest, as it is susceptible to denial of service attacks. Initially, the general signal model for modulation classification is specified, as it is the most common received signal classification approach.

3.2.1 General System Model

A general signal model can be applied to the wireless signal classification techniques in Section 3.2.3 to provide an understanding of the problem that needs to be solved. This signal model is a general representation and is stated to construct the problem of identifying the unknown modulation scheme. It differs from what is being developed in this thesis as the main aim of the signal classifier developed in Chapter 6 is to differentiate between known ISM radio frequency (RF) band signals. This thesis's methodology allows for the dominant signal in the WSN operating channels to be determined and interference detected if WSN packets cannot be received. As a result, this thesis's aim differs from the literature as the received signal model needs to be classified, even when the same modulation scheme is adopted. However, the automatic modulation classification problem is the most widely examined approach in the literature, resulting in the most relatable research to the signal classifier developed in Chapter 6.

A wireless communication system can be defined as a system that transmits information from one point to another point via a wireless medium. For a wireless communication system that sends binary messages from the transmitter to the receiver across a noisy channel, a matched filter can detect the transmitted pulses in the noisy received signal. The received baseband signal at the output of the matched filter, which is applied to increase the signal-to-noise ratio, can be expressed as:

$$r = \sum_{k} s_m(k).h(t-kT),\ m = 1,...,M,\ 1 \leq t \leq T \tag{3.1}$$

where the received signal r represents the received message as the sum of shifted unit pulses, s_m is the independent and identically distributed (i.i.d.) symbol stream carried by constellation $m \in M$ and T is the period of $h(t)$. The physical wireless link between a transmitting and receiving pair is classically modeled as a delayed tapped channel model expressed as the following function:

$$h(t) = A(t).e^{j(2\pi f_c+\phi)}.\delta(t-\tau) \tag{3.2}$$

where $A(t)$ is the amplitude, f_c represents the carrier frequency, ϕ is the phase and τ is the delay. The corresponding general form of a received modulated signal is then given by [67]:

$$r(t) = Re\Big\{\alpha e^{j2\pi\phi}e^{j2\pi\Delta ft}C(t)e^{j2\pi f_c(t-t_0)}\Big\} + n(t) \tag{3.3}$$

where $C(t)$ is the complex envelope of the modulated signal, $n(t)$ is the band limited noise, f_c is the carrier frequency, α is the channel amplitude, ϕ is the phase offset, Δf is the carrier frequency offset, and $Re\{.\}$ denotes the real part. The complex envelope is characterized by the constellation points C_i and pulse shaping function $p(t)$. For N symbols with periodicity T, the general form of the complex envelope can be expressed as [67, 68]:

$$C(t) = \sum_{i=1}^{N} C_i p(t-iT) \tag{3.4}$$

The modulation type and symbol energy of the received signal (3.3) are unknown and the signal is preprocessed blindly by assuming that it is drawn from a minimum-energy constellation, which is true for almost all the modulation types [69]. The noise model is assumed in most of the research to be additive-white-Gaussian-Noise (AWGN) [68]. As a result, the classification of wireless signals can be generally formulated as an n-class classification problem, where n is the number of different modulation schemes that the classifier can identify or how many potential modulation schemes are available. In this background review, only digital modulation schemes are considered, as these are the schemes analyzed in the methodology developed in Chapters 6 and 7. Furthermore, the wireless signal classifier developed in Chapter 6 focuses on an n-class classification problem, just like identifying the received modulation scheme.

3.2.2 Clear Channel Assessment

This thesis's work concentrates on classifying fluctuating operating wireless environments in real-time. Essentially, knowing what type of wireless environment (signal protocol) is dominating the wireless channel is a form of CCA. Generally, IEEE 802.15.4 based protocols (ZigBee) apply carrier sense multiple access with collision avoidance (CSMA/CA) to access the channel, which implements a CCA before transmitting a packet to check channel availability. Decisions are based on either energy detection or carrier sense, where if an IEEE 802.15.4 signal is sensed, then the channel may be busy, even if the energy is below the threshold (see Chapter 2). If the CCA identifies a busy channel, devices back off for a random period and reattempt, up to a user-defined maximum number of retries.

Previous improvements include splitting one eight-symbol CCA decision into two four-symbol CCA decisions [70], which allows the end of a packet transmission (or acknowledgement (ACK)) to be differentiated from a busy channel. The approach utilizes ACK packets' size and how the end of an ACK packet only takes up a small portion of the second back-off period. As a result, this technique identifies an idle channel when CCA is performed at the end of the ACK packet as the energy detection needs to be above the threshold for each four-symbol decision. This method increases the chance to transmit a packet when conventional techniques detect a busy channel. This proposed CCA method improved the throughput performance of IEEE 802.15.4 networks and was validated using simulations only. In [71] and [72], an interference aware adaptive CCA mechanism was introduced, which exploited packet loss information to change the CCA mode in use to improve ZigBee performance under WiFi interference. The CCA's energy detection mechanism can be affected by WiFi coexistence, which becomes interference when it prevents the CCA from identifying idle channels. This incidence of WiFi coexistence further validates the concept of spectrum coexistence being implemented as a malicious interference attack. The concept was introduced in Chapter 2, and the need to expand the CCA mechanism to manage WiFi coexistence issues proves its existence as a real-world threat.

The work in [71] demonstrated, using off the shelf components, that the CCA can impact ZigBee's performance when there is WiFi interference, particularly at high packets transmission rates. The proposed improvement uses a trade-off between the Transmit First In First Out Byte Register (TXFIFO) overflow of receiving radios and the packet collision loss to adapt the CCA mode or detection threshold to the sensed environment. This approach resulted in higher performance than conventional CCA modes and was tested in a hardware experimental testbed. These previous works that focus on the CCA show how WSNs and other networks implementing CCAs are susceptible to interference. This susceptibility motivates wireless edge nodes to be able to adapt to the operating environment independently. This thesis's developed methodology aims to enable edge devices to adapt to the classified channel, using features extracted from received I/Q samples and neglects network-level data. As a result, the developed methodology could be integrated into channel access techniques in the future. However, such integration is out of scope for this thesis and left for future expansions of this thesis's work (see Chapter 8).

3.2.3 Wireless Signal Classification

This section's central aspect surrounds wireless signal identification, which can also be termed as wireless technology classification or received modulation scheme identification. In terms of this thesis, the term wireless signal classification is applied as the critical classification required is the signal modal, such as WiFi, Bluetooth or ZigBee. In this thesis, wireless signal classification is required as a component of the overall designed interference diagnostic framework. As a result, the developed classifier leverages learnings from the literature regarding optimal approaches and techniques. However, it is not envisaged that the developed signal classification tool is a direct replacement for the most common use case of automatic modulation classification. As part of the classifier in Chapter 6, the developed features have the potential for broader signal classification as a future development (see Chapter 8). These signal classification methods and techniques are evolving with hardware and software enhancements. Typically, techniques are used to classify wireless signals based on their modulation schemes. This automatic modulation classification is a crucial technology that enables transceivers to utilize available resources efficiently [69]. This concept is essential in cognitive networks to recognize the received modulations and identify users and nodes in dynamic spectrum access applications. This modulation classification is a signal processing technique that estimates the modulation scheme of unknown noisy signals being received and involves two steps: pre-processing the received signal and classification algorithm design. In the literature, these signal classification techniques are broadly categorized into two main approaches, likelihood-based and feature-based, where feature-based is applied in this thesis for computational reasons that will be subsequently discussed.

Likelihood-based methods are based on hypothesis testing, by comparing the likelihood functions of received signals to classify different modulations. In statistics, the likelihood function measures the goodness of fit of a statistical model to a sample of received data for given values of the unknown parameters and is formed from the sample's joint probability distribution but viewed and used as a function of the parameters. These decision-theoretic approaches achieve optimal performance with the cost of high computational complexity. A maximum likelihood-based approach for coherent and non-coherent modulation estimations was proposed in [73], where optimal performance is achieved through the application of mathematical channel models with end-to-end settings. The approach is implemented by calculating the likelihood function of all modulations in the candidate pool and making the decision by choosing the class with the maximum likelihood value. These likelihood-based methods are, typi-

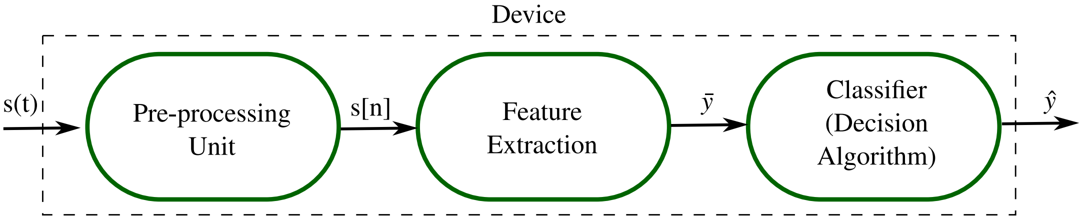


Figure 3.1: A simplified visualization of the architecture of feature-based wireless signal classification algorithm [69].

cally, unsuitable for real-world deployments due to high computational costs, typically, poor generalization in complex environments [69], and the requirement of a sophisticated parameter estimation process [73]. As a result, feature-based approaches are preferred in real-world practical deployments of signal classifiers. The feature-based classification architecture is provided in Fig. 3.1, where the feature extraction occurs on the data after preprocessing. The extracted features are used as the input to the designed classification algorithm. The traditional feature-based approaches primarily relied on expert knowledge to perform well under specific operating conditions. These classifiers suffer from high computational complexity requirements and poor generalization [69], which is an issue that this thesis focuses on. Machine learning developments in the design of classifiers have been employed to overcome these drawbacks. However, some machine learning classifiers still lack versatility and good generalization due to implementing single classifier predictions. Examples include support vector machines [74], k-nearest neighbor [75] and Naïve Bayes [76]. As a result, wireless signal classifiers based on combining the predictions of many weak component classifiers, for example, dependent decision trees (see Chapter 6), provide the highest chance of good generalization and accuracy [77]. These identified facts from the literature were used to select feature-based machine learning approaches as the desired classification approach. As the envisaged implementation platforms are resource-constrained embedded wireless edge nodes, the feature-based classifiers' choice was further validated. However, to validate the chosen optimal approach, this thesis developed classification algorithms based on single classifier predictions and classifiers that applied a combination of weak classifiers.

As specified in Fig. 3.1, feature-based wireless signal classification consists of receiving a time-series signal $s(t)$, data pre-processing to produce the sampled signal $s[n]$, feature extraction and classification (decision algorithm). Designed feature-based approaches, generally focused on digital modulation classification, produce robust performance and relatively low complexity [67]. This performance is achieved by efficiently extracting features from a statistical analysis of the signals. The work in [67] presents a compact overview of available features and classifiers, both pattern recognition and feature-based, used in automatic modulation classification. The available features can be categorized into five types: instantaneous time domain, transform domain, statistical, constellation shape, and zero-crossing features. The instantaneous features represent variations in the modulated signal and relate to the instantaneous amplitude, phase, and frequency. The transform domain features are typically extracted by transforming the received signal to Fourier and/or Wavelet domains [78], with distinct data processing techniques such as normalization and filtering. Higher-order moments (HOM), which are moments beyond 4th-order moments, higher-order cumulants, higher-order cyclic cumulants (HOCCs) and cyclo-stationarity produce the statistical features. Constellation features are extracted by analyzing the received constellation or by comparing it to a reference. As the name suggests, Zero-crossing features are related to counting the number of zero-crossings in the received signal. This thesis applies and extracts features from some of these categories to classify received signals, even when the same modulation scheme is applied. Additionally, new novel feature approaches are developed and the overall feature set is low-order, which contrasts with the typical use of higher-order moments and cumulants [67]. This thesis's work also extracts features from the spatial changes of received signals and the overall feature extraction is explained in detail in Chapter 6. Notably, the results in [67] state that SVMs attain a higher classification rate than artificial neural networks (ANNs) and that all pattern recognition techniques perform better than decision-tree methods. In Chapters 6 and 7, dependent decision-tree approaches will be shown to match the performance of ANNs and SVMs, where the decision-tree approach attains higher performance in specific cases. The decision-tree classifiers require fewer computational resources, which is a benefit compared to the neural network approaches. Furthermore, decision-tree approaches can produce a multi-class classification model and can be expanded by adding more decision points, as shown by transferring the optimal models developed in Chapter 6 to the work in Chapter 7.

The literature mentioned above provided an overview of available techniques and identified the most suitable and widely adopted approaches that are leveraged, in part, in this thesis. The following discussion expands on these identified techniques by specifying examples of use cases. In many existing studies, the previous features have been applied in different machine learning approaches, where the features are developed using various transforms. Cyclic features are employed in [79] in terms of higher-order cyclic cumulants, which typically require high-rate sampling. This requirement restricts the application of these features. However, the work in [79] applies compressive sensing by exploiting the sparsity of higher-order cyclic cumulants, the proposed algorithm is implemented with a (significantly) small number of nonuniform samples of the observed signal. Simulation results demonstrated the availability and robustness of this compressive approach. Wavelet transforms were implemented to extract coefficients to enable a robust Support Vector Machine-Decision Directed Acyclic Graph (SVM-

DDAG) classifier through numerical simulations in [78]. The main characteristic of wavelet approaches is that they can provide localized frequency information of a signal, which is very useful for classification. The authors in [69] employ spectral features and higher-order cumulants as input features to machine learning-based classifiers. In [80], a Matlab simulation approach is demonstrated, which uses higher-order cumulants derived from the received signals as the input features for various classifiers. The authors in [81] propose a method that calculates the correlation coefficients of second, fourth, and sixth-order cumulants. A deep learning-based approach is examined through simulations to show the usefulness of the extracted features in classifying different modulation schemes. The use of higher-order statistics (cumulants) is further validated in [82, 83, 84, 85]. The work in [82] applied the features to digital satellite communications through neural networks, while [83] applied high-order cumulants and deep neural networks to various shift-keying modulation schemes. The authors in [84] validate, through simulations, a novel approach for improving the correctness of the classification process with sixth–order cumulants and a simple two–step feature extraction structure. In [85], different machine learning algorithms (Naive Bayes, logistic regression, random forest, SVM, ANN, K nearest neighbors (k-NNs), Hoeffding tree and gradient boosted regression tree (GBRT)) were evaluated for automatic modulation recognition over the AWGN and the Rayleigh channel models. The results indicated that logistic regression and GBRT were the optimum approaches but incurred too much processing time, resulting in Naive Bayes and Random Forest as good alternatives. The work in [86] focused on combining features extracted from a spectral correlation analysis and SVMs. Other approaches that utilize these different features are employed across various machine learning approaches including decision trees [87], SVMs [78], k-NNs [75], Back Propagation (BP) neural networks [80] and Naive Bayes [76]. These were typically chosen for their performance in pattern recognition but their performance typically generalizes poorly to previously unseen data/scenarios [69]. In contrast to the conclusion in [69], the work in this thesis demonstrates that fundamental machine learning algorithms can indeed generalize to unseen data, given a sufficiently descriptive feature set and potent signal processing.

Deep learning methods have dramatically improved the state-of-the-art in speech recognition, visual object recognition, object detection and many other domains such as drug discovery and genomics [88]. The successful results are due to deep learning's ability to learn complex features automatically from large datasets. These successful deployments led to the development of applying deep learning-based algorithms for communication systems. O'Shea et al. [89] have outlined the compelling possibility of using deep learning techniques for radio signal identification, based on modulation schemes, and provide methods for real-world adoption. Other deep learning methods for signal classification based on modulation schemes include the work in [90], which examines fusion methods, in simulation, to further enhance convolutional neural networks (CNNs) to fully utilize the length of the received signal burst as the input size to the CNN is fixed, while the signal length can change. In [91], the authors explore CNNs as an automatic system to recognize the cognitive radio waveforms and produce simulation results that can identify eight distinct signals. The authors in [83] use deep neural networks and high order cumulants in an extensive simulation study to demonstrate the exceptional classification performance for digital modulation schemes. In [92], a software-defined radio (SDR) prototype for spectrum sharing is developed that utilizes a convolutional neural network for real-time wireless signal modulation classification. This concept is continued in [93], where deep learning methods are shown to outperform both a maximum-minimum eigenvalue ratio-based method and a frequency domain entropy-based method. In [77], the authors present a feature fusion scheme for automatic modulation classification using a CNN. The approach attempts to fuse different images and handcrafted features of signals to obtain more discriminating features. Simulation results produce high performance, for example, 92.5% accuracy with a signal-to-noise ratio of $-4dB$. A novel compressive convolutional neural network (CCNN) is proposed for automatic modulation classification in [94], where different constellation images are generated as network inputs from received signals. The associated simulation results demonstrated that CCNN displays superior classification and robustness than existing methods.

The majority of these approaches leverage deep learning's ability to automatically learn the features from the input data. This method is a data-driven approach and training such data-driven deep networks on large volumes of data typically requires appropriate computational resources and extensive time, both of which are rarely found in deployed communication systems [95]. The approach in [96] focuses on low-cost sensors and a reduced data-driven model, while, crucially for this investigation, identifying wireless classification over frequency, time and space dimensions as an active research problem. This analysis's main conclusion is that deep convolutional neural networks have strong representative abilities that can learn latent information repeatedly from signal samples. The signal classifier developed in Chapter 6 applies a deep neural network approach, in light of the existing literature suggesting neural networks being the optimal approach. Deep learning is investigated by applying manually extracted features due to the resource-constrained nature of the application area and limiting the required resources somewhat. However, more fundamental approaches achieve similar performance for a fraction of the required resources. Using the more fundamental machine learning approaches is more applicable to deployed communication systems consisting of resource-constrained devices. This thesis analyzes simulations to identify the foundational steps for live deployment and differs from specific literature by implementing simulation and live hardware experimentation.

The previous analyzed work, generally, concentrates on identifying the modulation scheme being implemented, which can infer the applied signal model, given specific circumstances. For example, identifying offset-quadrature phase shift keying (O-QPSK) at one of the ZigBee center frequencies infers the ZigBee protocol. In contrast, O-QPSK at the Bluetooth center frequency infers the Bluetooth signal model. Typically, statistical features are employed based on cyclic features, wavelet transforms and/or high order statistics and cumulants. This thesis's legitimate signal classification work is most comparable to the work in [97] as I/Q samples, received signal strength indicator (RSSI) samples and image-based spectrograms, based on fast Fourier transform (FFT) algorithms, are all used for signal classification. However, the authors in [97] focus on a different set of signals and require RSSI samples and image-based spectrograms. Notably, that study concluded that low complexity models need to be developed to reduce future intelligent devices' operational costs. In this thesis, a low complexity feature-driven approach that can generalize to unseen data for wireless signal classification aimed at edge device operation is produced. The analysis of the literature identified using feature-based machine learning approaches for classification. However, specific differences to the literature exist in terms of how the wireless signal classification is achieved. The work in this thesis (Chapter 6) is novel due to the exclusive use of raw I/Q samples and the developed low-order feature set, extracted entirely from received over-the-air I/Q samples in a typical domestic wireless operating environment. To the best of the author's knowledge, this study applies Hjorth parameters [20] in a novel application space and on new signals and FFT dynamics in a novel method, which is explained in detail in Chapter 6. Spatial features are used in terms of the probability density function of received samples. As a result, the developed feature set for wireless signal classification is novel and contributes to the literature. Furthermore, this thesis specifies that dependent decision-tree machine learning approaches are optimal and generalize well to unseen data. The need for, and importance of, this type of real over-the-air practical research is discussed in detail as a challenge that needs a solution in [98]. This thesis uses over-the-air experimentation to produce the low-complexity model for ISM RF band wireless signal classification. This development aims to perform a distinct task compared to the literature. It enables wireless resource-constrained edge devices to independently implement a specific interference diagnostic task, as explained in Chapter 7. As a result of this analysis of state-of-the-art signal classification techniques, feature-based machine learning was adopted for this thesis. Fundamental algorithms are examined, along with deep learning approaches, as time and computational resources are very limited in the application space of WSN and global positioning system (GPS) wireless applications and specific literature [85] indicated that these fundamental approaches could achieve high performance.

3.3 Interference Detection

WSNs and GPS applications are wireless communication systems, resulting in them being susceptible to the generic jamming attack strategies (e.g., constant jamming, reactive jamming, deceptive jamming, random jamming and frequency-sweeping jamming) presented in Chapter 2. The work in [1] surveys existing jamming attacks and anti-jamming strategies in several wireless protocols, including ZigBee networks and GPS systems. The survey paper aimed to provide a comprehensive knowledge landscape of existing jamming/anti-jamming strategies to stimulate more research efforts to secure wireless networks against jamming attacks. The work in this thesis aims to do exactly that by developing widely deployable machine learning interference diagnostic tools for resource-constrained wireless edge nodes. The authors in [1] outline that wireless technologies have significantly advanced over the past several decades. However, most wireless networks are still vulnerable to radio jamming attacks due to the open nature of wireless channels, and the progress in the design of jamming-resistant wireless networking systems remains limited. Chapter 2 expanded on this concept for WSNs by identifying unique vulnerabilities that WSNs incur. The added validation from such a recent survey paper supports the need for the work in this thesis. The stagnation in jamming resistant research can be attributed to the lack of practical physical-layer wireless technologies that can efficiently decode data packets in the presence of jamming attacks. Essentially, once a sufficient number of bit errors occur (forward error correction can detect and correct a limited number of errors in transmitted data without the need for retransmission), the packet is lost and needs to be retransmitted. ZigBee applies no forward error correction, so a single bit error will lead to a packet error. This thesis develops a practical over-the-air interference diagnostic framework to help solve this threat of malicious and unintentional jamming. These jamming attacks are becoming increasingly straightforward to deploy as hardware becomes increasingly available at more cost-effective prices. As discussed in Chapter 2, this enables potential attackers to prepare and develop attacks using off-the-shelf devices. For example, an off-the-shelf WiFi USB dongle device ($15) can disrupt WiFi services in a home or office scenario [99]. More advanced attacks can be implemented using SDRs, as demonstrated in Chapter 7 with an Analog Devices Pluto software-defined radio [100]. Notably, the authors in [1] state the following concise points about the current state of the art:

1. Jamming attacks are, typically, uncomplicated to develop and deploy, which produces an urgency to secure wireless networks against intentional and unintentional jamming threats.
2. Deployed jamming threats can only be thwarted at the physical (PHY) layer but not at the medium access control (MAC) or network layer. In typical jamming

attacks, legitimate wireless signals are overwhelmed by irregular or sophisticated radio jamming signals, resulting in legitimate wireless devices' inability to decode data packets. As a result, any MAC layer (or above) strategies are incapable of thwarting jamming threats, and innovative anti-jamming strategies need to be implemented on the PHY layer.

3. Effective anti-jamming strategies for real-world wireless networks remain limited, despite the significant advancements of wireless technologies in the 21st century. Most current wireless protocols are highly susceptible to jamming attacks due to the lack of protection mechanisms, including detection strategies.

This thesis tackles these points by developing an interference diagnostic framework for resource-constrained wireless edge devices. The framework aims to detect and classify interference exclusively using data always available to a functioning receiver across different circumstances. This approach results in this work directly focusing on the above three points. The analysis focuses on a practical anti-jamming approach using over-the-air signals, where the deployment is on the physical layer. This section's remainder looks at comparable work in the literature for detecting jamming attacks and implementing anti-jamming strategies. The main aspects of this thesis that differentiate it from the state-of-the-art are also specified.

Interference and intrusion detection is not a new area in wireless communication systems. However, it is an area that requires expansion and enhancements to match the current trend of WSNs and the previously discussed susceptibility to jamming. Vital aspects of WSN intrusion detection systems (IDSs) are discussed in [43], 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 a key conclusion. The authors in [101] describe the lack of traditional physical switches or gateways in WSNs as a vulnerability and emphasize the need for detection approaches. The optimal IDS should not degrade WSN performance or introduce new weaknesses but should be reliable and transparent to the system [101]. Flexible and reliable software-defined reactive jamming is shown to be feasible in [53], which provides attack deployment evidence for the previous descriptive studies. Denial of service attacks are outlined in [102], which also states that security is the linchpin of good sensor network design and detection can aid deployments. However, this type of research is not confined to WSNs as it is a current research topic across wireless networks, in general, including Global Positioning System signals [41], WiFi signals [103] and the coexistence of wireless systems [104]. Chapter 2 discussed specific threats to WSNs [50], how to secure WSNs [48] and existing security issues in WSN protocols [36]. In terms of this thesis, the work focuses on improving WSN security through anomaly detection in received WSN signals. However, the potential exists for malicious node identification, as explained in Chapter 7. The developed interference diagnostic framework provides high accuracy, generalizes to new data and can be implemented on resource-constrained and resource-abundant devices. As a result, this work adheres to the identified vital aspects of WSN IDSs. As a result, the work developed in this thesis, which is validated on ZigBee and GPS signals, has potential future uses in different wireless protocols, as discussed in Chapter 8.

Classic WSN techniques and IDSs typically analyze the RSSI and different forms of packet rates [44]. 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 [105] that evaluate packet delivery rates (PDRs) in a given area. This approach permits faster detection than end-to-end PDR and achieves jamming detection accuracy of over 97% [105], which sets the performance benchmark for this thesis. In contrast, chip sequence error patterns are used in [64] to identify the channel and, as a result, the emitting interference. This use of chips is a step above raw I/Q sample analysis. Four major chip error patterns were identified that allowed the distinction between interference 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, practically all the chips are incorrect and the type of interference can be one of a variety of transmissions, given enough power. SonIC was developed in [106], where the approach is based on sampling received RSSI values to extract features for a decision tree classifier for edge device applications. However, this process is limited as it requires the 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 [107] to develop an accurate and fast interference detection process using four SVMs and a logic decision stage. This approach suffers from the requirement of training additional classifiers if new interference signals need to be analyzed and the logic decision stage will, consequently, need updating.

In [108], the potential uses of machine learning in WSNs is discussed, where security and anomaly detection are identified as viable use cases. In [109], a framework for machine learning-based intrusion detection for WSNs is proposed. Network information, such as packet received signal strength, packet drop rate, and retransmission rate are used to detect intrusions. The proposed machine learning approach for the detection model is the SLIPPER algorithm [110]. The model consists of multiple binary classifiers, which incorporate a set of rules, as SLIPPER is a confidence-rated boosting algorithm. Each rule learned from its training dataset may not have very high prediction accuracy on new data, but the predictions based on the entire set of rules are expected to be accurate. This method relates to the wisdom of crowds concept where multiple weak learners combine to produce a strong learner. The use of received signal strength and descriptive packet rates are sufficient for specific applications. However, more information, such as malicious nodes, can be discovered by expanding into other frontiers. In [111], 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 [112], where the data is obtained from motes deployed in an actual living lab. These methods are outlier detection by active learning, identifying density-based local outliers and feature bagging for outlier detection. These machine learning techniques use features such as packet collision ratio, delivery waiting time and power consumption rate, to name but a few. An example of using decision trees as an intrusion detection method is provided in [113], 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 data set in [114]. The Random Forest technique was also previously shown to be capable of detecting jamming in GPS signals in [41]. By using a quarter-sphere centered at the origin, the drawback of the high computational requirements of traditional SVM could be alleviated [108]. For example, the authors in [115] introduced a one-class quarter-sphere SVM anomaly recognition technique, which can distinguish anomalies in data while minimizing communication overhead. Another SVM based IDS was proposed in [116], where an immune algorithm is used to pre-process the network data, while the SVM is adopted to classify the optimization data and recognize intruders. In [117], a real-time external interference source classification method for an 802.15.4-based wireless sensor network is proposed using deep learning and RSSI sampling in an office environment. The authors in [118] developed an anomaly and fault detection approach for medical WSNs. The framework utilized the Random Forests algorithm for classification and Additive Regression techniques for prediction.

Anti-jamming approaches exist and the authors in [11] proposed a Multiple-Input Multiple-Output (MIMO) based jamming resilient receiver to secure ZigBee communications against constant jamming attack. The approach focuses on the ZigBee preamble and utilizes online learning in a multi-antenna ZigBee receiver to mitigate the unknown jamming signal and recover the ZigBee signal. Specifically, the work in [11] proposed a scheme using the preamble field of a ZigBee frame, to train a neural network for jamming mitigation and signal recovery. This approach requires a MIMO receiver and, so, would require updating existing deployed nodes and implementing a more complex receiver design. In [14], a randomized differential direct sequence spread spectrum (RD-DSSS) scheme was proposed to recover ZigBee communication in the presence of a reactive jamming attack. RD-DSSS achieves anti-jamming broadcast communication without sharing spreading keys and the approach was validated in simulations. Dodge-Jam was developed by the authors in [12, 119] as a light-weight anti-jamming technique suitable for low-power and lossy wireless networks environments to address stealthy jamming attacks with small overhead. Stealthy attacks are defined as attacks that transmit short jamming signals to become less detectable with less energy, and yet powerful enough to ruin the entire packet transmission procedures. Dodge-Jam protects ACK packets by switching the ACK exchange channel to a channel calculated based on the content of a data packet. The procedure partitions a packet into multiple blocks and performs logical shifts of the blocks when retransmitting the packet. As a result, it helps the receiver recover the original packet from multiple received erroneous packets. DEEJAM was developed in [13] as a novel MAC-layer protocol for defeating stealthy jammers. Four defensive mechanisms are utilized to hide communication from a jammer, evade its search, and reduce its impact. DEEJAM offers four different countermeasures, namely frame masking, channel hopping, packet fragmentation and redundant encoding, to defend against different jamming attacks.

In [120], the authors propose the inclusion of a digital filter at the receiver side. This filter aims to effectively eliminate the spectral component caused by a periodically cycling jamming attack. Frequency Hopping Spread Spectrum (FHSS) and DSSS have been widely adopted to defend against jamming attacks. However, both approaches fail if the jammer jams all frequency channels or has high transmit power. To tackle this phenomenon, BitTrickle was developed in [121, 122] as an anti-jamming technique to defend low data rate wireless networks (WSNs) against high-power broadband reactive jamming attacks. The proposed system exploits the reaction time of reactive jammers, by transmitting packets in the unjammed time slots. A prototype of BitTrickle was developed using the universal software radio peripheral platform running the GNURadio open-source software. In [123], a method to detect reactive jammers in DSSS wireless communication systems was proposed. This detection approach extracts statistics from the jamming free symbols of the DSSS synchronizer to discern jammed packets from those lost due to bad channel conditions. Another detection method is proposed in [124] to cope with cross-technology jamming attacks. The approach is a novel ZigBee data extraction technique that can recover ZigBee data from the ZigBee packets that collided with WiFi packets. The technique consists of several steps, including multi-stage channel sensing, sweeping channel, and tracking the number of consecutive failed packets. Once the number of failed packets exceeds a certain threshold, the ZigBee device transmits its packets even if the channel is still busy, letting the signal recovery be made at the receiver side.

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 [125] and is only validated in simulations. In [126], another simulation-based detection approach is proposed that focuses on effective channel utilization (ECU) and interference power. The method detects power-based jamming attacks against wireless networks by implementing a detection approach using the ECU value, which is a widely used metric for channel utilization performance in wireless communications. This approach is validated in simulation and requires the thresholds to be selected using experience of the network. As a result, detection is limited to binary interference detection and can be susceptible to subtle reactive jamming. In [127], 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 [128] 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 results from live wireless signals and require information from high up in the network stack. In these cases, the approaches attempt to aid performance in the presence of jamming, however, being able to detect and classify the type of interference would still be beneficial to these networks, as the cause of errors needs to be identified.

This thesis differentiates itself from the wireless interference 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 approach contains both a simulation study and a hardware over-the-air wireless signal study. The Matlab simulations in Chapter 4 provided the initial motivation for solely using I/Q samples. The designed approach in this thesis is validated on multiple SDR receivers, resulting in no requirement for the receiver design to change or to be a MIMO receiver. All that is required is access to the received I/Q samples from the conventional receiver. 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. This is validated by developing a deep neural network and comparing performance to the more fundamental approaches. Furthermore, the developed detection solution is validated across different implementation platforms, SDR receivers, numerical ranges, signal models and frequencies. This cross-platform and cross-metric validation is another differentiating factor. The work in this thesis proves that deep learning is not the only method that achieves good generalization to new data. 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.

3.4 Exclusive use of I/Q Samples

This thesis's main contribution surrounds the exclusive use of raw received I/Q samples to produce a low-order feature set for wireless operating environment analysis. The aim of developing this feature set is to develop the following diagnostic tools for interference diagnosis on edge devices:

- Legitimate ISM RF Band Signal Classification.
- Legitimate XBee Commercial Node vs. Non-legitimate SDR Classification, where both signals have the same spectral image and transmit the same ZigBee PHY frame.
- Variations of ZigBee (WSN) Interference Detection and Classification (see Chapter 7).
- GPS Interference Detection and Classification.

These low-order I/Q features are extracted across the time- and frequency-domains and space, where the full feature set is explained in detail in Chapter 6. The hypothesis of solely using I/Q samples and the frequency domain's raw data is that it enables independent edge device operation, as no channel assumptions, network-level information, packet analysis or spectral images are required. The aim is that the developed I/Q samples based methodology can be applied across different implementation platforms, receivers, numerical ranges, signal models and frequencies. To achieve this approach, machine learning-based classification tools are applied. Non-deep learning methods that can achieve similar performance as deep learning approaches, but for a small fraction of the time and resource costs, are the desired design. This aspect is explored in Chapters 6 and 7, where the machine learning classifiers are developed using both non-deep learning and deep learning approaches.

Using the received I/Q samples in this thesis differs significantly from the literature, as focusing solely on using received raw I/Q samples for decisions is a relatively novel concept. Some examples do exist to make decisions in the 2.4 GHz RF band but differ from this thesis in the decision's purpose. In [129], the authors exploit I/Q component characteristics of a transmitter and deep learning techniques for uncooperative direction finding using a single uncalibrated directional receiver. The authors use an uncalibrated directional receiver to sense the 2.4 GHz channel and record the raw I/Q values from a directional as well as an omni-directional source. A fully connected deep neural network is implemented as an end-to-end regressor for predicting the bearing of the transmitter. Their system uses a neural network to learn from the I/Q data, resulting in the model automatically "learning" relevant features from the signal data as opposed to using custom-engineered features. This work motivated analyzing I/Q samples in this thesis, as in [129] the sole use of I/Q samples was unique from other radio directional finding techniques of using antenna arrays, known antenna models or received signal strength, for example. Typically, the non-machine learning methods for radio directional finding differ from the machine learning approaches in how the data is being modeled. Non-machine learning methods require distributional assumptions about the nature of the signal, while the machine learning techniques do not make such assumptions. Both feature-based [130] and non-feature [129] based approaches were successful. Based on this analysis, using received I/Q samples for decisions based on machine learning techniques, both with extracted features and "learned" features, is an effective approach.

This concept is expanded by the authors in [131], where the I/Q imbalance information is exploited to learn high dimensional features for transmitter identification. During the process of designing and manufacturing cheap radio hardware, specific imperfections have become the norm. The I/Q imbalance is one such imperfection and is caused at manufacturing due to the use of noisy mixers, oscillators and unbalanced low pass filters. The imbalance between the I and Q components of a signal results from the radio frequency interaction with the local oscillator frequency (required to create the intermediate frequency). As a result, the I and Q components of the modulator are not orthogonal. When a signal is transmitted using a particular radio transmitter, some I/Q imbalance is imposed over the complex-valued I/Q data, which can lead to performance degradation for higher-order modulations. The authors applied a generative adversarial network to detect rogue transmitters, while convolutional and fully connected deep neural networks classify different trusted transmitters. Other radio transmitter classification approaches use I/Q information, including the work in [132], where the authors used I/Q information and recurrent neural networks (RNNs) to predict primary user (SDR transmitters) activity in dynamic spectrum access networks. This method results in the secondary users opportunistically accessing the unused spectrum. This concept reinforces the wireless security aspects introduced in Chapter 2, where these secondary users can be blocked by malicious signals, leading to the need for detection scenarios on the wireless devices. In [132], the authors exploit both the spatial and temporal properties of the RF data and use them for a long-term prediction model for the primary user's presence or absence. Similarly, in [133], eight universal software radio peripherals (US-

RPs) are applied as SDR transmitters, where over-the-air raw I/Q time series data is collected using a DVB-T RTL-SDR receiver in a laboratory setting. Both the temporal variations and the inherent spatial dependencies in the collected I/Q time series data are exploited to learn unique feature representations for identifying the transmitters ("fingerprinting"). RNNs are leveraged with these features to identify the specific USRP transmitter. While in [134], deep learning and I/Q samples are used once again for radio device identification (fingerprinting) by learning unchanging hardware-based characteristics of individual transmitters. From this overview of using the I/Q imbalances and hardware characteristics imposed on the I/Q data, it is clear that this low-level data has promise for high-level decisions, specifically when machine learning is leveraged. In the later chapters, both the spatial and temporal aspects of the collected over-the-air I/Q time series data will be exploited for low-order feature extraction and interference detection and classification.

Most notably for the work in this thesis, raw I/Q samples are used as specific components in the design of a signal classification approach for Long Term Evolution (LTE), WiFi and Digital Video Broadcasting Terrestrial (DVB-T) signals in [97]. Although these signal models differ from the application space in this thesis, there is an overlap of WiFi signals. Furthermore, the aims differ as the work in [97] focuses on wireless technology classification, while this is only a part of this thesis, where the main goal is to establish an interference diagnostic approach. In contrast to this thesis's work, additional RSSI samples and image-based spectrograms, based on FFT algorithms and I/Q samples, are all required for signal classification in [97]. The study produces both deep learning approaches and decision tree-based classifiers, where the best model was a CNN. Other techniques included fully connected neural networks and Random Forest decision tree classifiers. Chapters 6 and 7 in this thesis will show that the low-order feature set developed in this thesis will achieve similar performance (98% or above) to the CNN developed in [97] through fully connected neural networks and dependent decision tree approaches. Additionally, the developed Random Forest approach significantly outperforms the Random Forest model in [97]. However, most importantly, the study in [97] proves the usefulness of using time-domain and frequency domain analysis of received I/Q samples and analyzes both manual and deep learning-based automatic feature extraction for deep learning and supervised machine learning approaches. Notably, the study concluded that low complexity models need to be developed to reduce future intelligent devices' operational costs, which is a crucial design requirement of this work. The work in [97] also reinforces the discussion in Chapter 2. The authors state that, due to increasing heterogeneity in wireless communications, often sharing the same spectrum band, sensing the environment and making intelligent decisions is crucial. In other words, continuously classifying fluctuating operating wireless environments in real-time can be crucial for successfully delivering authentic and confidential packets.

Additionally, the authors discuss that many of the previously developed or proposed methods target only resourceful devices. The success of the machine learning models developed in [97] motivates the proposed methodology in this thesis, as I/Q samples were successfully used for signal classification using machine learning. This application of I/Q samples bodes well for exclusively using I/Q samples for ISM RF band signal classification and WSN and GPS interference diagnostics.

It can be argued that most approaches use I/Q samples since that is how the data is received. However, the samples are generally used to gain access to higher levels of data for analysis, for example, symbols/chips and/or bits. These higher-level samples are then used in transforms (for example, Wavelet, Discrete Evolutionary etc.) to extract information for modulation scheme identification, as discussed in Section 3.2.3. A similar approach to using I/Q samples for interference detection is evident in the chip sequence error patterns utilized in [64]. This chip error work focused on channel identification and, as a result, the emitting interference/co-existing signal. Chip analysis is just above the level of I/Q samples, but the approach in [64] has certain implementation drawbacks as it requires devices to buffer known patterns for classification. Additionally, not all wireless protocols will use DSSS, so the scheme is limited to identification based on chip patterns. This thesis does not incur this limitation as the extracted features (see Chapter 6) are based on the received samples, which is how all wireless signals are received. This result means that the methodology developed in this thesis can expand to other signals where DSSS is not applied, if required. Chapter 6 examines this concept by including the IEEE802.11 signal model, which employs orthogonal frequency-division multiplexing (OFDM) signaling methods, in the wireless signal classification approach. Both of these pieces of work [64, 97], which compare with to this thesis's work, prove that using low-level data can be beneficial.

This section focused on using raw I/Q data for making decisions, where the use of raw received I/Q time-series samples typically focused on transceiver or directional identification. The work in [97] applied wireless technology classification using the received I/Q samples but focused on deep learning techniques and a different set of signal models than this thesis. The success of the work in [97], [64], the several radio transmitter identification approaches and the radio directional finding work proves the value of using raw I/Q samples and opens up new areas for investigation. The overall conclusion is that, to get the optimal performance (decision) from the raw received I/Q data, machine learning techniques should be adopted. The applied techniques can either leverage manually extracted features or "learned" features. As a result, this thesis's work contributes to signal classification, interference detection and crude radio classification for resource-constrained edge devices. The contributions are realized by focusing on low-order feature-based machine learning classifiers based exclusively on the time-domain, frequency-domain and spatial analysis of raw received I/Q samples.

An extensive Matlab investigation is undertaken in Chapter 4 to motivate the over-the-air experimentation and highlight that small deviations (low jamming-to-signal ratios) from the expected ZigBee signal can be identified. The hardware required to gain access to the necessary I/Q data and the associated data strategies are discussed in Chapter 5. Chapter 6 extracts the low-order features from received over-the-air data in a domestic environment and develops an ISM RF band signal classifier. Chapter 7 utilizes the extracted features to develop an interference detection framework for the ZigBee and GPS signal models and a commercial ZigBee node/SDR classifier. These developed classifiers result in multiple contributions to the field by exploring a novel investigation concentrated on exclusively using raw I/Q samples and low-cost open-source hardware and software for independent edge device interference diagnostics.

3.5 Conclusion

This thesis's work 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 wireless operating environment analysis. This analysis focuses on received signals, or samples from the wireless channel when no packets are received, to identify the presence of interference or the dominant signal being transmitted. The primary use case is WSN and GPS interference detection and classification, as jamming is an ever-present and destructive wireless network attack. Jamming is especially damaging to WSNs and GPS applications where the critical data in transit needs to be the most recent. Subtle approaches are more difficult to detect as packet analysis may require long analysis periods before detection, received power levels will be relatively unchanged and the RF spectrum can be as expected, at least visually. As a result, single device diagnostics, regarding the received signal and operating environment, that can detect subtle deviations from the expected reception contribute to the field. Furthermore, 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 ZigBee simulations in Chapter 4 will provide the initial motivation for solely using I/Q samples, while Chapters 6 and 7 explore real over-the-air signals in a typical domestic operating environment. The transferability of the designed feature set across different implementation platforms, receivers, numerical ranges, signal models and frequencies differs from the literature. This concept is explored in Chapter 7 by analyzing received GPS signals using a much lower-cost SDR than the ZigBee investigation. Finally, this chapter outlines that if fundamental supervised algorithms based on practical and descriptive data analytics/signal processing can be proved to be still fit for purpose, several benefits are accrued and further contributions to the field arise. These advantages would include achieving similar high performance for reduced execution time and computational resources than deep learning methods. This idea is investigated in Chapters 6 and 7, where both fundamental supervised approaches and deep-learning methods are investigated. This examination includes the achieved accuracy and how the model generalizes to unseen data. As a result, this thesis's main contribution is utilizing low-level I/Q data to formulate a low-order feature set to make typical higher-level decisions using supervised, fundamental machine learning approaches to detect and classify both malicious and unintentional interference on wireless edge nodes.

Chapter 4

Simulation Study of WSN Interference Detection and Classification: Initial Evaluation Exclusively using I/Q Samples

The information in this chapter is important for providing the initial motivation to pursue the over-the-air wireless transmissions. This work guided the hardware testbed development described in Chapter 5 and motivated and provided the foundation for the feature extraction in the experimental over-the-air work presented in Chapters 6 and 7. The work in this chapter has been published in part in the following:

- *G. D. O'Mahony, P. J. Harris and C. C. Murphy, "Identifying Distinct Features based on Received Samples for Interference Detection in Wireless Sensor Network Edge Devices", 2020 Wireless Telecommunications Symposium (WTS), Washington, DC, USA, 2020, pp. 1-7, doi: 10.1109/WTS48268.2020.9198724.*

- *G. D. O'Mahony, P. J. Harris and C. C. Murphy, "Detecting Interference in Wireless Sensor Network Received Samples: A Machine Learning Approach", 2020 IEEE 6th World Forum on Internet of Things (WF-IoT), New Orleans, LA, USA, 2020, pp. 1-6, doi: 10.1109/WF-IoT48130.2020.9221332.*

- *G. D. O'Mahony, K. G. McCarthy, P. J. Harris and C. C. Murphy, "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", Ad Hoc Networks, vol. 120, p. 102562, 2021, doi: 10.1016/j.adhoc.2021.102562.*

4.1 Introduction

An interference detection system can enhance the security of a communication link, as, typically, once an attack (or packet loss reason) is detected, it can be mitigated. This chapter explores the first examination of exclusively using received in-phase (I) and quadrature-phase (Q) samples to detect and classify interference in wireless sensor networks (WSNs). As described in Chapter 2, the ZigBee protocol is the chosen WSN signal model and this chapter provides an intensive Monte Carlo Matlab based simulation study. The necessary I/Q data is collected by applying Matlab Monte Carlo simulations, across a range of jamming-to-signal (JSR) ratios and interference types, including matched signal interference, continuous wave (CW), WiFi and thermal noise. The simulations initially focus on the full packet overlap case where the legitimate and interference packets fully interact. This approach is then extended by looking at different variations of legitimate and interference packet overlaps. These simulations evaluate the ideal case for using I/Q samples, as no hardware limitations exist. These limitations include, for example, the analog-to-digital converter (ADC) resolution and the reference voltage, which both limit the received samples in terms of resolution and maximum value before saturation. The developed features, as part of this investigation, will be leveraged as the foundation for hardware experimentation of wirelessly received over-the-air samples in later chapters.

Furthermore, this chapter uses the simulation study to identify the required data for developing machine learning-based interference detection models and investigates if a hardware wireless experimentation is warranted. The work differentiates itself from the literature, as discussed in Chapter 3, by developing a novel diagnostic framework concentrated on exclusively using raw I/Q samples for WSN interference detection and classification. This approach focuses on independent edge device decisions based entirely on the wireless channel's effects on received I/Q samples and makes no channel assumptions. As a result, no network-level data is required, enabling edge devices to make decisions based on data that is always available to a functioning receiver. This chapter discusses the simulation method and how it was evaluated by utilizing a software-defined radio (SDR) and a real-time spectrum analyzer. Initial bit-error location results motivate using I/Q samples and indicate under what circumstances an interference detection tool should operate. The initial features are established based on effective and descriptive data analytics and signal processing. These extracted features are applied to supervised discriminative machine learning classifiers, highlighting that heavily studied machine learning approaches are still fit for purpose in this context.

4.2 Experimental Method

This thesis establishes a simulation-based approach to interference detection as the "ideal" case without such hardware limitations as, for example, reference voltage and ADC resolution. Consequently, exclusively using I/Q samples for interference detection and classification can be investigated to determine if motivation exists for a live over-the-air examination. As explained in Chapter 2, 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 [21].

Here, designed and executed Matlab simulations describe bit error locations in ZigBee physical layer (PHY) frames (Fig. 4.1) under various jamming power levels to motivate and provide data for the development of an interference detection strategy. The Matlab simulations focused on node-to-node communications and applied ZigBee specifications [37], where possible. The ZigBee frame contained the required preamble of four zeros, a 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 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. This simulation approach was explored by translating the Matlab code into Python3, where equivalent Matlab functions from the "SciPy" library were implemented. The custom ZigBee samples were transmitted using an Analog Pluto SDR and compared to a commercial ZigBee transmitter, the DIGI XBee. The results are visualized in Fig. 4.2, where the simulated process correlates well with the commercial ZigBee transmission and, so, the proposed Matlab simulated setup is validated. The DPX (Digital Phosphor Technology) visualization is a software approach provided by Tektronix [35] and is used with a real-time spectrum analyzer to provide a pixel-based image of the radio frequency (RF) spectrum. This process is explained in detail in Chapter 5. It is applied here to show that the simulated approach for the ZigBee signal matches, spectrally, the signals transmitted from a commercial ZigBee device.

Different forms of interference were examined using varying power levels to under-

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

Figure 4.1: The simplified ZigBee PHY frame that is implemented in the Monte Carlo Matlab simulations to develop the WSN interference detection and classification tool. ZigBee-specific bytes are used where appropriate and a random payload, satisfying a zero-mean Gaussian distribution, is used in each iteration.

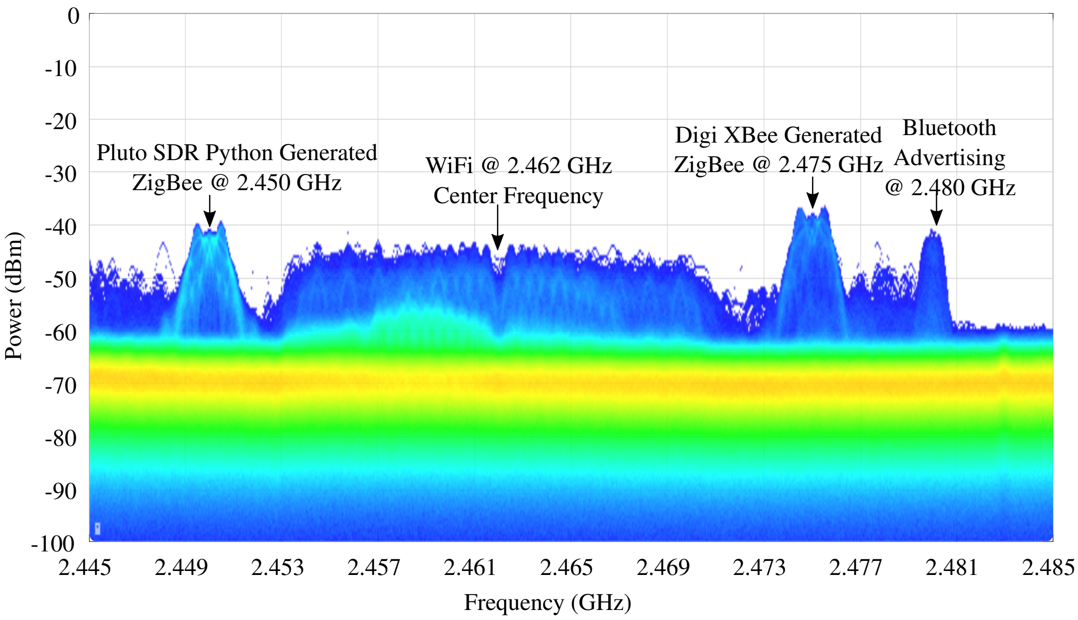


Figure 4.2: A Tektronix DPX visualization of the simulated ZigBee signal, transmitted from an SDR, compared to a commercially transmitted ZigBee signal in a typical domestic operating environment. Both ZigBee signals exhibit similar spectral images.

stand 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 based on a random number from the standard uniform distribution. The applied interference signals were CW jamming, matched signal interference (as explained in Chapter 2), thermal noise and WiFi (802.11b) coexistence. CW jamming forms the baseline interference model. It 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 due to the high correlation between spectral images, as discussed in Chapter 2. The thermal noise approach relates to noisy and hostile environments, which require 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 across a range of JSR values, which are statistically analyzed to identify differences, if any, between received error-free and erroneous transmissions (packets). As the probability of error (P_e) typically increases with JSR, the number of Monte Carlo simulations executed increases as the JSR decreases. As a result, simulations were executed using a logarithmic scale such as, for example, 2500 iterations at 30dB JSR, increasing to 60,000 iterations at -15dB JSR, where the JSR decreased in 1 dB decrements. This simulation method developed 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 1dB increments from -15dB to 30dB, 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.

A maximum likelihood decoder (MLD) is deployed as the simulated receiver using a sampling rate of $4MHz$. In the MLD, 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 (DSSS) PN codes $(PN_1, PN_2, ..., PN_{16})$. Here, the received samples are compared to an ideal set of samples for each PN code. In either case, the comparison produces a set of results, $(k_1, k_2, ..., k_{16})$, indicating the Hamming distances, H, of the received PN sequence and each sequence in the lookup table. Minimizing H maximizes the correlation and k denotes the index producing the minimum value in (4.1), where $H(P, PN_k)$ is the Hamming distance between the sequences, P and PN_k.

$$arg_k \; min \; H(P, PN_k), \; for \; k = (1, 2, ..., 16) \tag{4.1}$$

Each of the PN codes is designed to have a sharp autocorrelation peak, low cross-correlation values and to be 2-leveled with an equal number of 1's and 0's. This approach produces sequences resembling white noise, which increases resistance to both unintentional and intentional interference. 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/sample errors per PN code) is below a certain correlator error threshold (identified

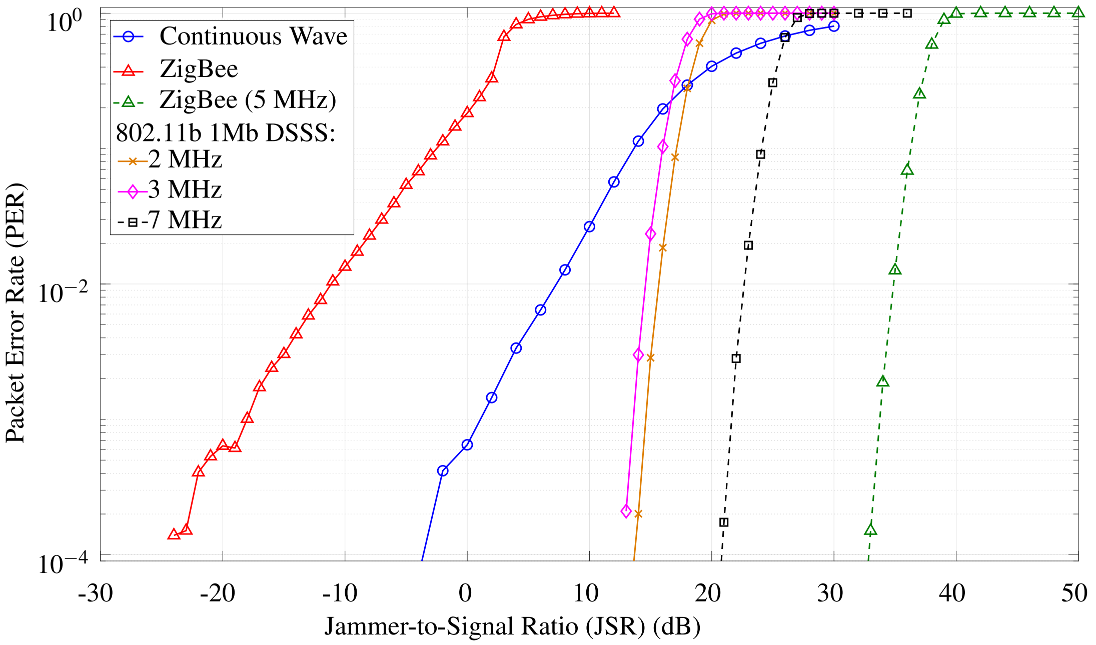


Figure 4.3: Matlab Monte Carlo simulation results for ZigBee packet error rates under CW, matched, offset matched and 802.11b coexistence interference for a range of JSR values for full packet overlap and a 40 byte packet length.

as ten chip errors in [64]), the correct symbol will be selected. Next, the question of ”What constitutes an error?” arises. For this study, a correlation failure, which is an incorrect symbol having the minimum Hamming distance, defines an error. A single-bit error causes a packet error since either the synchronization or the 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. Mainly, subtle jamming attacks are explored as, typically, these attacks are more difficult to detect using traditional packet error rate (PER) and received signal strength indicator (RSSI) methods.

By applying the described Matlab approaches, two distinct sets of experiments were performed, namely bit error location identification and feature analysis of error samples. The aim of the former procedure was to highlight where bit errors occur in the ZigBee PHY frame (Fig. 4.1), especially at lower JSR values. It was envisaged that this approach would provide sufficient support for the design of a detection scheme. Both error-free and erroneous received samples were then explored to detect if any statistical differences (features) exist, which could identify interference signals. Mainly, erroneous packets at JSR values of 15 dB and less were analyzed, as matched signal interference attains a PER of approximately 0.18 at 0dB and 1 at 15dB, as shown in Fig. 4.3, thereby being the most effective of the studied attacks. JSR values above this point would be readily detectable due to high power levels and packet loss rates. Hence, both subtle and brute force attacks can have destructive results.

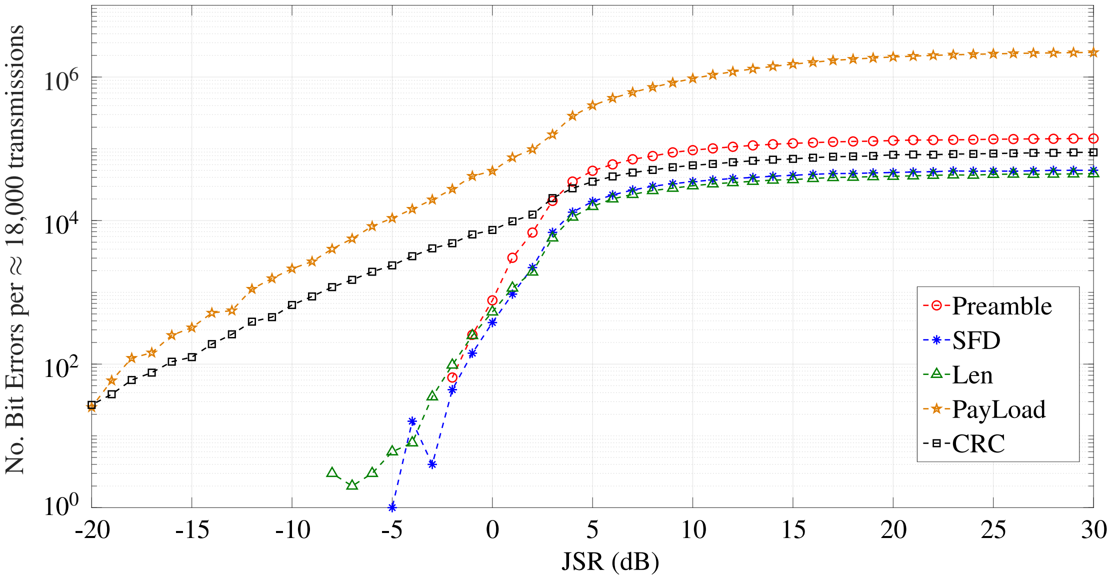


Figure 4.4: Number of ZigBee frame bit errors in five categories under matched signal interference for a range of JSR values, which provided the initial motivation for investigating the use of received I/Q samples in designing an interference detection strategy.

4.2.1 Results: Bit-Error Locations

The initial results focused on where the bit errors occur across the ZigBee frame for the simulated ZigBee packet transmissions under specific interference signals across a range of JSR values. The transmissions were investigated using $\approx 18,000$ simulations and three different jamming conditions. The results are expanded as per the packet segments outlined in Fig. 4.1 and provided in Figures 4.4, 4.5 and 4.6. These figures provide an insight into how bit errors vary in the ZigBee PHY frame as the jamming power increases. 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. As discussed in Chapter 2, an optimal interference detection framework needs to analyze packets with bit errors and when no packets can be received. Additionally, analyzing the channel before transmitting a packet has benefits as interference could be sensed before transmission. This bit-error analysis 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.

In these simulations, significantly more errors occur at low JSR values for matched signal interference compared to the other methods while, above $15dB$, high levels of packet corruption are evident in all but the 7 MHz WiFi interference, which requires a JSR of 22dB before errors begin to occur. The WiFi results suggest that at high JSR levels, the protocol can become malicious. In both the CW and matched signal cases, as the JSR value decreases, the probability of receiving an error-free preamble increases, which is evident at $\leq 10dB$ for CW and $\leq -5dB$ for matched interference. Overall, the results demonstrate that, at high levels of jamming, bit errors and, consequently, packet errors, occur across the frame, which is as expected. However, as the interference signal becomes more subtle, the probability of receiving an error-free preamble increases and bit errors are, likely, confined to the payload. Thus, nodes attempt to process erroneous packets, which eventually fail a frame check sequence and are rejected. This causes retransmissions and increased network and/or system latency, potentially having severe consequences for time-critical safety applications. The results illustrate that the cause of packet loss in the wireless channel becomes more challenging to identify, as power levels are as expected (JSR = 0dB). Retransmissions are also required at high levels of JSR, but it is typically easier to identify the presence of a jammer due to the high jamming power. Consequently, this bit error location work provided motivation to look at methods for identifying the presence of interference signals across the range of JSR values. For a distributed edge device investigative approach, it was decided to focus only on features based on the received I/Q samples and to neglect all network and packet rate information. This motivates examining I/Q samples, which are always available to a functioning receiver, as the source of data for interference detection features.

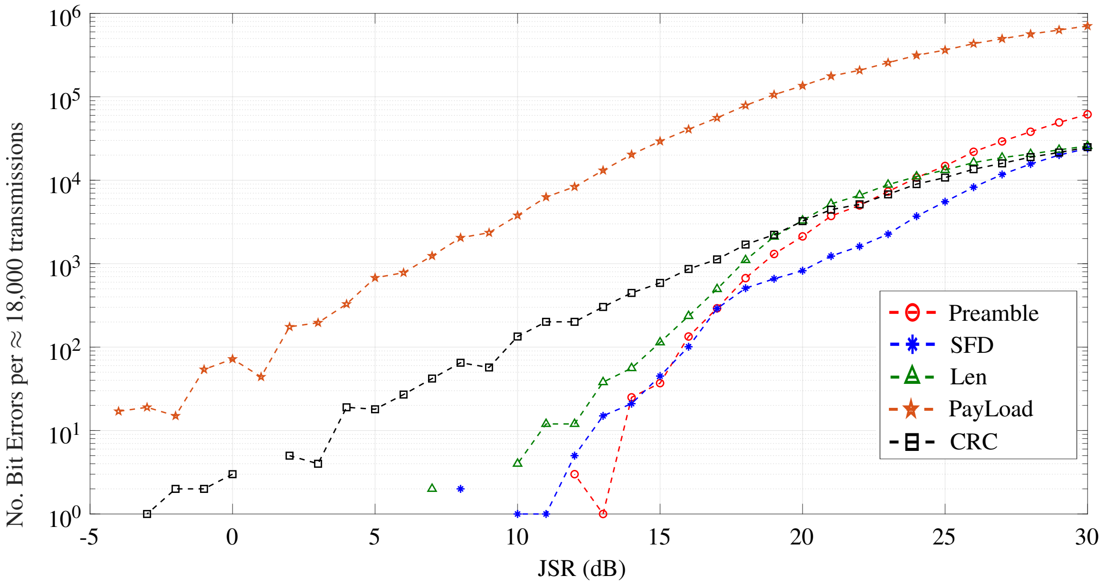


Figure 4.5: Number of ZigBee frame bit errors in five categories under CW interference for a range of JSR values, which upheld the initial insight of using received I/Q samples in the design of an interference detection strategy.

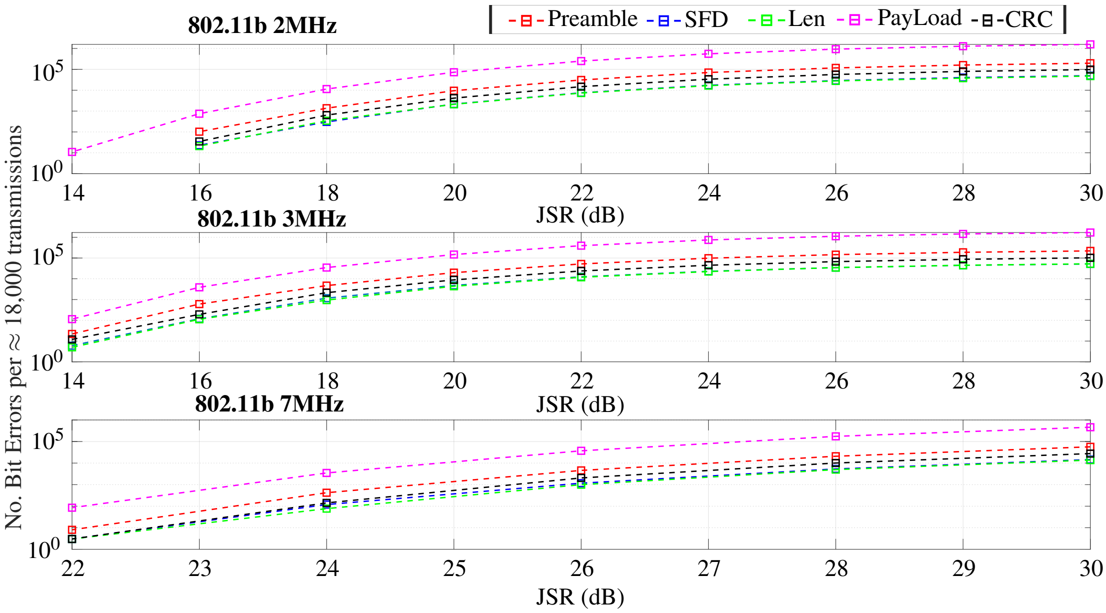


Figure 4.6: Number of ZigBee frame bit errors in five categories under 802.11b coexistence for each center frequency offset for a range of JSR values, which provided the initial motivation for investigating the use of received I/Q samples in designing an interference detection strategy.

4.3 Feature Extraction

The simulation method, depicted in Section 4.2, created a database of I/Q samples corresponding to ZigBee signals with and without errors, where the presence of an interference signal causes the errors. The interference signals included 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 SDRs, as shown in Chapter 5, or, possibly, in the device's debug mode, if otherwise unavailable. For the feature analysis, the results focused on matched signal interference, as it produced errors across the largest JSR range (Figures 4.4 $\rightarrow$ 4.6) and, as shown in Fig. 4.3, it can achieve a PER of ≈ 0.18, even at a JSR of 0dB. Attention was focused on determining features based exclusively on the analysis of received I/Q samples. As the signals are all mathematically created in Matlab, each received signal can be equated to the appropriate transmitted signal to determine the bit errors present, even if the received packet was erroneous. As the probability of error (P_e) increases with JSR, the number of Monte Carlo simulations executed increases as the JSR decreases. For the matched interference feature analysis method, simulations were executed on a logarithmic scale from 4,700 at 30dB JSR to 50,000 at -25dB JSR, and for transmissions without interference present, $10,000$ simulations were completed.

Initially, the statistical analysis focused on the measured probability density function (PDF) of the I/Q samples. For error-free packets, a low variance, relatively narrow bimodal sample distribution was expected, while for erroneous packets, a high variance and a pronounced bimodal distribution were anticipated. Fig. 4.7 indicates that the compact distribution becomes a wide bimodal shape as JSR values increase. Notably, the error-free PDF closely resembles what is seen in the spectrum, Fig. 4.2, as the zero bin is slightly smaller than its two nearest neighbors. This trend under increasing JSR levels allows features to be extracted from the distribution by analyzing the area within certain regions, determining the maximum peak and the number of non-zero populations. These results are provided in Fig. 4.8, where JSR values of 5dB and above can be clearly identified. The extracted PDF features include: 1) The area between bins -2 to +2, 2) The averaged area of the bins -128 to -3 and +3 to 127, 3) The number of non-zero bins and 4) The maximum peak. Matlab's *trapz* function calculates the areas and is shown in (4.2), 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, 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, resulting in no limitations in the maximum or minimum values.

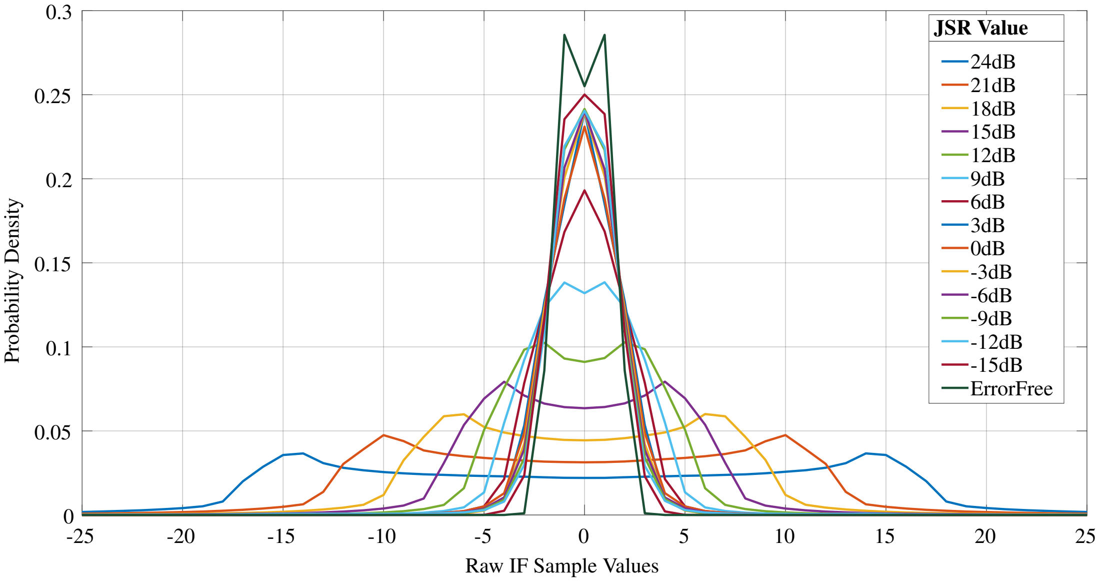


Figure 4.7: Measured PDF of simulated I/Q samples under matched interference for various JSR values. These results indicate that specific features can be extracted by analyzing the received PDF.

$$\int_{a}^{b} f(x)dx \approx \frac{b-a}{2N} \sum_{n=1}^{N} (f(x_n) + f(x_{n+1})) \tag{4.2}$$

The features are expanded by analyzing the received I/Q samples directly as a time series. As a result, the I/Q samples used to compute the PDF are individually analyzed to produce features as per Fig. 4.9. Derived features include: 1-2) The sample variance (and standard deviation to investigate which feature is more important), 3) The signal's entropy, 4) The mean sample value and 5) The absolute maximum value in the received

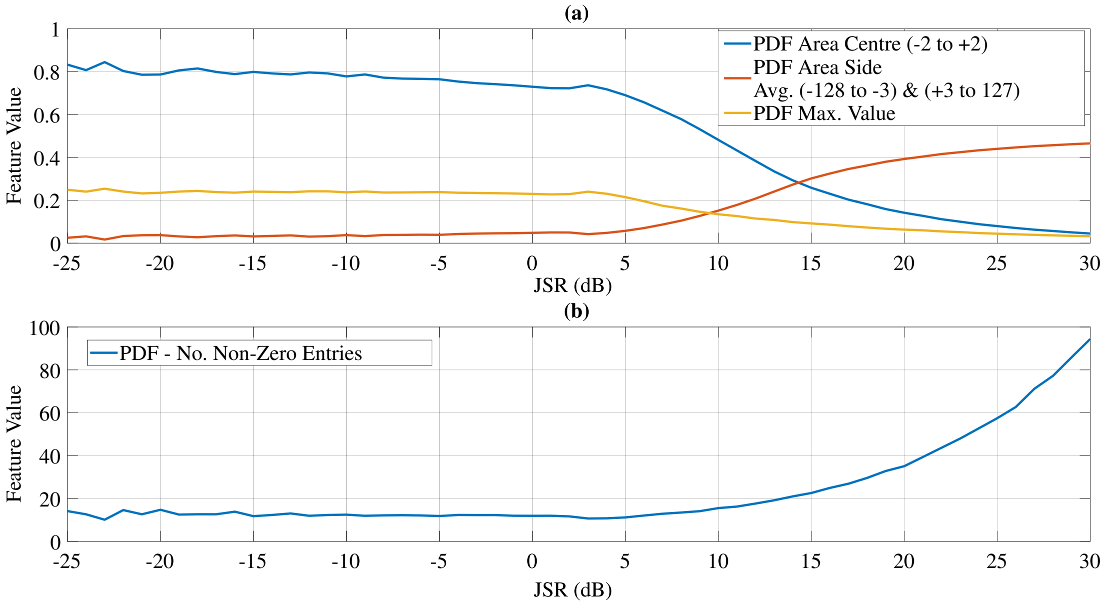


Figure 4.8: The extracted features from analyzing the received PDF, as a function of JSR. The extracted features include (a) area in the centre of the PDF, averaged area of the sides and the maximum PDF value. (b) The number of non-zero elements in the received PDF distribution using 256 bins.

sample set. Each of these features contains a useful trend that can identify the presence of an interference signal. As the jamming power increases, so does the variance, standard deviation, mean and absolute maximum of the I/Q samples. The entropy is calculated using (4.3), 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. This phenomenon is seen in Fig. 4.9 (d) as the JSR values increase. The extracted features should, theoretically, allow an edge node to determine why erroneous packets are being received by analyzing received samples.

$$H = -\sum_i P_i \log_2 P_i \tag{4.3}$$

Separately, Table 4.1 evaluates the same features for error-free packets using $10,000$ iterations, where the average, maximum and minimum values are provided to present value fluctuation. Table 4.1 demonstrates that the error-free transmissions are likely to contain different values than erroneous samples. For example, the minimum area of the center of the PDF for an error-free transmission is 0.9065, while the highest error value in Fig. 4.8 is 0.8445. Additionally, the maximum variance in error-free I/Q samples is 2.4407, while the minimum value in Fig. 4.9 is 3.677. Hence, the simulated results have extracted features which can be used to analyze received I/Q samples and, potentially, determine the presence of intentional and unintentional (WiFi) interference. The initial

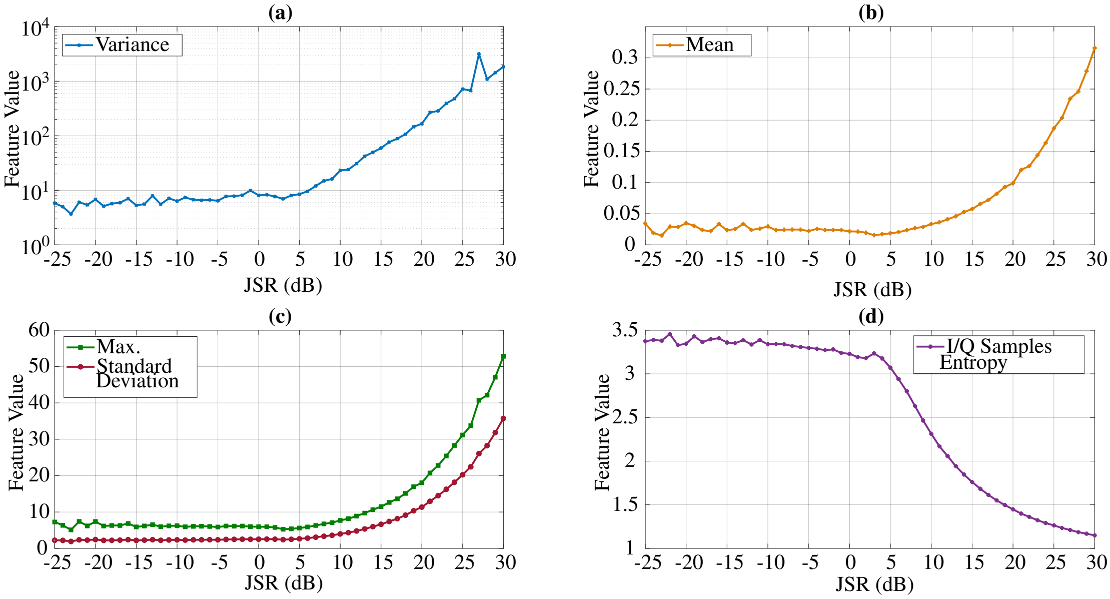


Figure 4.9: The extracted features from the time domain of the received I/Q samples. (a) The sample variance. (b) The mean value of the samples. (c) The maximum received sample and the sample standard deviation. (d) The entropy of received samples.

results implied a threshold of 5dB JSR but, by exploiting a machine learning approach, this threshold could be lowered based on the minimum and maximum values identified in Table 4.1 and the corresponding values in Figures 4.8 and 4.9.

Table 4.1: Error Free Features based on 10,000 packets

Feature:	**Error Free Packet**		
	Average Value	Max. Value:	Min. Value
Area Centre	0.9122	0.9189	0.9065
Area Side	5.0044 e-04	9.3700 e-04	1.56 e-04
PDF Maximum	0.2874	0.2964	0.2786
Non Zero Entries	7	8	7
I/Q Samples - Variance	2.3925	2.4407	2.3464
I/Q Samples - Standard Deviation	1.5468	1.5623	1.5318
I/Q Samples - Abs. Max.	3.2069	4.0274	2.888
I/Q Samples Mean	0.005	0.0196	7.4 e-05
I/Q Samples Entropy	3.6486	3.7412	3.5510

4.4 Machine Learning Models

Before investigating the usefulness of the extracted features from Section 4.3, various machine learning models and concepts need to be explained. Machine Learning is recognized as the discipline that allows computers to learn from data without being explicitly programmed. The term "learning" implies a progressive improvement in a specific task's performance and the ability to generalize on previously unseen data. These requirements are achieved by constructing a model based on the input training observations to generate a data-driven prediction of the output, rather than following predefined static program instructions. Here, the central concept of machine learning is introduced, why supervised machine learning applies to this study is discussed and applied supervised machine learning models for these simulations are described. As this thesis progresses in Chapters 6 and 7, different machine learning models will be required. These models are introduced and described when required.

4.4.1 Paradigms

In this thesis, the detection of interference in received I/Q samples is defined as a classification problem since the required output is categorical (discrete) when the received signal is classified. This problem identification is a categorization of machine learning as there is a dependence on the desired output that needs to be predicted. For a classification problem, the labels are divided into two (binary classification) or more (multi-class classification) classes and are usually solved in a supervised manner. Interference detection is an example of the classification problem, where the output classes for binary classification are "legitimate" and "interference", while the multi-class problem substitutes the interference type for the "interference" outcome. In contrast, when the system's output is a continuous value, rather than discrete, a regression approach is used. However, regression approaches are out of scope for this thesis. The classification approach suits the overall concept of interference detection as, here, the goal is to determine whether what is observed in the received I/Q samples (signal) is due to an intruder, or not.

Different types of machine learning approaches exist, where the specific type, and associated technique, depends on the problem being investigated. Supervised learning is the type of machine learning that learns a function that allows mapping an input data X to the output variable Y. In contrast, unsupervised learning aims at finding the patterns and structure in the data X without predefined variables or labels Y. Clustering is a prime example of unsupervised machine learning as the input data is divided into separate groups without the need for labels. This method is widely applied in different applications, such as, for example, market segmentation, social network analysis, and others. In Google news, for instance, clustering is used for grouping news into cohesive stories. Other types of machine learning include reinforcement learning, where learning is performed via interaction with an environment through trial and error to maximize long-term rewards. Active learning, which performs learning on a limited amount of training data, allows the learning algorithm to interactively query a user (or some other information source) to label new data points with the desired outputs.

Given that each machine learning type has various techniques that can be applied, the ML technique's choice is coupled with the given problem. This thesis focuses on detecting and classifying interference (classifying legitimate signals) in received I/Q samples where the focus is applied to WSN and industrial, scientific and medical (ISM) RF band signals. Based on the literature discussed in Chapter 3, for this classification problem, a supervised classification technique would be the most appropriate. In these supervised learning approaches, to build a model that predicts the response Y based on the explanatory variables (features) X, the dataset D is represented by $D = \{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$. To map every input $x \in X$ to a corresponding prediction $y \in Y$ an algorithm is employed to learn the mapping function (f) from the input variable (x) to the output variable (y); that is $\hat{y} = f(x)$. This generated function is the classifier and two main approaches exist to construct the classifier: generative and discriminative approaches.

The generative approach, given the output Y and features X, attempts to learn the joint probability distribution $P(X,Y)$. Specifically, the generative approach models how the data was generated and produces the most likely output $\hat{y}$ by making predictions using Bayes rule (4.4). The generative classifier models $P(Y)$ and $P(X|Y)$, which are called the class prior and the class conditional distributions, respectively, and examples of this type of classifier include Naïve Bayes and autoencoders, amongst others. However, as this work does not want to be limited to specific transmitters or environments, generative classifiers are not chosen due to the modeling of how the data was generated. No two transmitters will be exactly the same, due to I/Q imbalances and other electronic imperfections, and multiple different transceivers can be in operation in an active network which can be changeable.

$$\hat{y} = argmax_{y \in Y}\ P(y|x) = argmax_{y \in Y} \frac{P(x|y)P(y)}{P(x)} \tag{4.4}$$

In contrast, the discriminative algorithm makes no assumption on how the data is generated. This approach models the conditional probability of the label Y given the observations X, $P(Y|X = x)$. To discriminate classes Y, the discriminative approach directly learns the model $P(Y|X)$ depending only on the observed data. Examples of these classifiers include logistic regression, multilayer perception, support vector machine (SVM) and decision trees. These classifiers do not need to model the distribution of the observed data and, therefore, it may not be able to express the possibly complex relationship between observed variables and their labels. As a result, the discriminative models would not perform well on outliers, meaning the observed data needs to be large enough to provide a sufficient description of the problem.

When comparing the two approaches with test data that is generated by a different underlying distribution than the training data, it may be more straightforward to tune a generative model using the detected distribution changes. After fitting the generative classifier, it is also possible to generate the data which is similar to the observed one. However, if the relationship between X and Y only approximates the true generative process, a discriminative model may be preferred. In [22], the authors argue that the classification problem should be solved directly by modeling $P(Y|X)$, without any intermediate steps. In practice, discriminative classifiers have been shown to outperform generative ones, especially when the number of training examples is high [135]. In this thesis, the classification outcome is critical as performing mitigation once an attack (or packet loss reason) is detected compels the development of the interference diagnostic framework, as edge nodes can usually deliver packets to non-jammed neighbors [19]. As a result, supervised discriminative machine learning classifiers are investigated.

For this simulation study, two different supervised discriminative machine learning (ML) classifiers were examined as potential interference detection tools for WSN ZigBee signals. SVM [22] and Random Forest [23] were the chosen machine learning approaches. Both of these supervised ML algorithms are explained below and were used as classifiers to initially validate the usefulness of the features extracted in Section 4.3 and to provide an initial indication of how effective they are in detecting interference. Here, the classification groups would be error-free ZigBee signals (legitimate) and received signals with interference causing a range of bit errors (interference). The features outlined in Section 4.3 are used to try to define these distinct groups with as much mutual separation as possible.

4.4.2 Supervised Machine Learning: Support Vector Machine

A SVM [22] is a supervised binary discriminative classifier that aims to find the optimal hyperplane, as depicted in Fig. 4.10, that linearly separates the data points into two separate components by maximizing the margin. A hyperplane in an n-dimensional Euclidean space is a flat, n-1 dimensional subset of that space that divides the space into two disconnected parts. Typically, a SVM constructs a hyperplane or set of hyperplanes, in a high- or infinite-dimensional space, which can be used for classification, regression or other tasks, such as outlier detection, for example. A good separation (or margin) is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class. In general, the larger the margin, the lower the classifier's generalization error as the intuition is that a large margin on the training data will lead to good separation on the test data [136]. The classifier assigns new data points to one of the given two categories. Generally, the number of support vectors is much smaller than the total number of training instances. Hence, training a SVM can be resource-intensive, but the actual classification algorithm can be comparatively lightweight. In terms of this study, lightweight implementation is an important requirement as the desired imple-

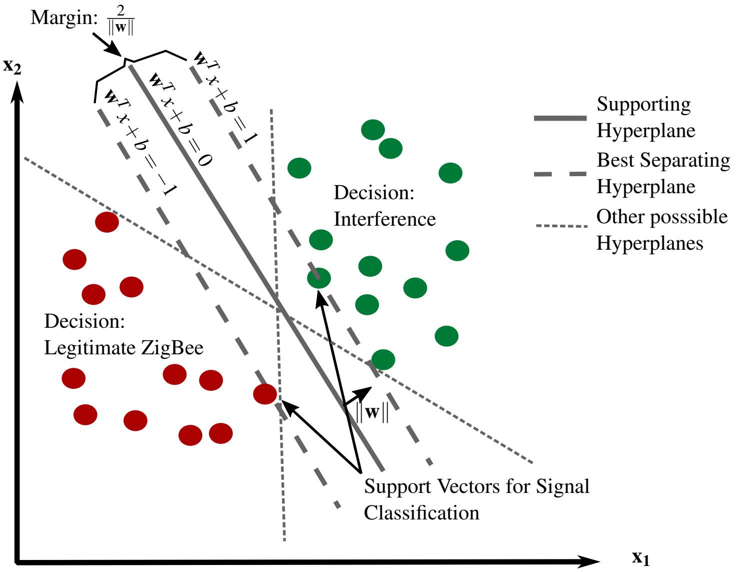

Margin: $\frac{2}{\|\mathbf{w}\|}$
$\mathbf{x}_2$
$w^T x + b = 1$
$w^T x + b = 0$
$w^T x + b = -1$
Decision: Interference
Decision: Legitimate ZigBee
$\|\mathbf{w}\|$
Support Vectors for Signal Classification
$\mathbf{x}_1$
Supporting Hyperplane
Best Separating Hyperplane
Other posssible Hyperplanes

Figure 4.10: A simplified visualization of the SVM in operation and the need for a large margin in the algorithm design. This binary classification implies a collaborative approach of multiple SVMs for a classification of input signals.

mentation platforms are resource-constrained WSN, Internet of Things (IoT) or Global Positioning System (GPS) edge devices.

In order to discriminate the data, which is not linearly separable, the SVM algorithm implements a function that maps data onto a higher dimensional feature space, where the data will become more linearly separable as compared to the input feature space. The mathematical function used to transform the data is called a similarity function or a kernel. Once the data is mapped, it can be separated using a hyperplane. Due to the way the SVM builds its decision boundary, it is also known as a maximum margin classifier. The SVM decision boundaries for two perfectly separable classes are illustrated in Fig. 4.10. For this linearly separable data, two supporting hyperplanes are selected so that the distance between them, called the margin, is as large as possible. The best separating hyperplane is the one that lies halfway between the supporting hyperplanes.

For a dataset of N points in the form of feature vectors $\{x_1, x_2, ..., x_N\}$ with corresponding labels $\{y_1, y_2, ..., y_N\}$, where $y_i \in \{-1, 1\}$ a hyperplane is defined as: $\mathbf{w}^T x + b = 0$, where $\mathbf{w}$ is the normal vector to the hyperplane. The parameter $\frac{b}{\|\mathbf{w}\|}$ determines the offset of the hyperplane from the origin along the normal vector $\mathbf{w}$ and $\|\mathbf{w}\|$ is the Euclidean norm of $\mathbf{w}$. Fig. 4.10 specifies that other possible hyperplanes exist that can perfectly separate the two classes, but none of them provide the maximum possible distance between the classes. The distance that has to be maximized is $\frac{2}{\|\mathbf{w}\|}$, which signifies minimizing $\|\mathbf{w}\|$. As a result, the two-class classification process with a SVM consists of assigning a positive/negative label to each input vector x using (4.5) [137], where k is a kernel function, $\alpha_i \geq 0$, $i = 1, ..., N$ are the Lagrange multipliers, N is the number of support vectors and the sign function determines whether the predicted value comes from the positive or negative class.

$$f_{svm}(x) = sign\left(\sum_{i=1}^{N} \alpha_i y_i k(x, x_i) + b\right) \tag{4.5}$$

Different kernel functions are available and, here, the linear, radial basis function (RBF) and polynomial kernels are compared using the validation data. The optimal kernel is then applied to the training data to produce the final SVM classification models. The kernel determined to be optimal most often in this thesis is the RBF kernel and it is defined by (4.6), where $||\mathbf{x_i} - \mathbf{x_j}||^2$ may be recognized as the squared Euclidean distance between the two feature vectors and σ is a free parameter. To classify the various received signals, either multiple SVM models are required, or one model classifies between legitimate and all other signals. The SVM method was chosen as it is relatively easy to apply, memory efficient and suits the binary classification problem of legitimate signal, or not.

$$k(\mathbf{x_i}, \mathbf{x_j}) = exp\left(-\frac{||\mathbf{x_i} - \mathbf{x_j}||^2}{2\sigma^2}\right) \tag{4.6}$$

4.4.3 Supervised Machine Learning: Decision Trees (Random Forest)

Random Forest [23] is a supervised decision tree discriminative machine learning approach. It is based on a large collection of individual decision trees, known as weak learners, consisting of binary intermediate nodes and operating as an ensemble, as visualized in Fig. 4.11. This ensemble concept forms the fundamental theory upon which the algorithm depends, as the “wisdom of crowds” concept implies that the mutual consensus of a group of individuals is usually more valuable than that of any single entity. Thus, this algorithm operates by combining a large collection of relatively uncorrelated models, sub-optimal decision trees, as a committee to produce a composite decision of higher quality that will outperform any of the individual constituent models. Using a tree-like model for the decision, the classifier allows the user to visually and explicitly represent the decision-making process.

Each weak learner (decision tree) is constructed through recursion and a typical tree structure can be defined as a root node followed by a set of internal nodes and final leaves. Each node is a logical divergent point where a particular explanatory variable (feature) splits the data according to a particular condition. All nodes are connected with branches showing the direction from a question to the answer. The leaf nodes are terminal nodes that have no child nodes and represent a value of a target variable. The trees use a deconstructed observed input to construct a series of binary intermediate nodes, that successively choose the attribute and associated threshold providing the best split into distinct groups. These groups are as different from each other as possible, but contain members which are as similar as possible. Decision making depends on a diverse group rather than a homogeneous approach, as each decision-tree is unique and specifies a vote. The output with the most votes is the overall prediction. This is visualized in Fig. 4.11, where two trees predict interference and four trees predict a clean signal, therefore the decision is that interference is not present.

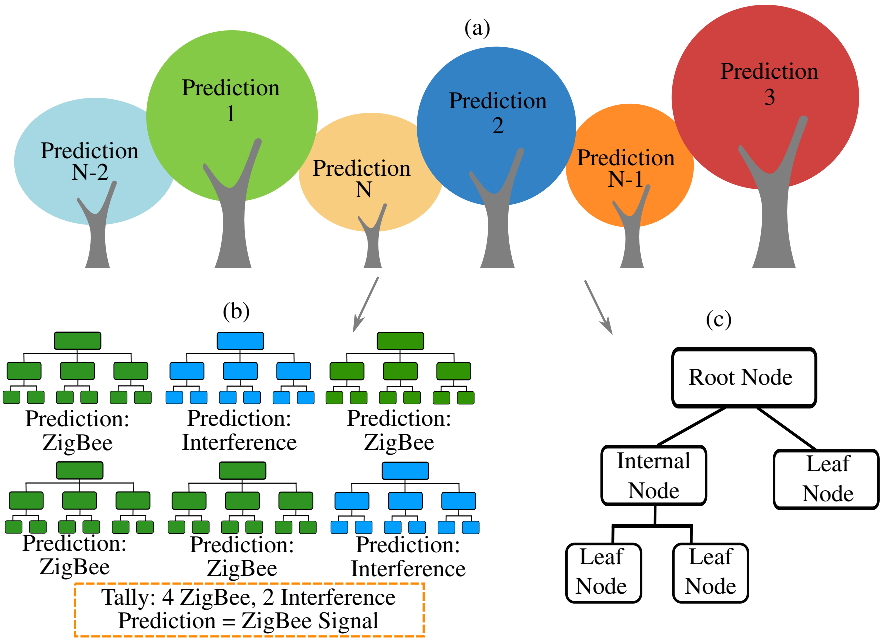


Figure 4.11: A simplified visualization of the Random Forest operating process, showing (a) The “wisdom of crowds” ensemble concept. (b) An example of the majority voting scheme. (c) Typical structure of an individual weak learner.

This idea depends on having low correlation between individual trees, as this protects each tree from their individual error [138]. Uncorrelated decision trees are ensured by two methods: bagging (bootstrap aggregating) and feature randomness. The former exploits each decision tree’s high sensitivity to the training data used and the latter ensures each tree can only pick from a random subset of available features. The bagging concept ensures unique trees by applying replacement, which allows each weak learner to be constructed from a random subset of the training samples and maintains the sample size by repeating previously used samples. Applying replacement allows examples to be repeated to maintain the sample size N, while, concurrently, allowing for a unique tree to be modeled. Thus, as each sample-set is randomly chosen from the total training sample set, the corresponding decision trees, known as weak-learners, contain different variations of the original classification data, which reduces variance and helps to avoid over-fitting. The random sample-set and feature set allow for the creation of uncorrelated trees that protect each other from their own errors and, once a set of decision trees has been computed, a new sample can be classified by performing a majority voting scheme, as visualized in Fig. 4.11.

Here, Random Forest, specifically decision trees, suits the identified problem for many reasons. This algorithm was used to develop an interference detection scheme in GPS signals [41], it is cited as being suitable for classification and intrusion detection [113], is fast, scalable, robust to noise, does not over-fit [139] and, importantly, can work with large datasets. As Monte Carlo experimentation, either through simulations and/or live data, is required for WSN transmission analysis, the chosen algorithm must be capable of working with large example datasets. Multiple iterations (likely in the thousands) are required to model the wireless channel sufficiently, as typical channels and environments change regularly. WSNs are commonly deployed in environments where the spectrum changes rapidly due to the number of connected devices, demand, packet size or services in operation and the physical channel typically changes due to varying fading levels, obstacles, path losses, and spurious interference. Furthermore, employing machine learning techniques on low power embedded systems by exploiting low-power micro-controllers is becoming more achievable in IoT applications [140, 141], meaning that optimizing machine learning algorithms for WSN nodes is possible. Hence, developing this type of algorithm for use in a WSN is an achievable task and is becoming more relevant as training begins to shift from the data centers to the edge nodes.

In contrast to other “black box” modeling techniques, the main advantage of tree-based classifiers lies in the possibility of finding the reasoning behind the model. This property makes trees a good candidate for problems that require an understanding of the decision-making process. While constructing decision trees, only features that are useful for a given problem are included, which enables the use of a tree-based classifier for feature selection. This will be explored in greater detail in Chapter 6, where the optimal feature set based on live over-the-air data is determined and dependent decision tree approaches are introduced.

4.4.4 Confusion Matrix and Receiver Operating Curve

This chapter introduces the concepts of a confusion matrix and a receiver operating characteristic (ROC) curve as methods for describing the designed classifiers’ performance. This thesis deals with both binary and multi-class classification problems. The binary approach needs to be described first as the predicted outcomes are labeled either positive or negative. Consequently, there are four possible outcomes from a binary classifier (Fig. 4.12): true positive (TP) – if the outcome is a positive prediction and the actual value is positive; false positive (FP) – if the actual value is negative but the predicted outcome is positive; true negative (TN) occurs when both the prediction and the outcome are negative; and false negative (FN) for the cases when the prediction is negative, while the actual value is positive. This concept can be extended for the multi-class case, where the same four outcomes are applied to each individual class.

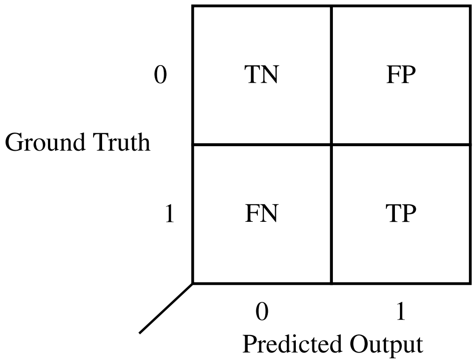


Figure 4.12: A visualization of a confusion matrix identifying the four possible outcomes of a binary classifier; TN, TP, FN, and FP.

The ROC curve is constructed with a set of sensitivity and specificity values obtained by adjusting the decision-making threshold. Sensitivity refers to the ability to correctly detect interference in the received I/Q samples, while specificity characterizes the ability to correctly identify (reject) a legitimate signal. These characteristics are computed as follows:

$$Sensitivity = \frac{TP}{TP+FN} \tag{4.7}$$

$$Specificity = \frac{TN}{TN+FP} \tag{4.8}$$

The ROC graph plots all sensitivity and specificity pairs resulting from continuously varying the decision threshold over the entire range of results. The area under the ROC curve (AUC) is then calculated and used as a single statistical measure. An AUC of 1 corresponds to perfect discrimination between the classes. Random discrimination is represented with an AUC of 0.5. If AUC < 0.5, then the predictions are negatively correlated with the ground truth. Accuracy is another widely used metric for evaluating classifier performance, which quantifies the percentage of correctly detected labels. When “unseen” data is available, the generalization error can be determined. This generalization error value outlines how the designed model would typically perform on instances it never encountered before. It is used as the primary verification of model suitability to the problem and, potentially, real-world operation. In this thesis, this set of evaluators is applied and the generalization error is used when “unseen” data is available. As each wireless transmission is unique, “unseen” data is achievable and this is discussed in detail in Chapters 6 and 7.

4.5 Results

This section focuses on validating the features extracted from the analysis of the I/Q samples and investigating whether this methodology should be examined using live over-the-air signals. For these simulated I/Q datasets, the SVM [22] and Random Forest [23] machine learning techniques were applied based entirely on the nine features, as detailed in Section 4.3. The Matlab “fitcsvm” and “TreeBagger” functions were employed to implement the SVM and Random Forest approaches, respectively. All necessary settings to implement the Breiman and Cutler Random Forest were applied. Fig. 4.13 provides the basic approach of the model using the Random Forest algorithm as an example. The Matlab simulated ZigBee signal, with(out) added interference, is received, deconstructed based on the defined feature set and classified by the designed procedure. The results initially focus on matched signal interference and, based on the success of those results, the methodology is expanded to include the remaining interference types of CW, WiFi and Thermal noise.

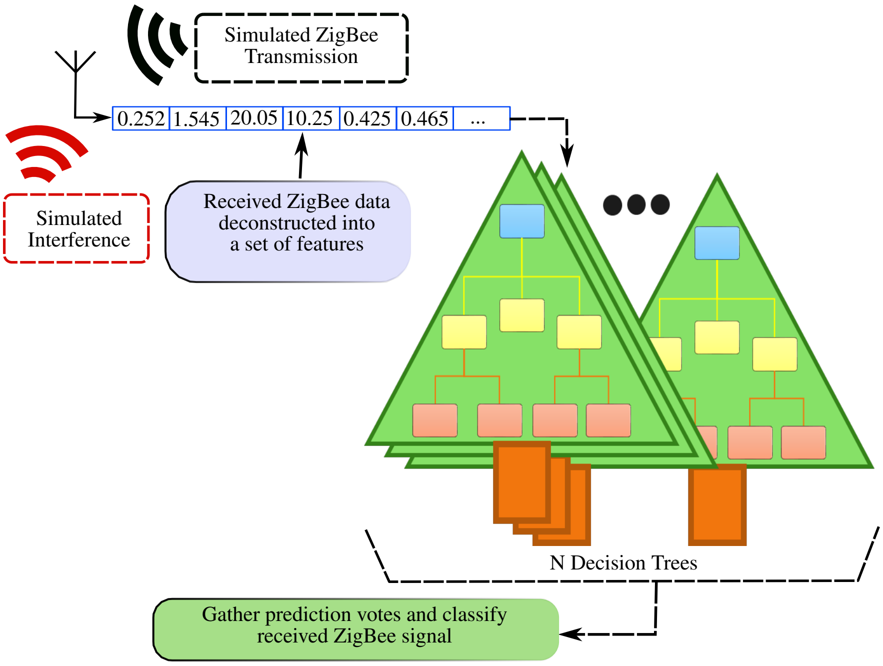

Simulated ZigBee Transmission
0.252
1.545
20.05
10.25
0.425
0.465
...
Simulated Interference
Received ZigBee data deconstructed into a set of features
N Decision Trees
Gather prediction votes and classify received ZigBee signal

Figure 4.13: Data flow diagram representing how the ZigBee data is collected, deconstructed and classified by the Random Forest algorithm.

4.5.1 Initial Results: Matched Interference

To begin, the procedure focused on matched signal interference and was validated using a SVM and expanded into a multi-class classifier by utilizing the Random Forest decision tree approach. For both approaches, the available feature data were divided into separate training (70%), validation (20%) and testing (10%) datasets, as sufficiently high volumes of data were available. For matched interference transmissions, erroneous

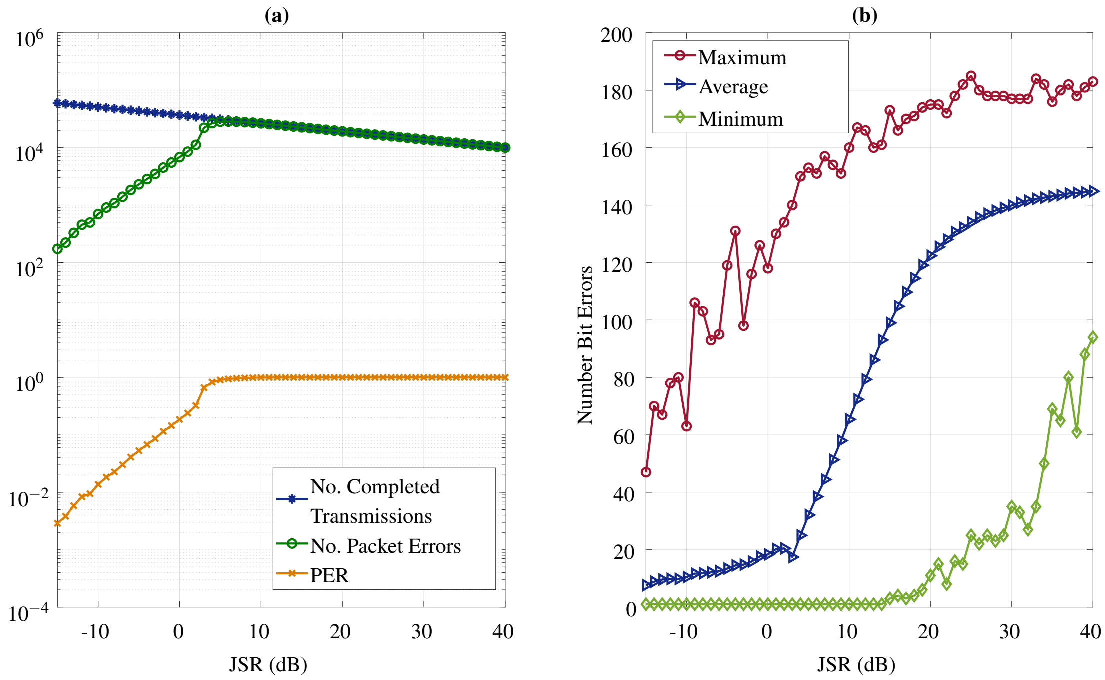


Figure 4.14: (a) The logarithmic approach to data collection used in the Monte Carlo simulations and associated packet error rate (PER), which is used to select the jamming regions to be classified. (b) A more in-depth visualization of the sporadic nature of the bit errors for each JSR point, showing the maximum, minimum and average bit errors at each point.

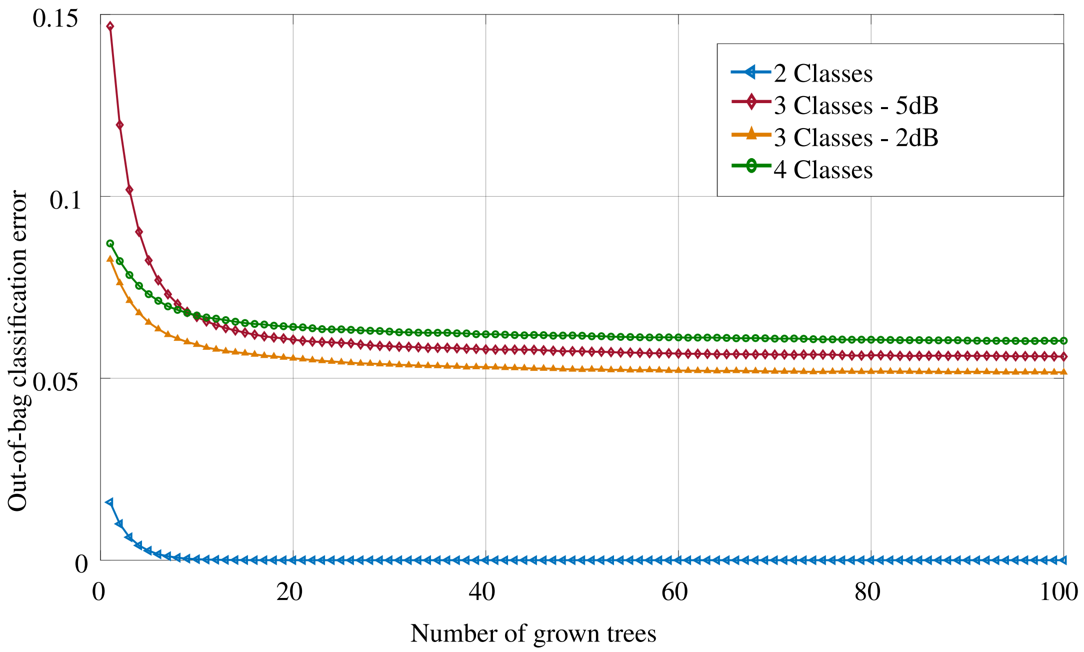


Figure 4.15: The Out-of-bag errors for the designed Random Forest algorithm for matched signal interference and full packet overlaps. Four separate scenarios are specified based on the analysis of the PERs and associated JSR value.

data includes JSR values decreasing, in steps of 1 dB, from 40 dB to -15 dB. Once a packet error occurs, the data is deconstructed and logged. Thus, the packet error data at each available data point is divided across the training, validation and testing datasets in the ratio 70:20:10. The number of executed simulations is on a logarithmic scale from 10,000 at 40dB JSR to 60,000 at -15dB JSR and visualized in Fig. 4.14 (a), where the performed number of trials and resulting packet errors are provided. A successful attack produces bit errors in a packet, as it results in either rejection or non-reception and requires retransmission. Here, a constant packet length of 40 bytes is used with an attack packet of matching length for a tight comparison across the JSR range. The designed algorithm attempts to identify why packets contain errors by simply analyzing the received I/Q samples. Training and validation were both implemented using the data from $-10dB \rightarrow 15dB$, which provided extra points for testing, and Matlab's ”fitcsvm” function. Also, as bit errors are sporadic when interference is supplied, as visualized in Fig. 4.14 (b), being able to identify different operating zones is advantageous. Therefore, different regions are defined and each classifier is used to classify these regions. The multi-class ability of the Random Forest algorithm is beneficial for this requirement and will be subsequently examined.

The initial matched signal SVM results examined JSR thresholds and the reasons for each threshold selection are provided in Table 4.2. The validation data and built-in Matlab functions were used to determine the appropriate SVM kernel to use. The decision was based on the 10-fold cross-validation error and the model training time. This approach encompassed the linear, RBF and third-order polynomial kernels. The RBF kernel was determined to be the optimal function, based on the achieved training time and model accuracy across the four zones specified in Table 4.2 and, so, the RBF kernel was utilized in the SVM analysis. The SVM was trained using data points from $-10dB \rightarrow 20dB$ and different binary detection thresholds to present a comprehensive study of the algorithm's performance. Table 4.2 identifies these thresholds, which are based on the bit-error analysis and feature trends outlined in Sections 4.2.1 and 4.3, respectively. 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 4.2 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 4.2. As a binary detection approach, the SVM achieves near-optimum performance, as per Table 4.4.

At first, the Random Forest results are provided through the out-of-bag (OOB) error. The OOB error is a method of measuring a decision tree algorithm's prediction error, utilizing bagging to sub-sample data samples used for training. OOB is the mean prediction error on each training sample x_i, using only trees that did not have x_i in

Table 4.2: 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
$\geq$ 5 dB	Identified Initial Threshold from Feature Trends	6.9619%	4.4508%
$\geq$ 0 dB	Expected Spectral Power	3.9628%	2.6515%
$\geq$ -5 dB	Below -5dB: Likely Error-Free SHR (Preamble and SFD)	0.9741%	0.8380%
$\geq$ -10 dB	Lowest Training JSR Point	2.35e-04%	0.0%

their bootstrap sample. The results are provided in Fig. 4.15 for four cases, including a two-class case for error-free and erroneous, an extended three-class case to separate the erroneous stage into PER regions of ≥ 0.32 and ≤ 0.32 and an erroneous case above and below a JSR of 5dB, which were identifiable during feature extraction. The PER regions relate to the decreasing slope towards low levels of PERs in Fig. 4.14, relating to, typically, unintentional interference. Finally, a four-class case is presented based on the packet/bit errors in Fig. 4.14. 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 $15 \rightarrow 20$ 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 $< -2dB$, $-2dB \rightarrow 2dB$ and $> 2dB$, 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. 4.15 specifies that the OOB error decreases with the number of trees and is much smaller for the two-class case. However, having such small differences between 'good' and 'bad' signals is, typically, not the best approach to ensure low false positives and high true positives. This algorithm's ability to define multiple cases is beneficial, as the high and medium jamming regions have a higher separation from error-free signals.

The four-class case outlines how a single Random Forest model can predict between different categories using a single model, which is an advantage over the SVM approach. The four-class case was validated using available validation and testing data to determine the optimal metrics, including the number of decision trees, feature depth and minimum percentage error. The feature depth, also referred to as predictor depth in this thesis, is the size of the random subset of features used in developing the unique weak learners. Validation data contained 20% of all available data and included varying the maximum feature depth from one to nine and the number of decision trees from one to 139. The four-class matched signal interference case results are shown in Fig. 4.16, where validation data determined the optimal metrics (hyper-parameters). The lowest

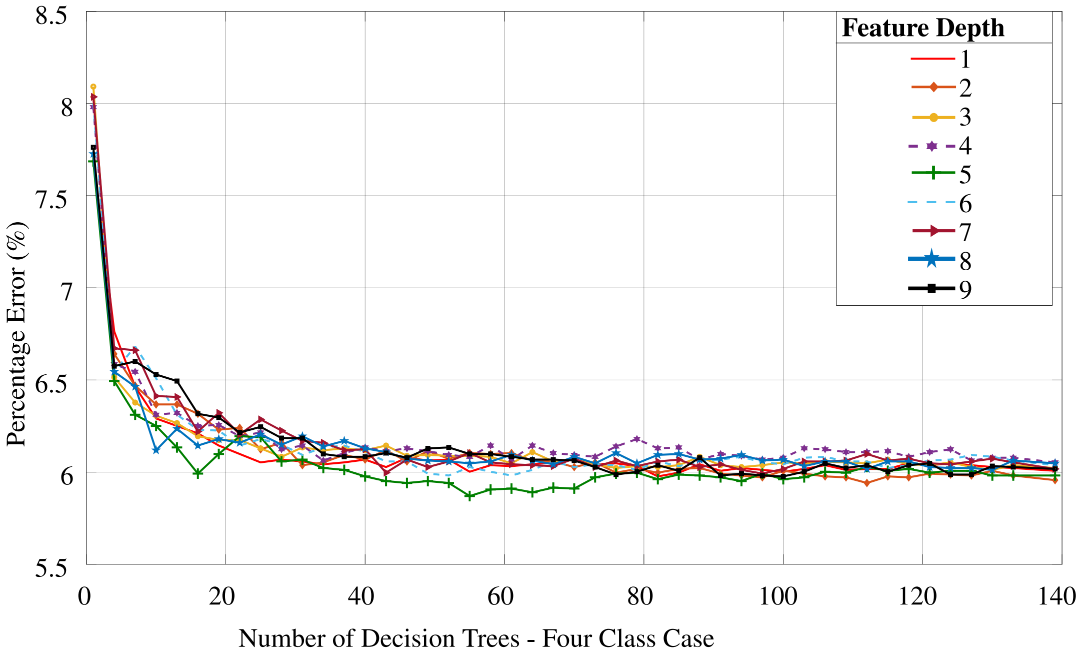
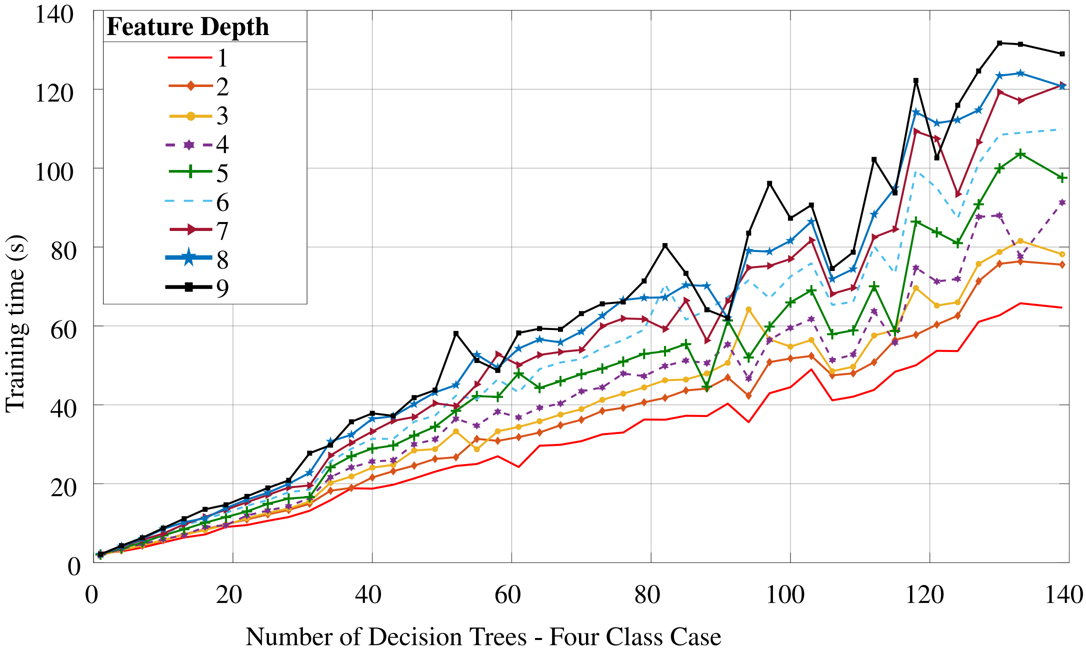


Figure 4.16: The generalization error investigation using the validation data for the random forest classifier. These results are for the four-class case under matched interference and full packet overlap.

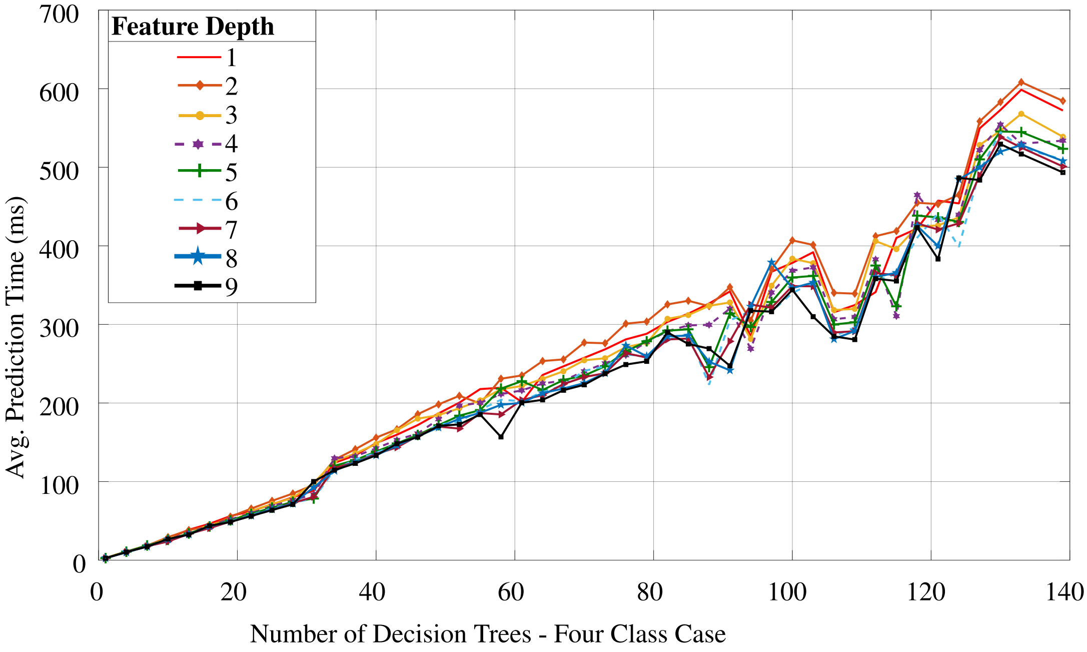
error level for the four-class case is $\approx 5.87\%$ using fifty-five decision trees and a maximum feature depth of five. The training time and average prediction time validation results are supplied in Figures 4.17 (a) and 4.17 (b), respectively.

Table 4.3 supplies the final calculated results for the algorithm design, where most errors occurred during classification into interference operating zones. A small deviation from the optimal occurs due to the use of a different seed in the random number generator, but differences are marginal. When this approach is applied as a binary classification, using the same metrics, the error was approximately 0% with an average prediction time of $43.1ms$. For the SVM, using the same data, the error was the same (0%), but the average prediction time was just $1.15ms$, see Table 4.4. 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 simulation results exhibiting low levels of error are artificially good as the data used to develop the models were simulated and, so, could not model live wireless signals exactly. Therefore, wireless channel variations, for example, fading levels, obstacles, path losses, spurious interference, etc., are not modeled with this approach. Only simulated noise and random center frequency offsets (a few tens of kilohertz) are modeled in these simulations. Therefore, the designed models and methods need to be adapted for wirelessly received I/Q samples. However, the objective of this study was to provide an initial validation of the usefulness of the extracted features and the varying thresholds show that enough differences exist between the error-free and erroneous



(a)



(b)

Figure 4.17: The (a) Training time and (b) Average prediction time investigations using the validation data for the random forest classifier. These results are for the four-class case under matched interference and full packet overlap.

Table 4.3: 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%

Table 4.4: 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 %

samples, even before the suggested 5dB threshold in Figures 4.8 and 4.9. Notably, as the threshold reduces, so too does the error, which suggests that features perform better when distinguishing between error-free and erroneous samples only. These promising simulation results suggest that this framework is a feasible solution. This result bodes well for a hardware approach that supplies real over the air live data signals and for the inclusion of different interference signals.

4.5.2 Expanded Results: Multiple Interference Types

The next phase examined using the designed features to classify the interference type, which included matched signal, CW, WiFi and 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). In each dataset, the proportion of each interference signal and error-free samples was consistent to avoid sampling bias, which is visualized in Table 4.5. The results here include the full set of 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, being achievable.

Table 4.5: 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

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 4.4, 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 (4.6) to be the optimal kernel. Based on the available testing data, the generalization error for full packet overlap was established as 0.00723% with an average prediction time of $1.67ms$.

For the Random Forest approach, optimal metrics (predictor depth and the number of trees) were determined using the validation and testing data. The optimal metrics are identified by analyzing the generalization error results in Fig. 4.18 and the training and average prediction time results in Fig. 4.19 (a) and Fig. 4.19 (b), respectively. Fig. 4.18 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 4.3, where the generalization error was 4.227% with an average prediction time of $714.24ms$. The associated confusion matrix is provided in Fig. 4.20, which provides an insight into the misclassification errors and suggests that the majority of cases can be correctly classified. When the classifier is analyzed in terms of ZigBee and ZigBee with interference causing bit errors, the sensitivity (4.7) and the specificity (4.8) are both approximately unity. This means that the multi-class classifier correctly detects interference and rejects legitimate ZigBee signals and the majority of errors occur when classifying between the interference types. The number of errors is due to inclusion of JSR values below 0dB, as this results in bit errors where the interference signal is not prominent and, so, the classifier detects the erroneous ZigBee signal but encounters difficulty in classifying the interference type. However, in general, interference signals will have sufficient power levels to be prominent and be classified correctly. 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

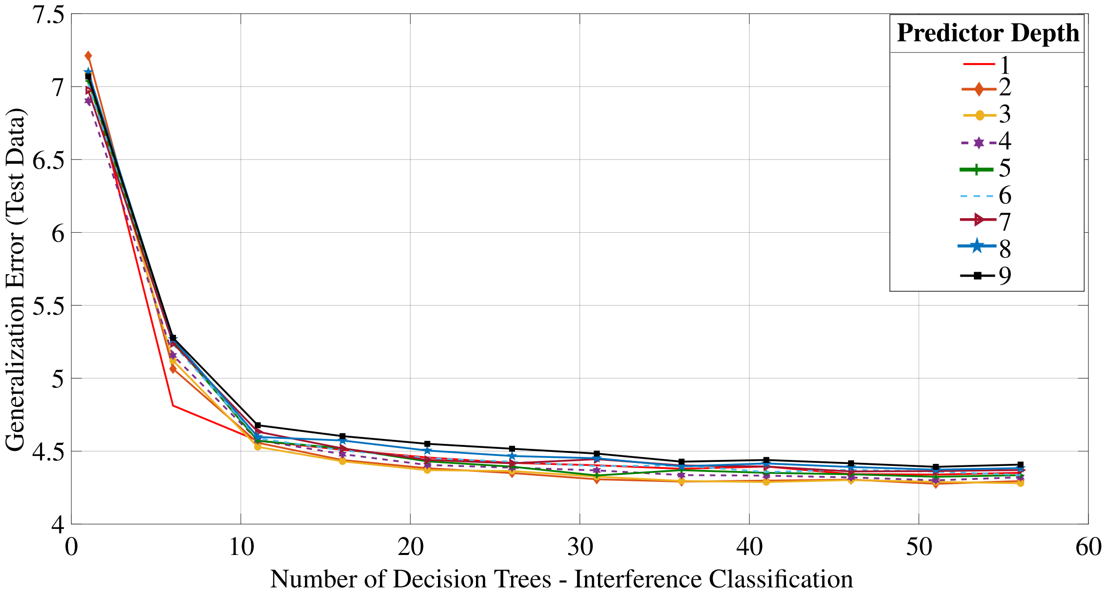
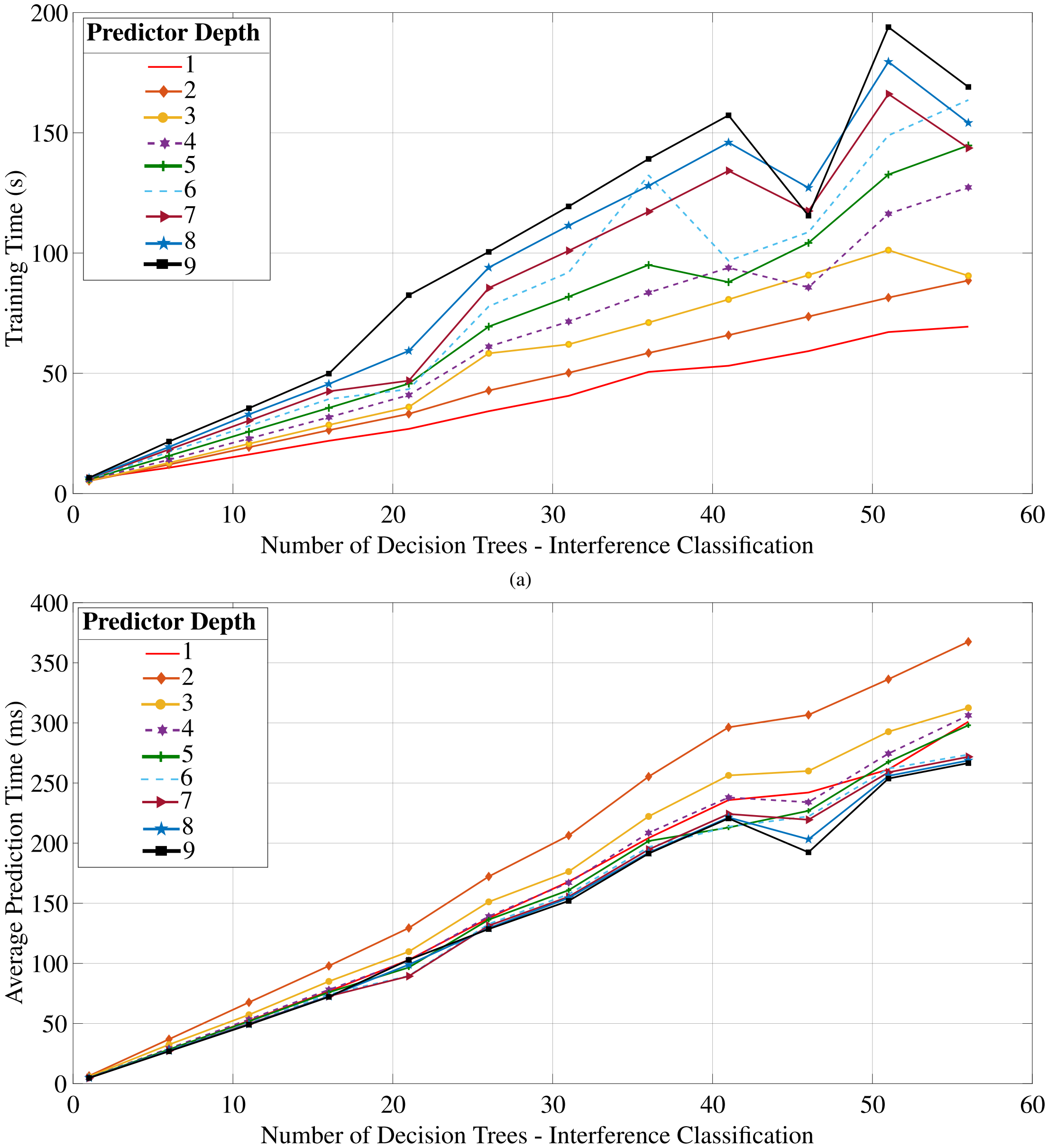


Figure 4.18: The Interference Classification results for the Random Forest algorithm using available validation and testing data. The optimal metric detection results for the designed Random Forest interference classification algorithm using full packet overlap, specifying the generalization error.

Fig. 4.21), 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 prediction time of 1.3*ms*. The confusion matrix for this SVM classifier was investigated and is supplied in Fig. 4.22, where the corresponding AUC of the ROC curve is approximately unity. This results in a sensitivity (4.7) of 0.9999% and a specificity (4.8) of 1, which means that the SVM classifier correctly detects interference and rejects legitimate ZigBee signals. 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. 4.23. As energy is typically limited on edge devices, the process of interference detection should only occur once a packet error has occurred and results should be determined as quickly as possible to enable real-time responses.

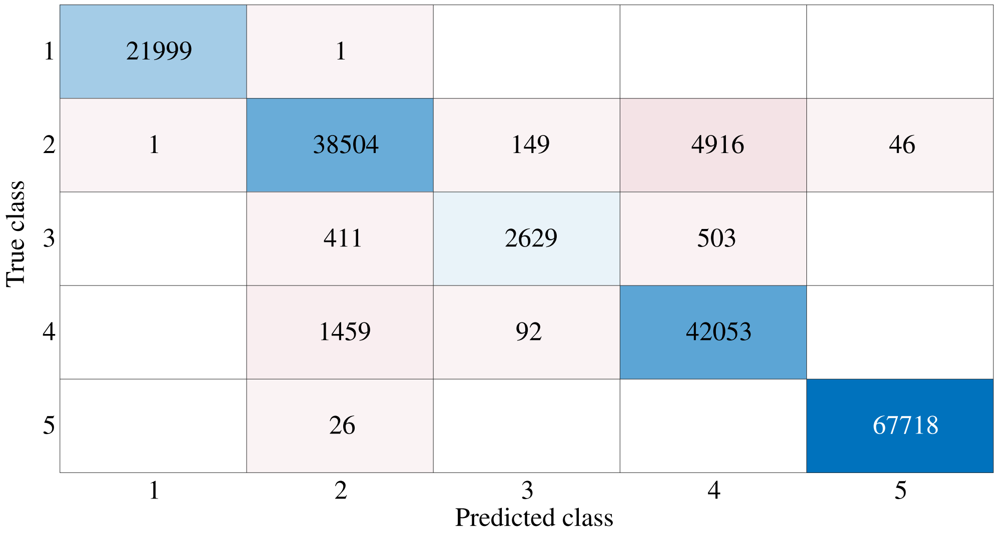
When the varying overlaps data were analyzed using the optimal Random Forest metrics for full overlap, the generalization error was 4.027%. The confusion matrix is shown in Fig. 4.24, using all available testing data to understand where the errors were occurring. Similar to the constant full packet overlap investigation, the decision tree classifier, when analysis focuses on ZigBee and ZigBee with interference causing bit errors, the sensitivity (4.7) and the specificity (4.8) are both approximately unity. As a result, the multi-class classifier correctly detects interference and rejects legitimate ZigBee signals and the majority of errors occur between classifying between the interference types. This is due to the low JSR values used in the simulations and, additionally, due to the small packet overlaps of 10% being implemented. The results show that the designed multi-class classifier detects interference for nearly every instance



(a)

(b)

Figure 4.19: The Interference Classification results for the Random Forest Algorithm using available validation and testing data. The optimal metric detection results for the designed Random Forest interference classification algorithm using full packet overlap, specifying (a) the training time and (b) the average prediction time.



True class \ Predicted class	1	2	3	4	5
1	21999	1			
2	1	38504	149	4916	46
3		411	2629	503	
4		1459	92	42053	
5		26			67718

Figure 4.20: 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.

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.

4.5.3 Discussion

When these simulation results, which investigate the ideal scenario without hardware limitations, are compared to fast jamming detection focused on collaborated packet rate information [105], the simulated performance achieves equal, if not better, results. This chapter's approach provides novelty over PDR-based systems as individual nodes can make decisions based on received I/Q samples. 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 [127].

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 not modeled. The absence of a real ADC means available resources do

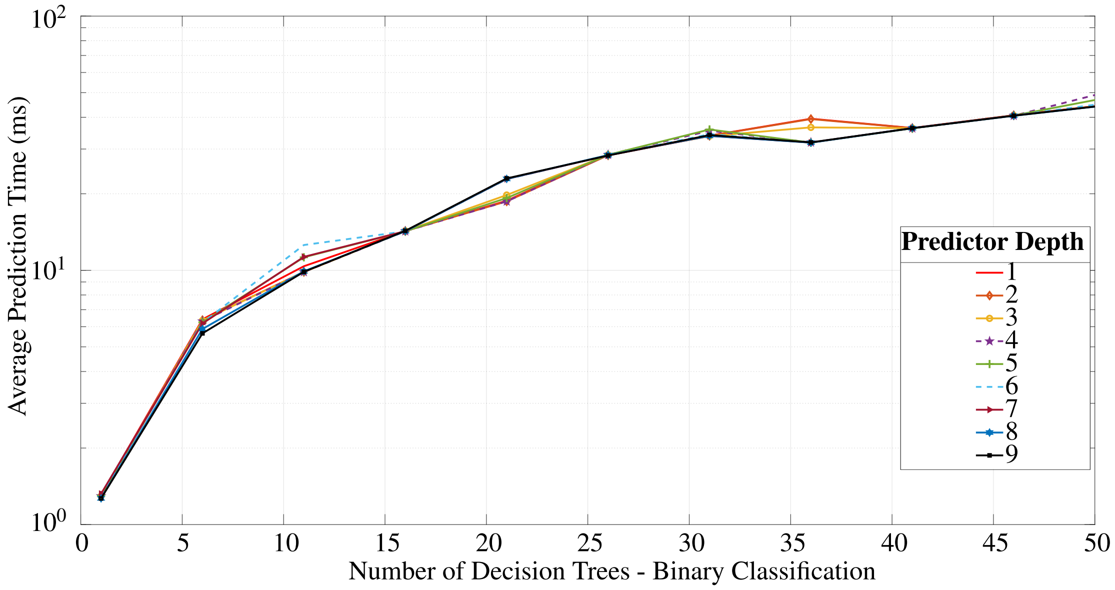
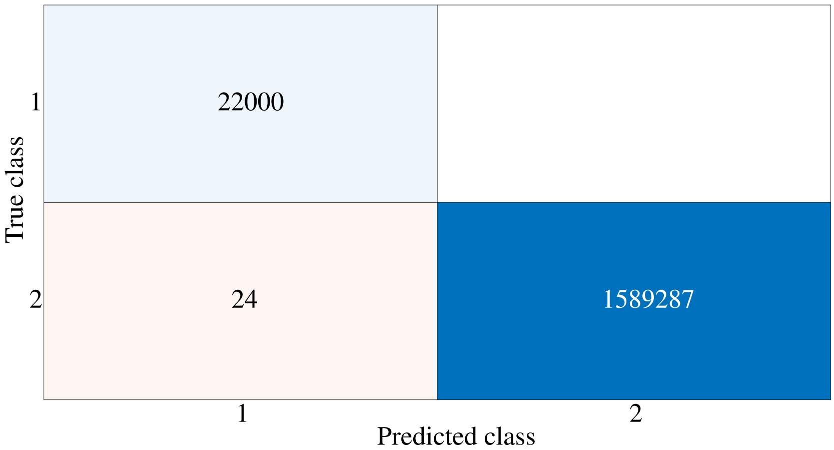


Figure 4.21: The Interference Detection results for the Random Forest Algorithm using available validation and testing data. Average prediction time results for binary classification between error free and interference signals using full packet overlap.

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 may be 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. Furthermore, the features extracted from these simulations will form the foundation for the hardware experimentation in Chapter 6, where hardware limitations and additional signal models with similar modulation schemes to ZigBee are studied.

Therefore, the designed models and methods need to be adapted for wirelessly received I/Q samples. However, the objective of this study was to provide an initial validation of the usefulness of the extracted features based exclusively on I/Q samples. The results of the varying thresholds show that enough differentiation exists between the error-free and erroneous samples, even before the suggested 5dB threshold in Figures 4.8 and 4.9. Notably, as the threshold reduces, so too does the error, which suggests that features perform better when distinguishing between error-free and erroneous samples only. These promising simulation results, using different packet overlaps and interference signals, suggest that this framework may be a feasible solution and bodes well for a hardware approach that supplies real over-the-air live data signals.



True class \ Predicted class	1	2
1	22000	
2	24	1589287

Figure 4.22: 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.

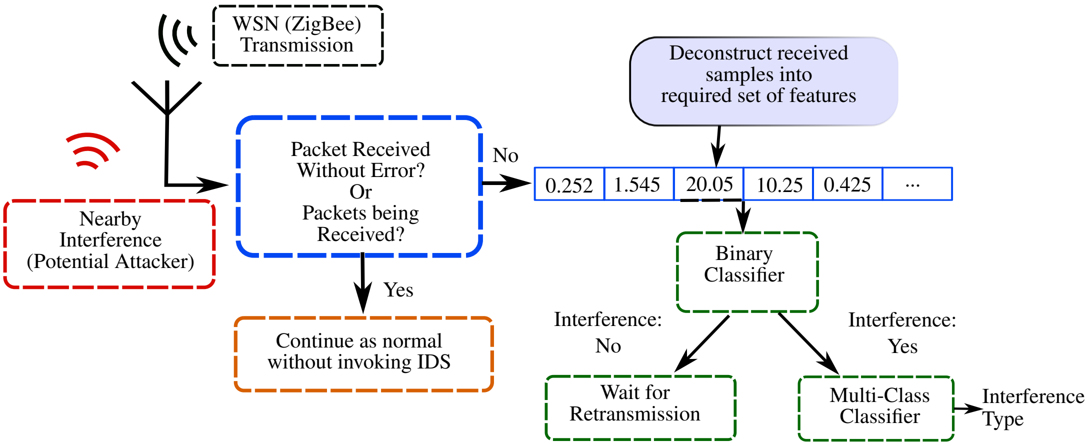
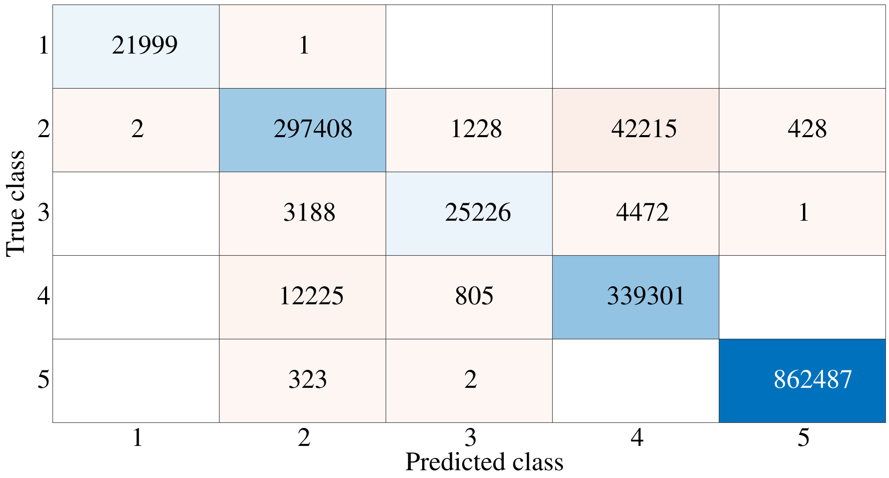


Figure 4.23: Data flow diagram representing the developed two model approach which leverages binary and multi-class classifiers.



True class \ Predicted class	1	2	3	4	5
1	21999	1			
2	2	297408	1228	42215	428
3		3188	25226	4472	1
4		12225	805	339301	
5		323	2		862487

Figure 4.24: 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.

4.6 Conclusion

This chapter focused on an extensive Matlab based simulation investigation using received I/Q samples for interference detection in WSNs. Both subtle, where the JSR values are causing PERs of 20% and below, and crude jamming attacks were examined. Bit error location analysis motivated a detection approach for subtle and crude interference attacks by specifying the requirements for an interference detection framework, including identifying interference when packets are received with errors and when no packet can be received. By focusing on received I/Q samples available on a single edge node, both of these cases are achievable. Features were extracted from the PDF and analysis of the time-series representation of the individual I/Q samples. Enough differentiation between error-free and erroneous samples existed to warrant an evaluation using a supervised machine learning classifier.

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 features extracted from the simulations are as a result of the ideal case where the PDF and samples have no numerical limitations caused by hardware restrictions. The simulation results established the detection methodology as a two-phase detection process. This process is visualized in Fig. 4.23 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 classification results using supervised discriminative machine learning classifiers validated the usefulness of the extracted features and highlight that heavily studied machine learning approaches are still fit for purpose, when potent data analysis and signal processing are employed. As a result of this chapter's work, live wirelessly received I/Q data is required as the simulation results provided sufficient evidence to warrant a hardware investigation. The next chapter focuses on developing hardware testbeds and associated data strategies for acquiring the necessary I/Q data. Chapter 6 will leverage the features extracted in this simulation study to establish an optimal set of features for legitimate wireless signal classification. Chapter 7 applies the optimal features from Chapter 6 for interference detection and classification using wirelessly received jammed data for WSNs and wired jammed data for GPS signals.

Chapter 5

Hardware: Developed WSN and SDR Testbeds and Data Collection Strategies

The information in this chapter is important for the experimental over-the-air work presented in Chapters 6 and 7. The specified hardware testbeds and data strategies were developed to gain access to the necessary live over-the-air wirelessly received I/Q data in a domestic wireless operating environment. The work in this chapter has been published in part in the following two conference papers:

- *G. D. O'Mahony, P. J. Harris and C. C. Murphy, "Developing Low-Cost Testbeds for Enhancing Security Techniques in Wireless Sensor Network Protocols," 2019 30th Irish Signals and Systems Conference (ISSC), Maynooth, Ireland, 2019, pp. 1-6, doi: 10.1109/ISSC.2019.8904967.*
- *G. D. O'Mahony, P. J. Harris and C. C. Murphy, "Analyzing using Software Defined Radios as Wireless Sensor Network Inspection and Testing Devices: An Internet of Things Penetration Testing Perspective," 2020 Global Internet of Things Summit (GIoTS), Dublin, Ireland, 2020, pp. 1-6, doi: 10.1109/GIOTS49054.2020.9119606.*

5.1 Introduction

The simulated results in Chapter 4, which exhibit low levels of error and "ideal" classifier performance, do not consider hardware restrictions and cannot model live wireless signals (and associated environmental interactions) exactly. These hardware restrictions include, for example, the receiver reference voltage and analog-to-digital (ADC) resolution, which both limit the numerical values of the received in-phase (I) and quadrature-phase (Q) samples compared to the simulations. Higher resolutions would allow for received signals to be extracted in greater detail from the channel. The reference voltage, which is the maximum voltage available to the ADC, determines the ADC conversion ceiling for received analog inputs. Essentially, a higher reference voltage allows for higher-powered signals to be received before saturation occurs. The absence of a real ADC in the simulations means there is no upper limit applied to the simulation values and extracted features only need to concentrate on the probability density function (PDF) and time series representations. Additionally, wireless channel characteristics such as, for example, fading levels, obstacles, path losses, spurious interference, etc., are inadequately modeled. As a result, it is essential to obtain wireless signal data from a typical operating environment to thoroughly investigate exclusively using I/Q samples to develop edge device diagnostic tools.

However, the promising simulation results from Chapter 4 suggest that focusing exclusively on I/Q samples may be a feasible solution that warrants a hardware approach rooted in real over-the-air signals. The simulation work provided insights for live data feature extraction and data analysis by identifying the type of data needed to train the jamming detection models. 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 results of Chapter 4 influenced the selection of hardware, developed testbeds and the data strategies established. This chapter discusses the hardware used in this thesis and explains why each device was selected. Each device's associated applications are specified, along with the developed data strategies utilizing the chosen wireless devices. The development of low-cost wireless sensor network (WSN) testbeds that transmit environmentally sensed data using the ZigBee protocol is the primary source of legitimate WSN data in this thesis. This chapter also demonstrates the advantages of utilizing software-defined radios, and available software packages, as WSN signal analysis and penetration testing tools. Extracting received I/Q samples in coexistence with both unintentional and malicious interference is essential for successfully developing the proposed methodology. SDRs enable access to the required I/Q data and provide the necessary jamming signals when needed, as the SDR can produce several signal models.

5.2 Hardware

The over-the-air wireless experimentation required different devices to successfully develop the proposed methodology as discussed in Chapter 1, namely, commercial

ZigBee nodes, a low-cost controlling device, a sensor for real data acquisition and an analysis and penetration device. These aspects were achieved by using the DIGI XBee [142] wireless nodes operating the ZigBee [37] protocol, the affordable Raspberry Pi computer, the Raspberry Pi SenseHat sensor and SDRs, respectively. Each of these devices and the chosen spectral analyzer (used to visualize transmitted wireless signals) are described in detail below.

5.2.1 XBee Devices

Here, the DIGI XBee wireless connectivity modules are the designated commercial ZigBee nodes. The devices operate using the ZigBee protocol but can also specifically run either the IEEE 802.15.4 or DIGI's DigiMesh 2.4 protocols. The radio frequency (RF) modules provide quick, robust communication in point-to-point, peer-to-peer, and multi-point/star configurations. These commercial-off-the-shelf (COTS) devices are low cost (3 pack kit ≈€90) and operate in deployments as a pure cable replacement for simple serial communication or as part of a more complex hub-and-spoke network of sensors. The devices have specifications as per Table 5.1 and each device is configured using DIGI's XCTU software [65]. Available parameters include the channel (center frequency), personal area network identifier (PAN ID), transmission (Tx) power and node use. The specific device types are [143]:

1. Coordinator: A full-function device (FFD) which is responsible for controlling the entire network, relaying messages and authenticating devices.
2. Router: A FFD responsible for forwarding and relaying data packets. This device type can communicate with the coordinator and end devices.
3. End Device: A reduced function device that communicates with the coordinator or a router only and cannot communicate with other end devices or relay packets.

The devices can utilize cluster topologies, which typically improves stability, reduces energy consumption and compresses the amount of transmitted data. Cluster head networks have many uses, for example, relay nodes (RN) which aggregate data and forward to Nanosatellites [8]. Here, the XBee devices are controlled and programmed through a development board that requires a USB connection and are programmed (after initial configuration using the XCTU software) using the "digi-xbee" Python3 library. Remote control and power of each XBee can be obtained by using a Raspberry Pi or equivalent, providing realistic deployment scenarios and real-time data analysis. The XBee device and associated development board are visualized in Fig. 5.1.

The signals transmitted from the XBee devices were validated by using a Texas Instruments **CC2531EMK** USB dongle. This ZigBee/IEEE 802.15.4 compliant System-on-Chip device employs TI's Packet Sniffer software [63] to both capture and decode

Table 5.1: XBee Device Specifications

Parameter:	Value
Data Rate	$250kbps$
Indoor Range	$60m$
Transmit Power	$-5dBm \rightarrow 5dBm$
Receiver Sensitivity	$-100\ dBm \rightarrow -102\ dBm$
Frequency Band / No. Channel	$2.4GHz$ / 16
Interference Immunity	DSSS
Encryption (Optional)	128-bit AES
Reliable Packet Delivery	Retries/Acknowledgements

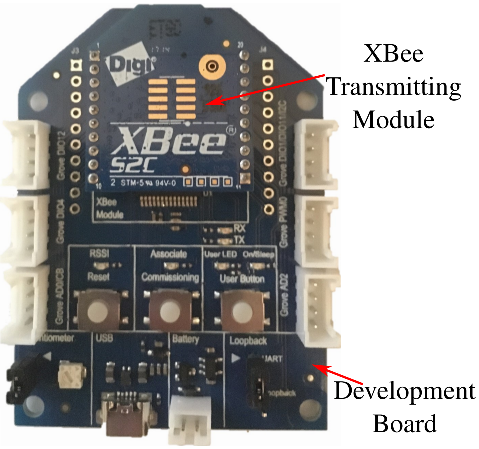


Figure 5.1: DIGI XBee device and associated development board used as the designated ZigBee node in the hardware experimentation of this thesis.

ZigBee packets. The packet sniffer successfully captured packets transmitted from XBee devices, validating using the chosen designated ZigBee node. The decoded information includes PAN ID, source and destination address, packet length, packet type (data or acknowledgement) and the payload, which is decoded if no encryption is used or if the key is known. This decoded information was employed to ensure the packets from the XBee devices were received as no encryption was used and the full payload was readable using the Packet Sniffer software.

5.2.2 Software-Defined Radios

Software-defined radios (SDRs) are reconfigurable radio systems whose characteristics are partially or fully defined via software or firmware [144]. A typical simplified SDR topology is provided in Fig. 5.2, where the main components are the RF antenna, RF front-end, field-programmable gate array (FPGA) and processing unit. A SDR interacts with the wireless environment using a hardware peripheral, whose capabilities characterize transceiver operation. The performance of the software component depends

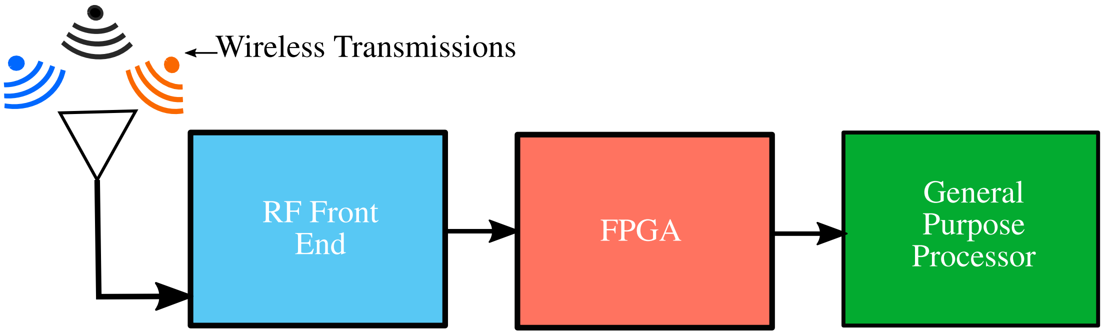


Figure 5.2: Simplified depiction of a typical SDR topology, specifying the main components that govern the operation of a SDR, including RF front-end, FPGA and general purpose processor. The RF front-end is the most significant component in terms of the work in this thesis.

on the proficiency of the RF front-end. Received analog RF signals are converted into a digital sequence, which depends on the available bandwidth and sampling rate in use. Hence, it is necessary to use SDRs with the appropriate hardware for analyzing the chosen RF signals. This thesis focuses on developing interference diagnostic tools in both the industrial, scientific and medical (ISM) RF band and the L1 Global positioning System (GPS) RF band. This approach results in receiving the I/Q samples of signals with baseband signal widths of 1.023 MHz (GPS) and 2 MHz (ZigBee) and implementing penetration tests on those signals.

By utilizing SDRs, raw received I/Q data can be obtained in real-time and analyzed off-line to gain an understanding of legitimate signals and the associated interactions with sources of interference. Using SDR architectures with hardware peripherals for the desired RF spectrum band, the software component can be manipulated to produce various jamming scenarios (or penetration tests) and legitimate signal structures for additional data acquisition by utilizing Matlab/Simulink or a Raspberry Pi/Python3 combination. This thesis used two specific SDRs to investigate the operation of the developed methodology and diagnostic framework using data collected from different receivers. This investigation provided an additional layer of validation as different hardware restrictions were implemented since the SDR receivers had different resolutions and available sampling rates. As a result, the sampling rate approach was chosen based on the most limited device, resulting in a sampling rate of twice the baseband signal frequency. This requirement will become more apparent in Section 5.4, where the developed data strategies are discussed.

The **Analog Pluto SDR** is an Analog Devices SDR ($\approx$ \$149) which has specifications as per Table 5.2 and is based on the Analog Devices AD9363 transceiver. This SDR is controlled and analyzed by either using Matlab/Simulink, through the Communications Systems Toolbox add on or using the Python3 library “pyadi-iio” in conjunction with the “libiio” package. The Pluto SDR can also be controlled using other SDR software programs such as, for example, GNU Radio, Pothos Flow, Cu-

Table 5.2: 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	20MHz	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

bicSDR, GQRX SDR, SDRConsole, etc., either through signal processing blocks or graphical user interfaces, where the SDR can, frequently, be exploited as a spectrum analyzer. In terms of this thesis, the focus is applied to the Python3 and Matlab approaches as this provides certain advantages, since a variety of attack styles can be tested quickly. The Pluto's Simulink/Matlab plug-in provides further benefits through the available toolboxes such as, for example, Signal Processing and Communications. These toolboxes, along with the transmitter and receiver Simulink blocks (or Matlab functions), provide efficient analysis methods spanning different modulation methods, filters, mixers, etc., which can be applied to create various physical layer (PHY) outputs and create signals which match specific protocols or attack styles. For lightweight Pluto operation, the Python3 "pyadi-iio" library is used in combination with "Scipy" to produce specific signal protocols or attack styles. Data collection and wired jamming are focused on lightweight operation, while wireless jamming attacks exploit the efficient and fast Matlab operation. In terms of the hardware specifications, which are a limiting factor compared to the simulations in Chapter 4, the ADC resolution is 12-bits and the reference voltage is 1.3 V. As a result, the Pluto receives samples in the region of [-2048:2047] [100]. Compared to other available SDRs and examples of the more expensive universal serial radio peripheral (USRP), in Table 5.3, the Pluto SDR provides the most cost-effective approach without forfeiting performance. The additional easy-to-use Matlab and Python3 resources, resulting in lightweight operation on, for example, a Raspberry Pi, further ratifies the choice of the Analog Pluto SDR. How the Pluto is used in the thesis is fully specified in Sections 5.3 and 5.4.

The **RTL-SDR** dongle was chosen as the GPS data collection device based on previous experimentation [41] and to investigate using a cheap SDR with a reduced ADC resolution compared to the Pluto SDR. This comparison of different receiving resolutions provides a more intensive validation of the proposed diagnostic methodology. The NESDR SMArTee RTL-SDR ($\approx$ \$32) [145] was the chosen receiver and receives samples in the range [-128:127]. This SDR has specifications as per Table 5.2, contains the required RF-suitable 4.5V regulator that provides DC output to power the active

Table 5.3: Additional SDR Specifications

SDR	Interface	Frequency Range	RF Bandwidth	ADC Resolution	Mode Mode	TX/RX Channels	Approx Price
LimeSDR	USB 3.0	$100kHz \rightarrow 3.8GHz$	$61.44MHz$	12 bit I/Q	Full Duplex	2/2	$299
LimeSDR Mini	USB 3.0	$10MHz \rightarrow 3.5GHz$	$30.72MHz$	12 bit I/Q	Full Duplex	1/1	$159
HackRF One	USB 2.0	$1MHz \rightarrow 6GHz$	$20MHz$	8 bit I/Q	Half Duplex	1/1	$299
BladeRF x40	USB 3.0	$300MHz \rightarrow 3.8GHz$	$40MHz$	12 bit I/Q	Full Duplex	1/1	$420
FreeSRP	USB 3.0	$70MHz \rightarrow 6GHz$	$61.44MHz$	12 bit I/Q	Full Duplex	1/1	$420
Ettus B200	USB 3.0	$70MHz \rightarrow 6GHz$	$61.44MHz$	12 bit I/Q	Full Duplex	1/1	$888
Ettus B210	USB 3.0	$70MHz \rightarrow 6GHz$	$61.44MHz$	12 bit I/Q	Full Duplex	2/2	$1472

GPS antenna (see Section 5.2.5) and is based on the Realtek RTL2832U chipset. The RTL2832U outputs 8-bit I/Q-samples, and the highest theoretically possible sample-rate is 3.2 MS/s. However, the highest sample-rate without lost samples that has been tested with regular USB controllers so far is 2.4 MS/s [146]. This SDR was chosen due to the availability of the "rtl-sdr" software package, which contains the "librtlsdr" library and several command-line tools such as rtl_test, rtl_sdr, rtl_tcp and rtl_fm. These command-line tools use the "librtlsdr" library to test for the existence of RTL2832 devices and to perform basic data transfer functions to and from the device. Because most of the RTL2832 devices are connected using USB, the librtlsdr library depends on the "libusb" library to communicate with the device. The rtl_sdr command-line program is used for I/Q data collection and can function on a Raspberry Pi through the Python3 command "os.system('rtl_sdr -f 1575420000 -s Sampling Rate -n Number of Samples -S File name.uint8')", where the "uint8" data can be converted to "int8" data. Notably, it produces data outputs that are compatible with "fastgps" [147], which is a GPS software receiver that performs the entire signal processing in software, allowing for smooth adjustments at all stages: correlation, acquisition, and navigation. 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. The Raspberry Pi and RTL-SDR combination produces an all-in-one GPS data collector and receiver.

By utilizing these devices, the necessary raw received I/Q data can be analyzed to understand why links fail and what is causing the interference levels to rise. Notably, the SDRs can receive I/Q samples even when ZigBee packets are erroneous and the packet error rate (PER) is close to 1. In contrast, many packet sniffers need to synchronize to the packet preamble and identify the start frame delimiter. This operation outlines why SDRs are useful as WSN or GPS signal/samples analysis tools and for a subsection of penetration testing on wireless signals operating in an environment under legitimate and attack situations.

5.2.3 Raspberry Pi

The Raspberry Pi is a small credit card-sized, affordable (≈€40) computer that can be utilized in several ways to provide remote control and low-cost operation of USB powered devices. As part of this thesis, the USB powered devices include the XBee ZigBee nodes and SDRs. Additionally, live environmental data is produced by connecting the Raspberry Pi SenseHat sensor (see Section 5.2.3.1). Once the initial setup is performed, a remote connection can be achieved through a secure shell (SSH) in Windows, using a putty terminal and the device's IP address, or Linux using the command "$ssh\ pi@IP-Address$". Different Raspberry Pi models are employed, including the Pi 2 Model B, Pi 3 Model B and Pi 3 Model B+, which is shown in Fig. 5.3, as it is the model that is predominantly deployed. The Raspberry Pi 3 Model B+ is applied as the relatively modest embedded platform in Chapters 6 and 7. The main specifications of the different Raspberry Pi models are provided in Table 5.4. These small computers run a Debian-based operating system, called Raspbian, and can run full Linux applications, like GNU Radio, python scripts, terminal commands, machine learning models, etc. Hence, each Raspberry Pi can use the "digi-xbee" Python3 library to control an XBee device, the "pyadi-iio" Python3 library and the "libiio" library to control the Pluto SDR and the "rtl-sdr" command-line software to operate the RTL-SDR. Additionally, received XBee packet data can be stored on the Pi and analyzed to track received and transmitted packet numbers. Built-in WLAN or a WLAN USB dongle provide remote access and the ability to update code and hardware without the need to be physically at the testbed developed using Raspberry Pi embedded devices. These devices support real-world deployment scenarios as, typically, WSN edge devices are left unattended and deployed where remote access or monitoring is the norm. Furthermore, to enable IoT capabilities, a WiFi-enabled device can use available tools, for example, DropBox Uploader exploiting a DropBox API application, to upload received data to the internet for remote analysis.

In terms of implementing the proposed methodology, initial investigations focused on using the leading machine learning programming language of Python on the Raspberry Pi Device. In Chapters 6 and 7, a subset of the developed classification approaches are computed on a RaspberryPi embedded device. The Raspberry Pi was chosen as it is an example of how low-cost hardware has advanced over the past decade. As we look to the future, it is not unreasonable to suggest that edge devices will have similar specifications. In this thesis, the versatility of the Raspberry Pi embedded device is utilized in the development of testbeds (Section 5.3), data strategies (Section 5.4) and in investigating the low-cost resource-constrained implementation of developed machine learning models.

Table 5.4: Raspberry Pi Specifications

Pi 2 Model B	A 900MHz quad-core ARM Cortex-A7 CPU 1 GB RAM 4 USB ports
Pi 3 Model B	Quad Core 1.2GHz Broadcom BCM2837 64 bit CPU 1 GB RAM 4 USB Ports BCM43438 wireless LAN Bluetooth Low Energy (BLE)
Pi 3 Model B +	1.4GHz Broadcom BCM2837B0, Cortex-A53 64-bit SoC 1 GB RAM 4 USB 2.0 Ports 2.4 GHz and 5 GHz IEEE 802.11.b/g/n/ac WLAN Bluetooth 4.2, BLE

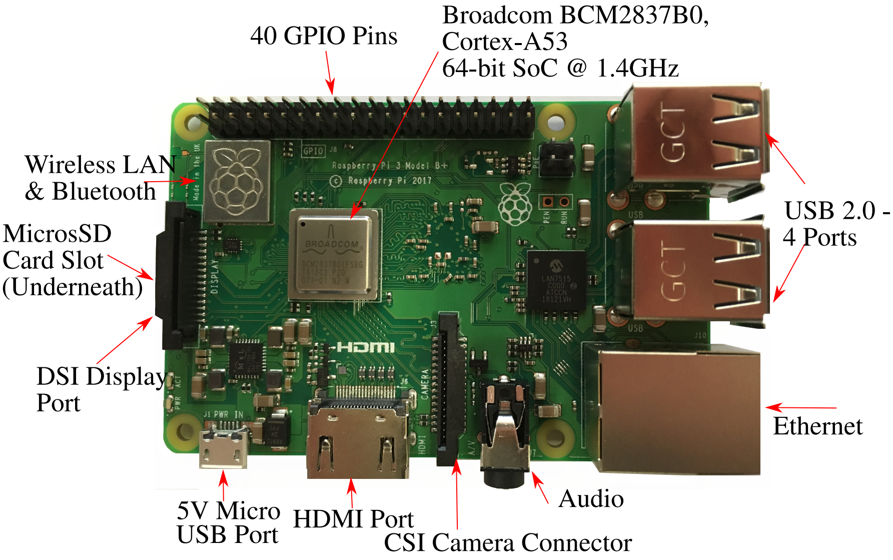


Figure 5.3: Raspberry Pi Model 3 B+ embedded device used as the main component in the developed hardware testbeds and data strategies. Some of the main elements are identified.

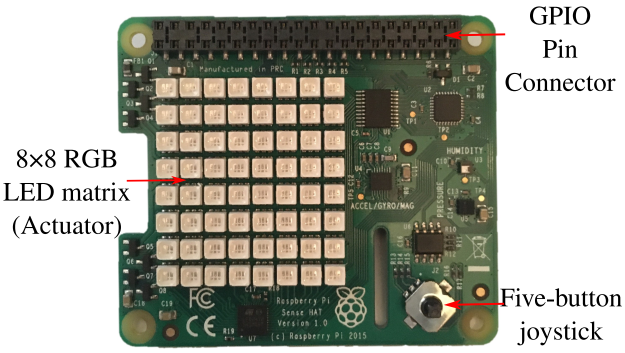


Figure 5.4: Raspberry Pi SenseHat Sensor used to acquire data from the operating environment with elements identified.

5.2.3.1 Raspberry Pi SenseHat

The SenseHat is an add-on board for the Raspberry Pi that connects directly to the available GPIO pins on the main Raspberry Pi board. The SenseHat has an 8×8 RGB LED matrix that can imitate an actuator, a five-button joystick and several sensors including Gyroscope, Accelerometer, Magnetometer, Temperature, Barometric pressure and Humidity. For this thesis, the environmental conditions of pressure, temperature and humidity were used as the source of real data, transmitted using the XBee RF module. The device is depicted in Fig. 5.4 and can be controlled using the Python3 library “sense-hat” running directly on the Raspberry Pi.

5.2.4 Tektronix RTSA 306B

This Tektronix device is a USB 3.0 powered Real-Time Spectrum Analyzer (RTSA) that is predominately used to visualize the RF spectrum. As a result, the RTSA can analyze network operation, identify nodes, estimate node type, visualize signal structure and analyze how signals co-exist with each other. Tektronix’s Digital Phosphor technology (DPX), which runs on the SignalVu-PC software package, acquires signals in real-time and visualizes the spectrum. DPX performs hardware digital signal processing and rasterizing of samples into pixel information, which can be plotted in real-time and as a bitmap image (instead of a conventional line trace). This software approach allows signals to be distinguished at the same frequency and a color scheme is used to identify signals which are more frequent than others. An example is provided in Fig. 5.5, where the XBee transmitted ZigBee signal is captured by the RTSA and plotted using the DPX software to gain further understanding of the signals in transit and the operating environment. This approach will be used in Chapters 6 and 7 to support the development of the proposed methodology. Consequently, legitimate protocol, jamming and coexisting signals are identifiable during an experiment to ensure the necessary data is acquired. The RTSA’s main specifications are provided in Table 5.5 and the device was controlled exclusively with the SignalVu-PC software package.

Table 5.5: Tektronix RTSA 306B Specifications

Frequency Range	9 kHz to 6.2 GHz
Acquisition Bandwidth	40 MHz
Typical Accuracy	$+/-\ 20ppm$
ADC Sample Rate	112 MSps
ADC Bit Width	14 bits

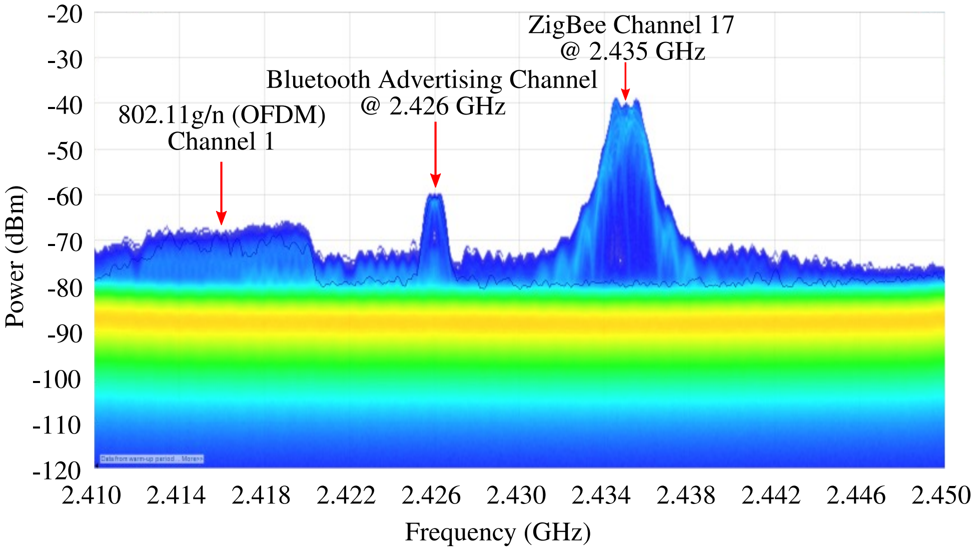


Figure 5.5: ZigBee signal, visualized using a RTSA and Tektronix's DPX software, in the ISM band at 2.435 GHz with coexisting signals.

5.2.5 Antennas

Typically, the chosen RF antenna is a critical element of the receiving device and, consequently, two specific antennas were used in this thesis. For the WSN interference study and ISM RF band legitimate signal analysis a ZigBee Siretta stubby antenna [148] was utilized. The Siretta antenna is designed for use in the 2.4→2.5 GHz range and has a 2 dBi gain, vertical polarization, a maximum VSWR of 2.0 and an input impedance of 50Ω. The experiments that utilize the Raspberry Pi/Pluto SDR approach to access received I/Q samples at various center frequencies adopt the Siretta antenna. Additionally, the Tektronix RTSA uses this stubby antenna to produce the DPX spectrum visualizations. A waterproof, magnetically mounted active GPS L1 (1.5754 GHz) low-noise amplifier (LNA) patch antenna that provides approximately 28 dB of gain was utilized for the GPS applications. This antenna requires a bias voltage between 3V→5V and connects to the RTL-SDR device through an SMA connection.

5.3 Developed Testbeds

For this thesis, two different testbeds were required: a live WSN ZigBee testbed with potential IoT operation and a GPS receiving testbed that allows access to available satellites in orbit. By designing these two approaches, the necessary legitimate I/Q data for each signal model was attainable.

The designed ZigBee testbed contains the Digi XBee devices being powered and controlled remotely using various Raspberry Pi models. Using a Raspberry Pi with XBee devices was previously described using the GPIO serial port in [149] and in

```
from digi.xbee.devices import ZigBeeDevice, RemoteZigBeeDevice
from digi.xbee.reader import XBee64BitAddress
import time
#Transmitter running on XBee Router
XBee_Router = ZigBeeDevice(“/dev/ttyUSB0”, 9600)
XBee_Router.open()
Remote_XBee_Coord = RemoteZigBeeDevice(XBee_Router,...
XBee64BitAddress.from_hex_string(Coord_64_Bit_Address))
while True:
Try:
Data = “Test Packet”
XBee_Router.send_data(Remote_XBee_Coord, Data)
time.sleep(120)
except:
XBee_Router.close()
#Receiver running on XBee Coordinator
XBee_Coord = ZigBeeDevice(“/dev/ttyUSB0”, 9600)
XBee_Coord.open()
while True:
Try:
Received_Message = XBee_Coord.read_data(1000)
Remote_XBee_Receiver = Received_Message.remote_device
Received_Data = Received_Message.data
except:
XBee_Coord.close()
```

Figure 5.6: Example Python3 code for controlling a Raspberry Pi connected XBee device, showing a basic understanding of how the “digi-xbee” library operates and how data is sent between two XBee nodes.

[150], [151], [152], [153] and [154], where one Raspberry Pi device is typically used as a base station attached to the network coordinator and provides client services, which allows applications, like environmental monitoring, to be established. Here, the approach is different as the development board is exploited to allow complete control and access to every node, which has been initialized using XCTU software. This concept authorizes full control of each node, packet analysis on each receiver and transmitter and data analytics at each node location. The node connections follow typical WSN operation as the nodes are static, use the cluster head network model and are sufficiently dispersed. This ZigBee-based testbed includes one coordinator (cluster head) and multiple receivers (cluster members). Each XBee device is connected to a Raspberry Pi using the “digi-xbee” Python3 library to control the XBee nodes and allow remote access through built-in wireless LAN or USB dongles. Example code which uses an XBee device to transmit a message every 2 minutes and the corresponding receiver code is provided in Fig. 5.6. Under regular operation, it is envisaged that each cluster member can send a data packet every x seconds/minutes, while each Raspberry Pi device monitors all packets. The remote access aspect is typical of WSN deployments, as they are usually deployed and left unattended. The data is uploaded through a gateway that can be accessed from clients monitoring the network. Additionally, this testbed can be used with a different number of nodes and varying packet sending rates or as a point-to-point communication model, which allows for a broader range of operation.

The full testbed incorporates six XBee nodes connected to six Raspberry Pi devices and transmits environmentally sensed data, including temperature, humidity and pres-

Table 5.6: ZigBee Testbed Validation: SenseHat Data

Node	One	Two	Three	Five	Six
Validation Test One - Node Four as Coordinator (Receiver)					
Approx. Operating Time (Hours)	64	64	64	64	64
Data Packets Transmitted	2042	2042	2006	2006	2042
Data Packets Received	2042	2042	2006	2006	2042
Validation Test Two - Node Four as Coordinator (Receiver)					
Approx. Operating Time (Hours)	24	24	24	24	24
Data Packets Transmitted	749	749	749	749	749
Data Packets Received	749	749	749	749	749

sure, obtained from the Raspberry Pi SenseHat sensor. Testbed validation occurred over multiple tests, where five SenseHats collected environmental data and XBee devices transmitted the data to a central coordinator. The operation was controlled using various types of Raspberry Pis and the results are provided in Table 5.6. To enable IoT abilities, a WiFi-enabled coordinator can use available tools, for example, DropBox Uploader exploiting a DropBox API application, to upload received data to the internet for remote analysis. Data were verified as ZigBee signals by sniffing the channel using the TI packet sniffer and for zero packet loss by tracking all transmitted and received data packets.

In terms of GPS data acquisition, a single GPS receiver is required as the transmitters (satellites) are freely available once suitable software and hardware are used. The GPS space segment in operation consists of a constellation of at least 24 satellites transmitting radio signals to users. To maintain the availability of at least 24 operational GPS satellites, 95% of the time, 31 operational GPS satellites are in orbit [42]. This constellation can be validated by using GNSSRadar [155], a software tool used to show the current GPS constellation for a specific location and the satellites' orbital speeds. The GPS testbed consists of the NESDR SMArTee RTL-SDR connected to the active GPS L1 patch antenna through a signal combiner, where a Pluto SDR is connected to the second port through a DC block. For regular operation, a 50Ω termination must be applied to the DC block to mimic the Pluto SDR and have comparable results with and without the SDR interferer connected. GPS data was collected across a two distinct full 24 hour periods without an interference signal to validate this setup. Using "FastGPS" as the software receiver, all available 31 satellites were visible at least once and each collected and analyzed dataset contained at least four satellites.

Both of these testbeds produce the required legitimate signal protocol I/Q data. For legitimate signal received I/Q data analysis, a SDR can simply receive on the required channel and acquire I/Q samples for off-line examination. This data analysis does require a data strategy, which is discussed in Section 5.4. SDRs are the critical element in translating and, potentially, validating the simulated approach in Chapter 4 to a real wireless application. The required I/Q samples can be received even when the channel becomes noisy and packets are lost or erroneous, since no preamble is required, as reception can be linked to known packet transmission times or periods where a sufficiently long data collection time is applied. This approach results in data collection for erroneous, error-free, and interference instances for feature extraction and analysis. Once analyzed, results can be implemented as part of the designed machine learning detection approach to highlight why a link has failed and packets have been lost.

The developed ZigBee and GPS testbeds can be penetration tested to gain access to the interference data. In terms of this thesis, penetration tests are performed to assess system responses to external interference. Thus, a SDR having an RF front-end capable of transmitting in the ISM and GPS L1 RF bands can be used as an interference response tester. The penetration testing approach uses Matlab functions and the "pyadi-iio" Python3 library to produce the necessary attack styles. A power amplifier is utilized to mimic typical scenarios where a power amplifier would be necessary to attack a sufficiently large network area. The Analog Devices CN0417 is chosen as it is designed for use with the Analog Pluto SDR. The power amplifier provides an additional 20dB of gain (approximately) and all ports are DC blocked and matched to 50 Ω. For wired jamming transmissions SDRs are connected through a DC block and power combiner to produce I/Q samples containing various interference signals without affecting neighboring networks. This operation is essential for the GPS applications as jamming signals cannot legally be wirelessly transmitted in the GPS L1 RF band. For the ZigBee WSN setup, wireless jamming is used as the network can be sufficiently isolated from any other potentially active IEEE802.15.4 systems. This isolation is ensured by operating the experiments in a typical domestic operating environment, where no other IEEE802.15.4 system is in use. This penetration testbed approach is a much lower cost option to traditional signal generators, is software configurable, is an open-source technique and has remote deployment potential. The Pluto libraries also allow Python's and Matlab's multitude of data science approaches to be applied to received data points, emphasizing using SDRs as interference transmitters and received response analyzers.

5.4 Developed Data Strategies

Before extracting any features or designing any diagnostic tool, data strategies accounting for data quality, quantity and source need to be established. These strategies are necessary as classifiers will generally not perform adequately if the training set is too small, or if the data is not representative, is noisy, or is "polluted" with irrelevant features. Each data strategy in this thesis incorporates SDRs, Raspberry Pi embedded devices

[156] and a variation of the testbeds described in Section 5.3. Each data strategy targets collecting typical wireless I/Q data in real-time for off-line feature extraction and machine learning-based classifier development. Each SDR interacts with the RF environment by utilizing a hardware peripheral, whose capabilities characterize transceiver operation. At the same time, 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. Four different data strategies were developed: wired legitimate signal classification, wireless legitimate signal classification, WSN ZigBee interference detection and classification and GPS signal interference detection and classification. A 4 MHz sampling rate is applied in all data collection experiments, except for the GPS data collection. This sampling rate is twice the baseband signal bandwidth of the ZigBee protocol, which is the primary signal of interest and allows for a comparison of the developed methodology on the 8-bit RTL-SDR dongle used for GPS data. The RTL-SDR provides a 2.048 MHz sampling rate, which is twice the GPS baseband signal bandwidth. As a result, the SDR I/Q data receiver's sampling rate is twice the baseband signal bandwidth of the application signal of interest, namely, ZigBee and GPS. The overall 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).

The Pluto SDR is used as the I/Q data receiver in each strategy involving signals in the 2.4 $\rightarrow$ 2.5 GHz ISM RF band. As a result, data is received in the range of [-2048:2047] and stored in ".txt" files in a two-column format, where the first column corresponds to the I channel and the second to the Q-channel. The data length of each data collection was 0.5 s and ensured at least one signal transmission would be received in every SDR reception, provided that the transmitting device produced signals of a sufficiently short period. The Pluto ADC specifications were then used to convert received I/Q samples into the range [-1,1] from the original Pluto range of [-2048, 2047]. The Pluto SDR uses the Analog Devices "AD9363" RF chip, which has a 12-bit ADC, where the 12-bit data from the ADC is stored in the lower 12 bits of the output value and sign-extended to 16 bits. This conversion supported developing features with similar ranges, even when the ADC is close to saturation. This technique typically results in higher-performing machine learning classification models.

Before the data is analyzed, it needs to be pre-processed. As shown in Fig. 5.7, the wireless channel is open to any accessible wireless transmitter and spurious interference. It is challenging to receive only a specific type of signal when collecting a data grab during a random time period using an unconnected device (SDR). As a result, most SDR data receptions (grabs) contained multiple signals of interest along with, potentiality, other "unwanted" signals. The data grabs required processing to obtain the correct

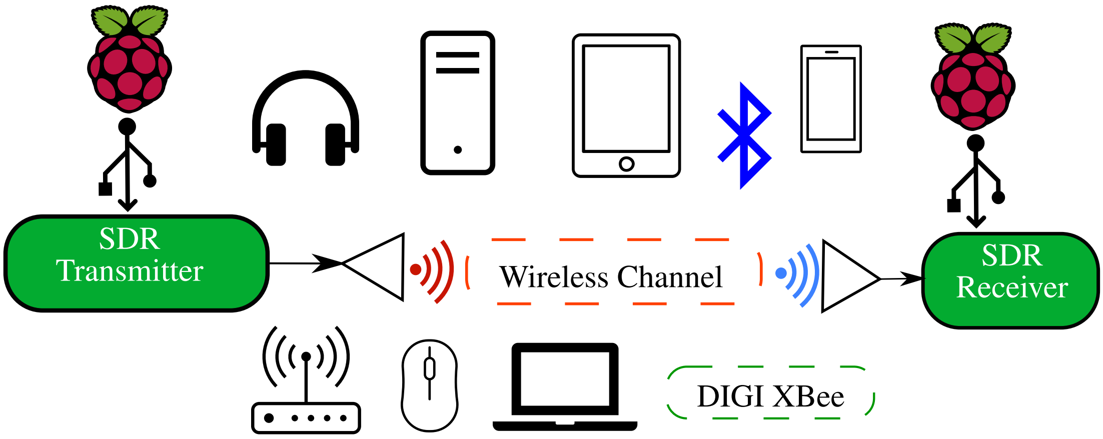


Figure 5.7: Proposed SDR approach for achieving the desired data strategy for I/Q samples in the ISM RF band between 2.4 GHz and 2.5 GHz. Typical commercial devices that are in operation in the typical domestic operating environment are also specified.

signal samples before the feature engineering stage. This processing stage was used to reduce the error/noise in the collected data and to discard obvious outliers. Without this data processing stage, poor-quality, unrepresentative, noisy, or polluted (with irrelevant features) measurements could be employed in the model development process.

The I/Q samples were initially examined in the time domain to visualize the signal patterns (I/Q samples) of interest and compare them with expected sequences. An example is shown in Fig. 5.8 to illustrate that the signals can be clearly identified compared to the received noise (or operating wireless channel), as signals have an "on-off" nature. The PDF, as introduced in Chapter 4, the Tektronix RTSA and the time series plots are then combined to identify any outliers. This analysis permits the removal of outliers and the acquisition of I/Q samples for each received signal type. The chosen sample length for analysis is 1250 I/Q samples and relates to the shortest signal length received as part of the data collection process in this thesis, the Bluetooth Advertising channel. After the samples are identified, the more considerable signal lengths are divided into multiple segments to ensure the data is from different parts of the received packet and results are not dependent on specific samples. This approach results in the required data being I/Q samples of length 1250 received from anywhere in the signal's transmitted packet.

In the data strategies focused on WSN data, the Pluto SDR is utilized to transmit a customized ZigBee packet. These ZigBee signals are based on the Matlab simulations in Chapter 4. The associated Matlab code was translated into Python3, where equivalent Matlab functions from the "SciPy" library were implemented. The custom ZigBee samples were transmitted using the Analog Pluto SDR and compared to the commercial ZigBee transmitter, the DIGI XBee. The results are visualized in Fig. 5.9, where the simulation based customized approach correlates well with the commercial ZigBee transmission and, so, the proposed customized ZigBee setup is validated. These SDR

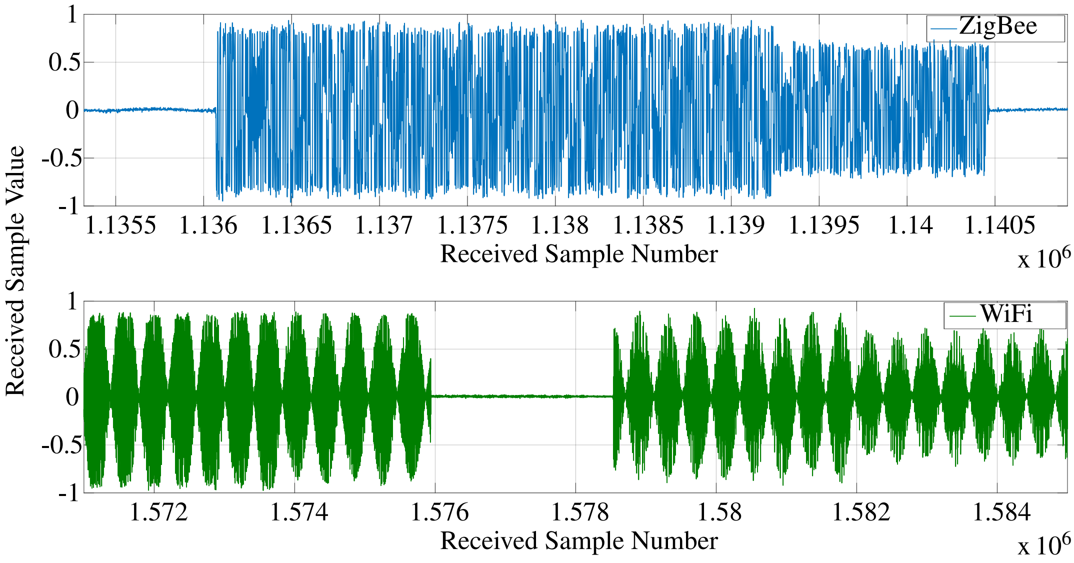


Figure 5.8: Example time series representations of received wireless signals on the I-channel, which establishes the need for correct sample identification.

customized ZigBee signals transmit a ZigBee PHY payload (based on random numbers from the standard uniform distribution or SenseHat environmental data), along with the required preambles, and are spread according to the ZigBee 32-bit PN sequences. Offset quadrature-phase shift keying (O-QPSK) and raised cosine pulse shaping are applied to the chips to match the ZigBee signal protocol. In some experiments, live environmentally sensed data from the Raspberry Pi SenseHat sensor is transmitted in the customized approach, which results in a more considerable variation in the transmitted packets.

5.4.1 Wired Signal Classification

As part of the initial live signal investigation, an I/Q data logger was designed, which incorporated a Raspberry Pi 3 B, an Analog Pluto SDR and the Siretta 2.4 GHz antenna. However, this data logger's initial stage accessed the wireless channel, but the signals of interest were transmitted over wires, as specified in Fig. 5.10. In these experiments, signals were transmitted over wires to avoid jamming any networks operating in the ISM RF band and transmission powers were sufficiently reduced to accommodate this wired approach, as the Pluto SDR can provide attenuation levels up to $89.75dB$. The wired approach also enabled a high signal to noise ratio with near-perfect samples being received at the receiver. It was designed to be the bridging investigation between the simulations and full wireless data signals.

For the data collection experiments, specific time intervals and data lengths were applied to collect data over one hour at each ZigBee center frequency. The data length of each data collection was 0.5 s and ensured at least one signal transmission would be received. As the wireless channel was simultaneously being received, unavoidable

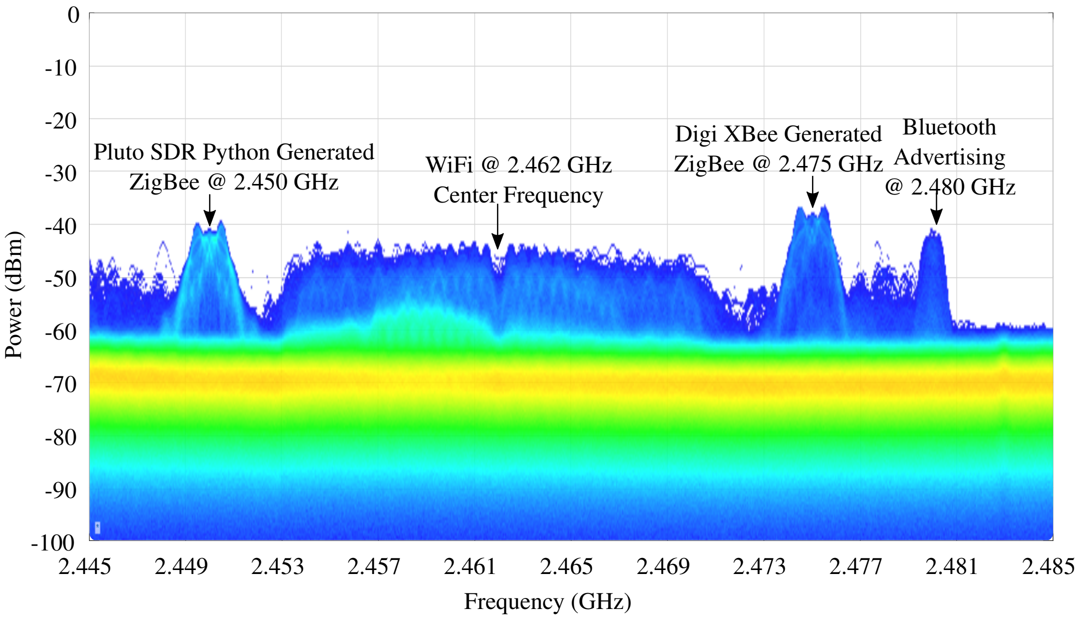


Figure 5.9: A Tektronix DPX image of the customized SDR and commercially transmitted ZigBee signals in a typical domestic operating environment. Both ZigBee signals exhibit similar spectral images.

spurious interference was received. Thus, each data grab is post-processed to obtain the correct signal samples. Both continuous wave (CW), which is simply a co/sine wave on the I and Q channels, and basic ZigBee signals, which were based on the ZigBee protocol specifications provided in Chapter 2 and the simulations in Chapter 4, were generated. Separate Raspberry Pi devices are required to control the SDR receiver and transmitter as it is difficult to provide real-time reception and transmission on a single platform, using either one or two serial connections, especially on embedded devices. The Pluto generated ZigBee signal is compared with a ZigBee signal from a commercial node, the DIGI XBee using the RTSA and the associated DPX graph is provided in Fig. 5.9. The Pluto SDR signals correlate well with the commercial XBee ZigBee transmissions and, so, are considered acceptable for use. These SDR customized ZigBee signals are used in preceding wired experiments and in the wireless experiments. For the noise channel to be received, a 50Ω termination must be applied to the DC block to ensure matched conditions. This data strategy ensured the collection of I/Q samples for the wireless noise channel (baseline), spurious wireless interference and wired transmissions of CW and SDR ZigBee signals. A summary of the collected data is provided in Table 5.7, where the overall data file is 0.5s long and contains four 125ms SDR received data buffers. The collected data is utilized in Chapter 6 to initially examine applying the extracted features from the simulated data in Chapter 4 to hardware transmitted signals and a typical lab operating wireless environment.

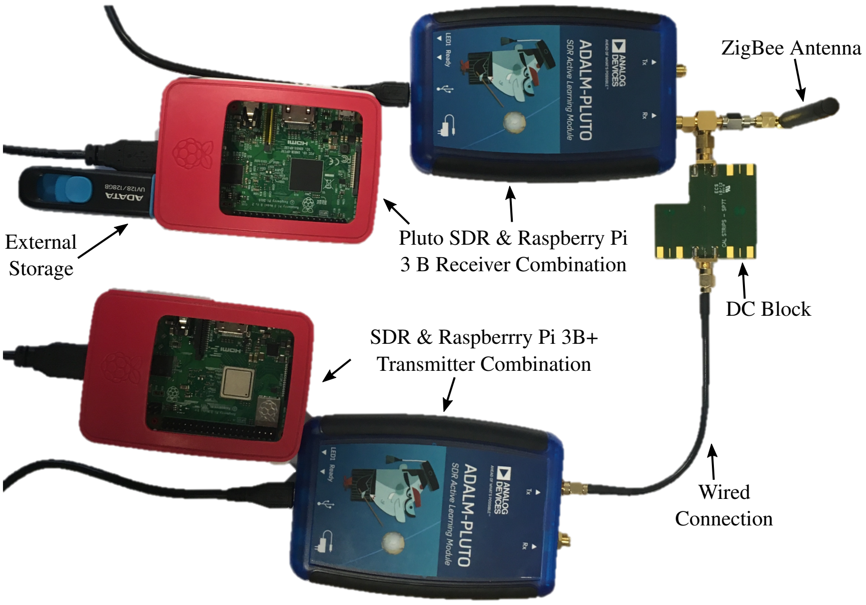

ZigBee Antenna
External Storage
Pluto SDR & Raspberry Pi 3 B Receiver Combination
DC Block
SDR & Raspberrry Pi 3B+ Transmitter Combination
Wired Connection
ADALM-PLUTO
ADATA

Figure 5.10: Initial experimental setup for signal addition, which minimizes impact on surrounding services using the 2.4 → 2.5 GHz RF band. This also allowed for the analyzed signals to be received with a high signal to noise ratio with near perfect samples.

Table 5.7: Wired Data Collection: 16 ZigBee Channel center frequencies

Channel Type	Data: Length / No. Frames	Time: Interval / Total	No. of Datasets	Tx. Gain
Noise	125 ms / 4	5 mins / 60 mins	11	N/A
ZigBee	125 ms / 4	5 mins / 60 mins	11	-20 dB
CW	125 ms / 4	5 mins / 60 mins	11	-20 dB

5.4.2 Wireless Signal Classification

As shown in Fig. 5.7, the legitimate wireless signal strategy was built around a Pluto SDR and Raspberry Pi design to maximize the data collection process using low-cost hardware. Data collection occurred in a wireless operating environment consisting of changeable service requirements for WiFi and Bluetooth, including the number of connected devices and service load (large download or constant music streaming, for example). The Pluto SDR, as discussed in Section 5.2.2, encompasses suitable parameters for operating as a data receiver and controlled data transmitter in this typical ISM RF band environment. Python3 is utilized to develop various signal types for transmission and the “pyadi-iio” library programs the necessary Pluto parameters, such as center frequency, gain, sample rate etc. The developed wireless data collection system is depicted in Fig. 5.11. All experiments in the legitimate signal classification investigation utilize the Raspberry Pi/Pluto SDR approach to access received I/Q samples at various center frequencies.

Wireless signals are received from both commercial and SDR sources. The commercial signals are typical ISM RF band transmissions and include WiFi both with

(IEEE 802.11ac & IEEE 802.11n) and without internet access (IEEE 802.11b & IEEE 802.11g), Bluetooth, where the advertising channels are targeted, and DIGI XBee ZigBee nodes. SDR sources produce Python3 generated CW and ZigBee signals based on Matlab simulations in Chapter 4, where the Matlab code has been translated to Python3. The commercial XBee nodes operate as per the WSN testbed in Section 5.3, where a point-to-point communication approach is operated. I/Q samples are received, using a 4 MHz sampling rate, in separate 0.5s data grabs, from which the required I/Q data is identified and extracted. This data strategy produces the necessary I/Q data for off-line analysis and feature investigation in Chapter 6. A summary of the collected data for training and testing ("unseen") is provided in Tables 5.8 and 5.9, respectively, which specifies the center frequency(ies), signal type, approximate number of data grabs and source. Approximately 400 test instances were collected for each of the six signal types and designated as an "unseen" test data set, to investigate how the designed classification approach generalizes to new data. Emphasis was given to the XBee ZigBee signals as these are WSN operating signals. However, the number of received signals in each data grab is protocol specific, as data grabs of certain protocols, for example, WiFi, can contain multiple signals for analysis. The collected data is fully explored in Chapter 6 to extract features and develop the first aspect of the edge device diagnostic tool, the ISM RF band signal classifier. Additionally, this data is used to develop the interference detection and classification tool in Chapter 7, by providing the legitimate ZigBee I/Q data.

Table 5.8: Summary of Collected ISM Signal Data

Signal Type	Center Frequency (MHz)	Total Data Grabs	Source
WiFi	2427	536	Commercial
Router	2447, 2462	270	Commercial
Bluetooth Advertising	2402, 2480	564	Commercial
ZigBee (XBee)	16 ZigBee Frequencies	944	Commercial
ZigBee (SDR)	16 ZigBee Frequencies	1120	SDR
CW	16 ZigBee Frequencies	1120	SDR
Noise	16 ZigBee Frequencies	492	Channel

Table 5.9: Summary of Collected ISM Signal Data - Unseen Test Data

Signal Type	Center Frequency (MHz)	Total Data Grabs	Source
WiFi	2427	144	Commercial
Router	2442	107	Commercial
Bluetooth Advertising	2402, 2480	206	Commercial
ZigBee (SDR)	Subset-ZigBee Frequencies	240	Commercial
CW	Subset-ZigBee Frequencies	240	SDR
Noise	Subset-ZigBee Frequencies	210	Channel

5.4.3 WSN Interference Detection & Classification

This thesis' primary focus is on WSN edge device security by implementing an interference detection and classification tool. The WSN data collection process adopted the developed ZigBee testbed containing live sensor data, as described in Section 5.3. This testbed consists of DIGI XBee ZigBee nodes connected to Raspberry Pi devices that incorporated the SenseHat environmental sensor. The operating environment is a typical domestic environment consisting of changeable service requirements for WiFi and Bluetooth, including the number of connected devices and service load (large download or constant music streaming, for example). To receive I/Q samples from, and to implement penetration tests on this WSN testbed, the Pluto SDR was utilized.

The commercial XBee ZigBee nodes transmit the required WSN commercial packets while the SDR receives the associated WSN data on the required channel, acquiring I/Q samples in the process. This approach provides data for commercial ZigBee devices and data were collected for five different XBee transmitters, where each transmission includes SenseHat produced data. The Pluto SDR is additionally utilized to provide constant ZigBee based signals while in the presence of jamming. Enough power in the jamming signal disrupts legitimate ZigBee network operation. As a result, an efficient method for the continuous transmission of signals that neglected the presence of jamming was required to gain access to the required data of O-QPSK (ZigBee) transmissions interacting with other signals. This method utilized SDRs to transmit the required ZigBee signal structure, which was shown to produce similar spectral images in Fig. 5.9. The transmitting and receiving process of this approach is shown in Fig. 5.12, where each Pluto is controlled and powered by a Raspberry Pi and an additional SDR or XBee device can be used as the legitimate transmitter and the SDR with the

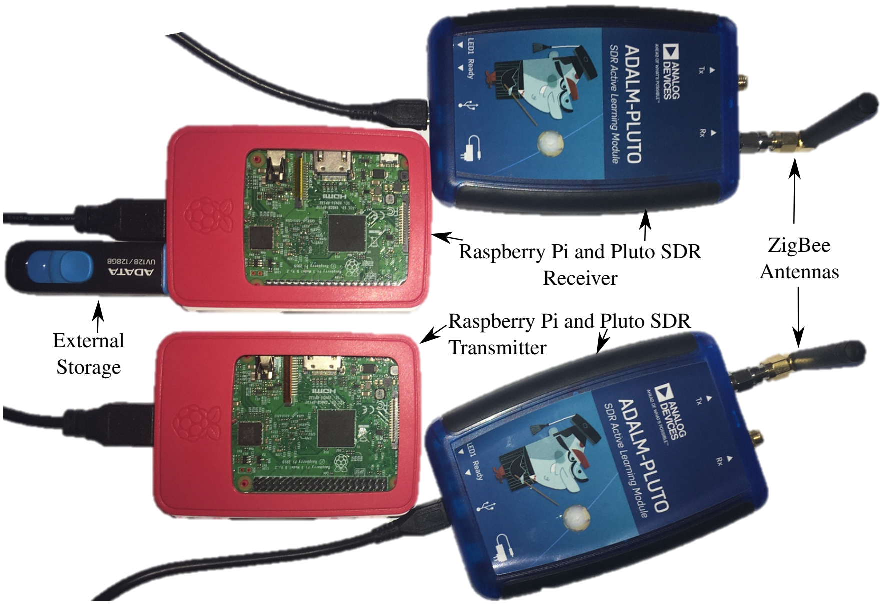

Raspberry Pi and Pluto SDR Receiver
ZigBee Antennas
External Storage
Raspberry Pi and Pluto SDR Transmitter
ADALM-PLUTO
SDR Active Learning Module
ANALOG DEVICES
ADATA

Figure 5.11: SDR and Raspberry Pi I/Q wireless data collector and signal transmitter for wireless transmissions in the 2.4-2.5 GHz ISM RF band.

power amplifier operates as the jammer.

This strategy permitted the collection of both SDR and XBee transmitted I/Q data in a typical domestic operating environment. The XBee data included legitimate data with no jamming signal and in the presence of subtle CW interference. CW jamming was chosen as the simulations predicated this jamming approach to be less effective than matched interference, resulting in a higher chance of collecting the necessary data using the SDR. The customized SDR transmitted ZigBee signals produced I/Q data with no jamming signal and in the presence of CW and matched signal interference. The SDR jamming signal power gains varied from -55 dB to -34 dB on the Pluto SDR, where the CN0417 power amplifier provided an additional 20 dB gain to the Pluto output, approximately. The CN0417 2.4 GHz, RF 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 so high as to block all transmissions. Matlab controls and operates the Pluto SDR when used as a jamming device as the associated computer's additional resources allow for continuous jamming signal transmission. 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. This utilization of SDRs and XBee devices provided the necessary I/Q data access under normal and interference conditions to develop edge device interference diagnostic tools, as specified in Chapter 7.

In summary, the above data strategies, focused on WSN and ISM RF band data, produce enough data instances for the following investigations, examined in Chapters

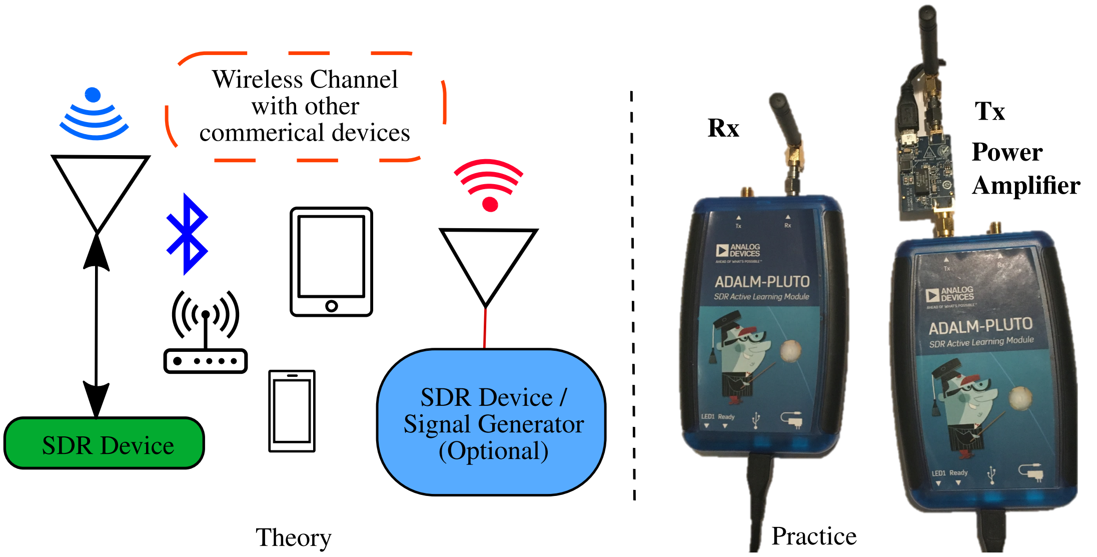


Figure 5.12: Pluto SDR approach, in Theory and Practice, being controlled and configured by a Raspberry Pi 3 embedded device for WSN analysis and SDR signal transmission. The interference signals are produced by a Pluto device connected to a PC and controlled using Matlab.

6 and 7, to develop the overall edge device diagnostic approach.

- Legitimate ISM RF band signals including XBee commercial ZigBee, SDR customized ZigBee, WiFi, Bluetooth, Noise and CW.
- Legitimate XBee node vs. non-legitimate SDR classification, where both signals have the same spectral image (Fig. 5.9).
- Artificial jamming of legitimate XBee node data using the legitimate ISM RF band signal data. This produces interference data for WiFi, CW and Matched signal interference. This concept is explained in detail in Chapter 7.
- SDR transmitted ZigBee live wireless jamming data under CW and Matched signal interference.
- Live subtle CW jamming of commercial XBee node transmitted ZigBee signals.

5.4.4 GPS Interference Detection & Classification

GPS signal data was acquired to investigate whether the developed diagnostic approach for ISM RF band legitimate signal classification and WSN interference detection and classification 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 thesis uses GPS signals, on account of previous work in GPS interference detection [41], 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 [41] and to analyze as many satellites as possible, GPS data was collected across a full 24 hour period to initially validate the antenna position and to ensure all available satellites were visible. This method was validated using the collected I/Q data and “fastgps” [147] as the GPS software receiver to identify the received satellites. The results are saved on the Raspberry Pi with associated timestamps and can be compared with GNSSRadar [155] to determine the accuracy of the identified satellites. This 24-hour methodology means that associated results are not dependent on a specific subset of satellites and validated the receiver’s antenna position.

As GPS signals are received at such low power levels (typically -125dBm), relatively low powered jammers, which broadcast noise on GPS frequencies, will readily block GPS signal reception. This fact implies that a jammer can have a large effective range, even though it might be a relatively low-powered signal or located at a relatively large distance from the receiver. Here, the primary source of interference was CW interference, while 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, and in this thesis, the ZigBee modulation scheme is leveraged as a GPS jamming signal. Both jamming signals were emitted into the GPS reception method using a wired approach to avoid jamming nearby GPS receivers. This procedure allows for collecting both good (four satellites or more) and interfered data for off-line analysis. The Pluto SDR received GPS signal I/Q data is in the RTL-SDR range of [-128:+127] and, in contrast to the ISM RF band data, it is not scaled to the range of [-1:+1]. This approach is undertaken to examine the proposed methodology and the developed features (Chapter 6) using data in a different numerical range from a different receiver.

The overall approach is specified in Fig. 5.13 and 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 supplies a maximum attenuation of $89.75dB$. 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. For successful GPS reception, four satellites need to be received, which means this is the metric used to identify when a jamming signal is effective. Each received data collection was analyzed for at least forty different time segments, where both the GPS signals and jamming signals were time-varying. Each data segment contained 10230 I and Q samples. This GPS sample size was due to significantly larger packets being received from the GPS signals than from the ISM RF band received packets. 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 associated developed

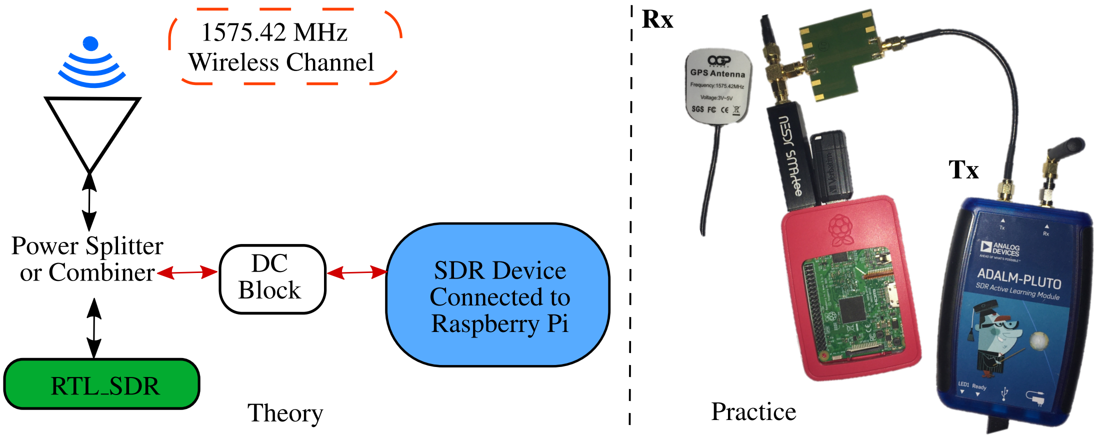


Figure 5.13: RTL-SDR dongle approach, in Theory and Practice, being controlled and configured by a Raspberry Pi 3 embedded device for GPS signal reception and interference addition.

models (see Chapter 7) were not dependent on any specific part of the received data outputted from the RTL-SDR dongle.

5.5 Conclusion

This chapter discussed the hardware and associated data strategies applied to obtain the required I/Q data from a typical domestic operating environment. Each hardware experimentation approach utilizes both Raspberry Pi embedded devices and SDRs to produce low-cost, high-performance testbeds and data collection approaches. The required I/Q samples for legitimate ISM RF band signals, including ZigBee, Bluetooth and WiFi, CW and noise, are collected. The interference detection approach requires data encompassing legitimate and jamming signal interactions. This data is acquired by utilizing XBee devices in the presence of subtle jamming from a Matlab controlled SDR and by exclusively using SDRs to provide data for higher-powered jamming approaches. For GPS signals, SDRs were utilized to transmit jamming signals over wires through a signal combiner connected to a GPS active L1 antenna. In each case, live environmentally sensed data is transmitted by using the Raspberry Pi SenseHat add-on board. The hardware described in this chapter and the associated applications produced the required data to develop legitimate signal classification, WSN and GPS interference detection and classification and legitimate node classification models. The associated feature extraction, model development and all associated results are described in Chapters 6 and 7.

Chapter 6

Legitimate Signal Classification - Feature Extraction and Optimal Machine Learning Model Development

This chapter addresses the first hardware and wireless over-the-air experimentation. The work focuses on legitimate signal classification for wireless signals in the ISM RF band. The work is important for the experimental over-the-air interference detection and classification work presented in Chapter 7. The developed optimal features and machine learning models in this chapter are leveraged in Chapter 7. The work in this chapter has been published in part in the following:

- *G. D. O'Mahony, K. G. McCarthy, P. J. Harris and C. C. Murphy, "Developing a Low-Order Statistical Feature Set Based on Received Samples for Signal Classification in Wireless Sensor Networks and Edge Devices", IoT, vol. 2, no. 3, pp. 449-475, 2021, doi: 10.3390/iot2030023.*
- *G. D. O'Mahony, P. J. Harris and C. C. Murphy, "Investigating Supervised Machine Learning Techniques for Channel Identification in Wireless Sensor Networks," 31st Irish Signals and Systems Conference (ISSC), Letterkenny, Ireland, 2020, pp. 1-6, doi: 10.1109/ISSC49989.2020.9180209.*

6.1 Introduction

This chapter utilizes raw in-phase (I) and quadrature-phase (Q) samples that were collected over time in distinct sessions using software-defined radios (SDRs) in a typical domestic wireless environment that contained different signal sources, devices, obstacles and service usage. Received I/Q samples are visualized and analyzed across time, frequency and space (probability density function (PDF)). Simulations in Chapter 4 motivated exclusively using I/Q samples to detect interference in ZigBee signals. Those simulations additionally supported interference classification, which manifests as received signal classification. This chapter expands on that concept by focusing on legitimate signal classification. Before interference can be detected, the legitimate wireless signals being transmitted in the environment need to be identified and when no packets are received, the dominant signal needs to be identified. This approach focuses on industrial, scientific and medical (ISM) radio frequency (RF) band signals and determines the wireless channel (signals in transit) when no packets are received. This results in the first stage of developing an interference diagnostic tool for wireless edge devices, as when packets cannot be received, signal model data is required. This requirement contrasts with the signal interaction data required for interference detection when packets are received with errors. Both approaches focus on I/Q samples and enable edge device decision-making. This ability is a consequence of no network-level data being used, requiring no prior knowledge of signals including relationships between samples and symbols and making no channel assumptions, except that signals can have different bandwidths.

This chapter's work leverages the results from Chapter 4 as the foundation to develop a feature set based on wireless data received using real hardware and can differentiate signals using similar modulation schemes and when the receiver becomes saturated. As discussed in Chapter 5, the analyzed signals are transmitted from both commercial and SDR sources. The signals being analyzed in this chapter are visualized using a Tektronix real-time spectrum analyzer (RTSA) and associated digital phosphor technology (DPX) software in Fig. 6.1, where channel noise, ZigBee, continuous wave (CW), WiFi and Bluetooth signal data are the chosen data for analysis. Classification decisions are based entirely on low-order features extracted from the raw I/Q samples. The contribution and novelty here surround the use of received raw I/Q samples and no channel assumptions to develop a feature set for ISM band wireless signal classification using a constant sampling rate based on twice the wireless sensor network (WSN) protocol baseband signal bandwidth, ZigBee [37]. This chapter's work is the thesis's first analysis of over-the-air wireless data and reinforces the results from the simulations implemented in Chapter 4. The use of I/Q samples will be shown to enable the classification of different wireless channels/signals and even distinguish between two different IEEE802.11 versions/transmitters. Utilizing the raw I/Q samples is validated by focusing on fundamental machine learning algorithms that are verified here as fit for purpose. The developed features are verified by analyzing the accuracy achieved and the ability to generalize to unseen data. The algorithms used are the previously introduced Support Vector Machine (SVM) [22] and Random Forest [23] models, while k nearest neighbors (k-NNs), XGBoost, Naive Bayes, artificial neural networks (ANNs) and deep neural

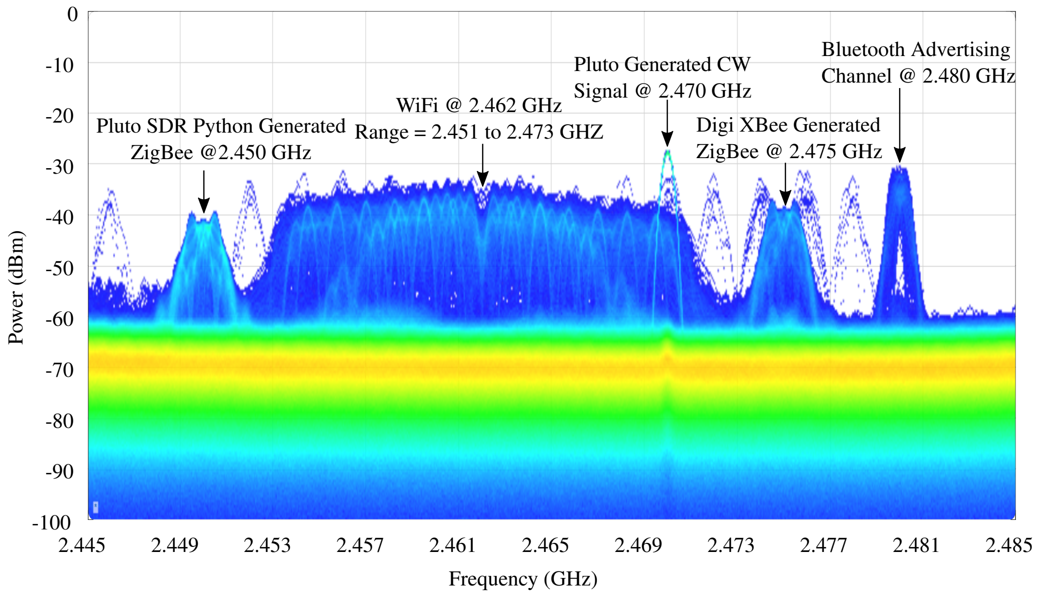


Figure 6.1: DPX visualization of noise, ZigBee (commercial node), ZigBee (Pluto SDR), CW (Jammer) and coexistence with Bluetooth and WiFi signals in a typical domestic operating environment. Each of these signal types are used in this chapter to develop an optimal feature set and models for legitimate signal classification.

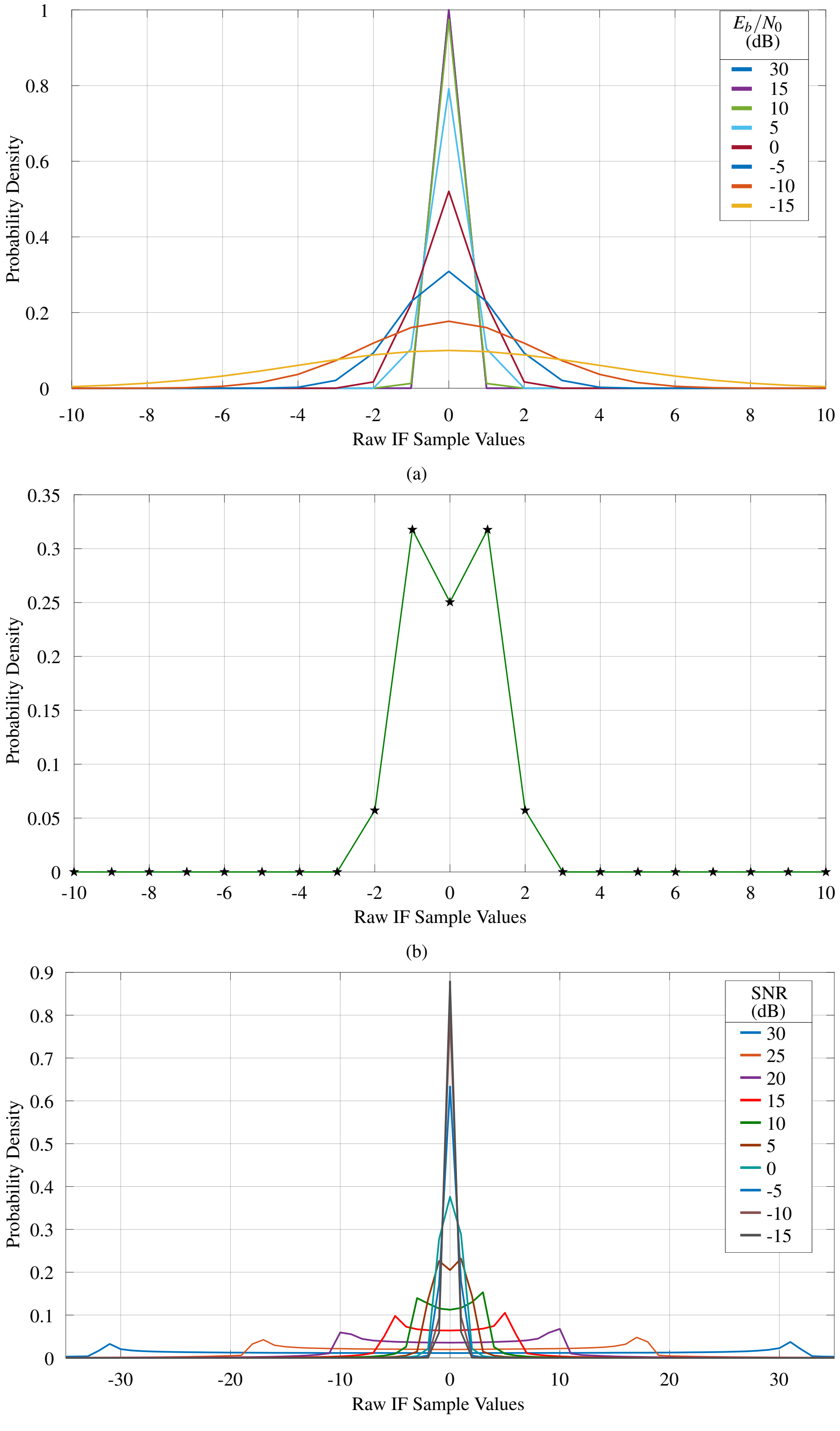
networks (DNNs) are also introduced. The results explore the hypothesis of making high-level edge device/network decisions based entirely on the lowest available data, raw I/Q samples, low-order statistics and machine learning.

The desired deployment is on embedded edge devices (i.e., typical WSN and IoT devices), which can use the designed signal classification model to adapt procedures in accordance with identified channel information autonomously. This approach will enable enhancements in edge device operation and efficiency, as wireless transmission performance is heavily linked to the wireless channel's quality. So, edge devices reacting autonomously to channel variations can accelerate the optimal response. This hypothesis can eradicate obsolete central controller responses due to transmission latency. It is enhanced if the data used is always available to a functioning receiver, hence raw received I/Q samples are employed. This chapter concentrates on open source platforms, with low-cost yet high performance, to enable interoperability. Machine learning is appropriate for this application as the computationally intensive model training and optimization stage can be done off-line. Thus, only the optimized model must be uploaded to the edge device. Here, models are fine-tuned using available test data, including unseen data, and implemented on a Raspberry Pi embedded device as an initial resource-constrained implementation study.

6.2 Initial Experimental Method: Simulations and Wired Transmissions

To gain an initial understanding of what was expected in the live wireless signals data collection and processing, Matlab was used to simulate signals in the presence of noise and specific test signals were transmitted over wires using SDRs. Wired test signal data is examined for the signal-free (noise), legitimate signal (ZigBee) and primary jamming signal (CW) cases in a live laboratory environment. Additionally, ZigBee, CW and the operating environment (noise) are briefly examined in a simulated approach. The simulation results are presented to predict what the signal structures should ideally adhere to, while signals were transmitted over wires to avoid jamming any networks operating in the ISM RF band. For wired transmissions, the applied SDR gain was sufficiently reduced to accommodate the use of wires. As discussed in Chapter 5, the chosen Pluto SDR can provide attenuation levels up to $89.75dB$, resulting in satisfactory operation for this approach. It was designed to bridge the investigation between the simulations from Chapter 4 and the full wireless data analysis in this chapter. Furthermore, the wired results were used to evaluate the features and classification approaches developed by analyzing the simulated transmissions in Chapter 4. Wired signals enable a high signal to noise ratio with near-perfect samples being received at the receiver.

The simulation results are presented in the form of the signal's PDF, and the samples were received in terms of the legitimate signal (ZigBee) center frequency. The simulations contained additional Gaussian noise and some random phase offset (compared to the ZigBee signal), as discussed in Chapter 4. The PDFs for each of the signals of interest are provided in Fig. 6.2. The simulated noise environment is provided in Fig. 6.2 (a), which is as expected since the noise samples will typically be received within a narrow range close to zero in typical operating conditions. The noise PDF is displayed in terms of a signal's energy-per-bit to noise ratio (E_b/N_0). As a result, lower E_b/N_0 values relate to a more hostile noise environment and the PDFs visualize the noise environment that a transmitted signal interacts with. Typical operating environments appear as a peak in the zero bin in the calculated PDF. This phenomenon is the expected occurrence once the signal-to-noise ratio (SNR) becomes small enough for the other signals. For the legitimate signal, ZigBee, shown in Fig. 6.2 (b), the center bin is lower than its two neighboring bins, which is also visualized by the Tektronix RTSA in Fig. 6.1. The jamming signal is a CW, which is a (co)sine wave with varying amplitude and possesses a PDF resembling a small peak at the received amplitude value ($+$ & $-$), as shown in Fig. 6.2 (c). As the CW signal decreases in power, it approaches being embedded in the noise and, as a result, the signal resembles the noise PDF at low power levels. These results provide guidance when determining if the received wired signals are appropriate when transmitted and received using SDRs, resulting in the accelerated



(a)

(b)

(c)

Figure 6.2: Calculated PDFs for Matlab simulated I/Q samples for example signals operating in the 2.4 GHz ISM band and analyzed in the wired investigation. (a) Noise signals of varying power levels (E_b/N_0), where the power is the simulated noise floor, (b) The expected PDF of a received legitimate ZigBee signal, (c) CW signals of varying power levels, where the power, SNR, is related to the simulated noise floor.

analysis of wired signals.

Using the simulation data as the baseline, the wired transmitted signals' PDFs were produced and can be visualized in Fig. 6.3. These results correspond to a subset of possible transmissions. In these experiments, the ZigBee signal saturates the receiver, while the CW does not. In practice, the receiver can become saturated for any signal with a high enough transmitting power. However, this subset of signals is sufficient for a concise investigation of the simulated extracted features applied to hardware transmitted signals. From these wired data PDFs, and the associated I/Q samples, identifying features were extracted by leveraging the simulation work in Chapter 4. The wired approach is the control environment in this investigation. It allows the initial machine learning models to be developed and further investigates whether a wireless approach has merit.

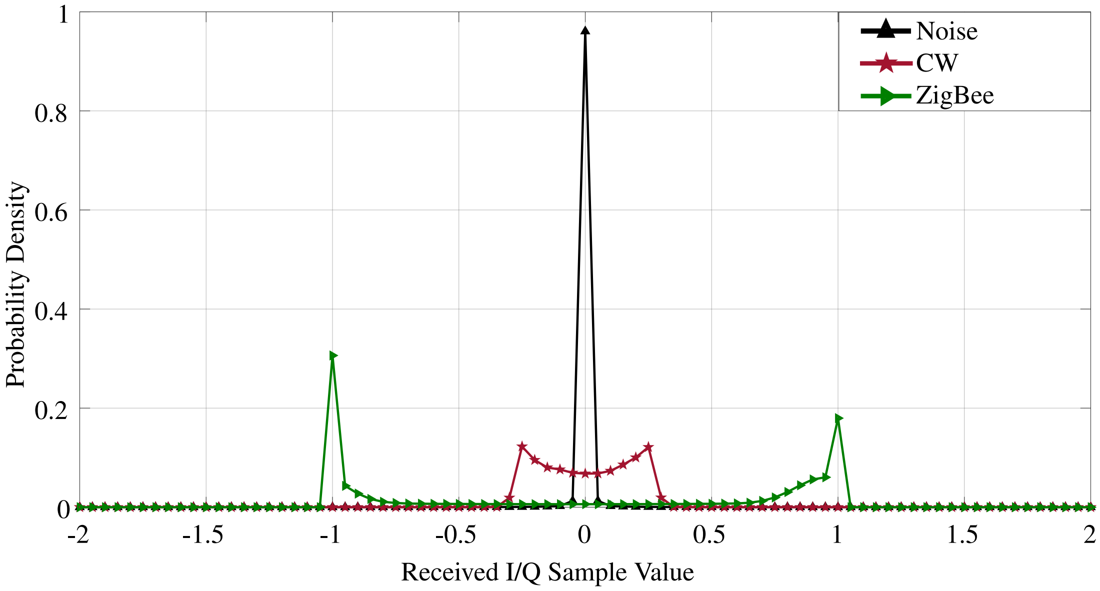


Figure 6.3: Averaged calculated PDFs for received I/Q samples for Noise, ZigBee and CW signals using the wired SDR and Raspberry Pi approach described in Chapter 5.

From the PDF, several features were extracted, including 1) The area in center bins, 2) the Averaged area of the two side bin ranges, 3) The zero bin value, 4) The number of non-zero entries and 5) The maximum value in the PDF. The feature set was expanded through statistical analysis of the received I/Q samples and resulted in the additional features of; 1) The sample variance, 2) the mean value of the received samples, 3) the absolute maximum value and 4) the sample entropy. These nine features extract all but one (sample standard deviation) from the simulation study and introduce the additional feature of the PDF center bin's value. This analysis formed the basis for the initial legitimate signal classification study, where legitimate corresponds to a specific transmitted signal. During the wired investigation, uncontrollable spurious interference signals were discovered in the collected noise data at random intervals and power levels. These spurious interference data instances needed to be classified along with noise, ZigBee and CW signals. For this discussion, the spurious interference is referred to as “non-ideal noise”. These extracted features, based on the collected noise data and wired signals, are summarized in Table 6.1, where the signal differences and similarities are indicated. These extracted features enable designing machine learning classifiers, namely, a SVM and a Random Forest ensemble approach, based on the success of applying these two methods in the simulations in Chapter 4.

Table 6.1: Extracted Features: Noise, Non-ideal Noise, CW & ZigBee

Signal	**Value Type**	Area Centre	Area Side	Centre Bin	Non-zero Entries	PDF Max.	Variance	Mean	Abs. Max.	Entropy
Noise	Average	1.0	0.0	0.9797	2.91	0.9797	4.83e-5	1.42e-5	0.0236	1.4854
	Maximum	1.0	0.0	1.0	9.0	1.0	0.0015	0.0018	0.1099	3.0414
	Minimum	1.0	0.0	0.2485	1.0	0.2871	6.02e-6	-0.0022	0.0085	0.6020
Non Ideal Noise	Average	0.9398	0.0	0.6947	25.74	0.6948	0.0079	-9.164e-5	0.4479	2.4605
	Maximum	1.0	0.0	0.9913	41	0.9913	0.0571	0.0045	0.9963	3.9915
	Minimum	0.6158	0.0	0.1163	7	0.1163	1.248e-4	-0.0031	0.1102	0.9843
CW	Average	0.8377	0.0334	0.0677	13.28	0.1496	0.0150	0.0013	0.2036	3.7298
	Maximum	1.0	0.0334	0.7349	31	0.7349	0.0241	0.04	0.4704	4.4902
	Minimum	0.6137	0.0334	0.0506	2	0.0845	7.335e-5	-0.0380	0.0342	2.9651
ZigBee	Average	0.0612	0.4597	0.0061	41	0.3065	0.3752	-0.0319	0.9999	3.7391
	Maximum	0.0967	0.4597	0.0111	41	0.3625	0.4310	-0.0040	1.0	4.6746
	Minimum	0.0531	0.4597	0.0036	41	0.1663	0.3107	-0.0592	0.9741	3.4378

The feature analysis of the received I/Q samples produced data which needed to be split between training, validation and testing, in the ratio of 70% : 20% : 10%, respectively. This data split procedure’s sampling bias was inspected by calculating the percentage of each classification type in the datasets using the scale of 0.0 → 1.0. Table 6.2 provides the results, where the percentage split is maintained across all developed datasets. As a result, the sampling probability is maintained. The “non-ideal noise” percentage is much lower than the other signals as the random interference was only evident in specific data captures, but the percentage in each dataset is maintained. Multiple SVM models were developed to classify the combinations of the four signal types, while one multi-class Random Forest classifier was developed. The validation and testing data were used to determine the Random Forest algorithm’s optimal metrics, which included the number of decision trees and predictor depth, which is the size of the random subset of features used in developing the unique weak learners. The same validation and testing data were used to determine the optimal kernel for the SVM models.

The optimal kernel was determined by using the available validation and test data and a ten-fold cross-validation approach. Each SVM was applied as a binary classifier for the six different combinations of noise, non-ideal noise, CW and ZigBee. As there is sufficient separation between the signals of interest, as shown in Fig. 6.3, the achievable margin is high in all cases, resulting in a low generalization error (approx. zero)

Table 6.2: Wired Data Split - Sampling Bias Comparison

	All Data	Training Data	Validation Data	Testing Data
Noise	0.30945	0.30945	0.30946	0.30940
Non-Ideal Noise	0.02388	0.02388	0.02387	0.02393
ZigBee	0.33333	0.33333	0.33333	0.33333
CW	0.33333	0.33333	0.33333	0.33333

for each kernel. The noise/non-ideal noise case is an outlier ($1 - 1.5\%$) as more spurious interference data is required to reduce the error. As a result, the SVM validation results suggest that each kernel is suitable for this signal classification task. Multiple binary SVM models are required to classify between the different combinations. The summarized SVM validation results are provided in Table 6.3, where non-ideal noise is abbreviated to "non-ideal". The results show the usefulness of the designed SVM approach for binary classification between a base model (noise) and a received signal. Based on the validation results, the chosen kernel is the Gaussian RBF and the associated training data results are specified in Table 6.4. The trained SVM classifiers were analyzed using the available testing data to produce the receiver operating characteristic (ROC) curves illustrated in Fig. 6.4. These results indicate that the method can produce "ideal" binary classifiers and stresses the need to adopt wireless signals since the wired approach is "ideal" by construction.

The wired results were extended using the Random Forest algorithm to develop a multi-class algorithm, which replaces the multiple SVM models with a single Random Forest model. A limited subset of optimal metrics was identified using the available validation and testing data, namely, the number of decision trees and predictor length. The optimal metrics investigation for the Random Forest approach is specified in Fig. 6.5, where the hyper-parameters were chosen to minimize the error. Initially, the single predictor model is the optimal (0.0825%) approach with nine decision trees, as specified

Table 6.3: SVM Optimal Kernel: Validation Error

Kernel:	Linear	Gaussian RBF	Polynomial:Degree 3
Combination:	Validation Error (%)		10 Fold Cross Validation Error (%)
Noise/Non-Ideal	1.4851 | 0.62	1.2376 | 0.37	1.2376 | 0.12
Noise/CW	0.0 | 0.0	0.0 | 0.0	0.0 | 0.0
Noise/ZigBee	0.0 | 0.0	0.0 | 0.0	0.0 | 0.0
Non-Ideal/CW	0.0 | 0.0	0.0 | 0.0	0.4619 | 0.0
Non-Ideal/ZigBee	0.0 | 0.0	0.0 | 0.0	0.0 | 0.0
CW/ZigBee	0.0 | 0.0	0.0 | 0.0	0.0 | 0.0

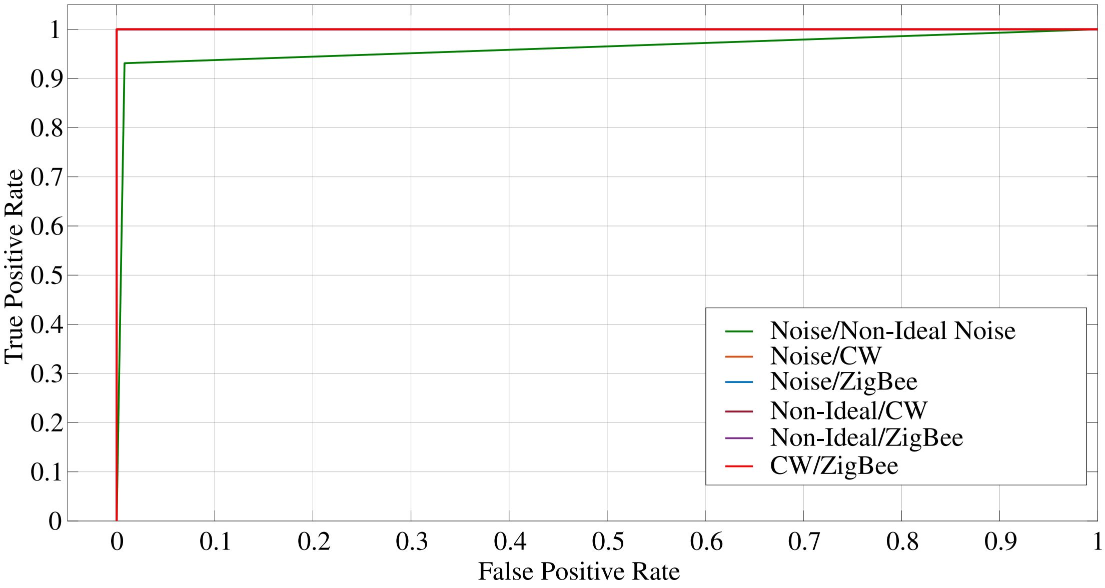


Figure 6.4: The receiver operating characteristic (ROC) curves for the designed SVM classifiers, where all binary classifiers except for the "Noise / Non-Ideal Noise" classifier are "ideal" as the area under the curve (AUC) is 1.0. Thus, the graph only contains two distinct curves. This graphs shows that the classifiers are not random and results are based on using the available testing data.

in Fig. 6.5 (a), but becomes erratic as the number of trees increases due to some features being more reliable and useful than others. The larger the predictor length, the higher the probability that useful features are being used and so the error fluctuates less and is asymptotic to approximately 0.33%. Essentially, some of the features in Table 6.1 are not as suitable as others and the overall process can be optimized. This phenomenon reoccurs when the optimization approach is reiterated in Fig. 6.5 (b), as the overall error reduces to 0.33% but, again, the single predictor is the most erratic. The wireless feature investigation will require identifying and removing obsolete features to gain a more accurate feature representation of the signals. This process is implemented in Section 6.3, where the optimal low-order features are extracted. However, the results outline the usefulness of using the Random Forest approach for signal/channel classification.

Table 6.4: SVM Signal Classification Generalization Error Results: Wired Training Data & Gaussian RBF Kernel

Combination	Training Time (ms)	Average Prediction Time (ms)	Test Data Error (%)	10 Fold Cross Validation Error (%)	AUC
Noise/Non-Ideal	93.1	1.32	1.2376	0.2823	0.9615
Noise/CW	130.47	1.25	0.00	0.00	1.0
Noise/ZigBee	90.04	1.18	0.00	0.00	1.0
Non-Ideal/CW	56.51	1.17	0.00	0.00	1.0
Non-Ideal/ZigBee	34.81	1.18	0.00	0.00	1.0
CW/ZigBee	93.51	1.18	0.00	0.00	1.0

Table 6.5: Random Forest Classification Results: Wired Signals

	Predictor Depth	No. of Trees	Training Time	Avg. Prediction Time	Test Data Error
Validation Data	2	35	188.83 ms	27.02 ms	0.33 %
Training Data	2	35	364.64 ms	27.14 ms	0.0825 %

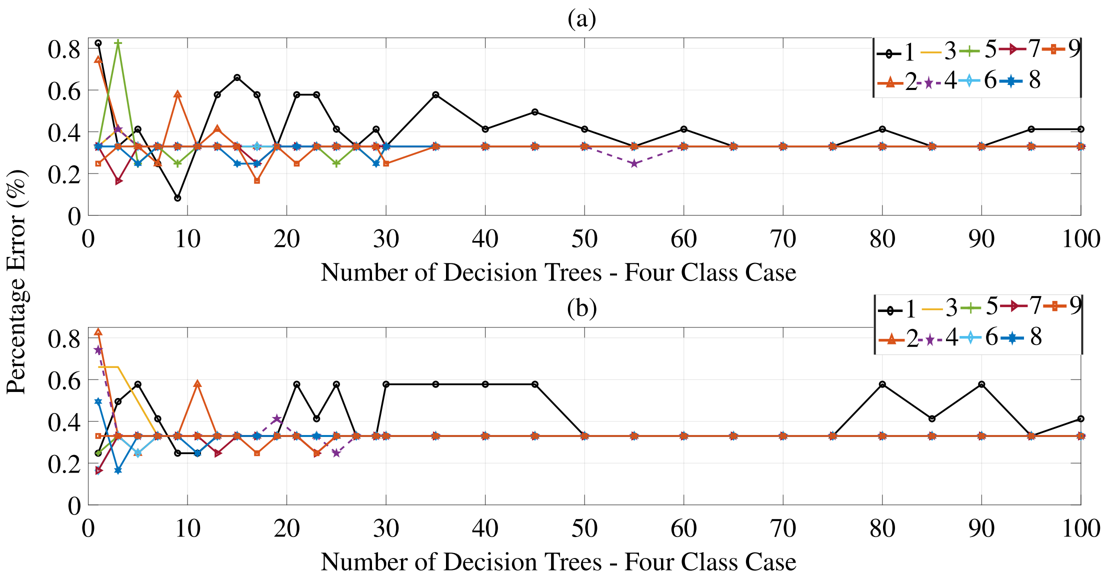


Figure 6.5: Random Forest generalization error investigation using available validation and test data for combinations of predictor length (1-9) and the number of decision trees. (a) Initial investigation, (b) The second investigation to determine if the erratic single decision tree operation was consistent.

A single model can classify multiple classes, which is more beneficial than the SVM approach, even though the performance is slightly less accurate, 0.33% compared to approximately 0%. The final random forest model, which provided the lowest error during the non-erratic stage, used 35 trees and two predictors in the random subset of features. The Random Forest model results are provided in Table 6.5 and specify that the final model obtained a training time of 364.64 ms and an average prediction time of 27.14 ms. The confusion matrix for this Random Forest multi-class classifier is supplied in Fig. 6.6 and specifies that only one instance of the wired testing data was misclassified.

The single model approach is desirable and is a critical factor for the wireless approach. This wired approach has limitations, specifically, the lack of environmentally created variations in the received signal data and the SDR receiver's saturation. These limitations result in received signals which are very different in this investigation. However, in reality, any received signals can saturate the receiver, given enough power. Consequently, the developed models will not generalize to new data instances, as the training data is not a comprehensive overview of the potential receptions of the analyzed signals. Nonetheless, these wired results motivate applying similar features and model development strategies to wirelessly received samples. Additionally, this wired approach outlined the focus for the feature engineering stage when examining live wirelessly received I/Q data. Furthermore, this wired investigation has validated the simulation work in Chapter 4 by applying the extracted features to real transmitted and received signals. Notably, the wired investigation provided the insight that a signal classification tool needs to process and classify signals that saturate the receiver.

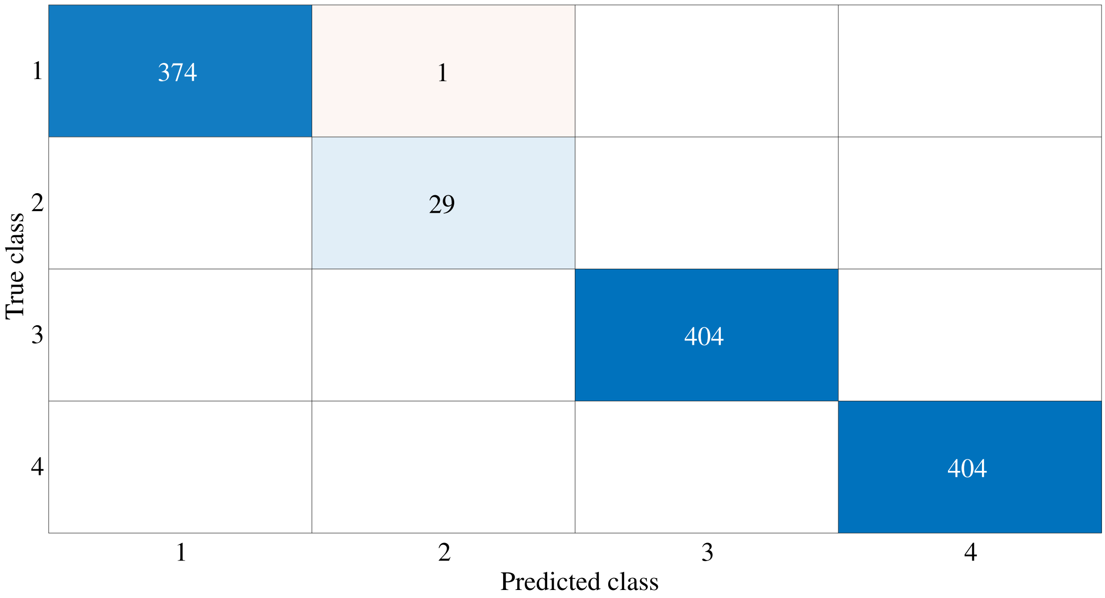


Figure 6.6: The confusion matrix for the optimal designed multi-class Random Forest Classifier, where the labels are as follows: 1- Noise, 2-Non-Ideal Noise, 3-CW and 4-ZigBee. This classifier only produces one False positive for the noise case, when noise is taken as the negative decision.

6.3 Feature Engineering

This chapter's primary focus is based on the hypothesis that I/Q samples are always available to a functioning receiver at the edge. This concept means that independent decentralized decision making is achievable if only I/Q samples are required to classify received transmissions. Legitimate signal classification is the first step towards developing an interference diagnostic tool. It is required to understand and develop a descriptive feature set of typical ISM RF band wireless signals that can enclose wireless edge devices. This approach results in being able to identify WSN signals and coexisting signals in a typical wireless operating environment that consists of changeable service requirements for WiFi and Bluetooth, including the number of connected devices and service load (large download or constant music streaming, for example). This classification approach can be leveraged for interference detection and wireless channel identification, given the development of a sufficiently descriptive feature set. The simulation work in Chapter 4 and the preceding wired investigation provided several critical insights for the wireless signal classification tool development, namely that decisions can be made based entirely on the PDF and time-domain analysis of received I/Q samples. Both approaches motivated investigating a live practical wireless implementation and the feature engineering here leverages the previously extracted features.

In this section, the data strategies, hardware and testbeds developed and described in Chapter 5 were utilized to provide model development data and “unseen” test data. The data corresponds to wirelessly received I/Q samples from commercially transmitted ZigBee signals, SDR transmitted ZigBee signals (based on the simulations in Chapter 4 and wired transmissions in Section 6.2 of this chapter), SDR transmitted CW signals, WiFi signals both with (IEEE802.11ac & IEEE802.11n) and without internet access (IEEE802.11b & IEEE802.11g) and Bluetooth advertising channel data. As per the previous investigations, features are entirely based on received I/Q samples and are initially extracted from the calculated PDF and statistical analysis of the I/Q data in the time domain. This wirelessly received I/Q data analysis and feature extraction was implemented and initially validated in Matlab, while the final implementation targets using available Python3 libraries on a suitable embedded platform. The initial validation targets the previously described machine learning approaches of the SVM and Random Forest models. Model development is expanded to include similar Python3 models and more modern approaches, as per Section 6.4.

As per Chapter 5, before the data could be analyzed, it needed to be pre-processed. The I/Q samples were initially examined in the time domain to visualize the signal patterns (I/Q samples) of interest and compare them with expected sequences. The Pluto SDR analog-to-digital converter (ADC) specifications were used to convert received I/Q samples into the range [-1,1] from the original Pluto range [-2048, 2047]. This conversion supports the development of features having similar ranges, even when the ADC is close to saturation, and may produce higher-performing machine learning classifiers. The PDF, the Tektronix RTSA and the time series plots were utilized to identify any outliers. The chosen sample length for analysis is 1250 I/Q samples (1250 I samples and 1250 Q samples) and relates to the shortest signal length received, the Bluetooth Advertising channel. As a result, the time series analysis granted access to the required received I/Q samples and the calculation of associated PDFs. As defined in Chapter 5, the individual data grabs can contain more than one signal and, so, multiple extractions are permitted, where possible. As the sample length was chosen to be 1250 I/Q samples, received signals were divided into 1250 sample segments. This segmentation process meant that the start, middle and end of the packets were analyzed. As a result, a specific part of a signal packet is not required, and only 1250 samples from the channel are required. Features are calculated using both the I and Q channel data, where the final feature value is the averaged result.

In certain circumstances, it was discovered that the ADC could become saturated and, so, the PDFs can become less distinctive (Fig. 6.7). However, as observed in the wired investigation, this phenomenon needs to be addressed in the feature set development to include high powered signals. Here, wireless devices were non-malicious and, so the ADC saturation was typical of this operating environment and associated receiver positions. Specific features are developed to identify signals, even when saturation occurs. The receiver's automatic gain control (AGC) can also affect the produced PDF as two spikes can occur at the limits since the initial receiver gain changed to a lower value after initial packet reception. However, the calculated PDF is extremely useful in identifying the distinct signals or, at the very least, narrowing the search to a smaller subset of possible signals.

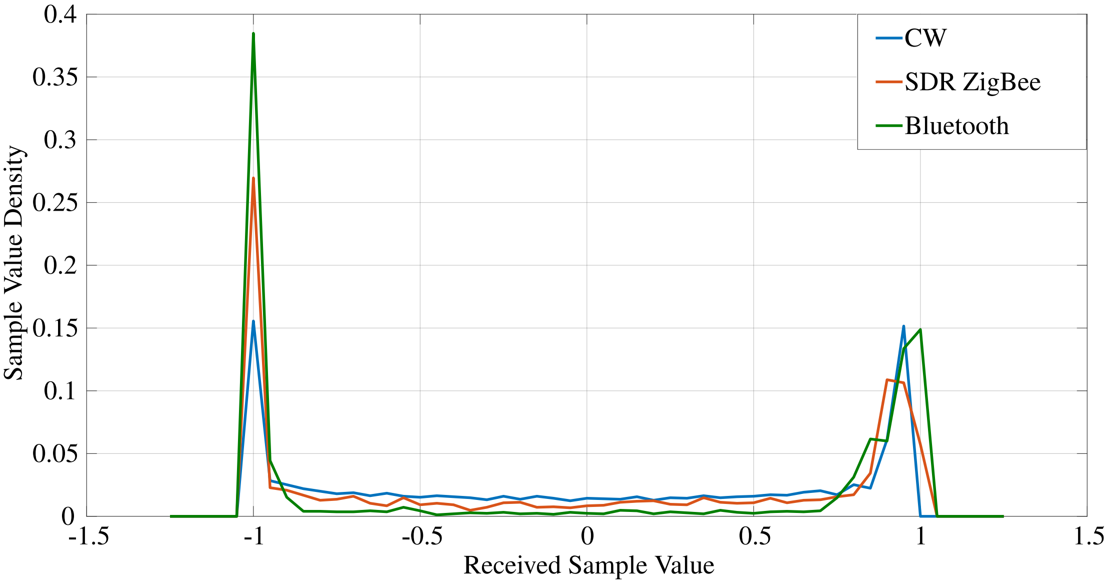


Figure 6.7: Example PDFs for different signals causing receiver saturation. These PDFs emphasize how the PDF becomes difficult to distinguish under saturated conditions and the need for additional features not dependent on the PDF.

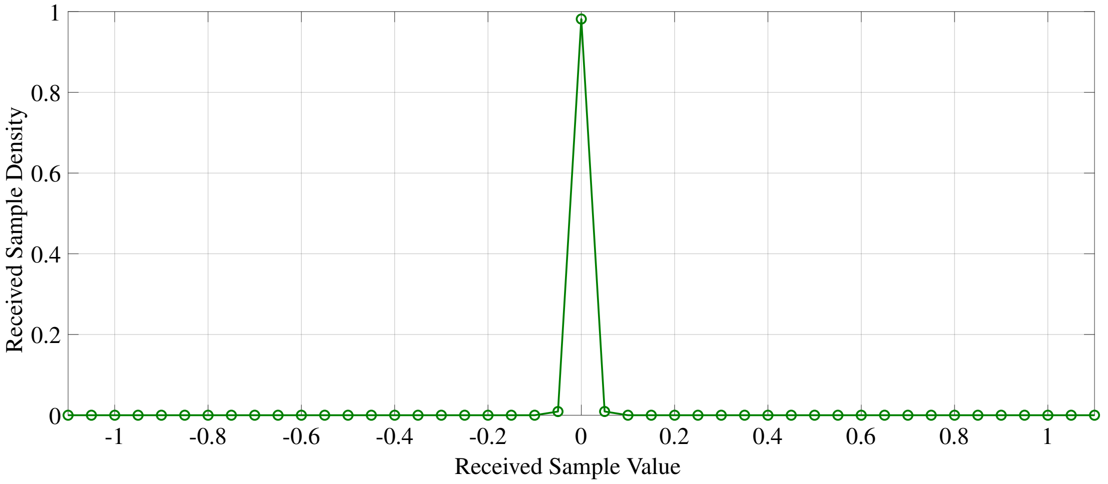


Figure 6.8: The calculated PDF for the inactive noise channel in the operating environment, illustrating the relatively non-hostile operating environment.

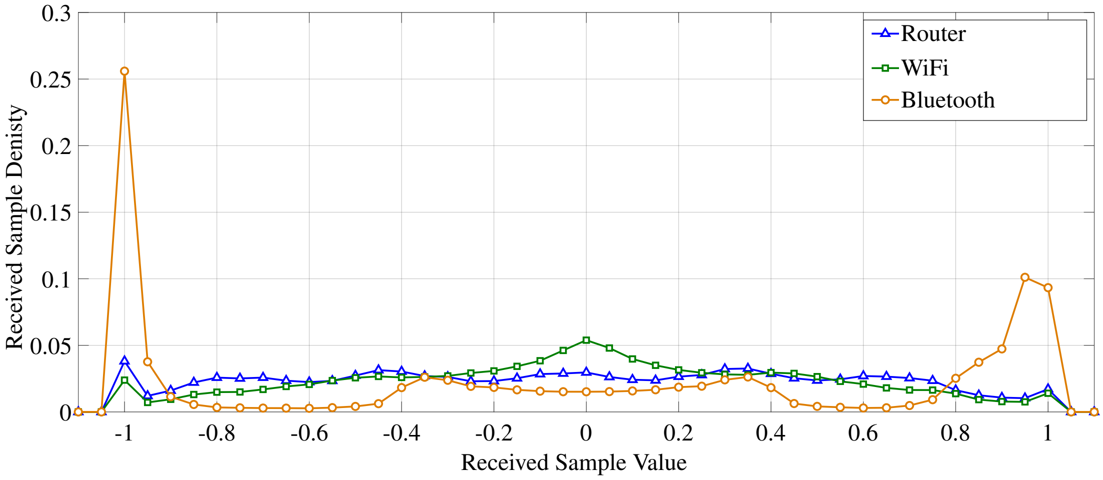


Figure 6.9: The computed PDFs for received WiFi, Router without internet connectivity and Bluetooth advertising signals.

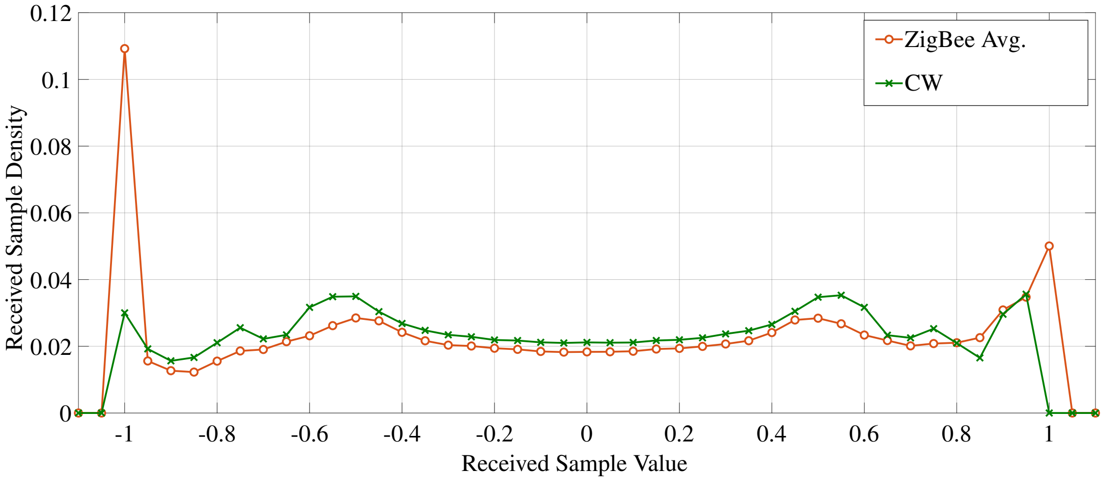


Figure 6.10: The PDFs for the received CW and ZigBee signals for various power levels.

Figures 6.8 - 6.11 present the calculated averaged signal PDFs for the received signal types including noise, WiFi, Router (no internet access), Bluetooth Advertising Channels, CW and ZigBee. The exhibited PDFs are the averaged PDFs for each signal, which is the average result for both the I and Q channels across all available data and ZigBee center frequencies. As the sample length is 1250 I/Q samples, the start, middle and end of the packets were accounted for and a specific part of a signal packet is not required. Thus, these PDFs provide as much variation as possible across all collected data. The PDFs were calculated in the range [-1.25:1.25] with a bin spacing of 0.05. This range was selected to be larger than the Pluto converted range of [-1:1] so as to visualize what occurs at the limits, which was useful for the comparison to the simulation results from Chapter 4. The PDFs include 41 useful bins, the maximum number of non-zero entries, and the spacing was selected based on trail and error and the requirement of achieving a low complexity operation. Calculating the PDFs using this method provided an opportunity to find the PDF distribution for each, even in

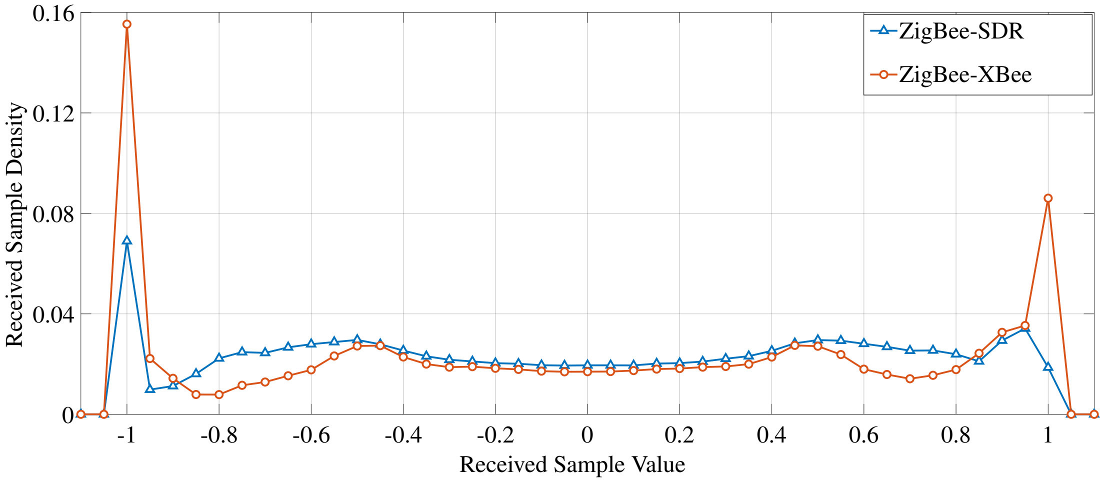


Figure 6.11: The calculated PDFs of the received legitimate ZigBee-XBee and ZigBee-SDR.

the presence of spurious wireless transmissions. The PDFs for some of the signals include multiple variations of what the PDF can exhibit, for example, the Bluetooth PDF displays both the receiver saturation and non-saturation cases, which is also reflected in the PDFs for the CW and ZigBee signals. Notably, distinct features are identifiable in most cases, using the simulation and wired studies' insights.

Analogous to the simulation and wired approaches, statistical analysis of the time domain expands the feature set to gain access to additional information regarding the I/Q samples of received signals. However, during this study, it was determined that if transmitted signals use similar modulation schemes (Bluetooth and ZigBee can both use phase-shift keying approaches) or saturate the receiver, additional features other than those extracted from the PDF and time domain are required. Receiver saturation is caused by the associated hardware restrictions, namely reference voltage and ADC resolution. The simulations in Chapter 4 were the ideal case and incurred no received value restrictions and, so, differences existed and became more apparent as signals were transmitted with more power. This aspect of the simulations made it possible for matched signal interference to be identifiable from legitimate ZigBee samples using the calculated PDFs and time series analysis. In the wired signal examination, the lack of similar modulation schemes being investigated allowed the time domain to identify the signals, even if the PDFs were similar. Based on those learnings, additional features are required to classify signals when the receiver is saturated or when signals use similar modulation schemes. These circumstances can occur in typical wireless environments and must be accounted for in this wireless experimentation.

Thus, the focus diversified to the frequency domain using, specifically, the fast Fourier transform (FFT). By utilizing the FFT, it was envisaged that the larger bandwidth signals could be distinguished from signals with a smaller channel width. For example, ZigBee has a 2 MHz wide channel while Bluetooth channels are 1 MHz wide. This concept allows, potentially, for a categorization to be made from the FFT of the received I/Q samples. This method exploits the information gained from using the RTSA, and its associated DPX technology (Fig. 6.1), to visualize the spectrum before grabbing the data and gaining a deeper understanding of the signals being transmitted. The FFT analysis in this work does not require any visualization or image recognition, in fact the concept focuses on analyzing the raw data from the calculated FFT. Thus, no spectrograms are developed for additional analysis, as seen in the literature [97] in Chapter 3.

Based on the above feature investigation process, a set of 28 features was extracted from the analysis of the raw received I/Q samples in the time domain, the FFT and the PDF. Every feature is calculated for both the I and Q channels separately and then averaged as the final feature. The developed 28 features could classify the received I/Q samples, as discussed in Section 6.5. However, this initial feature set included some comparable features and some features were included to evaluate their potential. However, by focusing on calculating the early results using this set, the informed decision to progress to model optimization and generalization performance, or remain in the feature engineering stage, could be made. As presented in Section 6.5, these 28 features provided the necessary classification results to warrant model development and optimization.

The 28 developed features are identified in Table 6.6, where the numbering corresponds to the subsequent feature importance investigation visualized in Fig. 6.12. Each area was calculated using the Matlab "trapz" function (6.1), while the Shannon Entropy used the formula as per (6.2). The different entropy approaches were investigated to understand if user programmed functions using (6.2) outperformed inbuilt Matlab approaches. The Hjorth parameters are particularly interesting as they have been useful in medical signals and the associated functions are: Activity (6.3), Mobility (6.4) and Complexity (6.5), where each equation can be simplified using square root rules and the relationship between variance and standard deviation. Most of the remaining features in Table 6.6 are self-explanatory, aside from the calculated number of peaks and PDF uniformity. In the time domain, the number of peaks relates to the number of values above 99% of the maximum value, where this was envisioned to help to distinguish between the higher bit rate signals and tones. The PDF number of peaks and curve uniformness were developed to identify signals that produce PDFs that incur multiple "bumps", e.g. WiFi, compared to PDFs that encompass two peaks at the edge points, i.e. CW. These PDF features and time-domain peak identification proved to be ineffective, which will be identified in the subsequent feature importance estimates.

$$\int_a^b f(x)dx \approx \frac{b-a}{2N} \sum_{n=1}^{N} (f(x_n) + f(x_{n+1})) \tag{6.1}$$

$$H = -\sum_{i} P_i \log_2 P_i \tag{6.2}$$

$$Activity = var(y(t)) \tag{6.3}$$

$$Mobility(y(t)) = \sqrt{\frac{var\left(\frac{dy(t)}{dt}\right)}{var(y(t))}} \tag{6.4}$$

$$Complexity = \frac{Mobility\left(\frac{dy(t)}{dt}\right)}{Mobility(y(t))} \tag{6.5}$$

The FFT functions need to be explained, as these approaches are mostly unique in this thesis. Each FFT is a 2048 point FFT, as this is the next power of 2 above the length of samples being analyzed (1250 samples). The FFT power is calculated by taking the summation of the square of the absolute value of each complex FFT point. 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 thesis's threshold corresponded to 30 and was determined through experimentation using the ZigBee and Bluetooth signals. Finally, 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 using the following function. This function takes the calculated FFT's absolute value and then analyzes the raw data to determine a single side 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.

Some of the developed features will have more distinctive characteristics and, so, will provide more useful information. Additionally, some developed functions were new and not pre-examined in either the simulation or wired approaches. Hence, the initial feature set was investigated to determine the most important/useful features and remove features that provide negligible value. The theory underlying what each feature was supposed to highlight and built-in Matlab functions for estimating predictor (feature) importance were used together to optimize the feature set to identify the most useful features. For this purpose, the Random Forest supervised machine learning approach was adopted. As discussed in Chapter 4, in contrast to other "black box" modeling techniques, tree-based classifiers' main advantage lies in the possibility of finding the reasoning behind the model. This property makes decision-trees a good candidate for problems that require an understanding of the decision-making process. This is the concept that was being examined during the feature importance experimentation.

Table 6.6: PDF, Time- and Frequency-domain features extracted from the raw received I/Q data

Domain	**Feature Number**	**Feature Description**
PDF	1	The number of non-zero entries
	2	The area in the center bins ([-0.1:0.1])
	3	The area in the left hand side bins (< -0.1)
	4	The area in the right hand side bins (> 0.1)
	5	The total area in the side bins
	6	The average area of the side bins
	7	The maximum value in the PDF
	25	The uniformity of the calculated PDF curve
	26	The number of peaks in the PDF
	28	The center bin (0) value
Time	8	Hjorth parameters [20] - Activity (Sample Variance)
	9	Sample Standard Deviation
	10	Absolute mean value
	11	Absolute maximum value
	12	The root-mean square (RMS) value
	13	Hjorth parameters [20] - Mobility
	14	Hjorth parameters [20] - Complexity
	15	Matlab's "entropy" function
	16	Shannon Entropy - using user-specific approach
	17	Shannon Entropy - using Matlab's "histogram function"
	18	Matlab's "approximateEntropy" function
	19	Sample Skewness
	20	Sample Kurtosis
	23	Number of peaks
	24	Number of zero crossings
Frequency	21	Number of FFT points over a predefined threshold
	22	The signal power in the FFT
	27	Unique function that uses the FFT points to estimate signal bandwidth

Furthermore, while constructing decision-trees, only features that are useful for a given problem are included. This understanding permits using a tree-based classifier for feature selection.

Two Matlab functions were used with the Random Forest specifications applied, namely, "TreeBagger" and "fitcensemble". Both functions were investigated using different numbers of trees with the full number of predictors available at each decision point. Hence, the model was trained using the available data and the out-of-bag predictor importance function was applied to each trained model. This approach was repeated for a few thousand iterations for each function, and the final result combined the averages of the two individual approaches. The results of this feature importance investigation are supplied in Fig. 6.12, where fourteen features were chosen, based on the results and the theory underlying the features, since, to be useful, the model will need to generalize to unseen data. The final feature set encompassed the following features as per Table 6.7, where the initial set corresponds to Table 6.6 and Fig. 6.12 and the final feature number corresponds to the feature position in the input vector $X = \{x_1, x_2, ..., x_{14}\}$. The input vector X contains the predominately investigated features and are the features leveraged in Chapter 7.

The final feature set contains elements that have theoretical justifications explaining

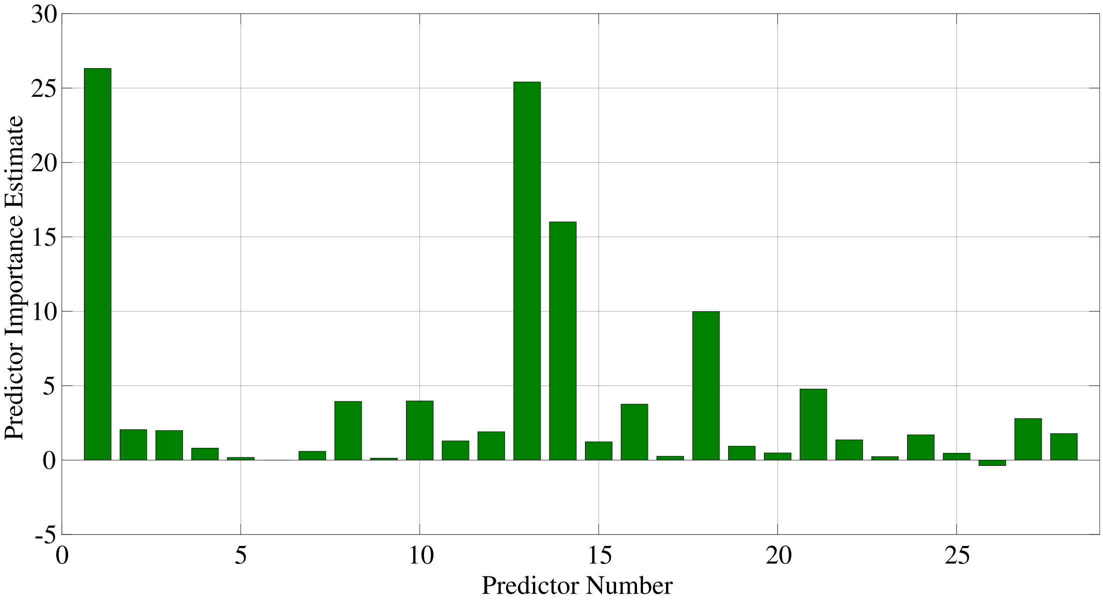


Figure 6.12: The estimated predictor importance measurements using the built in MatLab functions averaged across multiple iterations, feature depths and number of trees using two distinct functions, namely TreeBagger and fitcensemble.

why the corresponding importance estimates were high. For example, the PDF features classify the received signal into a specific pattern, which is useful in identifying a specific signal type directly (for example, noise or a group of signals). The FFT based features provide access to information regarding the received signal bandwidth, which can be used to help identify tones or signals with larger bandwidths and similar modulations schemes. The FFT size is currently the next power of two above the signal length, meaning an FFT of length 2048 was applied. This FFT size may be further investigated to see if this can be reduced, given that the motivation is to apply this methodology to low-power edge devices However, small and simple FFT libraries such as, for example, “kissfft” do exist and could be leveraged. Finally, the time series low-order statistical features help to identify the correct samples that correspond to the distinctive signal, as CW differs from IEEE802.11, for example. Section 6.5 verifies that the features are effective by focusing on the developed machine learning model’s generalization error. The feature count remains at fourteen. It is envisaged that the extra features will provide additional options in different wireless environments and allow the methodology to be applied with a greater probability of success.

Overall, when compared to the literature, a low-order statistical feature set was developed using data collected from an active wireless environment that is typically changeable. The raw I/Q data approach to create a concise novel feature set based on time, frequency, and space dimensions (PDF) is one of this thesis’s main contributions. To the best of the author’s knowledge, this application of Hjorth parameters [20] and FFT dynamics, as described in this section, is a novel method. Additionally, no assumptions are made about the wireless channel and network data is neglected. The developed

Table 6.7: PDF, Time- and Frequency-domain optimized features extracted from the raw received I/Q data

Domain	Initial Feature Number	Final Feature Number	Feature Description
PDF	1	1	Number of non-zeros entries of PDF
	2	2	The area in the center bins ([-0.1:0.1])
	3	3	The area in the left hand side bins (< -0.1)
	28	14	The center bin (0) value
Time	8	4	Hjorth parameters [20] - Activity (Sample Variance)
	10	5	Absolute mean value
	12	6	The root-mean square (RMS) value
	13	7	Hjorth parameters [20] - Mobility
	14	8	Hjorth parameters [20] - Complexity
	16	9	Shannon Entropy - using use specific approach
	18	10	Matlab's "approximateEntropy" function
	24	12	Number of zero crossings
Frequency	21	11	Number of FFT points over a predefined threshold
	27	13	Unique function that uses the FFT points to estimate signal bandwidth

features differ from the literature by only requiring access to raw received I/Q samples, permitting independent device decisions and using low-order statistics. Typically, the literature uses high-order statistics [81] and/or cumulants [82, 83, 84] when applying traditional techniques. These features were developed and extracted in Matlab, however, sufficient resources exist for adapting the feature extraction to other coding languages and platforms ("SciPy"). This concept was proved by the successful translation of the Matlab ZigBee simulation code to Python3 in Chapter 4.

6.4 Additional Machine Learning Models

In Chapter 4, the machine learning concept, classifier evaluation metrics, the SVM model and the Random Forest model were introduced. In this chapter, wirelessly transmitted over-the-air data is examined and more state of the art models are required to achieve high performance. As a result, several new supervised machine learning approaches are investigated. This section explains these models and why they were chosen. As discussed in Chapter 4, the choice of the machine learning method depends on a number of factors, namely, the problem that needs to be solved, the type of data available and the required interpretability of the obtained results. This chapter focuses on legitimate signal classification, as before interference can be detected, the legitimate wireless signals being transmitted need to be identified. The classification outcome is critical as performing mitigation once an attack (or packet loss reason) is detected compels the development of the interference diagnostic framework, as edge nodes can usually deliver packets to non-jammed neighbors [19]. As a result, supervised discriminative machine learning classification is required. The discriminative algorithm makes no assumption on how the data is generated, which is important when multiple different transceivers and signals are being analyzed.

6.4.1 Decision Trees

Tree-based models are easy to use as they are invariant to the input scale and accommodate categorical and missing data. Many different supervised machine learning decision-tree approaches exist and many of them have been successfully applied to wireless signal/modulation scheme classification, as discussed in Chapter 3. In the simulations in Chapter 4 and the wired investigation in Section 6.2 of this chapter, the Breiman and Cutler [23] Random Forest model was applied and produced high-performance results. However, this technique of using decision trees in an independent ensemble has been improved by using boosting algorithms to create an accurate and robust classifier from a set of weak classifiers, as previously identified in Chapter 3.

Before the boosting approaches applied in this thesis can be explained, typical decision tree operation, specifically the ensemble concept, needs to be re-introduced to provide an accurate distinction between the previously applied Random Forest and the more modern approachers. Decision trees learn the decision boundary by recursively partitioning the feature space into non-overlapping regions using some defined threshold. In general, adding a large number of decision trees, beyond a specific limit, does not improve the classifier's performance. This result occurs because each new tree is constructed to correct the sequence of previous trees' errors. As a result, at some point, the classifier stops reducing the classification error. This concept is visually demonstrated in Fig. 6.5, where the error stabilizes at approximately thirty decision trees. The number of trees to be used is usually optimized during the validation stage, along with other parameters. This approach will be demonstrated in the following results section of this chapter. However, typically, one tree is not sufficient to make an accurate prediction on an unlabeled input, which results in decision tree models implementing a tree ensemble approach. As discussed in Chapter 4 in terms of Random Forest, ensemble methods allow using multiple weak learning algorithms to get better predictive performance [23]. An example visualization of such an ensemble classification framework is provided in Fig. 6.13.

The two main types of approaches to building the ensemble framework are dependent and independent frameworks. In a dependent framework, each classifier's output is used for the construction of the next classifier. In contrast, each classifier is built independently in an independent framework and their corresponding outputs are combined to generate a single output. The advantage of the dependent framework lies in guiding future decisions based on the knowledge generated by the previous interaction. As a result, a new classifier is generated to improve on the instances for which the previous

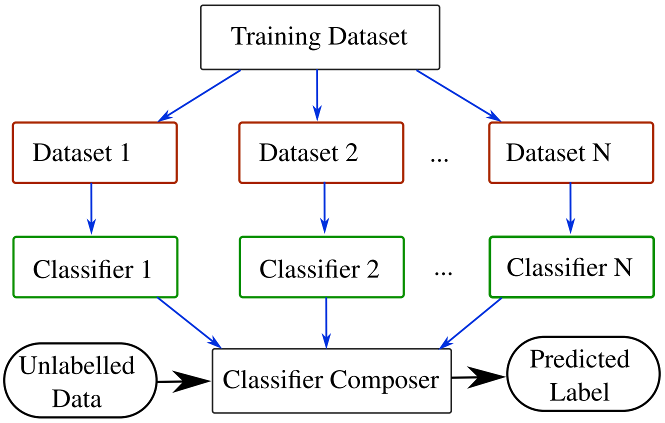


Figure 6.13: An example visualization of the ensemble classification framework applied in decision-tree based machine learning classifiers.

decision tree did not perform well. By discussing these two frameworks, the operational differences between the applied Random Forest approach and the boosting approaches in Section 6.5 are evident.

Decision trees include several node types which need to be defined before discussing the ensemble frameworks and associated approaches. A root node corresponds to the first or parent node, representing the entire population or sample. The root node gets divided (split) into two or more homogeneous sets, called internal nodes. Splitting is the term given to the process of dividing a node into two or more sub-nodes. Internal nodes of a decision tree include one or more child nodes, equivalently, a node that is not a leaf node. Leaf, or terminal, nodes do not split, while a decision node occurs when a sub-node splits into further sub-nodes. A subsection of the decision tree is called a branch or sub-tree. Any node divided into sub-nodes is called a parent node of the associated sub-nodes, whereas each sub-node is a child of the parent node. As discussed below, the size of decision trees can be reduced to avoid complex decision trees and overfitting, where this process is called pruning.

6.4.1.1 Independent Decision-Tree Methods

In the independent ensemble classification framework, the original dataset is divided into several smaller datasets (sub-sets), where the data in the sub-sets can be overlapped or mutually exclusive. Each dataset is used to train a single decision tree classifier and a classifier composer step produces the final decision. This classifier composer generates the final prediction by either averaging the output for regression problems or majority voting for classification tasks. However, if the training set's perturbation causes significant changes in the model, such a classifier is then referred to as unstable. Essentially, a stable classifier will be formed from individual classifiers that have similar performance given subsets of the training data.

Bagging, also known as a bootstrap aggregating method, was introduced in Chapter

4 and is a well-known example of the independent ensemble method [157]. This approach trains each classifier on a training set uniformly sampled with replacement from the original dataset. As a result, bagging produces unique decision-trees as a unique random subset of the training samples is used to train each tree and sampling with replacement maintains the sample size. The sampling with replacement method implies some observations may be repeated more than once, while others may not be included at all. This bagging procedure has been demonstrated to improve unstable classifiers, such as classification and regression trees [157]. The Random Forest method, employed and discussed in Chapter 4, is an example of an independent ensemble classification [23].

6.4.1.2 Dependent Decision-Tree Methods

Dependent ensemble classification frameworks are introduced in this Chapter and compared to the independent Random Forest framework. This dependent ensemble concept is based on boosting, a method for creating an accurate and strong classifier from a set of weak classifiers [158]. The initial algorithm applied in this thesis is AdaBoost, a well-known example of dependent ensemble classification [159]. The AdaBoost algorithm's primary concept resides in giving more focus, quantified by an assigned weight, to harder to classify instances. After each iteration, misclassified instances gain more weight, while the weights of correctly classified instances are decreased. Every classifier is then weighted according to its accuracy, which results in the decision trees that perform more accurate classification having higher weights. These higher-performing classifiers contribute greater to the final prediction, which is a majority voting among the weighted classifiers.

Gradient boosting is another powerful algorithm and the second dependent approach applied in this thesis. While AdaBoost uses up-weighting of the misclassified instances, gradient boosting identifies misclassification from the previous iteration's large residual. Gradient boosting regulates the importance of the misclassified instances by training weak classifiers on the remaining errors (residuals) of a strong classifier. These residuals are computed after each iteration and the process is carried out until the residuals are close to zero. Each weak classifier's contribution to the strong one is computed using the gradient descent optimization technique, allowing for the computation of the contribution, which minimizes the strong classifier's error. The residuals are calculated at the end of each iteration and the new model is represented as a sum of the previous model and the model obtained by fitting to the residuals.

Intuitively, in terms of decision trees, gradient boosting is a stage-wise additive model that generates learners during the learning process. Decision trees are added one at a time, and existing trees in the model are not changed. The first learner (decision tree) is built to predict the instances in the training dataset and calculates the loss, which is the difference between the first learner's outcomes and the actual labels. The second learner is trained on the residual error produced by the first learner to predict the loss after the first step and continues to do so until it reaches a threshold (residuals close to zero). By training the second learner on the gradient of the error with respect to the first learner's loss predictions, the second learner is being trained to correct the first model's mistakes. This concept is the core of gradient boosting, and it enables several simple learners to compensate for each other's weaknesses to fit the data better. Stopping the training process before the residuals are absolute zero avoids overfitting by building a very complex model. As a result, approaches that decrease the complexity of a model are advantageous, as it reduces the probability of overfitting and increases the chances to successfully generalize to new data.

6.4.2 XGBoost

From the analysis of the independent and dependent ensemble methods, dependent methods are known to outperform independent ones [23]. This thesis's methodology aims to achieve high performance on the available data while maximizing the probability of generalizing well to unseen data. Hence, a more powerful version of gradient boosting, in particular boosting with regularization, is applied to achieve the desired operation. The extreme gradient boosting classifier, XGBoost [160], is a prominent implementation of boosted decision trees with regularization and is available as open-source software library. Notably, XGBoost has been widely adopted in various Kaggle machine learning and data mining competitions, where state-of-the-art results for an extensive range of problems have been reported. Consequently, XGBoost is chosen to be the predominant machine learning model for the interference diagnostic framework in this study.

To explain the advantages XGBoost exhibits over typical gradient boosting approaches, specific aspects of the algorithm need to be defined. For gradient boosting, given a set of N training examples of the form $\{(x_1,y_1),...,(x_N,y_N)\}$, such that $x_i \in R^d$ is a d-dimensional feature vector of the i-th example and y_i is its label; the aim is to produce a model, f_k, to predict values in the form of (6.6). At each gradient boosting stage, $1 \leq k \leq M$, a new improved model is constructed, $f_{k+1}(x)$, that adds an estimator, $h(x)$ (6.7). This estimator is fitted to the residuals, $y - f_k(x)$, by minimizing the objective function (Obj) shown in (6.8), which measures how well the model fits the training data, where l is the training loss function measuring the difference between the predicted and actual outputs, f_k corresponds to an independent tree structure and Ω is a regularization function.

$$\hat{y} = \sum_{k=1}^{M} f_k(x) \tag{6.6}$$

$$f_{k+1}(x) = f_k(x) + h(x) \tag{6.7}$$

$$Obj = \sum_{i=1}^{N} l(\hat{y}_i, y_i) + \sum_{=1}^{M} \Omega(f_k) \tag{6.8}$$

Each decision tree, f_k, includes several tree parameters that define the learner, such as maximum tree depth. These parameters, discussed and optimized in Section 6.5, need to be defined to learn the decision trees represented with the function, f_k. However, learning a tree structure is much harder than traditional optimization problems where one can take the gradient, meaning it is intractable to learn all the trees at once. As a result, decision trees are built using an additive training strategy by adding one new tree at a time. The prediction value at step, t, is defined in (6.9), where at each described step, a new tree is added that optimizes a given objective function. This optimization is achieved using the gradient descent technique, specifically, gradient boosting of decision trees. This approach uses gradient updates by additional trees learned one at a time. XGBoost implements this concept by computing the second-order derivative using a second-order Taylor approximation. This provides a further improvement over the conventional gradient descent technique and the resultant regularized objective at step, t, is defined in (6.10), where $g_i = \partial_{\hat{y_i}^{(t-1)}} l(\hat{y_i}^{(t-1)}, y_i)$ and $h_i = \partial^2_{\hat{y_i}^{(t-1)}} l(\hat{y_i}^{(t-1)}, y_i)$ are the first (gradient) and second (Hessian) order derivatives of the training loss function, respectively, and ∂ is the partial derivative.

$$\hat{y_i}^{(t)} = \sum_{k=1}^{t} f_k(x_i) = \hat{y_i}^{(t-1)} + f_t(x_i) \tag{6.9}$$

$$\begin{aligned} Obj^{(t)} &= \sum_{i=1}^{N} l(y_i, \hat{y_i}^{(t-1)} + f_t(x_i)) + \sum_{i=1}^{t} \Omega(f_i) \\ &\approx \sum_{i=1}^{N} \left[l(y_i, \hat{y_i}^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \sum_{i=1}^{t} \Omega(f_i) \\ &\approx \sum_{i=1}^{N} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \end{aligned} \tag{6.10}$$

In machine learning, the developed model should exhibit high performance on the training data and generalize well to unseen instances. However, due to the bias-variance trade-off, high complexity models may result in high variance or overfitting. Here, to decrease the complexity of the decision tree, regularization methods are applied. In XGBoost, the regularization term $\Omega(f)$ in the objective function (6.10), aims to control the complexity of each individual decision tree. This regularization term is defined in (6.11), which is one way to define the complexity and works well in practice. Each

f corresponds to an independent tree structure with a vector of leaf weights w on the j_{th} leaf and T is the number of leaves in the tree. The number of terminal nodes is penalized with the γ parameter and weight optimization is performed using the $L2$ norm (Euclidean norm) to encourage smaller weights, where the associated $L2$ term in (6.11) is λ. This regularization term, Ω, helps to smooth the weights to avoid over-fitting, which is a key reason for choosing XGBoost in this thesis' methodology development.

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{6.11}$$

To further reduce the probability of overfitting, XGBoost includes two types of randomization: row and column sub-sampling [161]. Row sub-sampling improves classifier performance by using a randomly selected fraction of training examples without replacement. Column (feature) sub-sampling introduces randomness by constructing a decision tree at each iteration using a random subset of features. Furthermore, shrinkage (learning rate) is a regularization technique that scales newly added weights by a factor after each boosting step. This approach reduces each individual tree's influence and may enable future trees to improve the model. Finally, the Hessian of the objective function (6.10) regulates the number of points in the node and represents the level of purity in the node. Setting a minimum allowed number of instances that allow further splits in the node can reduce the complexity of the model by penalizing very deep trees. The XGBoost parameters optimized in Section 6.5 apply this concept in the "min_child_weight" parameter. This parameter is the minimum sum of instance weight (Hessian) needed in a child node. If the tree partition step results in a leaf node with the sum of instance weight less than "min_child_weight", then the building process will cease further partitioning. The larger the minimum child weight is, the more conservative the algorithm will be.

By analyzing the structure of a decision tree, the usage of the above concepts becomes clear. The structure of the decision tree is defined in (6.12), where the mapping q is a function which assigns each data point to the corresponding leaf, T is a number of leaves, and w is a vector of weights on the leaves. This process results in the regularized objective at step t corresponding to (6.13), where $I_j = \{i|q(xi) = j\}$ is a set of indices of data points assigned to the j-th leaf using function $q(x)$, where q is a tree structure. By defining $G_j = \sum_{i \in I_j} g_i$ and $H_j = \sum_{i \in I_j} h_i$, the regularized objective can be compressed to (6.14).

$$f_t = w_{q(x)},\ w \in R^T,\ q : R^d \to \{1, 2, ..., T\} \tag{6.12}$$

$$
\begin{aligned}
Obj^{(t)} &\approx \sum_{i=1}^{N} \left[g_i w_{q(x_i)} + \frac{1}{2} h_i w_{q(x_i)}^2 \right] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \\
Obj^{(t)} &= \sum_{j=1}^{T} \left[(\sum_{i \in I_j} g_i) w_j + \frac{1}{2} (\sum_{i \in I_j} h_i + \lambda) w_j^2 \right] + \gamma T
\end{aligned}
\tag{6.13}
$$

$$
Obj^{(t)} = \sum_{j=1}^{T} \left[G_j w_j + \frac{1}{2} (H_j + \lambda) w_j^2 \right] + \gamma T \tag{6.14}
$$

In equation (6.14), w_j are mutually independent, the form of $G_j w_j + \frac{1}{2}(H_j + \lambda) w_j^2$ is quadratic and an optimal w_j, for a region j and associated optimal objective reduction are given by (6.15) and (6.16), respectively. Now that a method to measure how good a tree is has been established, ideally, an algorithm would enumerate all possible trees and pick the best one. However, this is not feasible in practice, so optimization occurs on one level of the tree at a time. Specifically, learning the tree structure implies deciding on how to best split features (internal nodes). After each split, the originally defined leaf is converted to an internal node and new left (I_L) and right (I_R) nodes are generated, where $I = I_L \cup I_R$. The gain of each split is defined in (6.17), where the formula can be decomposed as 1) the score on the new left leaf, 2) the score on the new right leaf, 3) the score on the original leaf and 4) regularization on the additional leaf. If the calculated gain is smaller than γ, the optimal approach would be not to add that branch. Each decision tree is built to the predefined maximum depth and the nodes with negative gain are then pruned in bottom-up order. These defined approaches are used in Section 6.5 to identify the optimal XGBoost tree structures for this wireless signal classification problem and the resulting optimal parameters are leveraged in Chapter 7 as the foundation for the interference detection framework.

$$
w_j^* = -\frac{G_j}{H_j + \lambda} \tag{6.15}
$$

$$
Obj^* = -\frac{1}{2} \sum_{j=1}^{T} \frac{G_j}{H_j + \lambda} + \gamma T \tag{6.16}
$$

$$
Gain = \frac{1}{2} \left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right] - \gamma \tag{6.17}
$$

6.4.3 Neural Networks

The neural network (NN) machine learning model is a system that was inspired by the biological neural network and mimics brain function. This section provides a high-level overview of NNs, as the NN model is the state-of-the-art technique for many

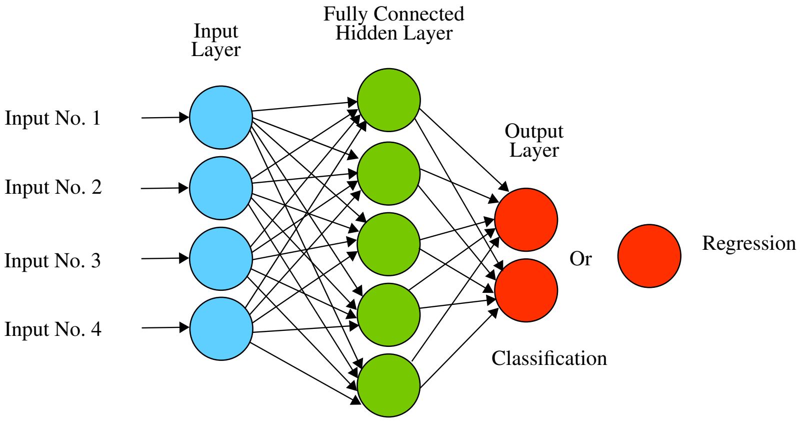


Figure 6.14: An example representation of a feedforward neural network with one hidden layer. Both the regression and classification concepts are visualized in the output layer.

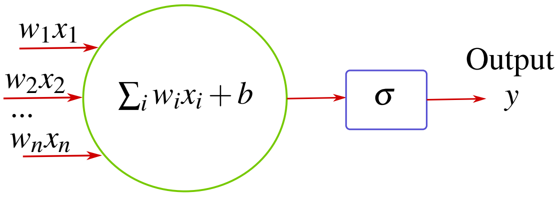


Figure 6.15: An example of a neuron, specifying the input and its corresponding weights and the activation function applied to the weighted sum over the inputs from the previous layer.

supervised ML problems. This concept is outlined in Chapter 3, where the general trend for wireless signal and received modulation scheme classification uses deep learning NN approaches. As influenced by biological brain function, a NN comprises neurons, whose outputs generate nonlinear functions from its input signal. An input signal, which is parameterized by weights, travels to the network's output through several layers, where each layer performs a different type of nonlinear transformation of the input signal. An example representation of a typical NN is shown in Fig. 6.14, where only one hidden layer is implemented.

The NN's primary concept is that the weights w are learnable and can control the influence of one neuron on another. This concept is visualized in Fig. 6.15, where the neuron's weight vector input is summed and, if the sum is above some definite threshold, the neuron can fire. This approach is modeled with a nonlinear activation function, σ, such as, for example, sigmoid, tanh, rectified linear unit (ReLU) or LeakyReLU. Each neuron performs a dot product of the input and its corresponding weights, adds a bias term, b, (this shifts the activation function to allow for better learning) and then applies the nonlinearity with different activation functions. As a result, rather than using original features, the NN uses new learned features, which are functions of the input. The input data is transformed into a more abstract and composite representation on each level of a multi-layered NN. Learning through a cascade of multiple layers of nonlinear processing elements is referred to as deep learning (DL). This deep learning method is applied to the features extracted in Section 6.3.

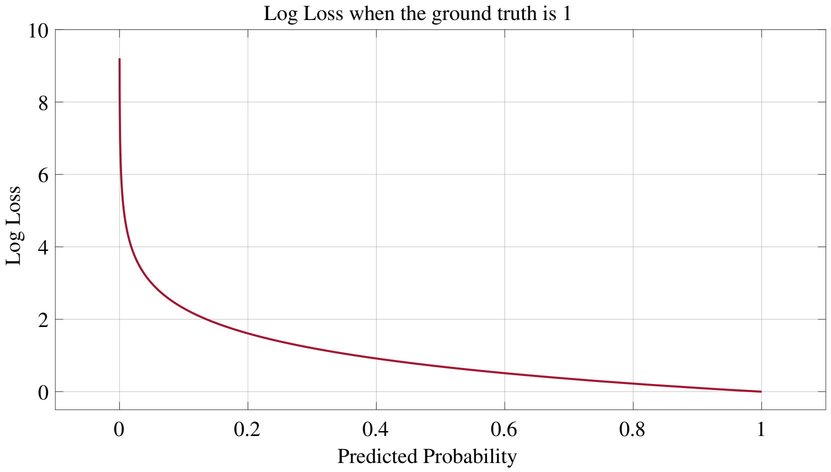


Figure 6.16: Visualization of the possible log loss values given a ground truth of 1.

The NN training process starts with the random initialization of the network parameters w, as it allows for the breaking of the symmetry and enables hidden nodes to learn different features. To investigate the performance of the initial hypothesis of the network, the initial parameters w and input x, forward propagation is performed. In contrast, during the process of training, backpropagation, the accuracy of prediction of the model is improved, or some specific loss function is minimized. The loss function evaluates how well the algorithm models the given data and the choice of loss function depends on the given task. This thesis focuses on binary and multi-class classification and the log loss (categorical cross-entropy loss) is the applied loss function for minimization. Notably, cross-Entropy is not Log Loss, but they calculate the same quantity when used as loss functions for classification problems. As a result, the classifier's output is a probability [0:1] over the number of classes and it is used for multi-class classification, where the "softmax" activation is applied. The input labels for training are in the form NxC, where N is the number of inputs and C is the number of classes, where the true class has a probability of 1. This approach is achieved using one-hot encoding to convert the input column vector of labels into an NxC matrix of probabilities. The applied multi-class categorical cross-entropy loss formula (log loss) rewards/penalizes the correct classes' probabilities only. For a binary case, the categorical cross-entropy function is the log loss function. As a result, the loss increases as the predicted probability diverges from the label, as visualized in Fig. 6.16. The loss formula for the categorical cross-entropy loss used in multi-class classification is defined by (6.18) and the log loss function for binary classification is defined in (6.19), where L is the loss function that measures the difference between the prediction $\hat{y}_i$ and the target y_i and p_i is the associated probability. This loss function measures the model's performance, where the output is a probability value defined between 0 and 1.

$$L(y,\hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_c^{(i)} log(\hat{p_c}^{(i)}) \tag{6.18}$$

$$L(y,\hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i log(p_i) + (1-y_i) log(1-p_i)\right] \tag{6.19}$$

The loss function can be optimized using different techniques. The gradient descent algorithm is one of the most popular algorithms to perform optimization and by far the most common way to optimize neural networks. Gradient descent iteratively updates the parameters w with a certain learning rate α until convergence is achieved. This approach is achieved by computing partial derivatives $\frac{\partial L}{\partial w}$ and $\frac{\partial L}{\partial b}$ of the loss function L with respect to weights w and bias term b, which were visualized in Fig. 6.15. Therefore, using the approach that each neuron contains the function $f(x) = \sum_{i=1}^{N} w_i x_i + b$, a single iteration of the gradient descent updates the parameters w and b in layer l according to equation (6.20) and (6.21), respectively, where i corresponds to the specific neuron and j relates to the input to that neuron. To understand the update procedure, let us assume that there is no bias. In this case, if the current value of the slope of w is positive, the current point is the right of optimal w^* and the update will be negative to start getting closer to the optimal values of w^*. However, if the current slope is negative, the update will be positive and will increase the current values of w to converge to the optimal set of values. This process continues until the cost function converges, that is, until the error curve becomes flat and does not change.

$$w_{ij}^{(l)} = w_{ij}^{(l)} - \alpha\frac{\partial L}{\partial w_{ij}^{(l)}} \tag{6.20}$$

$$b_i^{(l)} = b_i^{(l)} - \alpha\frac{\partial L}{\partial b_i^{(l)}} \tag{6.21}$$

However, each neuron's output depends on the weights, bias and activation function and, so, the overall loss function is minimized based on the neuron activation. As the back propagation algorithm performs the backward propagation of errors during training, the partial derivatives can be computed by applying the chain rule. During this procedure, the slope of the derivative of the activation function is calculated. Intermittently, the gradients can become very small when training deep networks and lead to the vanishing gradient problem. When the activation function includes regions where the function's slope is close to zero (sigmoid and tanh), the learning process may become very slow. However, the ReLU activation function has partially addressed the problem of vanishing gradients, as the ReLU gradient is equal to one for all positive inputs and, therefore, cannot shrink to zero as the neuron saturates. This function, $max(0, w.x+b)$ is not computationally expensive and is easy to use during back propagation. As a result, the ReLU activation function was applied here.

To implement the entire process, the loss function optimization approach needs to be selected. Different types (generally three variants) of gradient descent exist, which mainly differ in the amount of data they use. Depending on the amount of data, a trade-off between the parameter update's accuracy and the time it takes to perform an update is required. Batch gradient descent is the most common variant and calculates the error for each training example within a dataset and updates the model over the whole batch. As the gradients for the whole dataset need to be calculated to perform just one update, batch gradient descent can be very slow and is intractable for datasets that do not fit in memory. A single iteration is usually called a training epoch and this method generates a stable error based on all training examples. In contrast, Stochastic Gradient Descent (SGD), performs a model update for each training example $x^{(i)}$ and label $y^{(i)}$. This updating approach can result in noisy gradients and, therefore, complicates the process of minimizing the error, but it is typically faster than batch gradient descent. SGD's fluctuation has both positives and negatives, as it can jump to new, and potentially better, local minima. However, this method will ultimately complicate converging to the exact minimum, as SGD will keep overshooting. By slowly decreasing the learning rate, SGD achieves the same convergence behavior as batch gradient descent, almost certainly converging to a local or the global minimum for non-convex and convex optimization, respectively. The final variant is mini-batch gradient descent, which applies the best sections of the previous variants as it performs an update for every mini-batch of n training examples. This method encourages smoother gradients and allows for selecting a batch size suitable for the amount of memory available. This mini-batch approach is typically the algorithm of choice when training NNs.

However, the above methods incur challenges and as a result, optimized solutions have been developed. Choosing the learning rate and associated schedule to be applied can be difficult. Choosing a value that is too small leads to excessively slow convergence, while a learning rate that is too large can hinder convergence and cause the loss function to fluctuate around the minimum or even to diverge. The learning rate schedule can adjust the learning rate during training based on a predefined schedule or when the change in objective between epochs drops below a threshold. However, these approaches have to be defined in advance and, so, are unable to adapt to a dataset's characteristics. Additionally, the same learning rate applies to all parameters, which may be unwanted if the data is sparse and features have very different sampling frequencies. Finally, another key challenge surrounds minimizing highly non-convex error functions that are common for neural networks. The aim is to avoid getting trapped in the numerous suboptimal local minima of these non-convex error functions. To deal with the aforementioned challenges, several gradient descent optimization algorithms have been developed. These optimized approaches are the methods investigated in Section 6.5.

Adaptive Moment Estimation (Adam) [162] is one of these optimized gradient descent methods. Adam computes adaptive learning rates, which means it computes individual learning rates for different parameters. Adam is derived from adaptive moment estimation because Adam uses estimations of first and second moments of the gradient to adapt the learning rate for each weight of the neural network. The first moment is mean, and the second moment is uncentered variance (meaning the mean is not subtracted during variance calculation). To estimate the moments, Adam utilizes exponentially moving averages, computed on the gradient evaluated on a current mini-batch. It stores an exponentially decaying average of past squared gradients, v_t, and an exponentially decaying average of past gradients, m_t. Adam behaves like a heavy ball with friction moving along the slope of the loss function. As a result, the method prefers flat minima in the error surface. The decaying averages of past and past squared gradients, m_t and v_t, respectively, are computed using (6.22) and (6.23), respectively, where g_t is the gradient on the current mini-batch and β are hyperparameters for the decay rate.

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \tag{6.22}$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \tag{6.23}$$

m_t and v_t are estimates of the mean and the uncentered variance of the gradients, respectively, and are initialized as vectors of 0's. This initialization results in a bias towards zero, especially during the initial time steps and when decay rates are small (i.e. β_1 and β_2 are close to one). To counteract the biases, bias-corrected first and second moment estimates are calculated, as per equation (6.24) and (6.25), respectively.

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \tag{6.24}$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t} \tag{6.25}$$

The resulting Adam update rule is defined by (6.26), where default values of 0.9 for β_1, 0.999 for β_2, 0.002 for η and 10^{-8} for ε are applied.

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \varepsilon} \hat{m}_t \tag{6.26}$$

An extension of this approach is AdaMax, which is based on the infinity norm rather than the l_2 norm applied in Adam (the v_{t-1} term in (6.23)). To avoid confusion with Adam, u_t denotes the infinity norm constrained value of v_t. This results in the AdaMax update rule as per (6.27), where $u_t = max(\beta_2 v_{t-1}, |g_t|)$. Section 6.5 applies these approaches and AdaMax is identified as the optimal approach.

$$\theta_{t+1} = \theta_t - \frac{\eta}{u_t}\hat{m}_t \tag{6.27}$$

Different types of NNs exist, such as convolutional NN (CNN), long short-term memory networks (LSTM), recurrent neural network (RNN), residual NN (ResNet) and others. Very deep NNs may be challenging to train due to the possible problem of either the vanishing or the exploding of the gradients [163]. In this thesis, feedforward deep neural networks are investigated with relatively shallow hidden layer depths. The NN investigation is completed to compare against the determined XGBoost model in terms of achieved accuracy, generalization to new data and optimization and prediction times. As a result, the previous description of NNs is sufficient for this thesis.

6.4.4 K Nearest Neighbors and Gaussian Naive Bayes

The final two machine learning approaches are K Nearest Neighbors (k-NN) and Gaussian Naive Bayes, which are both briefly examined in Section 6.5 to validate using a decision tree classifier. As a result, only a succinct description of each approach is represented here. K-NN is a non-parametric (it can be used even when the variables are categorical) machine learning method that can be used for multi-class classification. This model is one of the most fundamental and simple classification methods and is suitable for a classification study when there is little or no prior knowledge about the distribution of the data. As a result, the k-NN model works on all kinds of data and no specific assumptions should be made concerning the data. The k-NN algorithm assumes that similar things exist in close proximity, meaning that similar values will be clustered together in a group. The primary concept is that the model is trained using a large amount of training data, where each data point is characterized by a set of variables (features). Each instance is plotted in a high-dimensional space, where each axis in the space corresponds to an individual variable. For a new (test) data point, the K nearest neighbors that are closest (most similar) to it need to be identified. The objects' neighbors are taken from a set of objects for which the class (for k-NN classification) or the object property value (for k-NN regression) is known. The concept can be visualized in Fig. 6.17.

Classification is implemented by considering the K-nearest neighbors, meaning the optimum value of "K", the number of nearest neighbors, needs to be determined. By default, the value of "K" is set to 5. In k-NN classification, an object is classified by its neighbors' plurality vote, with the object being assigned to the class most common among its K nearest neighbors, calculated using a distance measurement. As majority voting is applied, typically, an odd value for "K" is optimal. This concept relates to avoiding any condition where a two-class classification needs to be performed, for

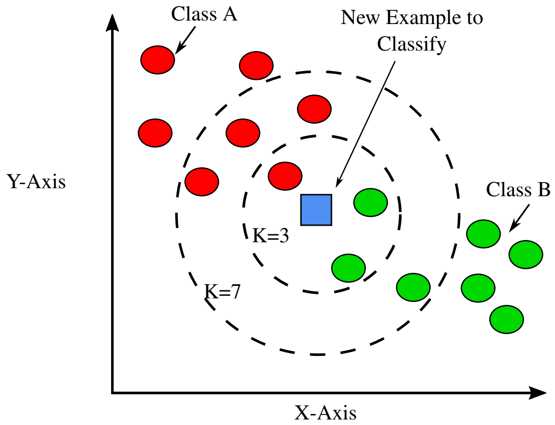


Figure 6.17: Visualization of the fundamental operation of the k-NN machine learning model.

example, if “K” is even, the classes can have the same number of votes. By choosing “K” to be odd, such cases are avoided.

k-NN is a form of instance-based learning, where the function is only approximated locally and all computation is deferred until function evaluation. The algorithm relies on distance for classification, where the typical distance functions are manipulations of Minkowski distance (6.28), which is set in the k-NN algorithm by setting the value of p. These distance measures include Manhattan distance (6.29) for $p = 1$ and Euclidean (6.30) for $p = 2$, where A and B are represented by the feature vectors $A = (x_1, x_2, ..., x_n)$ and $B = (y_1, y_2, ..., y_n)$ in a n dimensional space. For discrete variables (for example, text classification), an overlap metric, such as the “Hamming distance”, can be applied. The Hamming distance measures the number of misaligned elements of the text string, where the smaller the value, the more similar the objects are to each other. As a result, if the features include vastly different scales, normalizing the training data can dramatically improve its accuracy. Furthermore, due to the use of distances to an object’s nearest neighbors, a useful technique is to assign weights to the neighbors’ contributions. This approach ensures the nearer neighbors contribute more to the final classification than the more distant ones. An example of a common weighting scheme gives each neighbor a weight, $\frac{1}{d}$, based on the distance d to the object’s neighbor. However, when the number of data points is very large, specific methods must be employed to rapidly search the space and find the “most similar” objects. Typically, a form of pre-computation is employed, such as, for example, selecting data points that are representative of their associated cluster, which can be used to facilitate the search against a new item.

$$d(A,B) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}} \tag{6.28}$$

$$d_M(A,B) = \sum_{i=1}^{n} |x_i - y_i| \tag{6.29}$$

$$d_E(A,B) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{6.30}$$

Gaussian Naive Bayes is a variant of Naive Bayes that follows a Gaussian normal distribution and supports continuous data. Naive Bayes are a group of supervised machine learning classification algorithms based on the Bayes theorem defined in (6.31), where $P(A)$ is the probability of A occurring, $P(B)$ is the probability of B occurring, $P(A|B)$ is the probability of A given B, $P(B|A)$ is the probability of B given A and $P(A \cap B)$ is the probability of both A and B occurring. It is a simple classification technique but has high functionality, especially when the inputs' dimensionality is high. These Naive Bayes classifiers assume that a particular feature's value is independent of the value of any other feature. This concept indicates it may not be suitable to the signal classification study in this chapter, as certain features are dependent on others. For example, if the center area is high in the PDF, then the probability of having a low side area is high.

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(A).P(B|A)}{P(B)} \tag{6.31}$$

However, Naive Bayes Classifiers are trained very efficiently in a supervised learning approach and only a small training data size is required to estimate the parameters needed for classification. This concept could be useful for wireless networks where time and resources are limited. As a result, if smaller data sets can provide equivalent accuracy to the other approaches, it would be highly beneficial. Hence, analyzing the Naive Bayes classifier was assumed to be worthwhile. Gaussian Naive Bayes supports continuous-valued features and models, where each one conforms to a Gaussian (normal) distribution. In this case, the model can fit the data by simply finding the mean and standard deviation of the points within each label, y. These two calculations are all that is needed, as these values define the distribution (6.32). Thus, for a test point to be classified, the z-score distance (6.33) between that point and each class-mean is calculated. The input is assigned to the object class that induces the smallest distance (or, equivalently, largest probability from (6.32)).

$$p(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp\Big(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\Big) \tag{6.32}$$

$$z\text{-}score = \frac{(x - \mu_A)}{\sigma_A} \tag{6.33}$$

6.5 Results: Wireless Signal Classification

The features developed for the analysis of the raw I/Q data in Section 6.3 are used to classify received samples into one of six signals/channel types; noise, WiFi (IEEE 802.11ac/n), a WiFi router without internet access (IEEE802.11b/g/n), a Bluetooth advertising channel, CW and ZigBee. A feature-based approach was chosen to provide a relatively low complexity solution with near-optimal performance [67]. As the data from signals operating in the 2.4 GHz ISM RF band can be clustered relativity close-by, supervised machine learning techniques that incur relatively fast optimization and training times (compared to deep learning, see Table 6.12) are chosen. The fundamental approaches chosen include SVMs, Random Forest, k-NNs, Gaussian Naive Bayes and XGBoost [160], whilst a feature-based deep neural network (DNN) is studied to assist the validation of the selected approach. Since an emphasis is applied to developing a single multi-class classifier rather than multiple binary classifiers, most of the work focuses on multi-class classification. This study encompasses a classification problem since the required output is categorical (discrete). In these supervised learning approaches, to build a model that predicts the response Y based on the explanatory variables (features) X, the dataset D is represented by $D = \{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$. To map every input $x \in X$ to a corresponding prediction $y \in Y$ an algorithm is employed to learn the mapping function (f) from the input variable (x) to the output variable (y); that is $\hat{y} = f(x)$. This generated function is the classifier.

For this thesis, a vital characteristic to identify surrounds the designed machine learning model based on data collected in a specific domestic wireless environment under a unique set of channel fluctuations. As described in Chapter 5, the data collection process included a range of power levels and receiver positions. This approach ensured that the data limitations identified in the wired approach (Section 6.2) would not occur in the wireless investigation. As a result, the collected data both did and did not saturate the receiver, as visualized in the calculated PDFs for each signal in Figures 6.8 → 6.11. However, even with this data collection process and insights gained from the simulations and wired investigation, the developed models are specific to a domestic environment's training data. As a result more diverse data may yield improvements, as discussed in Chapter 8.

The proposed machine learning algorithms are adopted to fully validate the suitability of the developed feature set for classifying wireless ISM RF band signals. Even though more modern deep learning approaches are available, the results in this study, through a comparative approach, indicate that classical approaches are still fit for purpose. They incur relatively fast optimization times, compared to deep learning, have lower complexity and are shown to generalize well to new data. Thus, this investigation's overall strategy of data collection, feature extraction and model development can be applied to various deployed communication environments, where extensive time and computational resources are rare [95]. As a result, this section provides results identifying the value in the developed feature set and how well known and extensively studied machine learning approaches continue to be effective, especially when the desired deployments require off-line analysis to be as brief as possible. This process develops a low-order wireless signal classification methodology based exclusively on the analysis of received raw I/Q samples.

The collected data was split into training and testing datasets, where test data included an unseen testing dataset collected after the original. This approach allowed an estimate of the error rate on new data instances, known as the generalization error (or out-of-sample error), to be found. This generalization error value outlines how the designed model would perform on instances it never encountered before. It is used as the primary verification of model suitability to the problem and, potentially, real-world operation. For this discussion, it is worth noting that received wireless signals are, typically, unique on reception. The wireless channel varies over time and signals, generally, interact with other signals and obstacles differently on each signal transmission. The concept of signals incurring different interactions once transmitted is acceptable and data instances can be perceived as unique. Hence, testing data is relatively unseen to the training instances used in model development. However, as one packet of data can be divided into multiple instances and used in either training and/or testing, some test points may not be unseen. However, the use of test data collected after the original dataset ensures the use of unseen data.

The conventional data split between training and testing data of $80\% : 20\%$ was applied, respectively. This data split was chosen as the dataset was of a reasonable size for experimentation, meaning that tests could be completed in a suitable time-frame for model development and, so, a validation set was not required. Additionally, an “unseen” test dataset was collected to investigate how the developed model generalizes to entirely new data that was not used during model training. The data for each signal type was randomly split in the ratio of 80%:20% by initially creating a column array of unique random positive integers in the range of the number of individual signal instances. This array of random numbers was then split, using the $80 : 20$ ratio, and used to select the training and testing instances. The sampling bias of the data split procedure was examined by calculating the percentage of each classification type in the datasets on the scale of $0.0 \rightarrow 1.0$. The results are provided in Table 6.8, where it is clear that the percentage split is maintained across all developed datasets. Approximately 400 instances were added to each test signal type as unseen data. However, there is a slight variation, leading to a small deviation in the sampling for test data, including the additional unseen data.

Analysis initially focuses on Matlab machine learning functions, specifically the

Table 6.8: Sampling Bias Comparison in Developed Wireless Datasets

Signal	Model Data	Training Data	Testing Data	Testing + Unseen Data
Noise	0.0938	0.0938	0.0938	0.1335
WiFi	0.1511	0.1511	0.1511	0.1601
Router	0.1077	0.1077	0.1077	0.1405
Bluetooth Advertising	0.1099	0.11	0.1097	0.1372
CW	0.1805	0.1805	0.1805	0.1669
ZigBee	0.3569	0.3569	0.3571	0.2617

"fitcsvm" and both the "TreeBagger" and "fitcensemble" functions for SVM and Random Forest models, respectively. This approach was taken due to results obtained from the simulation study in Chapter 4 and the wired investigation in Section 6.2. Adaptive boosting is also investigated, based on an optimization study, and is applied in the "fitcensemble" case. Matlab operated as a continuation from the feature extraction environment, granted easy access to the I/Q data and used the same functions as the simulation study in Chapter 4.

Using the outcomes from applying SVMs to simulated and wired data, multiple binary classifiers are required to classify the different signal types. This procedure means potentially six classifiers using the one-versus-all method, or fifteen using the one-versus-one method. An additional logic decision stage would also be required, based on the SVM outputs [107]. Typically, multiple models would increase the computational load and be time-consuming, leading to potential implementation problems on resource-constrained WSN/IoT edge devices. Hence, a single multi-class classifier was the preferred approach. As a result, one SVM ZigBee versus all (other investigated signals) binary classifier was developed to enable a subsequent performance comparison to the multi-class approach. This comparison focuses on the requirement of only detecting legitimate WSN signals. As the RTSA-confirmed that SDR-transmitted ZigBee signals sufficiently matched XBee transmitted signals (see Fig. 6.1), both were classified here as ZigBee. This combination provided a greater ZigBee operating range, as the SDR transmit power was controllable, whereas the XBee's had limited control. This commercial ZigBee and SDR-transmitted ZigBee concept will be further discussed in terms of legitimate receivers as part of the interference detection study in Chapter 7. However, as this examination focuses on classifying received I/Q samples as a specific signal type, classifying all ZigBee transmissions in a single group is acceptable, given their similar spectral images.

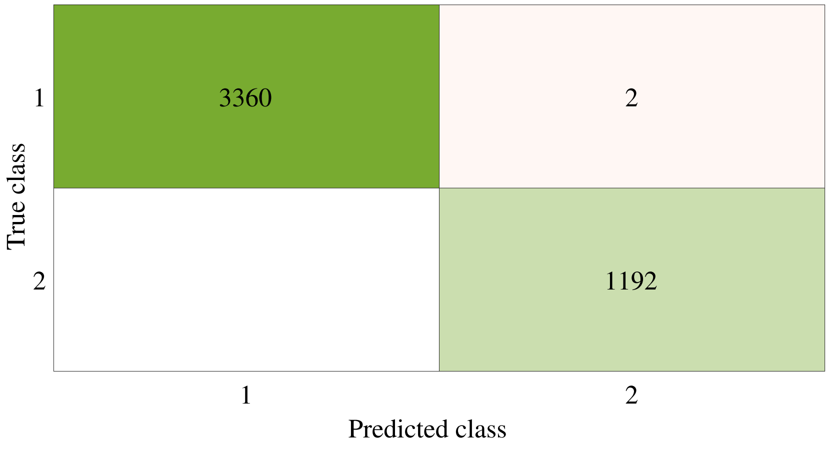
The aim of the binary classifier was to identify received samples as either being ZigBee or some other 2.4 → 2.5 GHz ISM RF band signal. In essence, this is taking the WSN signal focused binary classifiers from the one-versus-all classification method.

Table 6.9: SVM Signal Classification Generalization Error Results: ZigBee versus All

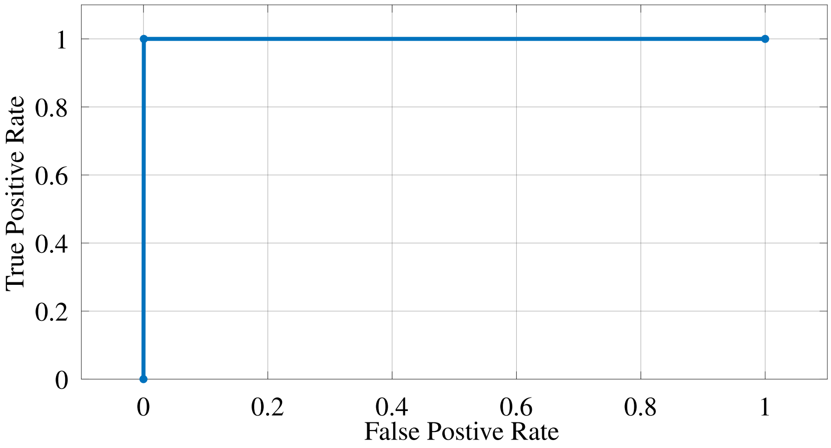
Kernel	Training Time (ms)	Average Prediction Time (ms)	Test Data Error (%)	10 Fold Cross Validation Error (%)	AUC
28 Features and All Test Data					
Linear	520	1.85	0.219	0.173	0.9969
Gaussian RBF	383	1.85	0.176	0.025	0.9975
Polynomial (3rd Order)	355	1.83	0.132	0.039	0.9985
14 Features and All Test Data					
Linear	508	1.84	0.329	0.4863	0.9956
Gaussian RBF	534	1.84	0.066	0.00	0.9993
Polynomial (3rd Order)	384	1.81	0.044	0.00	0.9997

This approach would be suitable for potential applications of this methodology, as discussed in the development of the interference diagnostic framework in Chapter 7. The results of this binary classifier development are shown in Table 6.9, where three different kernels were investigated using the “fitcsvm” function. The results focus on using all available testing data, including the additional “unseen” data, and the datasets containing both the original 28 and optimized 14 features were applied. Table 6.9 show that the reduction in features enhanced the model’s performance, which reiterates the idea that only the most useful features should be retained during feature optimization. The kernel with the highest performance was the third order polynomial, which provided an area under the curve of approximately 0.9997 when using the 14 element feature set, resulting in a near optimum ROC curve for a binary classifier. The confusion matrix and ROC for this kernel are provided in Fig. 6.18 (a) and Fig. 6.18 (b), respectively. These results indicate that the errors were false positives, where the ZigBee signals are positive results and the other signals indicate a negative result. These results indicate that the SVM can identify ZigBee signals with a relatively low generalization error and, so, this developed methodology could be used to identify ZigBee signals in new environments. This is advantageous as the methodology discussed in this study could then be applied to multiple operating environments. The SVM performance will be compared to identifying the same ZigBee signals in the subsequent multi-class approaches.

The requirement for developing multiple binary classifiers, either using the one-versus-all or one-versus-one method and the associated complexity, which increases as more signal types are added to the dataset, is undesirable. A single multi-class model is targeted, based on the previous success using the Random Forest decision-tree approach in both the wired and simulation investigations. Generally, if other signals are



(a)



(b)

Figure 6.18: In depth results for the selected 3rd order polynomial kernel for the ZigBee vs. all SVM classifier. (a) The Confusion Matrix for Testing Data, where the labels correspond to (1) Other Signals and (2) ZigBee, (b) The associated ROC curve showing near optimal classifier performance.

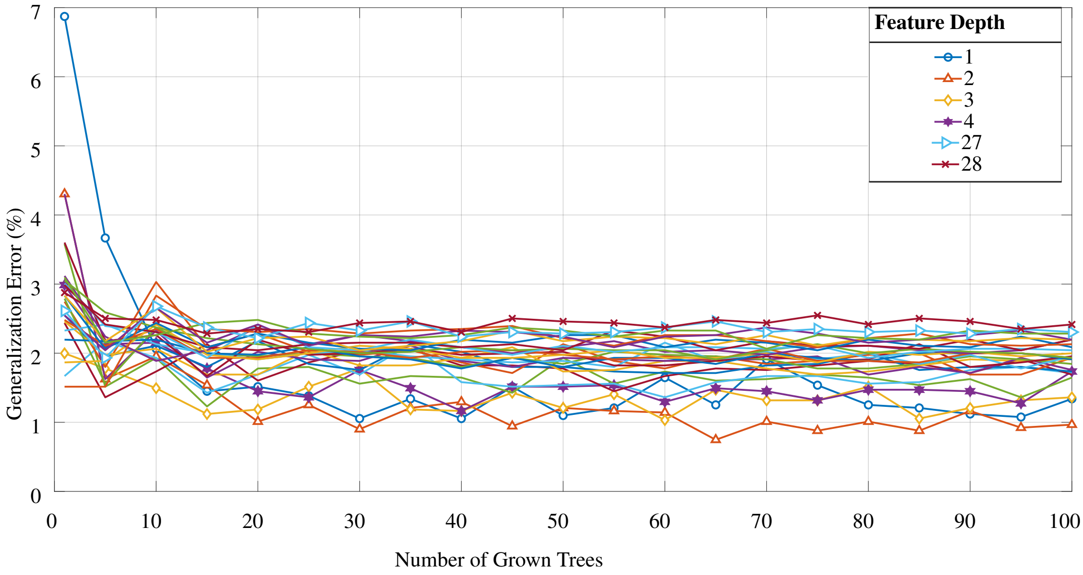


Figure 6.19: The generalization error results for the developed Random Forest approaches for a range of grown trees and feature depths (only certain depths marked as most follow a similar trend). The “TreeBagger” function was used along with the original set of 28 features, where the error stabilizes between 0.75-2.55 % as the number of trees increases.

discovered and require classification, additional samples can be included in the dataset and the multi-class model retrained. A similar process is available for expanding the developed model to include the 2.4 GHz ISM RF band signals operating in different environments. In comparison, the binary classification would require additional models to be developed and the associated logic decision approach to be updated.

Taking the single multi-class model approach, the Random Forest [23] classifier was chosen for the initial examination. The Matlab “TreeBagger” and “fitcensemble” functions were utilized to show that performance is not limited to a specific Matlab approach. These approaches are analyzed across a range of feature depths and number of decision trees, focusing on the generalization error, as available test instances were predominantly comprised of unseen data. These iterations implemented a feature depth range of either [1:28] or [1:14], depending on the feature set in use, and investigated the number of decision trees in the range [1,5,10,15 ..., 100]. The random seed was set to the same value at the start of each interaction, for reproducibility.

The primary generalization error results for these Random Forest approaches are provided in Figures. 6.19 to 6.22, where the total available testing data was used in each case (“TreeBagger” and “fitcensemble”) for the original 28 features and the optimized set of 14 features. In each model, the achievable generalization error was less than 2.5% and, in most cases, the smaller the feature depth, the smaller the error. The training and average analysis (prediction) times were also investigated. It was determined that both the training and average prediction times increase as the number of grown decision-trees increases, regardless of the feature depth. These timing trends are maintained across all of the investigated approaches and show a trade-off between reducing the error and minimizing the prediction and training times. Typically, the training time increases with the number of features to consider at each decision node when looking for the best split. These trends are expected for the Random Forest approach. As the training can be completed off-line on a resource heavy device, the main focus is on the average prediction time. Minimizing the prediction time is critical, as by the time data arrives at the data center, there is the potential that the data center’s control response will be obsolete on return to the edge device. As a result, for independent edge device operation, real-time decisions are essential for optimal performance.

The Random Forest results indicate that certain features contain more distinctive characteristics and, so, will provide more useful information. This concept is seen as the smaller feature depths incur a lower error rate. However, the optimized set of 14 features has, even if it incurs a slightly reduced performance compared to the 28 element feature set, the best potential to enable the developed methodology to generalize and be useful in new operating environments. This hypothesis is based on not selecting so few features that the models become overfitted to the available data. Hence, keeping these 14 features as the optimal feature set and using suitable optimization and

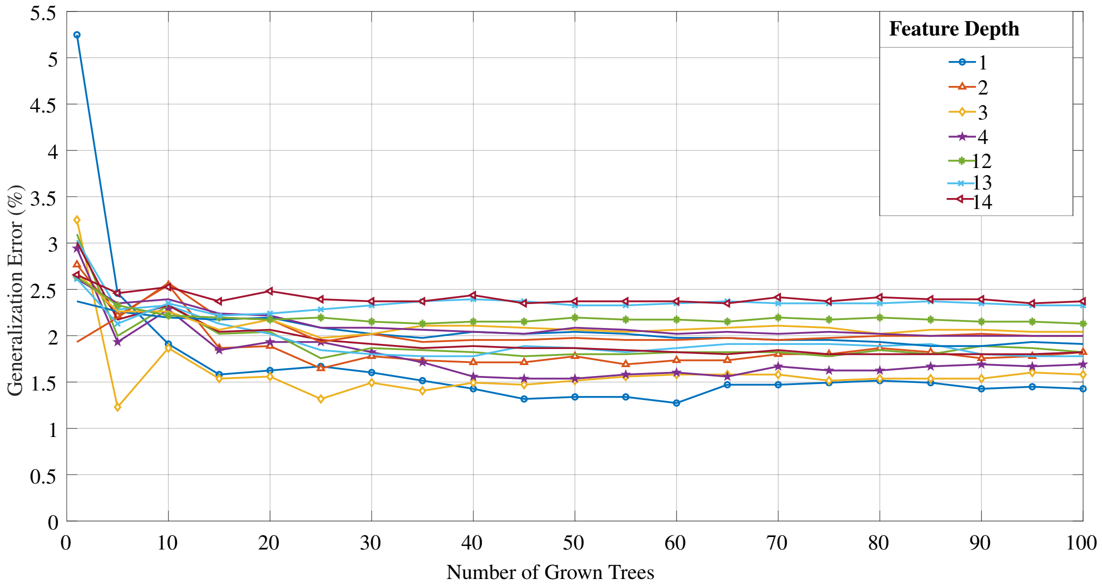


Figure 6.20: The generalization error results for the developed Random Forest approaches for a range of grown trees and feature depths (only certain depths marked as most follow a similar trend). The "TreeBagger" function was used along with the optimized set of 14 features, where the error stabilizes between 1.25-2.5 % as the number of trees increases.

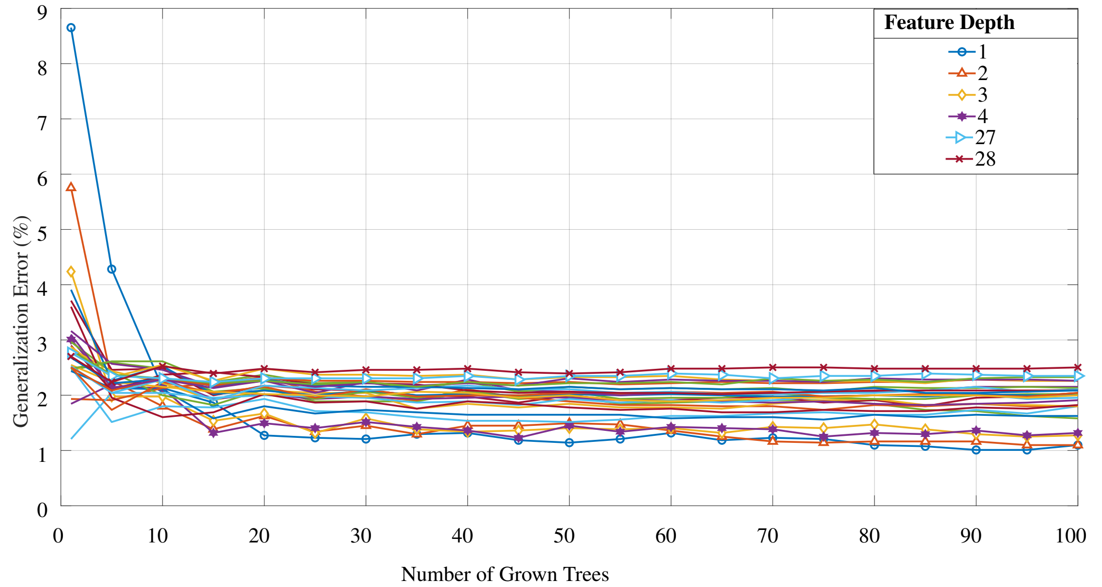


Figure 6.21: The generalization error results for the developed Random Forest approaches for a range of grown trees and feature depths (only certain depths marked as most follow a similar trend). The "fitcensemble" function was used along with the original set of 28 features, where the error stabilizes between 1-2.5 % as the number of trees increases.

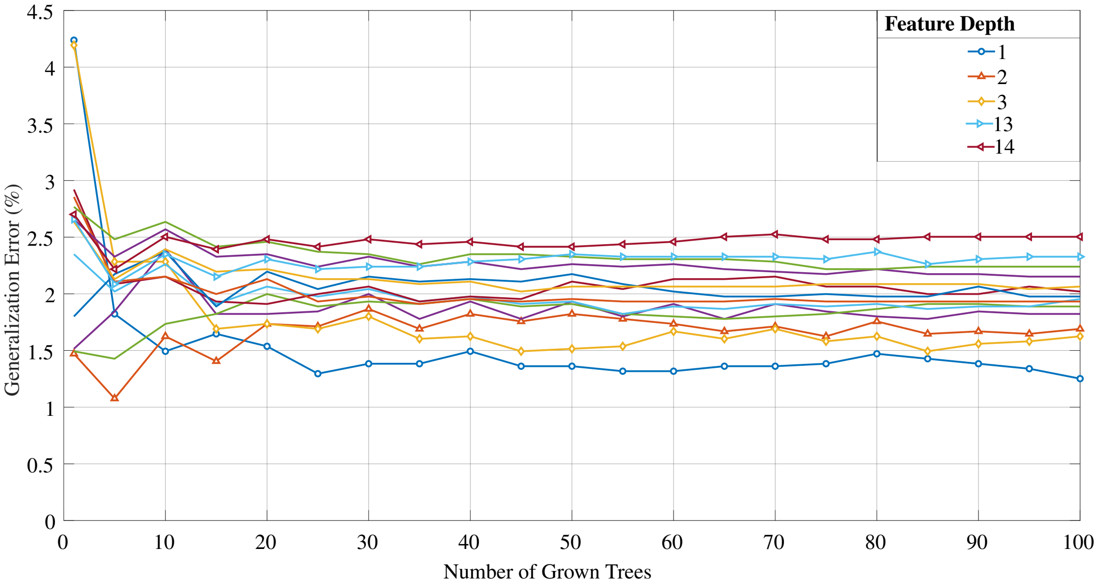


Figure 6.22: The generalization error results for the developed Random Forest approaches for a range of grown trees and feature depths (only certain depths marked as most follow a similar trend). The “fitcensemble” function was used along with the optimized set of 14 features, where the error stabilizes between 1.5-2.5 % as the number of trees increases.

performance-enhancing model development techniques such as, for example, boosting, is the implemented approach to allow the overall designed methodology to adapt to new signals and reduce the possibility of overfitting.

Using the “HyperparameterOptimizationOptions” input to the fitcensemble function, the available training data was used to obtain an optimized set of hyperparameters for this methodology. The results indicated using the “AdaBoostM2” function, an adaptive boosting dependent ensemble algorithm, where the “M2” is a Matlab specification for multi-class operation. Adaptive Boosting, or “AdaBoost” [159], is a specific method for a predictor (decision-tree) to correct its predecessor by focusing on the training instances that the predecessor under fitted, resulting in new predictors concentrating on the hard cases. After each iteration, the misclassified instances gain more weight, while the weight of the correctly classified example is decreased. This process is achieved in AdaBoost by initially training a base classifier (a Decision Tree, for example) and making predictions on the training set. The algorithm then increases the relative weight of misclassified training instances and trains a second classifier, using the updated weights, and again makes predictions on the training set, updates the instance weights, and so on. Essentially, the approach alters the distribution of the training dataset to increase weights on sample observations that are difficult to classify. Similar to Random Forest, the final prediction is based on a majority voting scheme, however, for AdaBoost, the weak learner’s predictions are weighted by their individual accuracy. As a result, trees which perform more accurate classification have higher weights and, consequently, have a higher contribution to the final prediction.

Here, the optimal “AdaBoostM2” used a learning rate of 0.61992 and a minimum

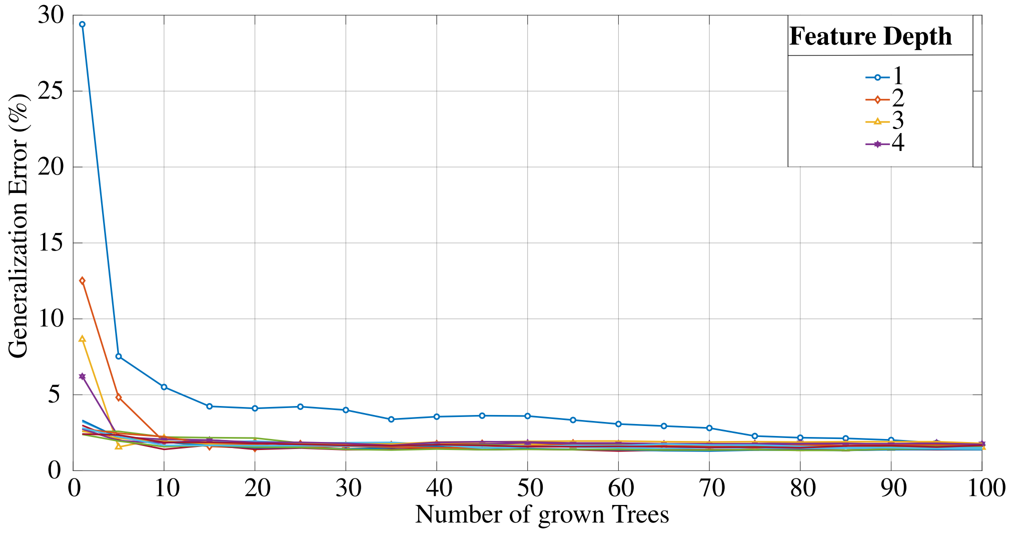


Figure 6.23: The generalization error results for the AdaBoostM2 ensemble method using the optimized set of 14 features with a minimum leaf size of 3 and a learning rate of 0.61992, for a range of decision tree learners and feature depths. The error stabilizes between 1.2-1.9 % and the unique feature depth trends are marked.

leaf size of 3. This setup was investigated for different numbers of decision tree learners in the range [1,5,10,15 ..., 100] and feature depths ([1:14]) available at the decision points. The results are visualized in Fig. 6.23, where the lowest error (1.2956 %) occurred for the combination of sixty learners and a feature depth of seven and seventy learners and a feature depth of eight. These results indicate the importance of using the 14 element feature set and not reducing the features based on the Random Forest results using basic hyper-parameter optimization. The sixty learners incurred a training time of 749.1 *ms* and an average prediction time of 20.94 *ms*. In contrast, the seventy learners incurred a training time of 993.58 *ms* and an average prediction time of 24.46 *ms*. As the training and analysis times were shorter for the smaller number of learners, it was chosen as the optimal “AdaBoostM2” Matlab combination. A more focused set of results was obtained, specifically the confusion matrix for this combination is depicted in Fig. 6.24. The majority of the errors occur when classifying between the two different IEEE802.11 signals. Thus, the model has generalized well to signals from different sources and has high accuracy in classifying the wireless signals being received and generalizes well to “unseen” data. This result further validates the feature set and methodology that has been developed in this study. Additionally, as this approach focuses on the hard instances and aims to improve on predecessors, it has a higher likelihood of aiding this methodology in adapting to different wireless operating environments. The results of this initial Matlab optimization study indicate that ensemble methods are suitable to this classification problem and dependent methods are optimal.

The results of this Matlab optimization investigation demonstrated the importance of analyzing the available parameters of the machine learning model being implemented.

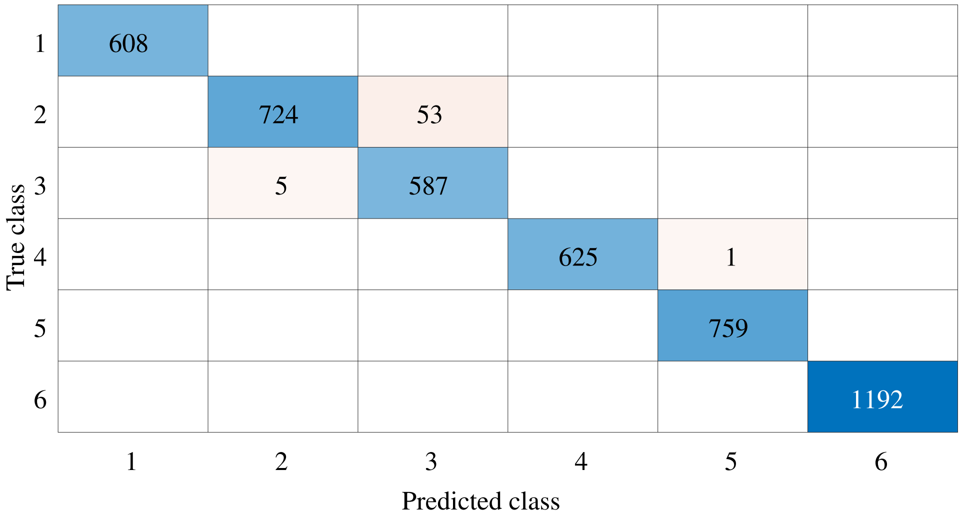


Figure 6.24: Confusion Matrix for the designed AdaBoostM2 ensemble method using sixty decision tree learners, a feature depth of seven, a minimum leaf size of three and a learning rate of 0.61992. The predictors are as follows: (1) Noise, (2) WiFi, (3) Router, (4) Bluetooth Advertising, (5) CW and (6) ZigBee.

The process indicated that different approaches could enhance machine learning optimization, and extensive optimization of hyperparameters results in higher performance. This lead to investigating parameter optimization and model development using the leading artificial intelligence/machine learning programming language, Python. The highly adopted libraries need to be used. The following work expands on the Matlab investigation by using the Python3 language and more modern techniques and models on both CPUs and GPUs. For the remainder of this chapter, each machine learning model is trained and tested using available Python3 libraries running on an Intel i7-9700 3 GHz CPU. DNNs are developed using Keras and TensorFlow on an Nvidia GeForce RTX 2060 graphical processing unit (GPU) with 6GB of RAM. Furthermore, this work's main objective is to permit decentralized computation and allow specific embedded devices to identify signals present in their surroundings. This approach is required to lead to methods that increase security by monitoring the wireless channel and adapting to real-time changes, such as frequency hopping when a CW jamming wave is detected. A resource-constrained implementation is required, and to initially investigate this approach, computations are examined on a RaspberryPi embedded device, as described in Chapter 5. For this Raspberry Pi operation, Python3 is required as the Raspberry Pi can execute machine learning using Python3. This Python3 implementation study focuses on the optimized 14 features and the same training and testing datasets as the Matlab approach.

At first, the "scikit-learn" machine learning library was utilized for its "RandomForestClassifier" function. A summary of the results of this approach are shown in Table 6.10 and in Fig. 6.25, where the Raspberry Pi results (Fig. 6.25 (a)) are compared with an implementation on the same Desktop PC that produced the Matlab results (Fig.

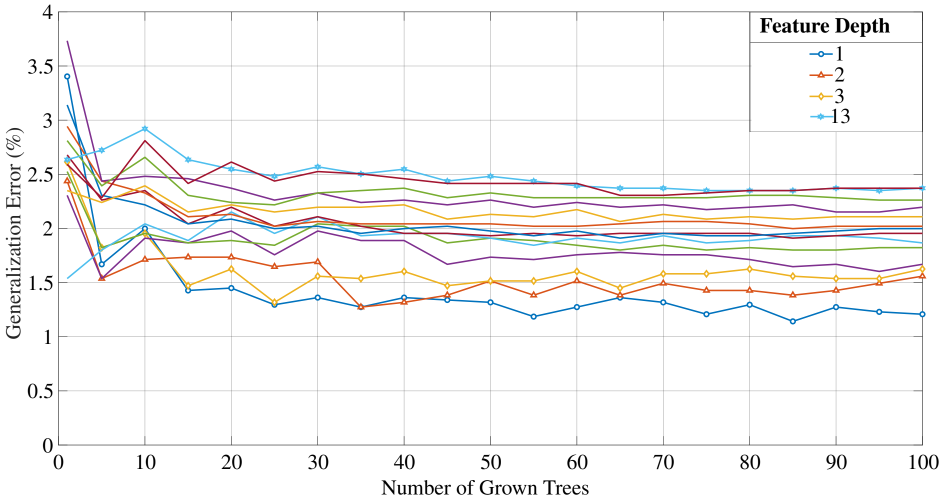
Table 6.10: Random Forest Classification Results: Device Comparison for Wireless Data

Device	Predictor Depth	No. of Trees	Training Time(ms)	Avg. Prediction Time(ms)	Test Data Error (%)
Raspberry Pi 3-B	1	85	1564	0.0679	1.142
Specifications:	1 GB of RAM and Quad-core Broadcom BCM2837B0, Cortex-A53 CPU @ 1.4 GHz				
PC	1	85	139.6	0.005	1.142
Specifications:	Dell XPS8930, 16 GB of RAM and an Intel i7-9700 CPU @ 3 GHz, 8 Cores				

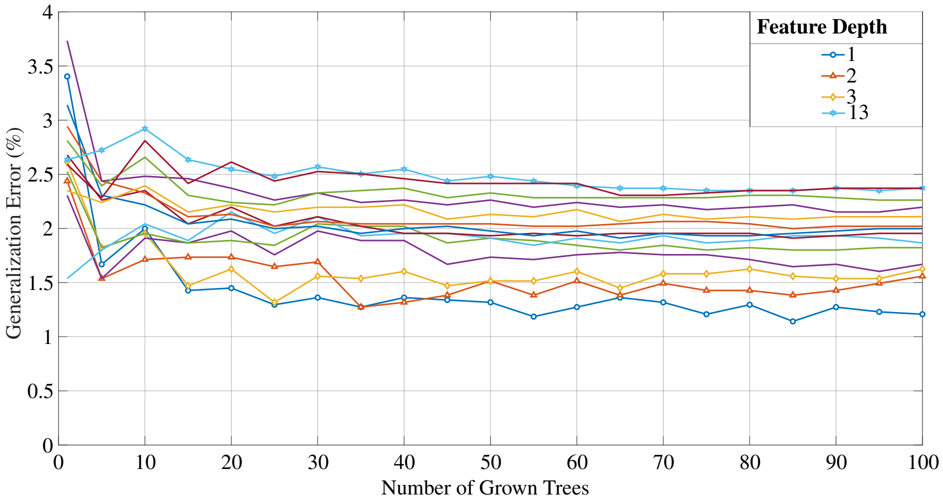
6.25 (b)). A Dell XPS8930 was used as the PC to compute the results, where 16 GB of RAM and an Intel i7 processor (3 GHz) were available, while the Raspberry Pi has 1 GB of Ram and a quad-core Broadcom Arm Cortex A53 processor (1.4 GHz). These results show that the Python3 approach aligns with the Matlab results while providing a faster implementation time. The optimal Random Forest approach here, when using the Desktop PC for consistency, incurred a training time of 139.6 *ms* and an average prediction time of 0.005 *ms*, compared to the Matlab AdaBoostM2 approach, which incurred a training time of 749.1 *ms* and an average prediction time of 20.94 *ms*. These results indicate that this investigation has provided sufficient evidence that the Raspberry Pi matches the PC developed models' accuracy but incurs training and average prediction time penalties.

As a result, further investigations can be implemented knowing that the Raspberry Pi will reach equivalent performance levels. Generally, training times are not a concern as it can be rectified by training and optimizing the Python models on a much more advanced machine. Only the optimally trained model must be uploaded and used on the lightweight Raspberry Pi embedded device. This process can be achieved using several model saving libraries, such as, for example, "pickle". However, the prediction time and required computational resources are important factors. The Raspberry Pi implementation did achieve a much faster average prediction time of 0.0679 *ms* compared to the AdaBoostM2's 20.94 *ms*. Hence, the Raspberry Pi implementation is sufficient to demonstrate the key findings that average prediction time penalties will exist, but other costs can be mitigated through optimization and training on a Desktop PC, for example.

Given the need to optimize on a Desktop PC and the associated potential performance improvements, the Python3 "RandomForestClassifier" function was analyzed further by investigating additional metrics using a specific random state for reproducibility. These parameters include 1) The number of decision trees, 2) The maximum number of predictors, 3) The maximum tree depth, 4) The maximum number of samples, 5) The minimum number of samples required to split an internal node and 6) The minimum number of samples required to be at a leaf node. In total, $197,568$ iterations were completed and the optimum generalization error was 1.098%, which shows the Python3



(a)



(b)

Figure 6.25: The generalization error for the designed scikit-learn Python Random Forest method using variable numbers of grown decision trees, variable feature depths and an unspecified maximum depth for each tree, where the error stabilizes between 1.14% and 2.57 %. Specific feature depth curves are annotated to provide detail of how the error changes over the range of available feature depths. The same python scripts were implemented on (a) A Raspberry Pi 3 Model B and (b) A Desktop PC.

benefits of delving deeper into the model development, as it outperforms the original Python model. This process indicated the need for an in-depth parameter optimization, which contrasts with the initial Matlab investigations and the simulations in Chapter 4. However, due to the simulations' lack of hardware limitations, the lack of in-depth optimization is not a significant factor. Here, this initial broad hyperparameter optimization demonstrates that it can be critical for the wireless approach.

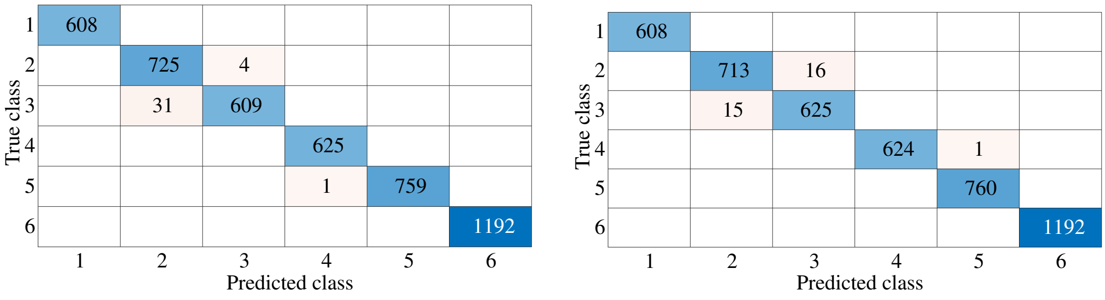
To further aid in minimizing the effect of overfitting and to heavily investigate a dependent ensemble approach, the widely adopted "XGBoost"[160] algorithm was investigated across typical parameters that enable the training of a decision tree boosted model. This algorithm stands for "eXtreme Gradient Boosting" and, in general, produces strong learners based on the correction of errors produced from weak learners. The model is explained in detail in Section 6.4.2. Based on the improvement seen in the

Matlab investigation when applying AdaBoost, which increased model performance, boosting (dependent ensemble approaches) is seen as an appropriate method for creating an accurate and strong classifier from a set of weak classifiers. XGBoost, which applies a gradient boosting approach, was chosen for this purpose and to provide a comparison to the AdaBoost method. The main idea behind AdaBoost lies in giving more focus, quantified by an assigned weight, to instances that are harder to classify. In contrast, as explained in Section 6.4.2, gradient boosting identifies misclassification from the large residual obtained on the previous iteration. Furthermore, as a result of the success of using the Random Forest ensemble method in the extensive simulation study, and the increased performance indicated in the Matlab AdaBoost approach, applying the boosted decision tree model of XGBoost was predicted to achieve high performance. For this investigation, several parameters were optimized, including 1) The number of decision-trees, 2) The applied learning rate, 3) The maximum tree depth, 4) The minimum child weight, 5) The percentage of available data used per decision-tree, 6) The applied booster algorithm, 7) The sub-sampling percentage of the feature columns and 8) The applied minimum loss reduction when determining if a further partition is required.

The optimized XGBoost approach produced the highest accuracy (and lowest generalization error of 0.7905%) in this study before developing deep neural networks. The confusion matrix for the designed approach is seen in Fig. 6.26 (a), where the majority of errors occur between classifications of different IEEE802.11 protocols. The final algorithm contained five trees, a learning rate of 0.8, a maximum tree depth of 10, a minimum child weight of 2, used 95% of available data per tree, used the "gbtree" booster, sub-sampled 75% of the feature columns and applied a minimum loss reduction of 0.5 when determining if a further partition is required. This produced a smaller error, when compared to the AdaBoost approach in the Matlab investigation, using the same data. By using XGBoost, an error reduction of 38.98% was achieved (1.2956% → 0.7905%), where the improvement occurred in the classification between the separate IEEE802.11 signals. This result indicates that gradient boosting is more beneficial for this classification problem and that using the decision tree approach is applicable.

To further validate choosing the XGBoost approach for this multi-class classification problem, other distinct machine learning approaches were briefly examined. Each model is investigated across a range of suitable parameters, specific to each machine learning approach. Analyzed algorithms included the "scikit-learn" neighbors-based classification functions, namely "KNeighborsClassifier", "RadiusNeighborsClassifier" and "NearestCentroid", and the "scikit-learn" Gaussian Naive Bayes ("GaussianNB") function.

For both the "KNeighborsClassifier" and "RadiusNeighborsClassifier" investigations, the examined algorithms were "ball_tree", "kd_tree", and "brute", the associated



(a)

(b)

Figure 6.26: Confusion Matrix for (a) Designed XGBoost method, which produced a generalization error of 0.7905% and (b) Designed DNN, which produced a generalization error of 0.7027%. The predictors are as follows: (1) Noise, (2) WiFi, (3) Router, (4) Bluetooth Advertising, (5) CW, and (6) ZigBee.

leaf size, where applicable, was an element of the set [5, 10, 20, 30, 40, 60, 80, 100], the weight function used in prediction was either "uniform" or "distance" and the exponent for the Minkowski metric was either 1 (equivalent to using Manhattan distance) or 2 (equivalent to using Euclidean distance). For the weights, "uniform" results in all points in each neighborhood being weighted equally and "distance" means points are weighted by the inverse of their distance, meaning closer neighbors of a query point will have a greater influence than neighbors which are further away. For the "KNeighborsClassifier" the number of neighbors was an element of the set [1, 2, 5, 10, 20, 40, 60, 80, 100]. For the "RadiusNeighborsClassifier" approach, the range of parameter space to use was an element of the set [85, 90, 95, 100, 110, 120, 140, 150], where 85 was the lowest radius that ensured each instance had at least one neighbor for each of the other parameters setups. The "NearestCentroid" approach was investigated, where each class is represented by its centroid, with test samples assigned to the class with the nearest centroid. The examined metric for calculating the distance between instances in a features array included: $\{cityblock, cosine, euclidean, haversine, l1, l2, manhattan, nan - euclidean\}$, which were all of the available distance metrics provided in the "metrics.pairwise.distance_metrics()" section of the "sklearn" library. For this approach, the centroids for the samples corresponding to each class are the points from which the sum of the distances (according to the metric applied) of all samples that belong to that particular class are minimized. If the "manhattan" metric is provided, this centroid is the median and for all other metrics, the centroid is set to be the mean.

The optimal parameters for the "KNeighborsClassifier", based on the parameter sets above, corresponded to using twenty neighbors, any of the three algorithms, a leaf size of 5, the uniform weight function and the Manhattan distance. This optimized parameter set produced an error of 3.4695%. However, as an odd number of neighbors negates the condition of multiple classes attaining the same number of maximum votes, this optimization of 20 neighbors allowed for further investigation. The odd numbers around twenty were investigated, while maintaining all other optimal parameters. The results indicated that seventeen neighbors was the most optimal result and produced an error of 3.38%, which indicates better performance compared to the even number of neighbors. For the "RadiusNeighborsClassifier" approach, the optimal parameters are a radius of eighty-five, any of the three algorithms, a leaf size of 5, the distance weight function and the Manhattan distance. These parameters resulted in a calculated generalization error of 3.6232%. The "GaussianNB" was briefly investigated across an array of values for the portion of the largest variance of all features that is added to variances for calculation stability. The examined values were [$1e^{-15}$, $1e^{-14}$,...,$1e^{-10}$, $2e^{-10}$, $5e^{-10}$, $1e^{-9}$, $2e^{-9}$, $5e^{-9}$, $1e^{-8}$, $5e^{-8}$, $2e^{-8}$, $1e^{-7}$, $1e^{-6}$, $1e^{-5}$]. The optimal approach applied a portion of $1e^{-10}$ or lower, and produced an error of 5.907%.

A valuable insight is gained from comparing the XGBoost approach to the SVM binary classifier. Compared to the ZigBee-versus-all case, this multi-class classifier achieved similar, if not better, performance than the SVM. In fact, all ZigBee signals were correctly classified in each case, but no signals were misclassified as ZigBee when XGBoost was applied (Fig. 6.26 (a)). The SVM classifier misclassified two instances as ZigBee (Fig. 6.18 (a)). However, the multi-class approach can be assisted by a binary classifier. The majority of the XGBoost errors occurred when classifying between the different IEEE802.11 signals. A SVM for this one-versus-one case was developed to examine if a higher performance was achievable. The designed SVM produced 53 errors (3.8714%) when using the same third-order polynomial kernel as the ZigBee case. By adopting the RBF kernel, this error is reduced to 23 mis-classifications (or 1.68%). The SVM obtained a 34% error reduction in the classification of the IEEE802.11 signals compared to XGBoost. This comparison discovered that the optimal approach for using these fundamental algorithms and the developed feature set was in a pipeline approach. The XGBoost method is optimal for all but the classification between IEEE802.11 signals. Hence, if an IEEE802.11 signal is detected for maximal performance, the data instance is passed to a separate binary SVM classifier focused on IEEE802.11 signals. This XGBoost/SVM approach achieves optimal performance (0.527% error) for the minimum amount of designed classifiers that generalize well to unseen data when resource management is key.

However, by focusing entirely on a resource-constrained operation, the most optimal approach may be neglected. Thus, deep learning was investigated. Applying deep learning examined how more traditional and less complex techniques compared to a fully connected neural network in terms of the generalization error, computation load, training times, parameter optimization and hardware resources. The DNN was optimized, using TensforFlow and the "KerasClassifier" function, over the number of epochs, batch size, optimizer, number of hidden layers and neurons. When selecting the optimal parameters, consideration was given to the training time, average prediction time and required resources. The batch size was examined using the values $[5,10,20,40]$, the

Table 6.11: 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	6 neurons	softmax

number of epochs were $[100, 200, 500, 1000, 2000]$, the investigated optimizers were [adam, rmsprop, Adadelta, Adagrad, Adamax], the initial hidden layer sizes were [50, 100, 200, 300] and the number of hidden layers were [1, 3, 5, 7, 9]. The hidden layers were calculated by decreasing the number of neurons in each layer based on the ratio of the number of neurons in the first hidden layer to the required number of hidden layers. The input layer was fourteen and the output layer was six, which correspond to the number of input features and output classes, respectively. The process was implemented using the Python “GridSearchCV” function and 5-fold cross-validation, where the K-Fold accuracy was the chosen metric.

The Adamax optimizer was the optimal approach in every case and the developed optimized DNN architecture for this study is provided in Table 6.11 and consists of three hidden layers, the “Adamax” optimizer, 2000 epochs and a batch size of five. A small performance improvement (approximately 0.128% when using GridSearchCV and 5-Fold cross-validation) is gained by using a single hidden layer of 200 neurons using a batch size of 10 and 2500 epochs. However, this increases the prediction and training times, and the performance increase was not sufficient to offset the timing trade-offs. The chosen DNN requires a training time of 1937.04s and achieves an average prediction time of 23.227 ms. These results were generated using Keras on an NVIDIA GeForce RTX 2060 with 6GB of RAM. In contrast, the XGBoost and SVM approaches’ average prediction times are 0.05 ms and 1.242 ms, respectively, while the training times are 78.1 ms and 44.44 ms, respectively. Hence, even when used sequentially, the XGBoost/SVM approach provides better performance and quicker prediction times.

Table 6.12 provides a summary of all the classification results using the various approaches, where it is evident that the XGBoost approach achieves the lowest generalization error for the traditional supervised machine learning approaches, while the DNN is the optimal approach. This is visualized in the associated confusion matrices for the developed XGBoost approach in Fig. 6.26 (a) and the DNN in Fig. 6.26 (b). However, as discussed previously, a combination of XGBoost and SVM outperforms the DNN in terms of achieved accuracy, training time and average prediction time. Notably, if a DNN/SVM approach is applied in a similar manner to the XGBoost/SVM method, the results will be equivalent. This is due to the SVM being the optimal ap-

Table 6.12: Selection Evidence - Supervised Approaches

Algorithm	Lowest Achieved Error (%)	Training Time	Iterations
XGBoost/SVM	0.527	122.54 ms	n/a
DNN	0.7027	1937.04 s	400
XGBoost	0.7905	78.1 ms	400000
Random Forest	1.098	265.57 ms	197568
AdaBoost (Matlab)	1.2956	749.1 ms	n/a
K Nearest Neighbors (17 Neighbors)	3.3816	26.93 ms	n/a
K Nearest Neighbors (20 Neighbors)	3.4695	15.6 ms	1152
K Nearest Neighbors (within Radius)	3.6232	35.934 ms	7166
Gaussian Naive Bayes	5.907	3 ms	17
Nearest Centroid	9.222	2 ms	8

proach but limited to reducing the number of errors in IEEE802.11 classification to 23, which is lower than the XGBoost and DNN models. The main finding surrounds the training and prediction times, which are much lower for the supervised traditional XGBoost/SVM approach. As a result, the parameter optimization for the non-deep learning approaches is orders of magnitudes faster, which pairs well with deployed wireless communication systems where extensive computational resources and time are rarely found [95]. As this thesis is focused on developing a methodology rather than a specific model, faster optimization times are critical for applying the developed methodology in various operating environments. The results indicating the equivalent performance to a feature-based DNN further benefits this methodology. Additionally, the desired deployment of edge devices requires low complexity and fast optimization times for new environments. These requirements validate the selection of the developed XGBoost/SVM machine learning approach. Particularly as the XGBoost/SVM design achieves equivalent accuracy and generalization error results as the developed fully connected neural network, for a small fraction of the computation requirements and in a vastly reduced timescale. This concept of optimization times will be discussed further in Chapter 7 when developing the interference diagnostic framework.

Furthermore, the implementation results from the Python experimentation on a Raspberry Pi embedded device indicate that the same prediction accuracy can be achieved on a much more lightweight device, but the training and prediction times increase. This increase in training and prediction times would be exceedingly more considerable for current deep learning approaches. However, the increased training time is generally not a concern as it can be rectified by simply training and optimizing the model on a much more advanced machine. Only the optimally trained model is required to be uploaded and used on the lightweight embedded device. The Raspberry Pi is suitable for this evaluation, 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. The Raspberry Pi implementation approach is studied further in Chapter 7.

6.6 Discussion

This study of live wireless signals in a typical domestic wireless operating environment, which contained different signal sources, devices, obstacles and service usage, incorporates some limiting factors. The developed two-stage model (XGBoost/SVM) approach would be heightened if data was captured across multiple industrial environments. This data collection would result in more diverse data being available for each signal and a higher probability of models generalizing to new data instances. However, this study's work has provided sufficient evidence to motivate applying the developed methodology to other wireless environments, as the results have generalized to unseen data and achieved high performance. The hardware specifications are also a limiting factor since the developed methodology's performance is linked to the ADC resolution (12-bit) and reference voltage (1.3 V). A higher resolution would allow for received signals to be extracted in greater detail from the channel. The reference voltage, which is the maximum voltage available to the ADC, determines the ADC conversion ceiling for received analog inputs. Essentially, a higher reference voltage allows for higher-powered signals to be received before saturation occurs. However, this study's novel feature set has proven its ability to differentiate between signals when receiver saturation occurs.

Despite these limitations, the wireless approach depicted in this study obtained high performance in classification accuracy and generalizing to unseen data. This study, which employs fundamental feature-based machine learning algorithms, is in contrast to [69], which states that traditional feature-based approaches lack generalization. The results prove 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. Typically, the literature uses high-order statistics [81] and/or cumulants [82, 83, 84] when applying traditional techniques. The achieved generalization error of below 1% is comparable to performance levels of other developed systems focused on the applied modulation scheme. However, unlike image classification, unified datasets for wireless signal classification are, generally, not yet available. So, the authors in [69] compared their system against various other feature-based schemes. At sufficient signal to noise ratios (SNR), the results vary from a classification accuracy of 80% up to almost 100%. In [97], similar results are achieved for a sufficient SNR when classifying wireless signals. Hence, this study's achieved results compete well with the literature and do so by using a low complexity novel feature set and without focusing on modulation schemes, using spectrograms or received signal strength indicator (RSSI) samples. Furthermore, all of the results in this study are based on real over-the-air signals from both SDR and commercial sources received with and without receiver saturation. Overall, despite the trend to use deep learning approaches, as specified in Chapter 3, this study proves that potent data analysis and signal processing permit traditional techniques to still be effective (Table 6.12) when paired with sufficiently descriptive feature sets based on time, frequency and space (PDF).

A use case for this investigation is the development of interference detection systems or edge device decentralized decision making. Edge devices making independent real-time decisions based on the operating environment are key developing points for this study. It can enable devices to react faster to the changing environment than centralized approaches, where packets need to be transmitted and received, increasing latency in the process. An example includes increasing the transmission power for a short period if certain signal types are identified in the channel. In terms of interference detection, the bit-error location analysis in the simulations in Chapter 4 identified two specific operating scenarios for interference detection approaches: 1) When packets are received with bit errors and 2) When no packets can be received during the radio "on" time. This signal classification covers the second option, as transmitters can be blocked from sending legitimate packets. As a result, the interference signal, for example, CW, can be identified and mitigation strategies implemented. Additionally, depending on the received signal type and associated power levels, a signal classification tool may be all that is required to detect interference. Both of these scenarios for using the developed signal classification methodology and low-order features are explored in Chapter 7. The signal classification model is integrated into an overall interference diagnostic framework and is applied to Global Positioning System (GPS) signals, which have received power levels equivalent to typical noise power.

However, in any use case that implements this designed methodology, especially edge devices, the energy usage will be a key performance metric. Thus, it is envisaged that the designed approach would not operate continuously and, instead, only operate on a specific duty cycle or when initiated. This concept is integrated into the overall diagnostic framework, as designed in Chapter 7. The energy usage work is beyond this thesis's scope and is left for future work, as discussed in Chapter 8.

6.7 Conclusion

This chapter exclusively uses raw received I/Q samples to develop a low-order statistical feature set for typical WSN and ISM RF band wireless signal classification. The signals included noise, IEEE802.11, Bluetooth Advertising, CW and ZigBee (IEEE802.15.4) signals transmitted from commercial devices and SDRs, where appropriate. Features were extracted from the calculated PDF of the received samples, statistical analysis of the time domain and from the frequency domain by implementing a FFT. The feature set differs from previous approaches due to the use of low-order statistics and novel uses of the raw data from the calculated FFT and Hjorth parameters. Analog Devices Pluto SDRs and Raspberry Pi embedded devices were exploited as a low-cost yet high performing analysis approach for obtaining the required I/Q samples. The designed novel feature set was validated by extensively investigating the Random Forest independent ensemble approach across various techniques, due to the success seen in the simulations in Chapter 4. Other models were briefly analyzed to validate the choice of ensemble decision-tree method including k-NNs, SVMs and fully connected DNNs. These machine learning concepts were explained in detail before being applied to the feature set. The selection of decision tree approaches was optimized through intensive parameter optimization and by applying boosting techniques, namely AdaBoost and XGBoost, to implement a dependent ensemble framework. Test data included unseen data that was used to examine how the developed models generalized to new data.

The optimal discovered approach for this low-order feature methodology is an XGBoost/SVM combination, which achieved an error of 0.527%. Most errors in the XGBoost model occurred between different IEEE802.11 signals and, so, a separate binary SVM classifier was developed to reduce the error if an IEEE802.11 signal was detected. Developed DNNs provided a comparison between deep learning and supervised traditional approaches. As a single model approach, the developed DNN with three hidden layers was the optimal approach. However, the XGBoost/SVM model achieved higher accuracy than the developed DNN and for reduced computational and time requirements. Notably, a similar DNN/SVM approach would only achieve the same accuracy as the XGBoost/SVM approach as the SVM provides the highest accuracy when classifying the IEEE802.11 signals. This result proved that traditional feature-based approaches are still fit for purpose, particularly for low complexity solutions, and achieve high performance when potent data analysis and novel descriptive feature sets are applied. A Raspberry Pi demonstrated that the designed model achieves the same results on an embedded device. Overall, this study showed that the lowest level of receivable data, I/Q samples, can be leveraged to make higher-level decisions.

This chapter provided an optimal set of 14 features for identifying legitimate signals, including the ZigBee WSN signal. These 14 features are used throughout Chapter 7 as the input features when developing the WSN and GPS interference detection strategies. The various investigated machine learning models identified that a gradient boosting dependent ensemble approach was the optimal decision-tree approach. The developed optimal approaches for the XGBoost and DNN models are key findings from this chapter. Due to these developed models' achieved accuracy and ability to generalize to unseen data, the two approaches are applied in Chapter 7 to expedite the development of an interference diagnostic framework. This chapter's optimal models are applied as the base models to develop the individual WSN machine learning diagnostic tools for interference detection and classification. Towards the end of Chapter 7, the designed signal classification approach will be combined with the models developed in Chapter 7 to establish a WSN edge device diagnostic framework.

Chapter 7

WSN and GPS Interference Detection and Classification

This chapter discuss this thesis' main contribution. The work in this chapter develops the overall interference diagnostic framework for wireless resource-constrained edge devices. The extracted optimal features and associated optimal developed machine learning models in Chapter 6 are leveraged as the foundation for developing the diagnostic framework. The overall framework is validated in separate areas of the RF spectrum, different receivers, distinct value ranges and for several different signal transmissions. As a result, the work in this chapter provides the overall contribution to the field in terms of an interference diagnostic framework. The work in this chapter has been published in part in the following:

- *G. D. O'Mahony, K. G. McCarthy, P. J. Harris and C. C. Murphy, "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", Ad Hoc Networks, vol. 120, p. 102562, 2021, doi: 10.1016/j.adhoc.2021.102562.*

7.1 Introduction

This chapter focuses on developing a low complexity interference diagnostic framework for wireless edge devices by focusing on wireless sensor network (WSN), ZigBee, and global positioning system (GPS) signals. The Matlab Monte Carlo simulations, investigated as the ideal case with no hardware limitations in Chapter 4, initially motivated exclusively using in-phase (I) and quadrature-phase (Q) samples for interference detection and identified signal interactions as the required data type. As discussed in Chapter

5, Analog Pluto software-defined radios (SDRs) collect the required live over-the-air data and transmit matched signal (ZigBee) and continuous-wave (CW) interference in designed hardware ZigBee wireless testbeds. An NESDR SMArTee RTL-SDR collects the required GPS data from both freely available over-the-air satellite transmissions, with and without Pluto SDR wired interference transmissions, at sufficiently low power levels. However, the legitimate signal classification results in Chapter 6 indicated the simulation feature set needed to expand to compensate for signals using similar modulation schemes and receiver saturation conditions. Hence, the developed fourteen low order 2.4-2.5 GHz wireless signal classification features, based on received I/Q samples, and associated optimal models from Chapter 6 are leveraged as the framework's foundation.

As discussed in Chapter 2, the jamming interference attack was studied as WSN and GPS systems are both highly susceptible to jamming. As the deployed WSN protocols and devices are generally publicly available (for interoperability), specific unavoidable security vulnerabilities exist, such as Denial of Service (DoS) attacks, more specifically, jamming. In addition, unintentional and malicious in-band interference is the single most significant threat to GPS applications and users [164]. These wireless attacks (denial, deception and/or destruction) have traditionally been the domain of Electronic Warfare [51]. However, these techniques are gradually being adopted for criminal activities as readily available hardware supports the development of effective systems [165] that can match jamming prevention techniques. As a result, WSN compromise, whether malicious or unintentional, is achievable and threat detection and analysis need to match advancing attack strategies [51]. Consequently, an interference detection approach that neglects network-level data, allows independent device operation and only requires data always available to a functioning receiver is an attractive concept.

Coupling WSN and GPS operation for developing an interference diagnostics framework is reasonable, as GPS signals are becoming increasingly crucial for civilians, services and industries due to the dependence on GPS-derived location and time measurements. Typically, WSNs, which can incorporate received GPS data, can be integrated into the overall Internet of Things (IoT) architecture and involve long-lived deployments consisting of low-cost, compact resource-constrained devices coupled to their operating environment, which prohibit using complex or computationally intensive security protocols. Due to the importance of these signal models, this chapter uses the ZigBee [37] protocol and available GPS signals to develop a novel low complexity interference diagnostic framework for WSN and GPS resource-constrained edge devices that exclusively utilizes consistently available received I/Q samples. This approach is based on the hypothesis that mitigation can be implemented once an attack (or packet loss reason) is detected, thereby improving device and system security in the process and applies the concept that edge nodes can usually deliver packets to non-jammed neighbors [19].

The real over-the-air data was collected by implementing a SDR experimental approach using ZigBee/SDR and GPS/SDR testbeds, where the focus was applied to matched signal (ZigBee) and CW interference. As per Chapter 6, the I/Q features are extracted across the time- and frequency-domains and space (PDF). In-depth analysis and validation of the low-order features for interference detection were achieved using machine learning-based classifiers, namely SVM, the dependent ensemble XGBoost approach and a deep neural network (DNN). The developed models were evaluated using available test data and K-fold cross-validation. The low complexity supervised machine learning interference framework based exclusively on the low order features achieves an average accuracy among the developed models above 98%, which matches or out-performs the literature as discussed in Chapter 3. The methodology development 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 an edge device interference diagnostic tool. Also, the developed methodology's transferability was investigated by applying the strategy to GPS signals using a different SDR and value range than the WSN investigation. This chapter's main contribution lies in the developed diagnostic framework which is a novel security approach that strengths the defense-in-depth approach for wireless edge devices. The developed framework enables independent operation, as no channel assumptions, network-level information, or spectral images are required. It differentiates itself by solely analyzing the raw 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. The overall diagnostic algorithm is formulated based on the developed interference detection models and the previous signal classification approach in Chapter 6. As a result, this interference diagnostic framework is the culmination of all of the work in this thesis thus far.

7.2 Experimental Discussion

The examination of specific procedures is required before delving into the developed interference diagnostic framework for either GPS or WSN signals. Applying the previously developed signal classification framework in this chapter is a form of transfer learning. A discussion regarding transfer learning establishes the reasoning for its application and how it can potentially lead to high performing interference detection models that generalize well to unseen data. The unique application of specific experimental methods during each of the signal models interference frameworks' development are established or reintroduced from previous chapters. The discussion on each method relates to how it was applied and why. This approach is taken to provide an comprehensive overview of the methodology developed in this chapter. Consequently, the work has potential applications for other signal models or wireless operating environments.

7.2.1 Transfer Learning Approach

Transfer learning is defined as a research problem in machine learning that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem. A common image analysis example is using knowledge gained while learning to recognize cars in the problem of recognizing trucks. In terms of relating transfer learning to how the generic machine learning task was introduced in this thesis, the general definition of transfer learning is given in terms of domains and tasks. A domain $\mathscr{D}$ consists of a feature space, $\mathscr{X}$ and a marginal probability distribution $P(X)$, where $X = \{x_1, x_2, ..., x_n\} \in \mathscr{X}$. Given a specific domain, $\mathscr{D} = \{\mathscr{X}, P(X)\}$, a machine learning task, as discussed in Chapters 4 and 6, consists of two primary components, a label space $\mathscr{Y}$ and an objective predictive function, $f : \mathscr{X} \to \mathscr{Y}$. The function, f, is used to predict the corresponding label $f(x)$ of a new instance x. This machine learning task, $\mathscr{T} = \{\mathscr{Y}, f(x)\}$ is learned from the training data consisting of pairs $\{x_i, y_i\}$, where $x_i \in X$ and $y_i \in \mathscr{Y}$. Then, given a source domain $\mathscr{D}_s$ and associated learning task $\mathscr{T}_s$, the new target domain is denoted $\mathscr{D}_T$ with associated learning task $\mathscr{T}_T$, where $\mathscr{D}_s \neq \mathscr{D}_T$, or $\mathscr{T}_s \neq \mathscr{T}_T$. Transfer learning aims to help improve the learning of the target predictive function $f_T(.)$ in $\mathscr{T}_T$ using the knowledge in $\mathscr{D}_s$ and $\mathscr{T}_s$.

From the practical standpoint, this thesis applies transfer learning by using the feature set and optimal XGBoost and DNN models obtained during the legitimate signal classification study in Chapter 6 for new problems in interference detection. The features are the same, but the data on which the features are calculated is different. A new numerical range and receiver were examined for the GPS signals, which results in slightly different feature calculations, but the feature descriptions are consistent. The differences include the numerical range, type of SDR receiver, frequency, and makeup of the data, which incorporates both signal interactions (mix of two or more signals) and signals incorporating a single source (satellites). Hence, the distribution of the data in the domain of this chapter is different from Chapter 6. Similarities exist, specifically in terms of the legitimate ZigBee signal, but the rest of the data and its labeling are different. As a result, this process can be identified as a domain adaptation.

Domain adaptation [166] is a subcategory of the machine learning concept of transfer learning. The domain adaption scenario arises from learning a well-performing model from a source data distribution and applying it to a different (but related) target data distribution. Hence, the source and target domains have the same feature space but different data distributions. In contrast, transfer learning includes the cases where the target domain's feature space is different from the source feature space or spaces [167]. Essentially, domain adaptation is the ability to apply an algorithm trained in one or more "source domains," $\mathscr{D}_s$, to a different (but related) "target domain," $\mathscr{D}_T$. A common example is the task of spam filtering, which consists of adapting a model from one user (the source distribution, $\mathscr{D}_s$) to a new user who receives significantly different emails (the target distribution, $\mathscr{D}_T$). A domain shift, or distributional shift, is a change in the data distribution between an algorithm's training dataset and a dataset it encounters when deployed. These domain shifts are common in practical applications of machine learning, and conventional algorithms often adapt poorly to domain shifts. Here, the model parameters and features remain consistent but are applied to a full dataset of newly collected data. The conventional process may not be fully applied. However, the underlying concept of applying what was learned in one task to a new task is adopted in this framework development.

Using the above approach, the features and associated optimal machine learning approaches from the legitimate signal classification in Chapter 6 are leveraged here to significantly improve the interference detection and classification model development efficiency. This process is valid due to their similar physical natures in the legitimate and interference induced signal domains. The best single multi-class classifier approaches were the XGBoost and DNN methods, where the determined parameters are provided in Tables 7.1 and 7.2, respectively, where "gamma" corresponds to the name given to the Python3 XGBoost minimum loss reduction parameter. This approach can be identified as a form of domain adaptation as the source (legitimate signal classification) and target (interference detection) have the same feature space but with different distributions due to the data being analyzed. In this interference detection work, the data is either signal interactions in the ISM RF band or GPS data received with/without interference at the L1 center frequency (1.57542 GHz). The developed signal classification models and associated features achieved high accuracy and, more importantly, generalized well to unseen data. These results indicated that the models and features could perform well when determining non-legitimate ZigBee and the type of interference in the received erroneous samples. As a result, this transfer learning is an appropriate advance of this work.

7.2.2 Experimental Methodology

The initial experimental methodology arose from the Matlab simulations in Chapter 4, where the features were extracted from the time domain and the calculated probability density function (PDF) of the received I/Q samples. These extracted features depended

Table 7.1: Optimized XGBoost Hyper-parameters

Parameter:	Value:
No. Decision Trees	5
Learning Rate	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 7.2: 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

on the ideal case, where the PDF and samples have no numerical limitations caused by hardware restrictions. The simulations cannot model live wireless signals (and associated environmental interactions) precisely, as wireless channel characteristics such as, for example, fading levels, obstacles, path losses, spurious interference, etc., are not modeled. These limitations meant that only the independent ensemble method of Random Forest and binary SVM methods were investigated as the smallest deviations from the norm were detectable. However, this setup was acceptable as the simulations were only an initial validation of using received I/Q samples to detect interference. The simulation results in Chapter 4 enabled jamming/interference detection for various regions, including an unintentional interference or high channel noise region, subtle jamming or signal collision region and a high impact jamming region. The simulated results motivated live wireless data. However, additional features are required due to the hardware restrictions (reference voltage and analog-to-digital converter resolution) and operating environmental conditions. This expansion of the methodology was highlighted in Chapter 6 when developing the ISM band signal classification tool. The additional features were required to classify signals when the receiver was saturated or when signals used similar modulation schemes.

The received data in this chapter was collected using an Analog Devices Pluto SDR and a NESDR SMArTee RTL-SDR dongle, as discussed in Chapter 5. For the WSN investigation, the I/Q samples are received in the range [-2048:2047] before being down-converted to the range [-1:+1] to provide features with similar value ranges. This down-conversion is the normalization approach taken in this study, and this technique typically results in higher-performing machine learning models. Here, the Pluto's received samples are converted to the range [-1:+1] for consistency with the signal classification approach in Chapter 6. In contrast, the GPS signal data remain in the RTL-SDR range of [-128:+127] for an additional examination of the features. The features and overall methodology can be further scrutinized by analyzing a second value range and a second type of receiver. Additionally, the sample sizes are altered for the GPS data to investigate the proposed methodology further. For the WSN signals, the analyzed samples remain at 1250 I and Q samples, where this sample size is based on the smallest received packet in the signal classification study, a Bluetooth Advertising Channel. However, for the GPS case, a much larger sample size is analyzed, 10230 I and Q samples. This GPS sample size was based on previous experimentation work in [41] and due to significantly larger packets being received from the GPS signals than from the ISM RF band received packets. If the features and developed methodology perform well across receivers, value ranges and analyzed sample sizes, the overall methodology gains an additional generalization metric. This concept can be inferred, as it would suggest the methodology's features have an underlying link to each classified data instance. Hence, successful results with multiple SDR receivers and numerical ranges provides the required evidence that the methodology is not dependent on specific hardware or numerical ranges. This result is critical as a methodology (interference diagnostic framework) that is independent of the applied hardware and numerical ranges is much more suitable for practical deployment. Typically, multiple different receivers and devices are in deployment in an operating environment. Additionally, this transfer to new data provides further validation of how the developed methodology performs with unseen data, resulting in model generalization validation.

The resultant fourteen optimal features extracted from the work in Chapter 6 were low-order and based on sample sizes of 1250 I and Q samples. The PDF calculations were between [-1.25:1.25] with a bin spacing of 0.05 for the WSN signals and between [-128:127] with a bin spacing of 1 for the GPS. The GPS signals did not require additional bins at the edge due to the expected GPS signals being confined to the PDF's middle. In comparison, the WSN signals were expected to cause saturation and visualization of the edge bins is beneficial. In GPS reception, saturation conditions correspond to jamming, only, while the WSN can be legitimate or jamming conditions. The final feature set in each signal model case 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 [20] - 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 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 $X = \{x_1, x_2, ..., x_{14}\}$. The slight differences in the WSN and GPS signals' calculated features enable examination of how the developed low-order feature set reacts to different numerical ranges of data, which is an important characteristic to determine.

As previously discussed in Section 7.2.1, the two optimized multi-class machine learning approaches that were developed based on these features were the XGBoost dependent ensemble method and a feedforward fully-connected DNN. The optimal XGBoost hyperparameters are specified in Table 7.1, while the DNN structure is provided in Table 7.2. 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. However, where appropriate optimization opportunities are apparent, they are investigated to develop the highest performing framework possible from the available operating environment and associated data. Later in the study, the previously designed signal classification approach that developed these optimal models will be combined with the models developed in this chapter 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.

This two-stage approach to the overall diagnostic methodology remains consistent with the insights gained from the simulations. This concept is visualized in Fig. 7.1, where the received I/Q samples are deconstructed into the set of fourteen 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. The analysis focuses on identifying the received samples as a legitimate signal, ZigBee for WSNs and receiving four or more satellite signals for GPS applications. If an erroneous packet, or less than four satellites, are received, the samples can be passed to the interference detection algorithm for jamming detection and/or classification. However, this two-stage approach assumes that the multi-class classifier has lower interference detection performance or a longer average prediction time. This two-stage methodology is examined in the results section to investigate whether a binary-detection followed by multi-class classification is better than a single multi-class classifier. A database of legitimate signal data and combinations of legitimate signals and jamming signals is required in either approach. The desired interference diagnostic framework requires high accuracy and model generalization to unseen data while simultaneously achieving low complexity through low-order features, fast optimization and prediction times and a computational requirement that is as small as possible. A machine learning diagnostic tool can be de-

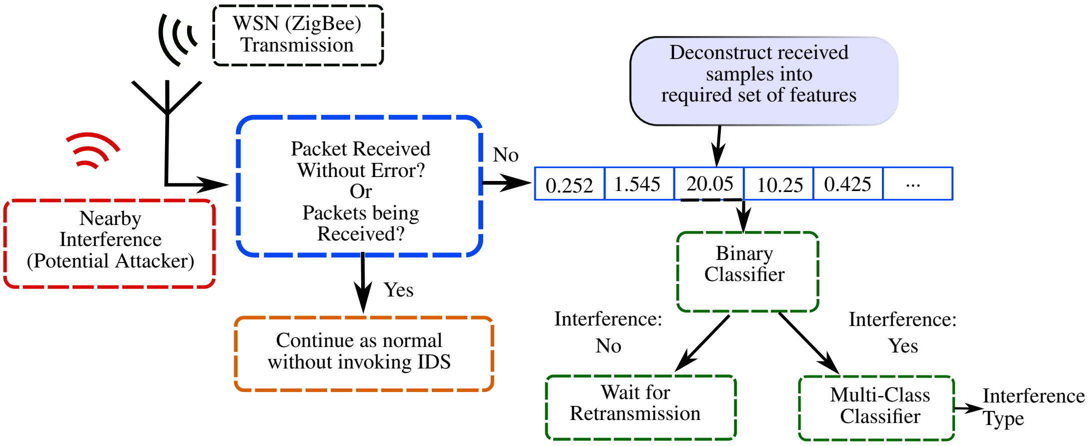


Figure 7.1: Data flow diagram representing the developed two model approach which leverages binary and multi-class classifiers.

veloped for use on low-power resource-constrained edge devices if these requirements are achieved. The Raspberry Pi embedded device examination provides an initial investigation of the developed model's fulfillment of these requirements. A comparison of the achieved performance to the literature provides the necessary evidence of required performance. The developed models must, at least, match the comparable approaches in the literature.

In this chapter's results, the training and testing of each model of the desired diagnostic framework uses available Python3 libraries running on an Intel i7-9700 3 GHz CPU. The development of the DNNs uses Keras and TensorFlow operating on the same CPU exploiting an Nvidia GeForce RTX 2060 graphical processing unit (GPU) with 6GB of RAM. An implementation study of a subset of the developed supervised machine learning algorithms occurs on a Raspberry Pi embedded device with 1 GB of RAM and a quad-core Broadcom Arm Cortex A53 processor (1.4 GHz). Notably, the achieved results aim to demonstrate 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 and potent signal processing are applied. However, a vital characteristic of the experimental methodology to observe relates to the designed models. As the data 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. The methodology includes the features, the data requirements, the identified machine learning approaches, the experimentation across platforms and the exclusive use of I/Q samples. The framework is the application of the methodology across several use cases that all affect a wireless edge node's security.

7.3 GPS Results

Initially, the focus is applied to GPS signals as their associated low received power levels, approximately -125dBm, are comparable to signal classification. This concept is applicable as received GPS signals resemble noise and relatively low powered jammers can readily block satellite reception. This phenomenon results in GPS interference signals being classifiable in the presence of noise-like signals, which is consistent with the approach in Chapter 6. However, in contrast, a relativity simple feature set focused on the calculated PDF can determine the presence of unwanted signals in GPS applications. This is due to the expected GPS PDF, which resembles a unimodal peak at the center bin (0) value. This typical unimodal shape of the GPS signals is provided in Fig. 7.2, where the PDF becomes distorted (peak lowers, PDF becomes broader and more bimodal) as the jamming power increases. This approach that focused on the PDF analysis was successful in a previous study [41], which applied the Random Forest machine learning method, a low complexity feature set and used data from a subset of available satellites. This previous study had limitations as the interference signal was not classified and data from all available satellites was not applied. The GPS data strategy outlined in Chapter 5 rectified the data limitation, as data from all thirty-one satellites in orbit was collected. This data collection results in the developed model being independent of the received satellites. The second limitation of interference classification is amendable if a more advanced feature set is implemented, as the jamming signal can, potentially, be classified. For this purpose, the previously developed optimal machine learning models and associated features in Chapter 6 are implemented to investigate whether a GPS jamming detection and classification approach can be designed. Data was collected for clean GPS signals (all thirty-one satellites) and GPS signals in the presence of CW and offset quadrature phase-shift keying (O-QPSK) modulated signals (ZigBee), as discussed in Chapter 5.

Based on the simulations in Chapter 4, the envisaged 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. The two-stage approach was the proposed design strategy, as jamming signal classification should only be implemented once interference is detected, to minimize computation on resource-constrained devices. A linear binary classifier seems appropriate as the detection is associated with the PDF changes. Low complexity and minimum response times (prediction and optimization) were the essential design requirements. To validate the developed approach's choice, the DNN was designed and compared with less complex SVM and XGBoost approaches. This work was undertaken due to the current trend in using deep learning for interference detection and signals classification, as discussed in Chapter 3. This GPS work expands on the previous study [41] by looking at satellite data across a full

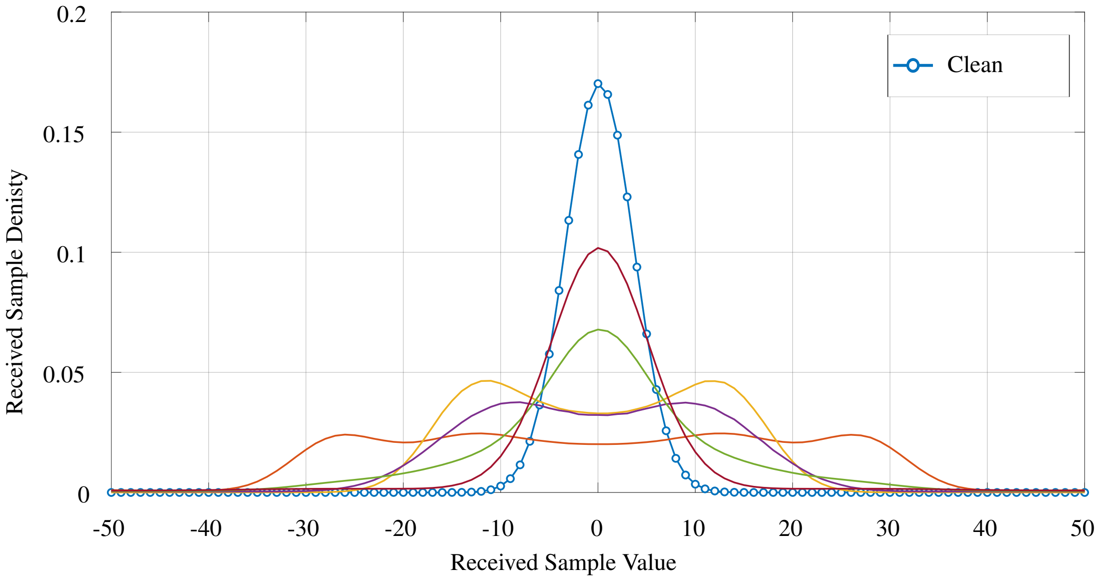


Figure 7.2: The probability density of the raw received I/Q samples collected under a range of interference conditions and signals, with the clean case marked and all others being GPS signals in the presence of increasing levels of jamming power.

24-hour period (31 satellites in total) and classifying the jamming signal. This investigation also examines if the proposed methodology is transferable across the RF spectrum, SDR receiver (hardware independent) and numerical value range on which the features are calculated. Successful results would signify that the features, data requirements (raw I/Q samples only) and the machine learning approach are compatible across platforms, signal models and data normalization (numerical range).

The ISM RF band wireless signal classification algorithm, based on received I/Q samples, is applied to GPS signal data and associated data for jamming scenarios over wires at 1.57542 GHz. The XGBoost hyperparameters, as per Table 7.1, were applied and the DNN structure in Table 7.2 was applied with two output neurons for detection and three neurons for classification (Clean, CW and O-QPSK). GPS signals leverage the full previously developed scheme as GPS signals resemble noise under typical operating conditions. The test data can be classified as unseen since the data was collected over 24 hours, where the transmitting satellites are in orbit at a constant velocity, and the interference signals used several different power levels for the two jamming signal types. 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 more than once, and the satellites are in constant motion. 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 data were randomly allocated

Table 7.3: GPS Jamming Detection and Classification Results

Model	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

to training and testing in the conventional 80% : 20% split.

The GPS binary jamming detection and multi-class classification results are provided in Table 7.3. The detection results clearly show that the designed approach can detect variations from the expected received signal even when Pluto jamming signals incurred a “-70dB” gain. The Pluto transmission gain increased by 2dB every hour over the 24 hour period, where the RTL-SDR incurred saturation conditions at the end of the experiment. At the 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 (RBF) was determined to be the optimal kernel. However, the most accurate overall model was the developed XGBoost (dependent ensemble decision tree approach), 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 fewer required 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 scores 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 prove that the developed features in Chapter 6 are not confined for use in the 2.4-2.5 GHz RF band, nor do they depend on specific hardware or numerical data ranges. This conclusion results from the GPS examination using the fourteen developed features in the received range of [-127:128], compared to the original range of [-1:+1]. The developed features provide an underlying description of the received signal being analyzed and this results in the features being compatible across platforms, signal models and data normalization (numerical range). Additionally, the developed, optimized models being successfully transferred to new data and, subsequently, being useful for jamming detection and classification in GPS signals signifies a successful adoption of a form of transfer learning. Overall, this GPS signal investigation was a bridging investigation between the previously designed signal classification approach and the WSN jamming detection and classification. The achieved results suggest that adopting the approach for use in WSN interference detection is promising. 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 suitably descriptive feature set and optimal model hyperparameters. These GPS data results adds another layer of evaluation to the developed feature set and interference diagnostic framework under development.

7.4 WSN Results

The designed GPS approach leverages the signal classification models fully on the newly collected GPS data with and without interference. WSN signals cannot apply this approach due to the received power levels, which are much higher in comparison. This observation is why the simulation study in Chapter 4 was critical, as it provided essential contributions to the development of the WSN interference detection strategy. The simulations both consider the ideal scenario, which neglects hardware restrictions, and identify the data needed for jamming identification and classification. 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 applicable multi-class machine learning model was identified to be the independent ensemble approach of Random Forest. These simulations require features from the calculated PDF and time-domain analysis of the I/Q samples to identify interference. However, in Chapter 6, the addition of frequency domain features and the use of dependent ensemble approaches, namely XGBoost, were determined to be more optimal when analyzing over-the-air wireless transmissions. Consequently, as the methodology aims to achieve high performance on the available data while maximizing the probability of generalizing well to unseen data, the learnings from Chapter 6 are employed to increase the prospect of successful interference detection and classification in live over-the-air wireless signals.

Additionally, the simulation results in Chapter 4 established the detection methodology as a two-phase detection process. This process is visualized in Fig. 7.1 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. Based on the simulation results, the hypothesis is that binary detection incurs a quicker prediction time, on average, than the multi-class approach.

In this section, the fourteen previously developed and extracted features in Chapter 6 are explored to determine whether an existing ISM band signal classification scheme applies to a new data distribution of signal interactions. For the WSN jamming detection and classification study, several different investigations, using the fourteen features and optimized models mentioned above, are implemented. These different studies aimed to develop an interference diagnostic framework for edge devices that can operate under several conditions. Furthermore, by focusing on the same features for multiple models and classification, the overall diagnostic strategy's complexity is reduced. 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 7.1 and 7.2, the lower complexity XGBoost approach will be examined.

The following results (and implementation study) demonstrate the developed diagnostic framework and feature set's usefulness. The collected data is split into training 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. The data was of a sufficient size that using the entire training set for model development incurred limited time penalties. The work included the five steps mentioned 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. 7.3).
- Artificial jamming of legitimate XBee node data.
- SDR transmitted ZigBee live wireless jamming.
- Live subtle CW jamming of commercial XBee nodes.

7.4.1 Legitimate Node Classification

The first concept investigated was whether the fourteen features and associated models could identify individual commercial nodes (radio identification), DIGI XBee nodes in this study. This classification would enable the identification of a malicious commercial node transmitting data in the network. Each XBee node transmitted SenseHat (see Chapter 5) data and associated acknowledgement packets. Data was received using a

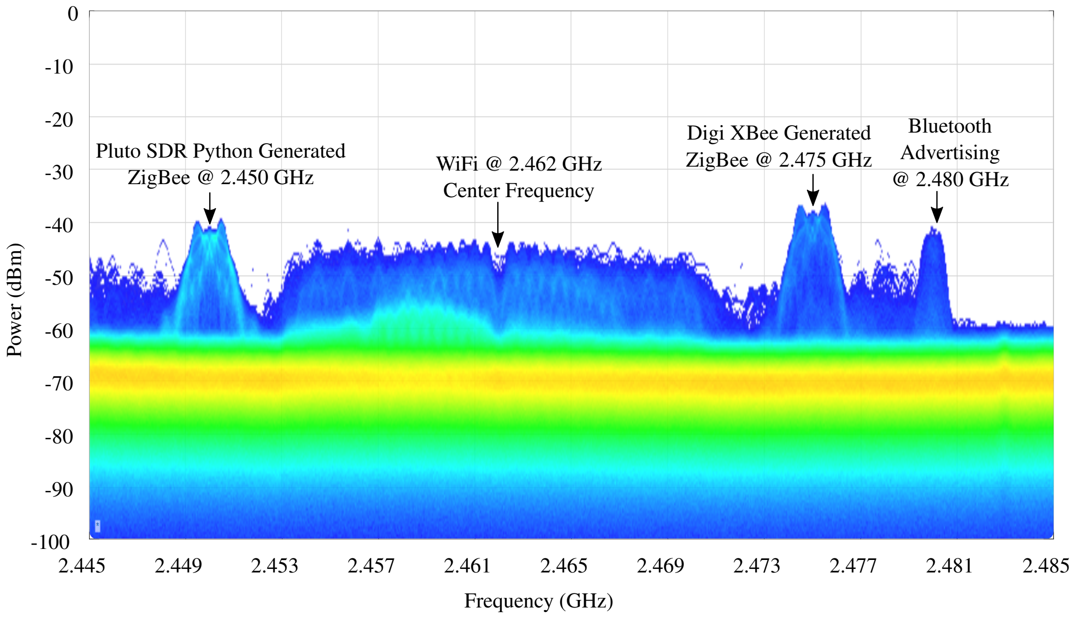


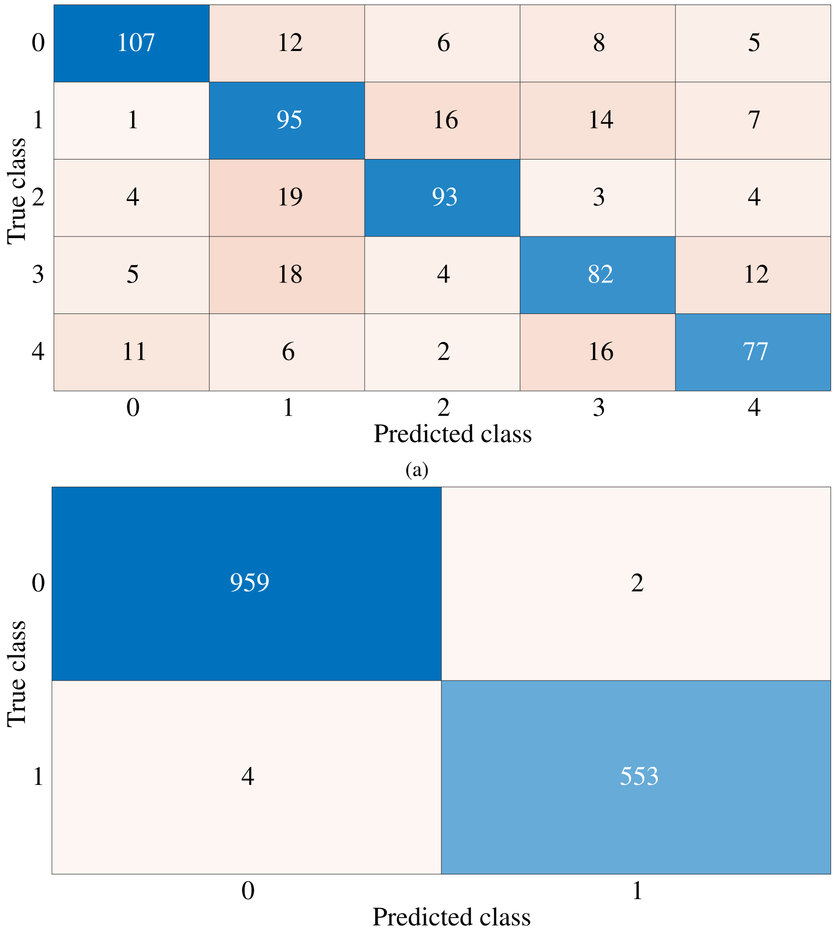
Figure 7.3: A Tektronix DPX image [35], which is a digital signal processing software that rasterizes samples into pixel information, of the customized SDR and commercially transmitted ZigBee signal.

Pluto SDR and analyzed using the fourteen features on the down-converted data range [-1:+1]. Since the fourteen features were designed to identify legitimate ZigBee transmissions, the results provided in Table 7.4 are as expected. The DNN and XGBoost models are compared here as this is a multi-class classification, where a single model is desirable. 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 7.1. The confusion matrix for this approach is provided in Fig. 7.4 (a) where the classifier fills every classification possibility. These features cannot develop a radio classification model. However, the results suggest that legitimate radios as a group may be classified, as the results indicate the fourteen features do characterize the commercial nodes together.

As the individual XBee radios could not be identified with sufficient accuracy, the next phase aims at classifying legitimate (XBee device) and non-legitimate (SDR) nodes. This method involved using data from five individual XBee nodes which transmitted SenseHat data and network operation signals, such as, for example, acknowledgements, and SDR ZigBee transmissions using SenseHat and random data. By visualizing

Table 7.4: 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
XGBoost (Sub-optimal)	36.9 ms	0.057	99.54	50	98.996	1.03



True class \ Predicted class	0	1	2	3	4
0	107	12	6	8	5
1	1	95	16	14	7
2	4	19	93	3	4
3	5	18	4	82	12
4	11	6	2	16	77

(a)

True class \ Predicted class	0	1
0	959	2
1	4	553

(b)

Figure 7.4: 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 and (b) The Legitimate ZigBee Signal Identification, where predictor 0 is the XBee Signals and 1 is the SDR ZigBee signals.

Table 7.5: IEEE 802.15.4 PHY Frame Layout

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

the RF spectrum using the Tektronix RTSA, these transmissions can be compared. The SDR transmitted (non-legitimate) ZigBee signals have the same spectral visualization as the commercial (legitimate) ZigBee signals. The DPX visualization validates this observation in Fig. 7.3 and the SDR achieves this performance by implementing the ZigBee protocol and deploying the ZigBee frame (Table 7.5). As a result, both signals apply similar packet structures and modulation schemes, which results in comparable spectral shapes, but the SDR devices are not explicitly designed for WSN ZigBee operation. Being able to identify explicit ZigBee radio devices from software configurable radios is advantageous from a security perspective. As observed in Fig. 7.3, the spectrum alone is insufficient to identify outlier radio devices, resulting in a need for identification upon reception.

The fourteen features for WSN signal classification are applied to investigate the development of a diagnostic tool on edge devices for implementing malicious node identification. The classification results are provided in Table 7.4, where the SVM, DNN and XGBoost models are compared. The XGBoost model, using the parameters in Table 7.1, outperformed the DNN and SVM, where the RBF kernel was determined to be the optimal approach. As the XGBoost achieved the best accuracy, a further investigation was implemented to examine 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. However, the complexity can be reduced significantly by applying 25 decision tree estimators, a learning rate of 0.7, a maximum tree depth of 10, a minimum child weight of 1, used 50% of available data per tree, used the “gbtree” booster, sub-sampled 90% of the feature columns and applied a minimum loss reduction of 0.5. The model’s overall complexity is reduced by implementing this set of parameters, and the resultant accuracy penalty is approximately 0.06%. These results show that the SDR can be identified, with relatively low error, from amongst the commercial XBee nodes as a malicious external node. These results are visualized in the confusion matrix for the optimal classifier (200 decision trees) in Fig. 7.4 (b). Using the positive result as an SDR transmission, the optimal classifier incurs two false

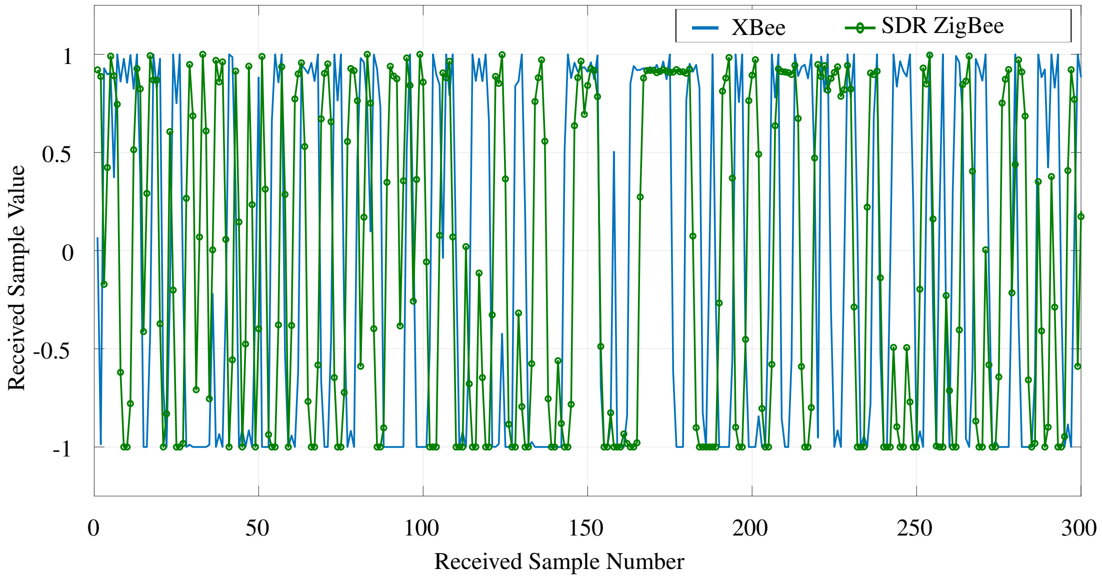


Figure 7.5: A comparison between received samples from a ZigBee signal transmitted from an XBee device and a SDR.

positives and four false negatives. These results indicate a sensitivity of 0.993 and a specificity of 0.998, which means that the classifier can correctly detect non-legitimate signals and reject legitimate ZigBee signals. For the sub-optimal classifier (25 decision trees), the false-positive rate increased by one, meaning that the classifier can correctly detect non-legitimate signals.

However, if the received samples are visualized, a critical characteristic for interference detection becomes apparent. Both the commercial ZigBee and SDR ZigBee signals portray similar properties, as shown in Fig. 7.5, and spectral shape (Fig. 7.3). 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 this feature set identifies the small deviations between the SDR samples and XBee samples, it suggests that subtle interference can be detected. This use of SDR samples will be useful in later sections, especially when applied with XBee samples.

7.4.2 Artificial Jamming Scenario

Before investigating the live signal jamming scenarios using the SDRs to continuously transmit ZigBee signals without the need to be 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 artificial jamming technique randomized the previously collected XBee data and added another previously collected signal to the received ZigBee I/Q samples segment. This addition occurred in software as the received 1250 I and Q samples of each signal type were added as array additions. This method utilized the data samples collected in Chapter 6 to investigate the data distributions associated with signal interactions. 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 receiver's maximum values were maintained by limiting all "jammed" samples to the region of [-2048:2047] or [-1:+1] when down-converted. Each signal's I and Q samples are stored as two single row vectors, each containing 1250 columns. When the data is randomized, the pair of I and Q vectors corresponding to the XBee signals are added to I and Q vectors of a different signal acting as interference. This approach replicates the interaction of two signals, both collected under typical operating conditions. It provides an initial insight into the usefulness of applying the features and models in Tables 7.1 and 7.2 to WSN interference detection.

The investigation results are provided in Table 7.6, where each investigated model performs well for the jamming detection and classification. 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, subsampled 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 fourteen features should be applicable to jamming detection in live experimentation, and the classifier's confusion matrix is specified in Fig. 7.6. The results of the confusion matrix indicate that only one WiFi interfered signal is misclassified as matched interference. However, the classifier performs perfect classification between non-jammed and jammed conditions. Additionally, as this artificial jamming examination utilizes the original fourteen features, their usefulness is validated further.

The features have been adapted to WSN jamming detection when they were initially developed for legitimate signal classification, consisting of a different data distribution.

Table 7.6: 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

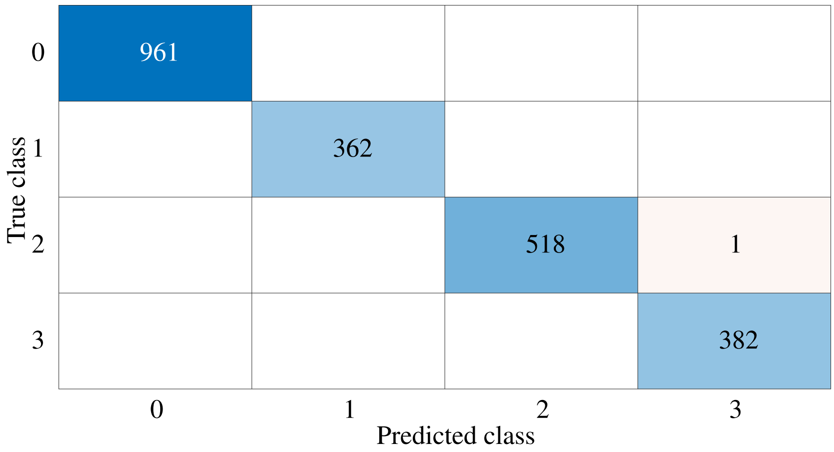
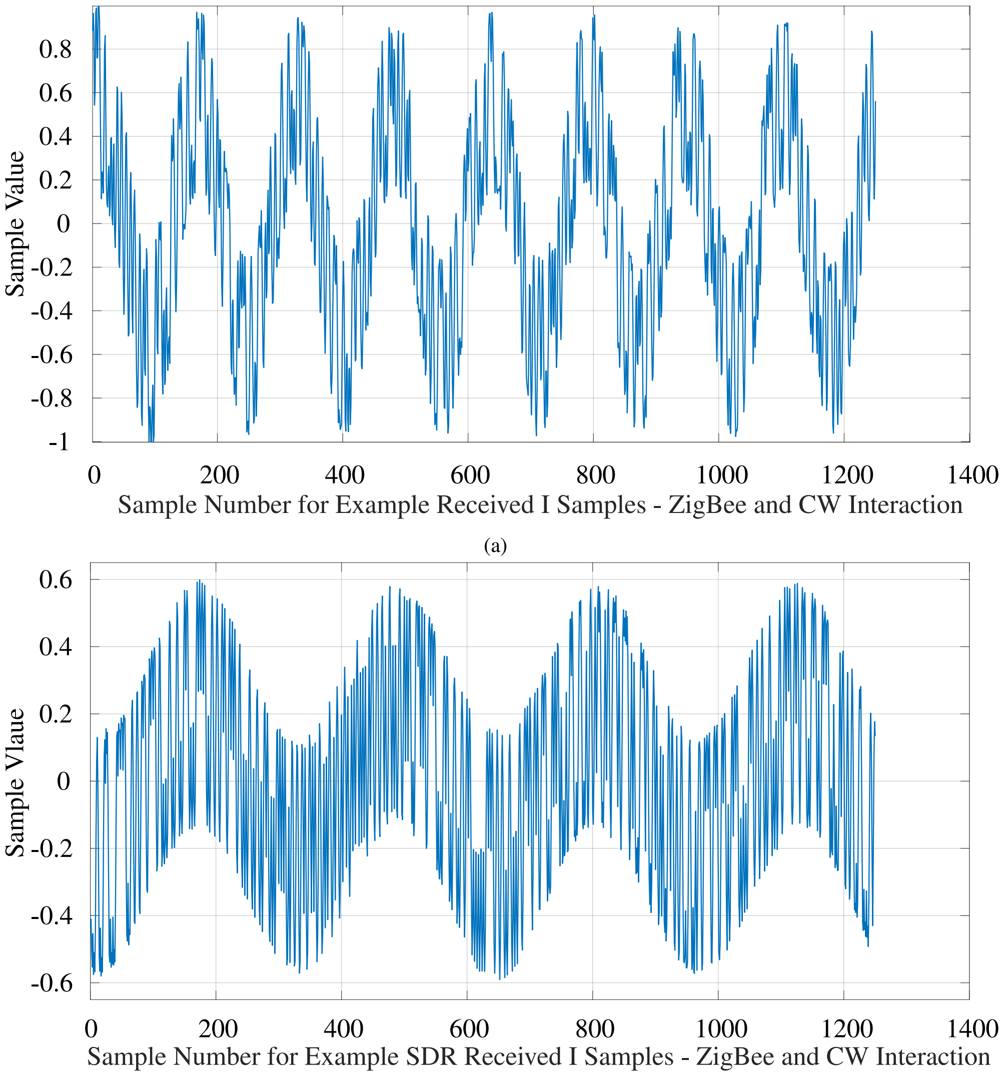


Figure 7.6: The Confusion Matrix for the optimal XGBoost approach for the Artificial Jamming Signal Classification where the predictors are (0) ZigBee XBee, (1) CW Jammed, (2) WiFi Jammed and (3) Matched Signal Interference.

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 7.1) can be implemented without incurring a loss of performance. 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 fourteen features outperform the original simulation features even with hardware restrictions, potentially discovering the real-world differences that were a simulation limitation. Notably, these results do motivate live over-the-air experimentation and outline expected types of signal interactions (Fig. 7.7 (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. Additionally, this artificial jamming mimics the scenario of signal coexistence, whether malicious or unintentional.



(a)

(b)

Figure 7.7: 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 and each case includes a CW trend.

7.4.3 SDR Jamming Scenario

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 and causes a denial of service. This phenomenon occurs for reactive, which only jams once a legitimate signal is detected in the channel, constant, deceptive, random and intelligent jamming approaches. As a result, an efficient method for the continuous transmission of signals that neglected the presence of jamming was required to gain access to the required data of O-QPSK (ZigBee) transmissions interacting with other signals. This method, as mentioned in Chapter 5, utilized SDRs to transmit the required ZigBee signal structure. Even though the SDR and XBee ZigBee signals can be distinguished from one another, 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 received I-data due to the ZigBee (O-QPSK) signal interacting with a CW interference transmission. This visualization is provided in Fig. 7.7, where Fig. 7.7 (a) specifies the artificial situation and Fig. 7.7 (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 transmissions and it can be hypothesized with sufficient confidence that the XBee over-the-air interactions would correlate well 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 dB to -34 dB on the Pluto SDR, where the CN0417 power amplifier provided an additional 20 dB gain to the Pluto 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. Using a power amplifier for the relatively low power signals here provides data that would mimic the process of being transmitted through a power amplifier. These signal powers were sufficient to cause signal interactions while not being so high as to block all transmissions in a typical network or to completely saturate the receiver with the interference signal. 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. If no packets are being received, the legitimate signal classification approach developed in Chapter 6 can be implemented to detect the interference (dominant) signal in the channel, even if the receiver is saturated. 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. This observation further validated the need for a classifier based on signal interactions, as it incurs a different data distribution. As a result, 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 Chapter 5, 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 jammer stops transmitting briefly when changing gain or frequency values. As a result, some instances may not include jamming or only include some relatively 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 distant from (“unseen”) the training data, due to each data segment’s unique timestamps and interactions.

The results are specified in Table 7.7 for both jamming detection and classification. In contrast to the artificial jamming study, the XGBoost model achieved the highest accuracy of 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 trees, but the 25 decision tree approach was chosen for its lower complexity. 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

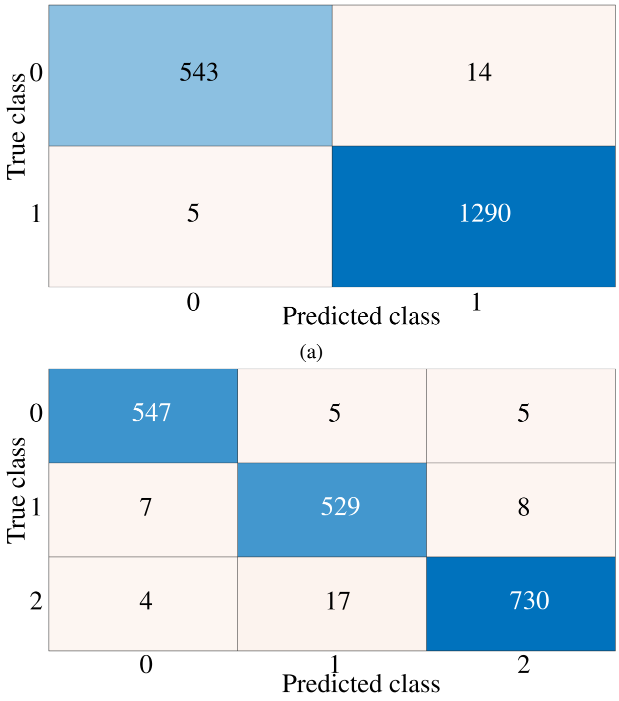
Table 7.7: Over-the-air SDR Jamming Results

	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

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 Fig. 7.8 (a) and Fig. 7.8 (b), respectively. Both classifiers exhibit high performance in terms of identifying the presence of interference (positive case) and rejecting legitimate clean signals (negative case).

7.4.4 Commercial Jamming Scenario

This study's final interference detection 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. The choice of CW jamming is as a result of the simulations (in Chapter 4) predicting this jamming approach to be less effective than matched interference. This interference method results in a higher chance of collecting the necessary data using the SDR. An example of the collected data on the received I-channel, labelled as a CW interference signal interaction, is shown in Fig. 7.9, which indicates a similar trend to the artificial and SDR jamming approaches in Fig. 7.7. This data strategy resulted in received ZigBee 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 legitimate signals were being transmitted. Those legitimate I-Data segments followed the previously collected legitimate signal classification data in Chapter 6. This approach contrasts with the SDR approach, as there was no packet monitoring during that investigation. The legitimate ZigBee packets are received as the SDR jammer briefly stops transmitting due to the Matlab functions being used. As a result, the CW interactions were iden-



(a)

(b)

Figure 7.8: The Confusion Matrices for the optimal XGBoost approach for (a) The SDR ZigBee Jamming detection where the predictors are (0) ZigBee Clean, (1) Jammed and (b) The SDR ZigBee Jamming Classification where the predictors are (0) ZigBee Clean, (1) CW Jammed, and (2) Matched ZigBee Interference.

tifiable using the methods discussed in Chapter 5. A set of CW jammed XBee data was collected and used to develop a machine learning classifier based entirely on XBee data. This CW jammed XBee collected data was also used as testing data for the model designed using SDR ZigBee data in Section 7.4.3. The previously collected XBee data in Section 7.4.1 was employed as the legitimate ZigBee data, while an additional test data set was collected as “unseen” data. The CW jammed XBee data was split between training and testing in the ratio of 80% : 20%, respectively.

The XBee models’ results are specified in Table 7.8, where 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 five 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. 7.10. 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

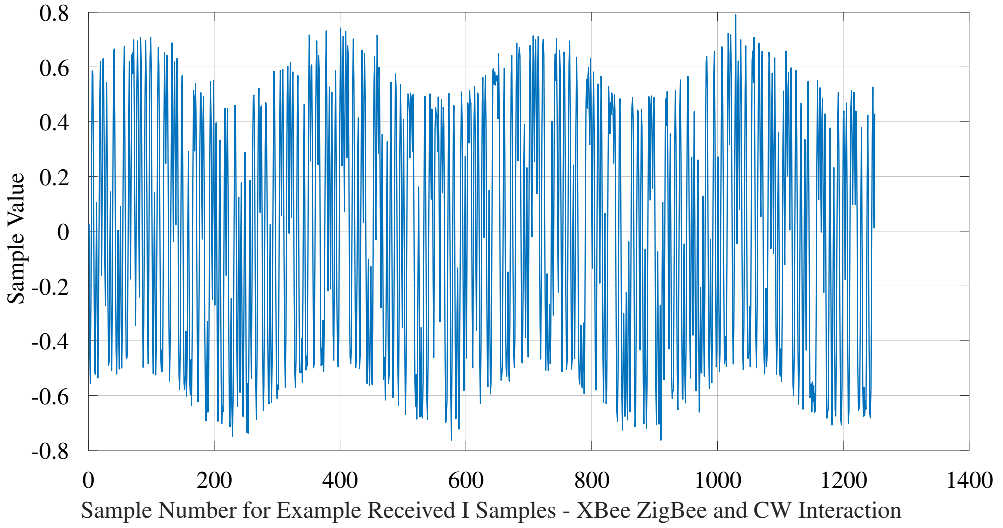


Figure 7.9: An example of a ZigBee signal interacting with a CW Jamming signal for the live XBee ZigBee and CW jamming investigation, which incurs a similar interaction to that observed during the artificial and SDR jamming situations.

Table 7.8: 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

XGBoost model, which was tested using only unjammed XBee and XBee CW jammed data. This model investigated whether the SDR jammed data could detect interference in the XBee data. The optimal XGBoost hyperparameters achieved an accuracy of 90.62%. These parameters included five decision tree estimators, a learning rate of 0.4, a maximum tree depth of 5, a minimum child weight of 1, 50% of available data per tree, the “gbtree” booster, sub-sample 10% of the feature columns and apply a minimum loss reduction of 5. 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, which resembles more closely the XBee CW 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%. The optimal XGBoost hyperparameters

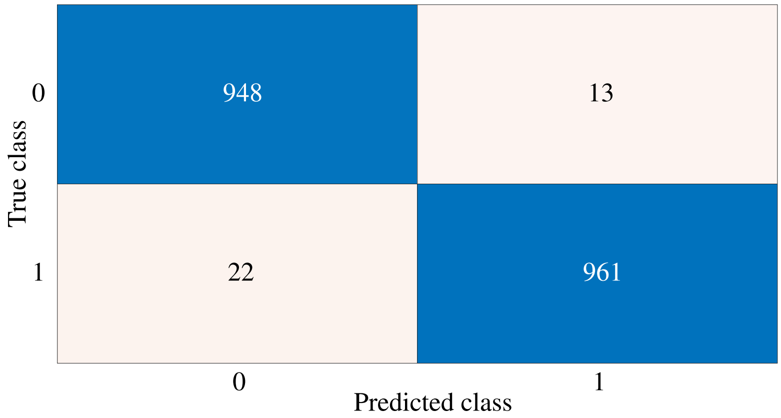


Figure 7.10: The confusion matrix for the optimal XGBoost approach for XBee CW Jamming Detection where the predictors are (0) ZigBee XBee and (1) CW Jammed.

included twenty-five decision tree estimators, a learning rate of 0.4, a maximum tree depth of 20, a minimum child weight of 2, 95% of available data per tree, the "gbtree" booster, sub-sample 75% of the feature columns and apply a minimum loss reduction of 0.5. This model incurred a training time of 329.11 ms and an average prediction time of 0.067 ms. The associated confusion matrix is supplied in Fig. 7.11, which indicates that the classifier performs well when all data is included. These results validate focusing on SDRs for a more efficient WSN jamming signal investigation. Additionally, the result indicates that the SDR approach, where higher powered 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 loss of accuracy.

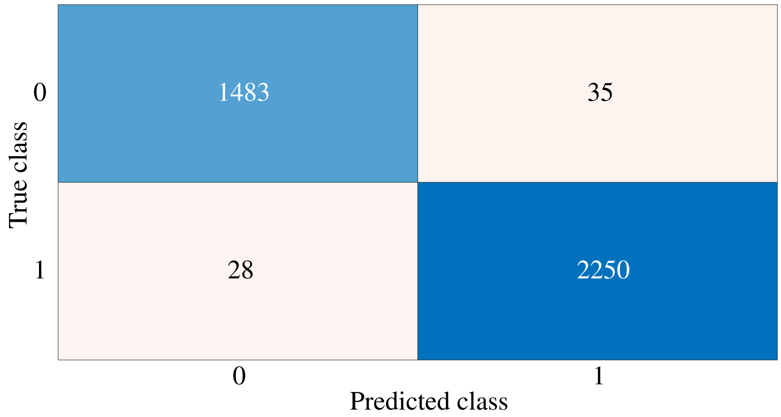


Figure 7.11: The confusion matrix for the optimal XGBoost approach for Jamming Detection model which includes all collected SDR and XBee data where the predictors are (0) ZigBee XBee and (1) ZigBee Jammed.

7.4.5 Embedded Implementation

A subset of the developed supervised machine learning algorithms was implemented on a Raspberry Pi embedded device, introduced in Chapter 5, 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. The results are provided in Table 7.9, where, for each model, similar accuracy was achieved, where any differences resulted from the available multi-threading on the different devices. In computer science, multi-threading is a CPU's ability (or a single core in a multi-core processor) to provide multiple threads of execution concurrently. The results in all cases provide an average greater than 98%, which matches the comparative literature. 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 the 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.

Implementing machine learning at the edge on resource-constrained devices is feasible due to advances in hardware, like the Raspberry Pi development over the past ten years or so, and low numeric approaches for machine learning. In terms of the Raspberry Pi, wireless edge nodes containing similar hardware specifications as the Pi seems to be an appropriate vision for future IoT and WSN applications. As a result, the framework design was not limited by the design of current low-cost edge nodes. This concept is further developed by the work in [168], where 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, which is unattainable by full precision networks, is achieved via low-numeric approaches. The techniques gained from such low numeric experimentation will enable machine learning at the edge. Low precision networks, which suit

Table 7.9: 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

computationally constrained devices, typically incur a classification accuracy penalty. However, this can be recovered through increased computation [168]. As a result, this study and the developed diagnostic framework is a feasible solution for WSN applications and wireless GPS edge devices. However, future work is required to fully convert the framework to low-power embedded systems. If Python capability is available, the models can be implemented as proved by the Raspberry Pi implementation.

7.5 Discussion: Interference Diagnostic Framework

The models developed in this chapter provide sufficient evidence that the developed low-order features and machine learning model structures are applicable for WSN ZigBee jamming detection and classification. The evidence is provided through examining the problem in terms of an artificial jamming study, a SDR ZigBee transmission study, a commercial ZigBee transmission study and a study combining the SDR and commercial transmissions. The developed models were tested on a Raspberry Pi device and achieved similar performance. This result, combined with the GPS investigation, proves that the developed I/Q samples based methodology can be applied across different implementation platforms, SDR receivers, numerical ranges, signal models and frequencies. Therefore, an underlying description of the received signals is evident in the developed low-order features, which is exploited across the developed models. These features are novel as they are low-order and apply unique uses of Hjorth parameters, the FFT and entropy functions. The interference detection models, on average, require a larger number of decision trees compared to the legitimate signal classification study. Typically, the number of trees increased from five to 25 decision trees, where the performance matched the developed DNNs, but incurred a smaller average prediction and training time. Additionally, the two-model approach, suggested by the simulations in Chapter 4, proved not to be the optimal approach. A single multi-class XGBoost classifier is the optimal model determined in this thesis.

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. Notably, the entire framework is based on data that is always available to a functioning receiver, its received I/Q samples. The required implementation algorithm for the edge node is provided in Algorithm 1. This algorithm outlines when each of the designed models should be implemented and why using the same fourteen features in each developed algorithm is crucial for the diagnostic framework's overall design. Each model can use the same fourteen features as 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. For one set of feature calculations, several different decisions can be investigated, depending on the signal reception. For example, if no legitimate packets' signals are being received, the signal classification model can be implemented to detect the channel's dominant signal. In contrast, if packets are received with errors, the interference detection algorithm can be implemented and, subsequently, the classification algorithm, if required. Finally, if a legitimate packet is received, the legitimate node classification model will determine if a commercial node transmitted the packet or not.

This developed framework is relevant as each designed machine learning model achieves similar, if not better, accuracy to the approaches introduced in Chapter 3. For example, the collaborative packet rate analysis system in [105] achieves jamming detection accuracy of over 97% and the chip sequence error pattern approach in [64] achieves accuracies of over 96% for all signals analyzed and an average of 98.29%. In [97], where raw I/Q samples are the main focus, the highest accuracy achieved for the designed signal classification approaches is up to 98%, using convolutional neural networks and only 88% when applying decision tree methods. As a result, the framework developed in this thesis uses a less complex methodology, has several applications, is transferable across the RF spectrum, signal models, implementation devices, numerical ranges and receivers, while achieving 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 fourteen 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

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
```

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.

Overall, this chapter, through a GPS investigation and several ZigBee over-the-air data examinations, has 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 designed methods' effectiveness. The methodology differs from the literature by only requiring access to raw received I/Q samples and focusing on decision tree approaches. This procedure permits independent device decisions as no channel assumptions, network-level information or spectral images are required. Despite the trend to use deep learning approaches, as discussed in Chapter 3, this thesis employed potent data analysis and signal processing to prove that traditional techniques effectively facilitate parsimonious interference detection and classification.

7.6 Discussion: Low-Complexity

This thesis focused on developing a low-complexity solution for interference detection and classification on wireless edge devices. To date, the work in this chapter has proved that the developed low-order features and machine learning model structures have developed widely deployable decision support systems for intelligent interference detection for WSN and GPS signals. The overall designed interference diagnostic framework involves legitimate node classification and interference detection and classification solely using data that is always available to a functioning receiver, its received I/Q samples. This discussion section brings together the different strands on computational complexity and highlights how the work presented in this thesis achieves the low complexity aspect of the research. Low-complexity is warranted to reduce the operational costs of future intelligent devices [97] and is achieved in this thesis by solely using I/Q samples, using non-deep learning machine learning algorithms and developing low-order features (compared to the related literature).

The sole use of I/Q samples results in wireless edge devices being able to make independent decisions by utilizing low-level data to make typical higher-level decisions. This is crucial in modern applications due to increasing heterogeneity in wireless communications, often sharing the same spectrum band, resulting in a need for devices to sense the environment and make intelligent decisions [97]. Classic WSN interference detection approaches typically analyze the RSSI and different forms of packet rates [44], which would require edge devices to be aware of network operation and use resources to determine the necessary network data. The process designed in this thesis allows the edge device to neglect network-level data, high-order features, received signal strength indicator (RSSI) samples and image-based spectrograms and to avoid the requirement to buffer known patterns or previous receptions. The developed approach alleviates some of the complexity in terms of accessing data, as the only data required is always available to a functioning receiver.

Deep learning methods have dramatically improved the state-of-the-art in several application areas [88], where the successful results typically leverage deep learning's ability to automatically learn complex features from large input datasets. This method is a data-driven approach and training such data-driven deep networks on large volumes of data typically requires appropriate computational resources and extensive time, both of which are rarely found in deployed communication systems [95]. For example, the chosen DNN in this chapter required the use of Keras on an NVIDIA GeForce RTX 2060 with 6 GB of RAM, where 5.9 GB of RAM was utilized across the testing. To analyze the model complexity, the model training time and average prediction time were used as measures of complexity throughout this thesis. It is clear from the results of Chapters 6 and 7 that the main finding surrounding the training and prediction times is that they are much lower for the supervised traditional machine learning approaches. The average training time for the deep learning approach is orders of magnitude larger than the other adopted machine learning algorithms. For example, in Table 7.8, the DNN training time is 17788.88 s and the optimal XGBoost training time is 205.42 ms, while the same trend is seen in Table 7.7, as the training times for the DNN and optimal XGBoost are 2618.56 s and 481.71 ms, respectively. The same trend is seen across the results in this thesis and when analyzing the average prediction times, as the DNN typically requires 10's of milliseconds, while the XGBoost requires a fraction of that time (typically 10's of microseconds). Using these measures of complexity, to adapt the developed framework to new wireless environments, the traditional non-deep learning machine learning approaches are optimal and less complex to achieve, due to the reduced training and prediction times. The smaller average training and prediction times result in more efficient and faster optimization times, which enables a more efficient and less complex adaption to new environments. Additionally, as shown in the GPS results (Table 7.3), the model sizes are larger for the deep learning approaches, which can be an issue for resource-constrained devices.

When discussing resource-contained devices, the complexity can be expanded to include the fast adaption to embedded devices that do not have access to large levels of resources. This was investigated through an implementation study of a subset of the developed supervised machine learning algorithms on a Raspberry Pi embedded device with 1 GB of RAM and a quad-core Broadcom Arm Cortex A53 processor (1.4 GHz) (see Section 7.4.5). The Raspberry Pi results were compared to the results on a Desktop running an Intel i7-9700 3 GHz CPU with 16 GB of RAM, where both devices leveraged available Python3 libraries. The supervised non-deep learning approaches were ported to the embedded devices quickly and similar accuracy to the desktop implementation was achieved, where any differences resulted from the available multi-threading on the different devices. As a result, it is clear that parameter optimization for the non-deep learning approaches for both a new environment or device is orders of magnitudes faster and incurs a lower level of complexity, which pairs well with deployed wireless communication systems where extensive computational resources and time are rarely found [95]. Notably, the achieved results demonstrate that the less complex machine learning approaches can match the performance of DNNs, but for a small fraction of the required time and resources. As this thesis focused on developing a methodology rather than a specific model, faster optimization times are critical for applying the developed methodology in various operating environments.

The final aspect of low-complexity operation surrounds the development of the novel low-order statistical feature-set. The concise novel feature set is based on time-domain, frequency-domain and spatial analysis of the received I/Q samples collected from an active wireless environment that is typically changeable. When compared to the related literature, as discussed in Chapter 3, it contrasts with the typical use of high-order statistics [81] and/or cumulants [82, 83, 84], where the high order is, generally, beyond 4th-order. The developed features utilize low-order functions, as discussed in Chapter 6, which can easily be ported between programming languages, as shown by the available libraries in Python that correspond to Matlab functions and in Section 7.4.5. As the features are all extracted from the I/Q data and normalized to the range [-1:+1] for the WSN investigation, the implementation supports low complexity operation through low numerical operations. The lack of spectral images, received signal strength indicator samples, high-order cumulants, network data, buffering known patterns and/or received signals and transforms other than the FFT also support a lower complexity compared to the literature. In terms of the FFT, low-complexity approaches exist and can be implemented on a compiled programming language such as, for example, C++, using low-complexity approaches, such as Kiss FFT, to ensure low-complexity operation of the transform. Evidently, through the different aspects discussed in this section, it is clear how the work presented in this thesis achieves the low complexity aspect of the research.

7.7 Conclusion

This chapter employed low-order features extracted entirely from received I/Q samples to develop a WSN interference diagnostic framework for resource-constrained edge devices. The Matlab Monte Carlo simulations provided in Chapter 4 motivated employing I/Q samples to detect interference in ZigBee signals. The required data for interference detection was ZigBee and jamming signal interactions. SDRs, XBee commercial ZigBee nodes and Raspberry Pi embedded devices were employed in hardware testbeds to facilitate access to the required live over-the-air signals. Both WSN ZigBee and GPS signals were investigated using a Pluto SDR as the jamming device, transmitting both CW and matched ZigBee (O-QPSK in the GPS case) interference.

The previously designed low-order feature set and optimal XGBoost and DNN architectures from the signal classification study in Chapter 6 were leveraged in this jamming detection and classification study. This procedure was a form of transfer learning and was applied to increase the efficiency of developing the interference diagnostic framework. 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. Typically, the number of decision trees increased from five to 25 for the interference detection model, compared to the legitimate signal classification approaches. The performance of the XGBoost classifiers matched the developed DNNs, but incurred a smaller average prediction and training time. Additionally, the two-model approach, suggested by the simulations in Chapter 4, proved not to be the optimal approach. A single multi-class XGBoost classifier is the optimal model determined in this thesis.

The framework development included applying the low-order features and associated design methodology to various interference situations. Initially, GPS interference detection and classification were investigated using a different SDR receiver (RTL-SDR), numerical range and operating frequency. For WSNs, 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. The designed models were combined to develop an edge device low complexity, low-order WSN interference diagnostic framework, which is the thesis's main contribution. 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, channel assumptions or spectral images are required. This novel framework has low complexity, high accuracy ($\geq 98\%$) and fast optimization and prediction times, critical for real-time edge device operation.

The overall results indicate that the developed methodology, which includes the low-order features, the data requirements, the identified machine learning approaches, the experimentation across platforms and the exclusive use of I/Q samples, along with the developed framework, has the potential for real-world adoption. The GPS and WSN investigations combine to demonstrate that the developed machine learning methodology is applicable across different SDR receivers, numerical ranges, signal models and frequencies. The Raspberry Pi implementation provided the evidence for cross-platform operation. These results culminated in the development of a novel low complexity widely deployable machine learning diagnostic tools for wireless edge devices, focusing on interference detection and classification.

Chapter 8

Conclusions and Future Work

This thesis's primary aim was to improve security on resource-constrained wireless edge devices by only using data consistently available to a functioning receiver. The research focused on interference attacks as wireless networks are still inherently susceptible to external interference. As wireless communications and the Internet of Things penetrate every aspect of modern society and become deployed in safety-critical applications, this work's importance intensifies. This concept is further enhanced by society becoming dependent on wireless communications, such as remote patient health monitoring and location services. This chapter summarizes what was achieved in this thesis regarding the main contributions to the field and re-iterates the work's novelty. Furthermore, along with the overall conclusion, some potential limitations of the work are discussed. Although a fully working methodology on an embedded system was presented in this thesis, there are areas in which further research might yield improvements to the overall design and potential real-word adoption.

8.1 Concluding Summary and Contributions

This thesis's central research question focused on determining if received in-phase (I) and quadrature-phase (Q) samples could be utilized to improve security on resource-constrained wireless edge devices. This thesis's main contribution is utilizing low-level I/Q data to formulate a low-order feature set to make typical higher-level decisions using supervised, fundamental machine learning approaches. This procedure enabled the development of low complexity, widely deployable decision support systems for intelligent interference detection on wireless edge devices implementing wireless sensor network (WSN) and global positioning system (GPS) signals. The designed interference diagnostic framework involves developing legitimate signal/channel classifiers and detecting and classifying both malicious and unintentional interference on wireless edge nodes. These tools permit decentralized edge device decision-making, like implementing appropriate security and transmitting mechanisms and reducing retransmissions and energy usage. The aim is for edge nodes to react to the operating environment's influence on the received legitimate signal and to extract high-level interference information from low-level received samples. This section will briefly summarize Chapters 2-7 and outline the associated contributions, where the four main objectives, introduced in Chapter 1, were accomplished across the chapters. These summaries will demonstrate through practical implementations that I/Q samples can indeed be used to improve wireless edge device security.

Chapter 2 introduced the application area for this thesis as WSNs, which have become an integral part of modern technology and can benefit from location and time measurements through the application of GPS signals. Analyzing received GPS signal samples provides a feasibility study examining the transferability of this thesis's work to a different application, area of the radio frequency (RF) spectrum and hardware receiver. ZigBee was identified as the chosen WSN protocol as it is the de-facto standard for WSNs. An extensive overview of security was provided using four pillars; vulnerabilities, requirements, attacks and defenses. Furthermore, as the number of innovative solutions requiring a WSN deployment diversifies and incorporates safety-critical applications, attacker incentive increases. The more critical the application, the higher the probability of sensitive data being transmitted and attacks incurring a higher takeaway. The security analysis concluded that due to the ever-present threat of jamming attacks and its effectiveness not being limited to WSNs, this thesis focuses on improving security by enabling edge devices to implement interference diagnostic tools. These tools have to detect interference when packets are received with errors and when no packets can be received, while also characterizing the intrusion and distinguishing between intentional and unintentional interference.

Chapter 3 specified an overview of the current technical methods for the three primary topics addressed here. This thesis contributes to wireless signal classification methods, interference detection in wireless communications networks, namely WSNs, and the exclusive use of raw I/Q samples. It was shown that this thesis's work 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 wireless operating environment analysis. The literature motivated the choice of jamming in Chapter 2. Jamming attacks are uncomplicated to develop and deploy due to wireless channels' open nature, threats can only be thwarted at the physical layer, and effective anti-jamming strategies for real-world wireless networks remain limited. The more modern subtle jamming approaches are more difficult to detect as traditional packet analysis may require long analysis periods before detection, received power levels will be relatively unchanged and the RF spectrum can be as expected, visually. This overview outlined that focusing on low-order statistical features extracted from received samples differs from the signal classification literature and interference detection by neglecting network-level data, high-order features, received signal strength indicator (RSSI) samples and image-based spectrograms. In contrast to previous interference detection work, only the designed, optimized machine learning model is required on the device, and both malicious and unintentional interference can be classified. Other approaches require the reception of a "clean" packet or specific patterns for comparison. This chapter's information ensured this thesis's novelty and enabled selecting suitable techniques, where feature-based machine learning approaches were identified as optimal.

Chapter 4 focused on an extensive Matlab-based simulation investigation using received I/Q samples for interference detection in WSNs (ZigBee signals). Subtle and crude jamming attacks were examined by implementing continuous-wave (CW), matched signal, WiFi and thermal noise interference. Features were extracted from the calculated PDF and time-domain analysis of the I/Q samples, which incurred no numerical limitations caused by hardware restrictions. Enough differentiation between error-free and erroneous samples existed to warrant an evaluation using a supervised machine learning classifier. This chapter introduced the machine learning concepts of Support Vector Machines (SVMs), Random Forest and this thesis's applied classifier performance metrics. These simulations concluded that both jamming detection and signal classification were achievable using features extracted exclusively from I/Q samples and intelligent classifiers. The required data is ZigBee and jamming signal interactions. These simulations implemented a smaller feature set than the hardware experimentation due to the lack of hardware restrictions. As a result of this chapter's work, live wirelessly received I/Q data is required as the simulation results provided sufficient evidence to warrant a hardware investigation. The developed models produced near-ideal binary detection classifiers ($\approx 0\%$ error) and $\geq 95\%$ accuracy for the multi-class classifiers that detected and classified the interference signal.

Chapter 5 discussed the utilized hardware components and associated developed data strategies for obtaining the required I/Q data from a typical domestic operating wireless environment. Each hardware experimentation approach utilized Raspberry Pi embedded devices and software-defined radios (SDRs) to produce low-cost, high-performance testbeds and data collection approaches. As the interference detection approach requires data encompassing signal interactions (penetration testing), XBee ZigBee commercial devices were set up in the presence of subtle jamming from a Matlab-controlled SDR. Higher-powered jamming approaches were achieved by exclusively using SDRs to provide continuous data transmissions. The Analog Devices Pluto SDR is exploited for the industrial, scientific and medical (ISM) RF band signal experimentation, while the NESDR SMArTee RTL-SDR was applied for GPS signal analysis. In all ZigBee transmissions (XBee and SDR), live environmentally sensed data is transmitted using the Raspberry Pi SenseHat add-on board. This chapter discussed how the received data was processed as part of the data strategy before feature analysis. The hardware and data strategies described in this chapter, and the associated applications, produced the required data to develop legitimate signal classification, WSN and GPS interference detection and classification and legitimate node classification models.

Chapter 6 dealt exclusively with using raw received I/Q samples to develop an optimal set of fourteen low-order statistical features for typical WSN and ISM RF band wireless signal classification. This thesis analyzed noise (typical channel conditions), WiFi (IEEE 802.11), Bluetooth Advertising (IEEE 802.15.1), CW and ZigBee (IEEE 802.15.4) signals. These signals were transmitted from commercial devices and SDRs, where applicable. Features were extracted from the time-domain, frequency-domain and spatial analysis of the received I/Q samples. The frequency-domain analysis focused on the raw data produced by the fast Fourier transform. The feature set differs from previous approaches due to low-order statistics and novel uses of Fast Fourier Transform (FFT) samples and Hjorth parameters. This chapter also introduced the primary machine learning concepts applied in the over-the-air data experimentation. Validation of the extracted novel feature set was obtained by developing several machine learning classifiers including Random Forest, SVMs, K Nearest Neighbors, XGBoost and a fully connected deep neural network (DNN). Test data included unseen data that was used to examine how the developed models generalized to new data. The optimal approach was an XGBoost/SVM adaption, which achieved an error of 0.527%, where the binary SVM classifier was developed to reduce the error if an IEEE802.11 signal was detected. The development of a DNN provided a comparison between deep learning and supervised traditional approaches. The XGBoost/SVM model achieved the same accuracy as the DNN but required significantly less computational and time requirements. This chapter proved that traditional feature-based approaches are still fit for purpose, particularly for low complexity solutions, and achieve high performance when potent data analysis and novel descriptive feature sets are applied. A Raspberry Pi demonstrated that the designed model achieves the same results on a relatively modest embedded device. These results indicate the usefulness of using received I/Q samples for operating environment analysis.

Chapter 7 employed the low-order features and optimal machine learning models developed in Chapter 6 to develop a WSN interference diagnostic framework focused on resource-constrained edge devices. Both WSN ZigBee and GPS signals were investigated using an SDR as the jamming device, transmitting both CW and matched ZigBee interference. The optimal XGBoost and DNN architectures from Chapter 6 were leveraged in this jamming detection and classification study as a form of transfer learning to increase the efficiency of developing the interference diagnostic framework. The XGBoost algorithm outperformed or matched the DNN in each case for a small fraction of the required time and computational resources. This result further validates that feature-based methods are still fit for purpose, particularly for low complexity solutions, and can achieve high performance. Typically, the number of decision trees increased for the interference detection model compared to the legitimate signal classifiers. Optimal 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. GPS jamming detection and classification models proved that the framework is transferable across the RF spectrum, SDR receivers and numerical ranges for received I/Q samples. The same features were applied in each model development, resulting in the designed models being combined to develop an edge device low complexity, low order WSN interference diagnostic framework. The framework provides multiple detection scenarios, benefiting from requiring the same features in each case, and enables independent decisions on edge devices as no network-level data is required. This novel designed framework has low complexity, high accuracy ($\geq 98\%$) and fast optimization and prediction times, critical for real-time edge device operation. A Raspberry Pi implementation provided evidence for cross-platform operation on a relatively modest embedded device.

In summary, this thesis has developed and validated a novel low-order feature set based exclusively on the time-domain, frequency-domain and spatial analysis of received I/Q samples. To the best of the author's knowledge, the application of Hjorth parameters [20] in this thesis is novel and the use of the FFT dynamics produces novel features. These features are applied in feature-based machine learning classifiers to improve security on resource-constrained wireless edge devices, with the benefit of only requiring data consistently available to a functioning receiver. Real-world validated intelligent, novel, low-complexity and widely deployable interference diagnostic tools are developed. Models are based on hardware experimentation of wireless over-the-air received samples in a typical domestic operating environment. The developed tools include ISM RF band signal classification, WSN and GPS interference detection and classification and classifying between a ZigBee signal from a commercial node and a SDR. The designed interference diagnostic framework for independent compact wireless devices can detect interference when packets are received with errors and when no packets are received, while also differentiating between unintentional and malicious sources. The contribution is validated and enhanced by successfully implementing the developed intelligent classification methodology across different, relatively low-cost, open-source

SDR receivers, numerical ranges, signal models (GPS investigation), frequencies and implementation platforms (Raspberry Pi embedded examination).

This thesis's designed methodology includes the low-order features, the data requirements, the identified machine learning approaches, the experimentation across platforms and the exclusive use of I/Q samples, along with the developed framework using real-world signals. The overall framework is validated by implementing deep neural networks and lower complexity, more fundamental, machine learning solutions using open-source software programs. The developed supervised optimal XGBoost approach detects and classifies malicious and unintentional interference with the same performance as deep learning for a small fraction of the time and resource requirements. The average achieved accuracy of the developed models is $\geq 98\%$, when unseen testing data is applied, demonstrating the framework's ability to generalize to new data. The models use the received signal and operating environment interactions to detect subtle deviations from the expected reception to enable high-level decisions based on low-level data. As a result, the developed framework contributes to the defense-in-depth strategy for WSN edge nodes as an additional layer of security has been developed.

8.2 Possible Limitations

This thesis focused on live wireless signals in a typical domestic operating environment, which contained different signal sources, devices, obstacles and service usage, incorporates some limiting factors. One such limitation could be the adopted SDR hardware specifications introduced in Chapter 5 and implemented in Chapters 6 and 7. The received samples are heavily linked to the receiver's analog-to-digital converter (ADC) resolution and reference voltage performance. A higher resolution would allow for received signals to be extracted in greater detail from the channel. The reference voltage, which is the maximum voltage available to the ADC, determines the ADC conversion ceiling for received analog inputs. Essentially, a higher reference voltage allows for higher-powered signals to be received before saturation occurs, enabling a more precise representation of the signal. However, this thesis's novel feature set has proven its ability to differentiate between signals when receiver saturation occurs and has implemented both 12-bit and 8-bit receivers. The features produced accurate results in both cases and for different numerical ranges. As a result, this hardware limitation concern can be determined to be resolved.

A further limitation could be attributed to the data collected from a typical domestic operating wireless environment that is changeable. Although the collected data corresponds to an active environment, the developed intelligent models would be heightened if data from multiple industrial environments were integrated into this thesis's dataset. More data will, typically, produce a model which can generalize to new instances in new environments with a reduced error. Hence, this additional data collection would enable a more general representation of the signals to be established. However, this thesis's work provides sufficiently descriptive features based on time-domain, frequency-domain and spatial analysis of received I/Q samples. The achieved results compete successfully with the literature and use a low complexity novel feature set without focusing on spectrograms or RSSI samples. This thesis provides sufficient evidence that the developed features and models can detect jamming signals, classify received signals and generalize to new data. The extensive validation of the developed models provides sufficient evidence that the developed features and model structure are useful. The additional data would enhance what was developed in this thesis.

8.3 Future Work

Several areas of the developed intelligent diagnostic tools for deployment on resource-contained edge devices can be researched further. This further research might yield improvements to the overall design, a real-world deployment and application in other device decisions and signal models.

The initial future work encompasses adapting the developed models and techniques for deployed low-power embedded edge devices, including power usage optimization and hardware metrics. The power usage needs to be optimal and not degrade the deployment capabilities, so future work should focus on optimizing the energy usage corresponding to the feature extraction and model usage. It is envisaged that the designed approach would not operate continuously and, instead, only operate on a specific duty cycle, or when called upon. As we look to the future for the hardware metrics, it seems reasonable to suggest that edge devices will have similar specifications as the Raspberry Pi device utilized for embedded implementation in Chapters 6 and 7. As a result, if Python capability is available on the edge nodes, the models can be easily integrated into the system. Additionally, embedded devices typically utilize C/C++ code, which can embed a Python interpreter to run Python programs. However, suppose this embedding process or Python were not available. In that case, the models and feature extraction methods would need to be modified for the software and code being applied. As shown in Chapter 4, Matlab code can be successfully translated to Python3, providing evidence that code translation is possible. This concept expands to integrating the designed, evaluated approaches into the device operation to make real-time edge decisions. Further exciting research surrounding the developed diagnostic framework is customizable hardware architectures, such as field-programmable gate arrays (FPGAs). This hardware provides opportunities for data width-specific computation by implementing unique logic configurations [168]. The techniques gained from such low numeric experimentation may enable otherwise resource-heavy machine learning at the edge without incurring a classification accuracy penalty [168].

To expand the capabilities of the designed interference detection framework, future work should analyze a more extensive set of potential interference attack styles and collect data on edge devices in different operating environments. As discussed in Section 8.2, additional data is beneficial to the model development. The different interference attacks will yield improvements to the overall design by broadening the classification range. 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 fourteen 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 under an expanded set of interference signals. It is envisaged that the machine learning models and features could be optimized further by focusing on reducing the applied FFT size, if possible.

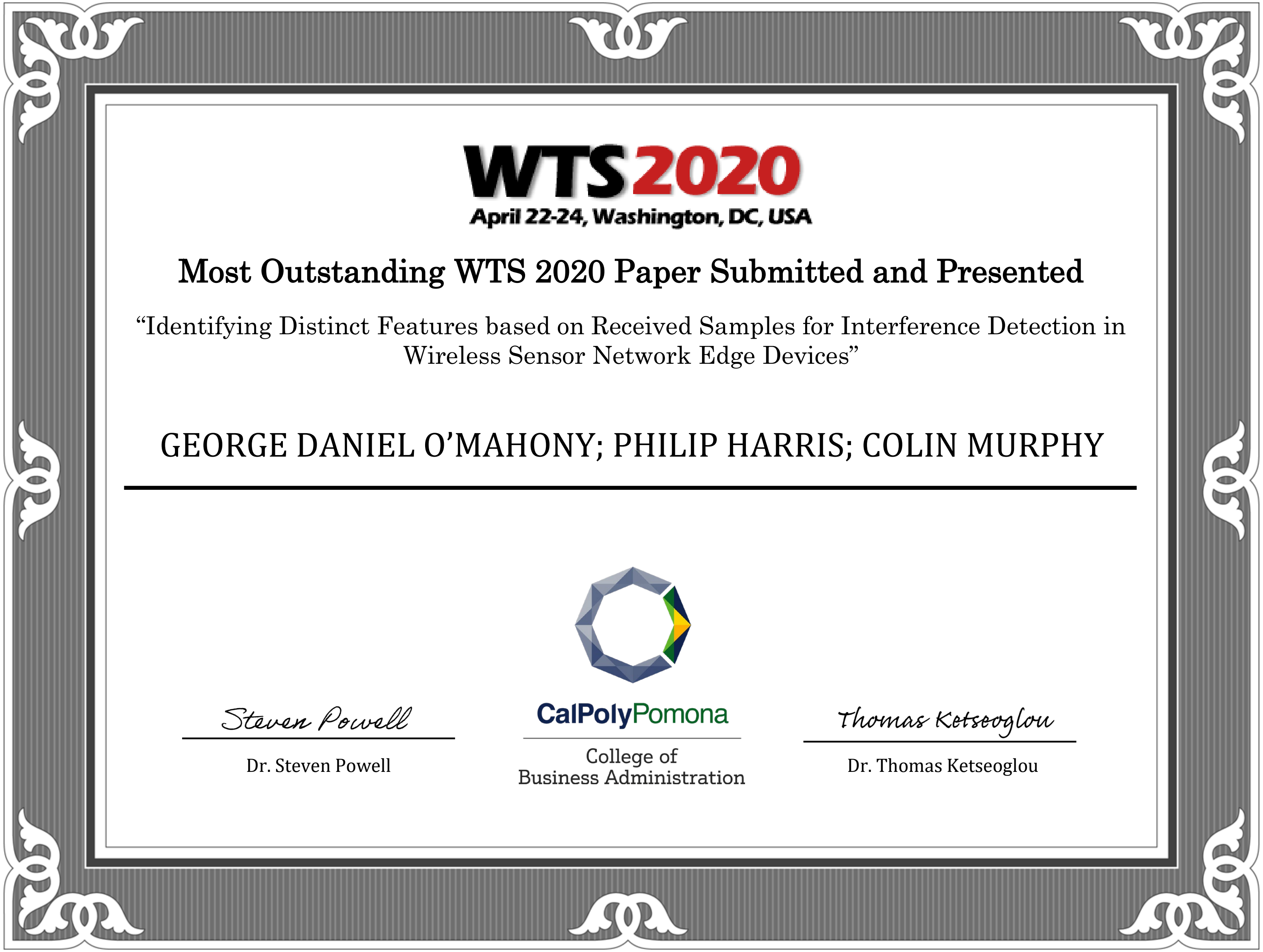
The final potential future research area is the collaborations with other applications. As discussed in Chapter 3, the standard clear channel assessment technology looks for a ZigBee signal or compares the received energy against a predefined threshold. The developed signal classification model in Chapter 6 could be integrated into channel access techniques in the future to provide a more accurate representation of the wireless channel. This collaboration has benefits for operating in environments with high coexistence levels. Depending on the sensed signal, packets can be transmitted to reduce latency, or with higher transmission gain, to aid in successful packet delivery. As the signal classifier was successfully deployed for legitimate signals in the ISM RF band, the potential exists for broader signal classification. As a result, other wireless protocols can be investigated to determine if the developed low order feature can identify the signal. These signals can include, as suggested in [97], Long Term Evolution (LTE) and Digital Video Broadcasting Terrestrial (DVB-T) protocols. Another potential use-case is identifying legitimate signals to update interference detection mechanisms operating in new wireless environments. An interference detection approach is typically developed based on data that is available to the system designer. This data can be leveraged from a known dataset, commercial nodes, designed testbed, etc. However, the training data may be collected from a wireless environment that differs significantly or marginally from the proposed deployment. Hence, having another approach that can generalize well and identify specific legitimate signals can leverage available confidence scores to adapt an interference detection model to new operating environments.

Appendix A

Research Awards

In Chapter 1, the list of publications arising from this thesis was introduced. Three of the published conference papers received awards and the associated certificates are supplied in this section. The award winning papers were as follows:

- "Identifying Distinct Features based on Received Samples for Interference Detection in Wireless Sensor Network Edge Devices" received the "Most Outstanding WTS 2020 Paper Submitted and Presented" award at the 2020 IEEE sponsored Wireless Telecommunications Symposium (WTS).
- "Detecting Interference in Wireless Sensor Network Received Samples: A Machine Learning Approach" received an "Outstanding Student Paper Award" at the IEEE 6th World Forum on Internet of Things (WF-IoT 2020).
- "Investigating Supervised Machine Learning Techniques for Channel Identification in Wireless Sensor Network" received the "Best Student Paper" award at the IEEE sponsored 31st Irish Signals and Systems Conference.



WTS2020
April 22-24, Washington, DC, USA

Most Outstanding WTS 2020 Paper Submitted and Presented

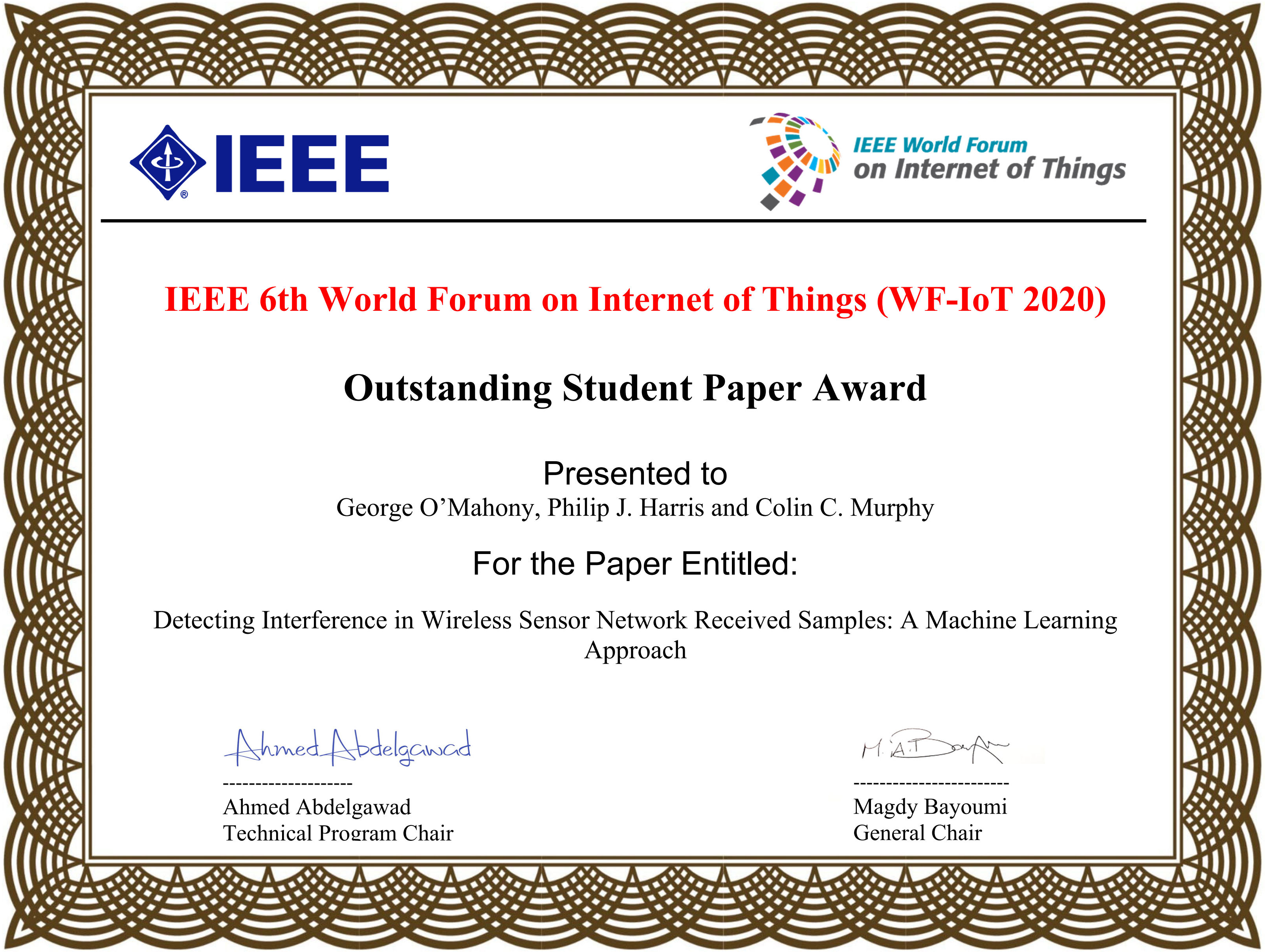
“Identifying Distinct Features based on Received Samples for Interference Detection in Wireless Sensor Network Edge Devices”

GEORGE DANIEL O’MAHONY; PHILIP HARRIS; COLIN MURPHY

Steven Powell
Dr. Steven Powell

CalPolyPomona
College of
Business Administration

Thomas Ketseoglou
Dr. Thomas Ketseoglou



IEEE

IEEE World Forum
on Internet of Things

IEEE 6th World Forum on Internet of Things (WF-IoT 2020)

Outstanding Student Paper Award

Presented to
George O'Mahony, Philip J. Harris and Colin C. Murphy

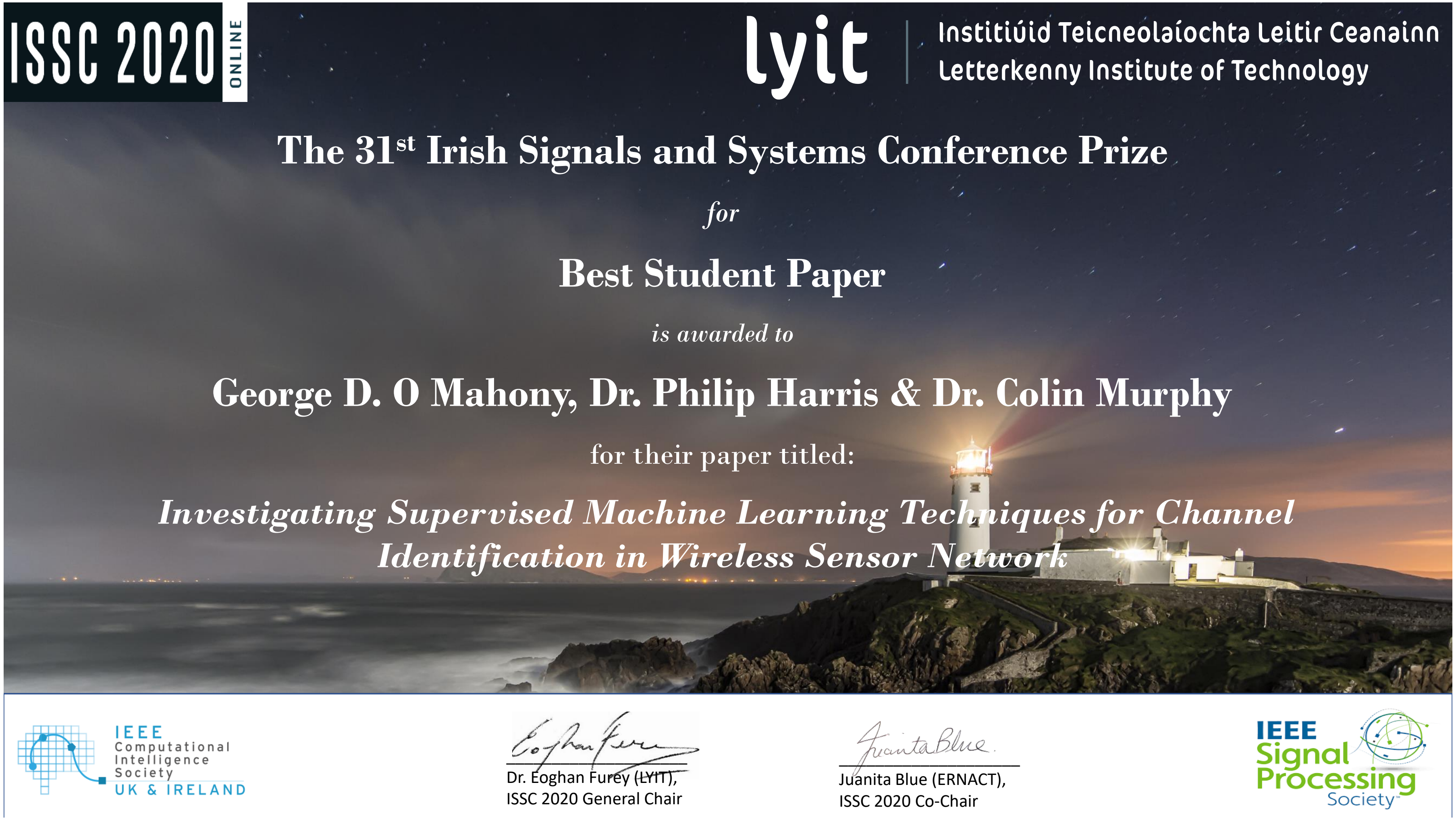
For the Paper Entitled:

Detecting Interference in Wireless Sensor Network Received Samples: A Machine Learning Approach

Ahmed Abdelgawad

Ahmed Abdelgawad
Technical Program Chair

Magdy Bayoumi
General Chair



ISSC 2020 ONLINE

lyit | Institiúid Teicneolaíochta Leitir Ceanainn
Letterkenny Institute of Technology

The 31st Irish Signals and Systems Conference Prize

for

Best Student Paper

is awarded to

George D. O Mahony, Dr. Philip Harris & Dr. Colin Murphy

for their paper titled:

Investigating Supervised Machine Learning Techniques for Channel Identification in Wireless Sensor Network

IEEE Computational Intelligence Society UK & IRELAND

Eoghan Furey
Dr. Eoghan Furey (LYIT),
ISSC 2020 General Chair

Juanita Blue
Juanita Blue (ERNACT),
ISSC 2020 Co-Chair

IEEE Signal Processing Society

Bibliography

[1] H. Pirayesh and H. Zeng, “Jamming Attacks and Anti-Jamming Strategies in Wireless Networks: A Comprehensive Survey,” *arXiv.org*, no. 2101.00292, Jan., 2021. [Online]. Available: http://arxiv.org/abs/2101.00292

[2] A. Förster, *Introduction to Wireless Sensor Networks*. Wiley, ISBN:9781119345343, 2016, doi: 10.1002/9781119345343.

[3] S. Tennina, M. Santos, A. Mesodiakaki, P.-V. Mekikis, E. Kartsakli, A. Antonopoulos, M. Di Renzo, A. Stavridis, F. Graziosi, L. Alonso, and C. Verikoukis, “WSN4QoL: WSNs for remote patient monitoring in e-Health applications,” in *2016 IEEE Int. Conf. Commun. (ICC), May*, 2016, pp. 1–6, doi: 10.1109/ICC.2016.7511597.

[4] P. Katopodis, G. Katsis, O. Walker, M. Tummala, and J. B. Michael, “A Hybrid, Large-scale Wireless Sensor Network for Missile Defense,” in *2007 IEEE Int. Conf. Syst. Syst. Eng., Apr.*, 2007, pp. 1–5, doi: 10.1109/SYSOSE.2007.4304261.

[5] H. J. Beestermöller, J. Sebald, M.-C. Sinnreich, H.-J. Borchers, M. Schneider, H. Luttmann, and V. Schmid, “Wireless-Sensor Networks in Space Technology Demonstration on ISS,” in *Dresdner Sensor-Symposium, Dec.*, 2015, pp. 99–102, doi: 10.5162/12dss2015/7.3.

[6] T. Vladimirova, C. P. Bridges, J. R. Paul, S. A. Malik, and M. N. Sweeting, “Space-based wireless sensor networks: Design issues,” in *2010 IEEE Aerosp. Conf., Mar.*, 2010, pp. 1–14, doi: 10.1109/AERO.2010.5447031.

[7] R. K. Yedavalli and R. K. Belapurkar, “Application of wireless sensor networks to aircraft control and health management systems,” *J. Control Theory Appl.*, vol. 9, no. 1, pp. 28–33, 2011, doi: 10.1007/s11768-011-0242-9.

[8] A. Addaim, A. Kherras, and Z. Guennoun, “Design of WSN with Relay Nodes Connected Directly with a LEO Nanosatellite,” *Int. J. Comput. Commun. Eng.*, vol. 3, no. 5, pp. 310–316, 2014, doi: 10.7763/IJCCE.2014.V3.341.

[9] L. Atzori, A. Iera, and G. Morabito, “The Internet of Things: A survey,” *Comput. Networks*, vol. 54, no. 15, pp. 2787–2805, 2010, doi: 10.1016/j.comnet.2010.05.010.

[10] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, “Internet of Things for Smart Cities,” *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, 2014, doi: 10.1109/JIOT.2014.2306328.

[11] H. Pirayesh, P. K. Sangdeh, and H. Zeng, "Securing ZigBee Communications against Constant Jamming Attack Using Neural Network," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4957–4968, 2020, doi: 10.1109/JIOT.2020.3034128.

[12] J. Heo, J. Kim, J. Paek, and S. Bahk, "Mitigating stealthy jamming attacks in low-power and lossy wireless networks," *J. Commun. Networks*, vol. 20, no. 2, pp. 219–230, 2018, doi: 10.1109/JCN.2018.000028.

[13] A. D. Wood, J. A. Stankovic, and G. Zhou, "DEEJAM: Defeating Energy-Efficient Jamming in IEEE 802.15.4-based Wireless Networks," in *2007 4th Annu. IEEE Commun. Soc. Conf. Sensor, Mesh Ad Hoc Commun. Networks, Jun.*, 2007, pp. 60–69, doi: 10.1109/SAHCN.2007.4292818.

[14] Y. Liu, P. Ning, H. Dai, and A. Liu, "Randomized Differential DSSS: Jamming-Resistant Wireless Broadcast Communication," in *2010 Proc. IEEE INFOCOM, Mar.*, 2010, pp. 1–9, doi: 10.1109/INFCOM.2010.5462156.

[15] J. Park, W. Nam, J. Choi, T. Kim, D. Yoon, S. Lee, J. Paek, and J. Ko, "Glasses for the Third Eye," in *Proc. 15th ACM Conf. Embed. Netw. Sens. Syst., Nov.*, 2017, pp. 1–14, doi: 10.1145/3131672.3131690.

[16] H. Huang, S. Xiao, X. Meng, and Y. Xiong, "A Remote Home Security System Based on Wireless Sensor Network and GSM Technology," in *2010 Second Int. Conf. Networks Secur. Wirel. Commun. Trust. Comput., Apr.*, 2010, pp. 535–538, doi: 10.1109/NSWCTC.2010.132.

[17] S. Chiang, C. Huang, and K. Chang, "A Minimum Hop Routing Protocol for Home Security Systems Using Wireless Sensor Networks," *IEEE Trans. Consum. Electron.*, vol. 53, no. 4, pp. 1483–1489, 2007, doi: 10.1109/TCE.2007.4429241.

[18] Zhuo Lu, Wenye Wang, and C. Wang, "Hiding traffic with camouflage: Minimizing message delay in the smart grid under jamming," in *2012 Proc. IEEE INFOCOM, Orlando, FL, USA, Mar.*, 2012, pp. 3066–3070, doi: 10.1109/INFCOM.2012.6195760.

[19] A. Wood, J. Stankovic, and S. Son, "JAM: a jammed-area mapping service for sensor networks," in *24th IEEE Real-Time Syst. Symp., Dec.*, 2003, pp. 286–297, doi: 10.1109/REAL.2003.1253275.

[20] B. Hjorth, "EEG analysis based on time domain properties," *Electroencephalogr. Clin. Neurophysiol.*, vol. 29, no. 3, pp. 306–310, 1970, doi: 10.1016/0013-4694(70)90143-4.

[21] B. Stelte and G. D. Rodosek, "Thwarting attacks on ZigBee - Removal of the KillerBee stinger," in *Proc. 9th Int. Conf. Netw. Serv. Manag., Oct.*, 2013, pp. 219–226, doi: 10.1109/CNSM.2013.6727840.

[22] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995, doi: 10.1007/BF00994018.

[23] L. Breiman, "Random Forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[24] M. Li and H. Lin, “Design and Implementation of Smart Home Control Systems Based on Wireless Sensor Networks and Power Line Communications,” *IEEE Trans. Ind. Electron.*, vol. 62, no. 7, pp. 4430–4442, 2015, doi: 10.1109/TIE.2014.2379586.

[25] P. K. D. Pramanik, A. Nayyar, and G. Pareek, “Chapter 7 - WBAN: Driving e-healthcare Beyond Telemedicine to Remote Health Monitoring: Architecture and Protocols,” in *Telemed. Technol.* Academic Press, 2019, pp. 89–119, doi: 10.1016/B978-0-12-816948-3.00007-6.

[26] S. Li, B. Chen, and L. Yu, “A modified 802.11 protocol applicated in space wireless local area network,” *Int. Conf. Comput. Des. Appl. ICCDA, Jun.*, vol. 2, pp. 585–588, 2010, doi: 10.1109/ICCDA.2010.5541313.

[27] N. Celandroni, E. Ferro, A. Gotta, G. Oligeri, C. Roseti, and M. Luglio, “A survey of architectures and scenarios in satellite-based WSN,” *Int. J. Satell. Commun. Netw.*, vol. 31, pp. 1–38, 2012, doi: 10.1002/sat.1019.

[28] K. F. Haque, K. H. Kabir, and A. Abdelgawad, “Advancement of Routing Protocols and Applications of Underwater Wireless Sensor Network (UWSN)—A Survey,” *J. Sens. Actuator Networks*, vol. 9, no. 2, 2020, doi: 10.3390/jsan9020019.

[29] T. Bowler, “The low-cost mini satellites bringing mobile to the world,” 2018. [Online]. Available: http://www.bbc.com/news/business-43090226

[30] C. P. Kruger and G. P. Hancke, “Implementing the Internet of Things Vision in Industrial Wireless Sensor Networks,” in *12th IEEE Int. Conf. Ind. Informatics (INDIN), July*, 2014, pp. 627–632, doi: 10.1109/INDIN.2014.6945586.

[31] Z. Qu, G. Zhang, H. Cao, and J. Xie, “LEO Satellite Constellation for Internet of Things,” *IEEE Access*, vol. 5, pp. 18 391–18 401, 2017, doi: 10.1109/ACCESS.2017.2735988.

[32] M. Marszalek, M. Rummelhagen, and F. Schramm, “Potentials and limitations of IEEE 802.11 for satellite swarms,” *IEEE Aerosp. Conf., Mar.*, pp. 1–9, 2014, doi: 10.1109/AERO.2014.6836320.

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

[34] IEEE, *IEEE Standard for Low-Rate Wireless Networks*, 2016, doi: 10.1109/IEEESTD.2016.7460875.

[35] Tektronix, “DPX Overview,” 2019. [Online]. Available: https://www.tek.com/dpx-overview

[36] I. Tomić and J. A. McCann, “A Survey of Potential Security Issues in Existing Wireless Sensor Network Protocols,” *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1910–1923, 2017, doi: 10.1109/JIOT.2017.2749883.

[37] ZigBee Alliance, “ZigBee Specification. ZigBee document 05-3474-21,” Tech. Rep., 2015. [Online]. Available: https://zigbeealliance.org/wp-content/uploads/2019/11/docs-05-3474-21-0csg-zigbee-specification.pdf

[38] G. Shi and K. Li, “Signal Interference in WiFi and ZigBee Networks,” 2017, doi: 10.1007/978-3-319-47806-7.

[39] S. Farahani, *ZigBee Wireless Networks and Transceivers*. ScienceDirect, 2008, doi: 10.1016/B978-0-7506-8393-7.X0001-5.

[40] A. Reziouk, E. Laurent, and J.-C. Demay, “Practical security overview of IEEE 802.15.4,” *Int. Conf. Eng. MIS, Sept.*, pp. 1–9, 2016, doi: 10.1109/ICEMIS.2016.7745382.

[41] G. D. O’Mahony, S. O’Mahony, J. T. Curran, and C. C. Murphy, “Developing a low-cost platform for GNSS interference detection,” in *Eur. Navig. Conf., Apr.*, 2015, pp. 1–8.

[42] Gps.gov, “Space Segment,” 2020. [Online]. Available: https://www.gps.gov/systems/gps/space/

[43] A. Abduvaliyev, A.-S. K. Pathan, J. Zhou, R. Roman, and L. W.-C. Wong, “On the Vital Areas of Intrusion Detection Systems in Wireless Sensor Networks,” *IEEE Communiations Surv. Tutorials*, vol. 15, no. 3, pp. 1223–1237, 2013, doi: 10.1109/SURV.2012.121912.00006.

[44] D. Liu, J. Raymer, and A. Fox, “Efficient and Timely Jamming Detection in Wireless Sensor Networks,” in *IEEE 9th Int. Conf. Mob. Ad-Hoc Sens. Syst., Oct.*, 2012, pp. 335–343, doi: 10.1109/MASS.2012.6502533.

[45] G. D. O’Mahony, P. J. Harris, and C. C. Murphy, “Analyzing the Vulnerability of Wireless Sensor Networks to a Malicious Matched Protocol Attack,” in *2018 Int. Carnahan Conf. Secur. Technol. (ICCST), Montr. QC, Oct.* IEEE, 2018, pp. 1–5, doi: 10.1109/CCST.2018.8585681.

[46] T. Hamza, G. Kaddoum, A. Meddeb, and G. Matar, “A Survey on Intelligent MAC Layer Jamming Attacks and Countermeasures in WSNs,” in *IEEE 84th Veh. Technol. Conf., Sept.*, 2016, pp. 1–5, doi: 10.1109/VTCFall.2016.7880885.

[47] S. Shanthi and E. G. Rajan, “Comprehensive Analysis of Security Attacks and Intrusion Detection System in Wireless Sensor Networks,” in *IEEE 2nd Int. Conf. Next Gener. Comput. Technol., Oct.*, 2016, pp. 426–431, doi: 10.1109/NGCT.2016.7877454.

[48] Y. Zhou, Y. Fang, and Y. Zhang, “Securing Wireless Sensor Networks: A Survey,” *IEEE Commun. Surv.*, vol. 10, no. 3, pp. 6–28, 2008, doi: 10.1109/COMST.2008.4625802.

[49] A. P. Abidoye and I. C. Obagbuwa, “DDoS attacks in WSNs: detection and countermeasures,” *IET Wirel. Sens. Syst.*, vol. 8, no. 2, pp. 52–59, 2018, doi: 10.1049/iet-wss.2017.0029.

[50] A. Tyagi, J. Kushwah, and M. Bhalla, “Threats to security of Wireless Sensor Networks,” *7th Int. Conf. Cloud Comput. Data Sci. Eng., Jun.*, pp. 402–405, 2017, doi: 10.1109/CONFLUENCE.2017.7943183.

[51] Keysight Technologies, “Electronic Warfare: Vying for Control of the Electromagnetic Spectrum,” *Whitepaper*, pp. 1–5, 2020.

[52] G. Bullard, “Securing the Digital EcoSytem – A war we’re losing,” 2018.

[53] M. Wilhelm, I. Martinovic, J. B. Schmitt, and V. Lenders, “Short paper: Reactive jamming in wireless networks - How realistic is the threat?” *4th ACM Conf. Wirel. Netw. Secur., Jun.*, pp. 47–52, 2011, doi: 10.1145/1998412.1998422.

[54] S. Dehnie, V. Chakravarthy, Z. Wu, C. Ghosh, and H. Li, “Spectrum Coexistence Issues : Challenges and Research Directions,” in *IEEE Mil. Commun. Conf., Nov.*, 2013, pp. 1681–1689, doi: 10.1109/MILCOM.2013.285.

[55] Z. Yang, P. Cheng, and J. Chen, “Learning-based Jamming Attack against Low-duty-cycle Networks,” *IEEE Trans. Dependable Secur. Comput.*, vol. 14, no. 6, pp. 650–663, 2015, doi: 10.1109/TDSC.2015.2501288.

[56] N. Ahmed, S. S. Kanhere, and S. Jha, “The holes problem in wireless sensor networks: a survey,” *ACM SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 9, no. 2, pp. 4–18, 2005, doi: 10.1145/1072989.1072992.

[57] A. Tayebi, S. Berber, and A. Swain, “Wireless Sensor Network attacks: An overview and critical analysis,” in *Seventh Int. Conf. Sens. Technol., Dec.*, 2013, pp. 97–102, doi: 10.1109/ICSensT.2013.6727623.

[58] A. S. Rawat, P. Anand, H. Chen, and P. K. Varshney, “Collaborative spectrum sensing in the presence of byzantine attacks in cognitive radio networks,” *IEEE Trans. Signal Process.*, vol. 59, no. 2, pp. 774–786, 2011, doi: 10.1109/TSP.2010.2091277.

[59] J. P. Anderson, “Computer security threat monitoring and surveillance,” *James P Anderson Company, Fort Washington, Pennsylvania*, 1980.

[60] O. Can and O. K. Sahingoz, “A Survey of Intrusion Detection Systems in Wireless Sensor Networks,” in *6th Int. Conf. Model. Simul. Appl. Optim., May*, 2015, pp. 1–6, doi: 10.1109/ICMSAO.2015.7152200.

[61] I. S. Kocher, C.-O. Chow, H. Ishii, and T. A. Zia, “Threat Models and Security Issues in Wireless Sensor Networks,” *Int. J. Comput. Theory Eng.*, vol. 5, no. 5, pp. 830–835, 2013, doi: 10.7763/ijcte.2013.v5.806.

[62] M. A. Mahmood, W. K. Seah, and I. Welch, “Reliability in wireless sensor networks: A survey and challenges ahead,” *Comput. Networks*, vol. 79, pp. 166–187, 2015, doi: 10.1016/j.comnet.2014.12.016.

[63] Texas Instruments, “SmartRF Protocol Packet Sniffer.” [Online]. Available: http://www.ti.com/tool/PACKET-SNIFFER

[64] K. Wu, H. Tan, H. L. Ngan, Y. Liu, and L. M. Ni, “Chip error pattern analysis in IEEE 802.15.4,” *IEEE Trans. Mob. Comput.*, vol. 11, no. 4, pp. 543–552, 2012, doi: 10.1109/TMC.2011.44.

[65] DIGI, “XCTU Software.” [Online]. Available: https://www.digi.com/products/xbee-rf-solutions/xctu-software/xctu

[66] Y. F. Zhu and X. M. Tang, “Overview of swarm intelligence,” *2010 Int. Conf. Comput. Appl. Syst. Model. (ICCASM 2010), Oct.*, pp. 400–403, 2010, doi: 10.1109/ICCASM.2010.5623005.

[67] A. Hazza, M. Shoaib, S. A. Alshebeili, and A. Fahad, “An overview of feature-based methods for digital modulation classification,” in *2013 1st Int. Conf. Commun. Signal Process. their Appl., Feb.*, 2013, pp. 1–6, doi: 10.1109/ICCSPA.2013.6487244.

[68] A. Hazza, M. Shoaib, A. Saleh, and A. Fahd, “Robustness of digitally modulated signal features against variation in HF noise model,” *EURASIP J. Wirel. Commun. Netw.*, vol. 2011, no. 1, p. 24, 2011, doi: 10.1186/1687-1499-2011-24.

[69] A. H. Wahla, L. Chen, Y. Wang, R. Chen, and F. Wu, “Automatic Wireless Signal Classification in Multimedia Internet of Things: An Adaptive Boosting Enabled Approach,” *IEEE Access*, vol. 7, pp. 160 334–160 344, 2019, doi: 10.1109/ACCESS.2019.2950989.

[70] K. J. Son, H. Cho, S. H. Hong, S. P. Moon, and T. G. Chang, “New enhanced clear channel assessment method for IEEE 802.15.4 network,” *Int. SoC Des. Conf. (ISOCC), Nov.*, pp. 251–252, 2015, doi: 10.1109/ISOCC.2015.7401742.

[71] Y. Tang, Z. Wang, T. Du, D. Makrakis, and H. T. Mouftah, “Study of clear channel assessment mechanism for ZigBee packet transmission under Wi-Fi interference,” *IEEE 10th Consum. Commun. Netw. Conf. (CCNC), Jan.*, pp. 765–768, 2013, doi: 10.1109/CCNC.2013.6488545.

[72] Y. Tang, Z. Wang, D. Makrakis, and H. T. Mouftah, “Interference aware adaptive clear channel assessment for improving zigbee packet transmission under Wi-Fi interference,” *IEEE Int. Conf. Sensing, Commun. Networking, (SECON), Jun.*, pp. 336–343, 2013, doi: 10.1109/SAHCN.2013.6645003.

[73] J. L. Xu, W. Su, and M. Zhou, “Likelihood-Ratio Approaches to Automatic Modulation Classification,” *IEEE Trans. Syst. Man, Cybern. Part C (Applications and Reviews)*, vol. 41, no. 4, pp. 455–469, 2011, doi: 10.1109/TSMCC.2010.2076347.

[74] L. Wang and Y. Ren, “Recognition of digital modulation signals based on high order cumulants and support vector machines,” in *2009 ISECS Int. Colloq. Comput. Commun. Control. Manag., Aug.*, vol. 4, 2009, pp. 271–274, doi: 10.1109/CCCM.2009.5267733.

[75] M. W. Aslam, Z. Zhu, and A. K. Nandi, “Automatic Modulation Classification Using Combination of Genetic Programming and KNN,” *IEEE Trans. Wirel. Commun.*, vol. 11, no. 8, pp. 2742–2750, 2012, doi: 10.1109/TWC.2012.060412.110460.

[76] M. L. D. Wong, S. K. Ting, and A. K. Nandi, “Naïve Bayes classification of adaptive broadband wireless modulation schemes with higher order cumulants,” in *2008 2nd Int. Conf. Signal Process. Commun. Syst., Dec.*, 2008, pp. 1–5, doi: 10.1109/ICSPCS.2008.4813755.

[77] Z. Zhang, C. Wang, C. Gan, S. Sun, and M. Wang, “Automatic Modulation Classification Using Convolutional Neural Network With Features Fusion of SPWVD and BJD,” *IEEE Trans. Signal Inf. Process. over Networks*, vol. 5, no. 3, pp. 469–478, 2019, doi: 10.1109/TSIPN.2019.2900201.

[78] C. Park, J. Choi, S. Nah, W. Jang, and D. Y. Kim, “Automatic Modulation Recognition of Digital Signals using Wavelet Features and SVM,” in *2008 10th Int. Conf. Adv. Commun. Technol., Feb.*, vol. 1, 2008, pp. 387–390, doi: 10.1109/I-CACT.2008.4493784.

[79] L. Xie and Q. Wan, “Cyclic Feature-Based Modulation Recognition Using Compressive Sensing,” *IEEE Wirel. Commun. Lett.*, vol. 6, no. 3, pp. 402–405, 2017, doi: 10.1109/LWC.2017.2697853.

[80] Y. Zhang, G. Wu, J. Wang, and Q. Tang, “Wireless signal classification based on high-order cumulants and machine learning,” *2017 Int. Conf. Comp. Tech. Elec. Commun. (ICCTEC),*, pp. 559–564, 2017, doi: 10.1109/ICCTEC.2017.00127.

[81] S. H. Lee, K.-Y. Kim, and Y. Shin, “Effective Feature Selection Method for Deep Learning-Based Automatic Modulation Classification Scheme Using Higher-Order Statistics,” *Appl. Sci.*, vol. 10, no. 2, pp. 1–14, 2020, doi: 10.3390/app10020588.

[82] A. Smith, M. Evans, and J. Downey, “Modulation classification of satellite communication signals using cumulants and neural networks,” in *2017 Cogn. Commun. Aerosp. Appl. Work. (CCAA), Jun.*, 2017, pp. 1–8, doi: 10.1109/C-CAAW.2017.8001878.

[83] W. Xie, S. Hu, C. Yu, P. Zhu, X. Peng, and J. Ouyang, “Deep Learning in Digital Modulation Recognition Using High Order Cumulants,” *IEEE Access*, vol. 7, pp. 63 760–63 766, 2019, doi: 10.1109/ACCESS.2019.2916833.

[84] M. S. Pajic, M. Veinovic, M. Peric, and V. D. Orlic, “Modulation Order Reduction Method for Improving the Performance of AMC Algorithm Based on Sixth–Order Cumulants,” *IEEE Access*, vol. 8, pp. 106 386–106 394, 2020, doi: 10.1109/ACCESS.2020.3000358.

[85] M. A. Hazar, N. Odabasioglu, T. Ensari, Y. Kavurucu, and O. F. Sayan, “Performance analysis and improvement of machine learning algorithms for automatic modulation recognition over Rayleigh fading channels,” *Neural Comput. Appl.*, vol. 29, no. 9, pp. 351–360, 2018, doi: 10.1007/s00521-017-3040-6.

[86] H. Hu, Y. Wang, and J. Song, “Signal classification based on spectral correlation analysis and SVM in cognitive radio,” *22nd Int. Conf. Adv. Inf. Netw. Appl. (AINA), Mar.*, pp. 883–887, 2008, doi: 10.1109/AINA.2008.27.

[87] Z. Zhang, Y. Li, X. Zhu, and Y. Lin, "A Method for Modulation Recognition Based on Entropy Features and Random Forest," in *2017 IEEE Int. Conf. Softw. Qual. Reliab. Secur. Companion (QRS-C), Jul.*, 2017, pp. 243–246, doi: 10.1109/QRS-C.2017.47.

[88] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.

[89] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-Air Deep Learning Based Radio Signal Classification," *IEEE J. Sel. Top. Signal Process.*, vol. 12, no. 1, pp. 168–179, 2018, doi: 10.1109/JSTSP.2018.2797022.

[90] S. Zheng, P. Qi, S. Chen, and X. Yang, "Fusion Methods for CNN-Based Automatic Modulation Classification," *IEEE Access*, vol. 7, pp. 66 496–66 504, 2019, doi: 10.1109/ACCESS.2019.2918136.

[91] M. Zhang, M. Diao, and L. Guo, "Convolutional Neural Networks for Automatic Cognitive Radio Waveform Recognition," *IEEE Access*, vol. 5, pp. 11 074–11 082, 2017, doi: 10.1109/ACCESS.2017.2716191.

[92] C. Gravelle and R. Zhou, "SDR demonstration of signal classification in real-time using deep learning," *2019 IEEE Globecom Work. (GC Wkshps), Dec.*, pp. 1–5, 2019, doi: 10.1109/GCWkshps45667.2019.9024661.

[93] S. Zheng, S. Chen, P. Qi, H. Zhou, and X. Yang, "Spectrum sensing based on deep learning classification for cognitive radios," *China Commun.*, vol. 17, no. 2, pp. 138–148, 2020, doi: 10.23919/JCC.2020.02.012.

[94] S. Huang, L. Chai, Z. Li, D. Zhang, Y. Yao, Y. Zhang, and Z. Feng, "Automatic Modulation Classification Using Compressive Convolutional Neural Network," *IEEE Access*, vol. 7, pp. 79 636–79 643, 2019, doi: 10.1109/ACCESS.2019.2921988.

[95] H. He, S. Jin, C. Wen, F. Gao, G. Y. Li, and Z. Xu, "Model-Driven Deep Learning for Physical Layer Communications," *IEEE Wirel. Commun.*, vol. 26, no. 5, pp. 77–83, 2019, doi: 10.1109/MWC.2019.1800447.

[96] S. Rajendran, W. Meert, D. Giustiniano, V. Lenders, and S. Pollin, "Deep Learning Models for Wireless Signal Classification With Distributed Low-Cost Spectrum Sensors," *IEEE Trans. Cogn. Commun. Netw.*, vol. 4, no. 3, pp. 433–445, 2018, doi: 10.1109/TCCN.2018.2835460.

[97] J. Fontaine, E. Fonseca, A. Shahid, M. Kist, L. A. DaSilva, I. Moerman, and E. De Poorter, "Towards low-complexity wireless technology classification across multiple environments," *Ad Hoc Networks*, vol. 91, p. 101881, 2019, doi: 10.1016/j.adhoc.2019.101881.

[98] Y. E. Sagduyu, Y. Shi, T. Erpek, W. Headley, B. Flowers, G. Stantchev, and Z. Lu, "When Wireless Security Meets Machine Learning: Motivation, Challenges, and Research Directions," 2020. [Online]. Available: https://arxiv.org/abs/2001.08883

[99] M. Vanhoef and F. Piessens, “Advanced Wi-Fi attacks using commodity hardware,” in *Proc. 30th Annu. Comput. Secur. Appl. Conf. - ACSAC ’14, Dec.* New York, New York, USA: ACM Press, 2014, pp. 256–265, doi: 10.1145/2664243.2664260.

[100] 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

[101] I. Butun, S. D. Morgera, and R. Sankar, “A Survey of Intrusion Detection Systems in Wireless Sensor Networks,” *IEEE Commun. Surv. Tutorials*, vol. 16, no. 1, pp. 266–282, 2014, doi: 10.1109/SURV.2013.050113.00191.

[102] A. D. Wood and J. A. Stankovic, “Denial of Service in Sensor Networks,” *IEEE Comput. Mag.*, vol. 35, no. 10, pp. 54–62, 2002, doi: 10.1109/MC.2002.1039518.

[103] O. Puñal, I. Aktas, C.-J. Schnelke, G. Abidin, K. Wehrle, and J. Gross, “Machine learning-based jamming detection for IEEE 802.11: Design and experimental evaluation,” in *IEEE 15th Int. Symp. a World Wireless, Mob. Multimed. Networks, Jun.* IEEE, 2014, pp. 1–10, doi: 10.1109/WoWMoM.2014.6918964.

[104] A. Sikora and V. F. Groza, “Coexistence of IEEE 802.15.4 with other systems in the 2.4 GHz-ISM-band,” in *IEEE Instrum. Meas. Technol. Conf., May*, 2005, pp. 1786–1791, doi: 10.1109/IMTC.2005.1604479.

[105] K. Siddhabathula, Q. Dong, D. Liu, and M. Wright, “Fast jamming detection in sensor networks,” in *2012 IEEE Int. Conf. Commun., Jun.*, 2012, pp. 934–938, doi: 10.1109/ICC.2012.6363672.

[106] F. Hermans, O. Rensfelt, T. Voigt, E. Ngai, L.-Å. Nordén, and P. Gunningberg, “SoNIC : Classifying Interference in 802.15.4 Sensor Networks,” *ACM/IEEE Int. Conf. Inf. Process. Sens. Networks, Apr.*, pp. 55–66, 2013, doi: 10.1145/2461381.2461392.

[107] S. Grimaldi, A. Mahmood, M. Gidlund, and M. Alves, “An SVM-based method for classification of external interference in industrial wireless sensor and actuator networks,” *J. Sens. Actuator Networks*, vol. 6, no. 2, pp. 1–25, 2017, doi: 10.3390/jsan6020009.

[108] M. A. Alsheikh, S. Lin, D. Niyato, and H.-P. Tan, “Machine learning in wireless sensor networks: Algorithms, strategies, and applications,” *IEEE Commun. Surv. Tutorials*, vol. 16, no. 4, pp. 1996–2018, 2014, doi: 10.1109/COMST.2014.2320099.

[109] Z. Yu and J. J. P. Tsai, “A Framework of Machine Learning Based Intrusion Detection for Wireless Sensor Networks,” in *IEEE Int. Conf. Sens. Networks, Ubiquitous, Trust. Comput., Jun.*, 2008, pp. 272–279, doi: 10.1109/SUTC.2008.39.

[110] W. W. Cohen and Y. Singer, “A Simple, Fast, and Effective Rule Learner,” in *Proc. Sixt. Natl. Conf. Artif. Intell. Elev. Innov. Appl. Artif. Intell. Conf. Innov. Appl. Artif. Intell.*, ser. AAAI ’99/IAAI ’99. USA: American Association for Artificial Intelligence, 1999, pp. 335–342, doi: 10.5555/315149.315320.

[111] Z. Xiao, C. Liu, and C. Chen, "An anomaly detection scheme based on machine learning for WSN," in *1st Int. Conf. Inf. Sci. Eng. ICISE, Dec.*, 2009, pp. 3959–3962, doi: 10.1109/ICISE.2009.235.

[112] H. Ayadi, A. Zouinkhi, B. Boussaid, and M. N. Abdelkrim, "A machine learning methods: Outlier detection in WSN," in *16th Int. Conf. Sci. Tech. Autom. Control Comput. Eng., Dec.*, 2015, pp. 722–727, doi: 10.1109/STA.2015.7505190.

[113] K. A. Jalil, M. H. Kamarudin, and M. N. Masrek, "Comparison of machine learning algorithms performance in detecting network intrusion," in *Int. Conf. Netw. Inf. Technol., Jun.*, 2010, pp. 221–226, doi: 10.1109/ICNIT.2010.5508526.

[114] M. C. Belavagi and B. Muniyal, "Performance Evaluation of Supervised Machine Learning Algorithms for Intrusion Detection," *Procedia Comput. Sci.*, vol. 89, pp. 117–123, 2016, doi: 10.1016/j.procs.2016.06.016.

[115] S. Rajasegarar, C. Leckie, M. Palaniswami, and J. C. Bezdek, "Quarter Sphere Based Distributed Anomaly Detection in Wireless Sensor Networks," in *2007 IEEE Int. Conf. Commun., Jun.*, 2007, pp. 3864–3869, doi: 10.1109/ICC.2007.637.

[116] Y. S. Chen, Y. S. Qin, Y. G. Xiang, J. X. Zhong, and X. L. Jiao, "Intrusion Detection System Based on Immune Algorithm and Support Vector Machine in Wireless Sensor Network," *Info. and Auto. ISIA 2010. Commun. in Comput. and Informat. Sci.*, vol. 86, pp. 372–376, 2011, doi: 10.1007/978-3-642-19853-3_54.

[117] S. Yi, H. Wang, W. Xue, X. Fan, L. Wang, J. Tian, and R. Matsukura, "Interference Source Identification for IEEE 802.15.4 wireless Sensor Networks Using Deep Learning," in *2018 IEEE 29th Annu. Int. Symp. Pers. Indoor Mob. Radio Commun., Sep.*, 2018, pp. 1–7, doi: 10.1109/PIMRC.2018.8580857.

[118] G. Pachauri and S. Sharma, "Anomaly Detection in Medical Wireless Sensor Networks using Machine Learning Algorithms," *Procedia Comput. Sci.*, vol. 70, pp. 325–333, 2015, doi: 10.1016/j.procs.2015.10.026.

[119] J. Heo, J. Kim, S. Bahk, and J. Paek, "Dodge-Jam: Anti-Jamming Technique for Low-Power and Lossy Wireless Networks," in *2017 14th Annu. IEEE Int. Conf. Sensing, Commun. Netw., Jun.*, 2017, pp. 1–9, doi: 10.1109/SAHCN.2017.7964926.

[120] B. DeBruhl and P. Tague, "Digital Filter Design for Jamming Mitigation in 802.15.4 Communication," in *2011 Proc. 20th Int. Conf. Comput. Commun. Networks, Jul.*, 2011, pp. 1–6, doi: 10.1109/ICCCN.2011.6006020.

[121] Y. Liu and P. Ning, "BitTrickle: Defending against broadband and high-power reactive jamming attacks," in *2012 Proc. IEEE INFOCOM, Mar.*, 2012, pp. 909–917, doi: 10.1109/INFCOM.2012.6195840.

[122] S. Fang, Y. Liu, and P. Ning, "Wireless Communications under Broadband Reactive Jamming Attacks," *IEEE Trans. Dependable Secur. Comput., May*, vol. 13, no. 3, pp. 394–408, 2016, doi: 10.1109/TDSC.2015.2399304.

[123] M. Spuhler, D. Giustiniano, V. Lenders, M. Wilhelm, and J. B. Schmitt, "Detection of Reactive Jamming in DSSS-based Wireless Communications," *IEEE Trans. Wirel. Commun.*, vol. 13, no. 3, pp. 1593–1603, 2014, doi: 10.1109/TWC.2013.013014.131037.

[124] Z. Chi, Y. Li, X. Liu, W. Wang, Y. Yao, T. Zhu, and Y. Zhang, "Countering cross-technology jamming attack," in *Proc. 13th ACM Conf. Secur. Priv. Wirel. Mob. Networks, Jul.* New York, NY, USA: ACM, 2020, pp. 99–110, doi: 10.1145/3395351.3399367.

[125] N. Rouissi, H. Gharsellaoui, and S. Bouamama, "A Hybrid DS-FH-THSS Based Approach Anti-jamming in Wireless Sensor Networks," in *2016 World Symp. Comput. Appl. Res. (WSCAR), Mar.*, 2016, pp. 93–97, doi: 10.1109/WSCAR.2016.19.

[126] J. Ng, Z. Cai, and M. Yu, "A New Model-Based Method to Detect Radio Jamming Attack to Wireless Networks," in *2015 IEEE Globecom Work. (GC Wkshps), Dec.*, 2015, pp. 1–6, doi: 10.1109/GLOCOMW.2015.7414032.

[127] S. G. Hymlin Rose and T. Jayasree, "Detection of jamming attack using timestamp for WSN," *Ad Hoc Networks*, vol. 91, p. 101874, 2019, doi: 10.1016/j.adhoc.2019.101874.

[128] P. Bhavathankar, S. Chatterjee, and S. Misra, "Link-Quality Aware Path Selection in the Presence of Proactive Jamming in Fallible Wireless Sensor Networks," *IEEE Trans. Commun.*, vol. 66, no. 4, pp. 1689–1704, 2018, doi: 10.1109/TCOMM.2017.2736550.

[129] V. A. Shanthakumar, C. Banerjee, T. Mukherjee, and E. Pasiliao, "Uncooperative RF Direction Finding with I/Q Data," in *Proc. 2020 4th Int. Conf. Inf. Syst. Data Mining, May*, 2020, pp. 6–13, doi: 10.1145/3404663.3404668.

[130] T. Mukherjee, M. Duckett, P. Kumar, J. D. Paquet, D. Rodriguez, M. Haulcomb, K. George, and E. Pasiliao, "RSSI-Based Supervised Learning for Uncooperative Direction-Finding," in *Mach. Learn. Knowl. Discov. Databases. ECML PKDD 2017. Lect. Notes Comput. Sci. vol 10536. Springer, Cham*, 2017, pp. 216–227, doi: 10.1007/978-3-319-71273-4_18.

[131] D. Roy, T. Mukherjee, M. Chatterjee, and E. Pasiliao, "Detection of Rogue RF Transmitters using Generative Adversarial Nets," in *2019 IEEE Wirel. Commun. Netw. Conf. (WCNC), Apr.*, 2019, pp. 1–7, doi: 10.1109/WCNC.2019.8885548.

[132] ——, "Primary User Activity Prediction in DSA Networks using Recurrent Structures," in *2019 IEEE Int. Symp. Dyn. Spectr. Access Networks (DySPAN), Nov.*, 2019, pp. 1–10, doi: 10.1109/DySPAN.2019.8935716.

[133] ——, "RF Transmitter Fingerprinting Exploiting Spatio-Temporal Properties in Raw Signal Data," in *2019 18th IEEE Int. Conf. Mach. Learn. Appl., Dec.*, 2019, pp. 89–96, doi: 10.1109/ICMLA.2019.00023.

[134] T. Jian, B. C. Rendon, E. Ojuba, N. Soltani, Z. Wang, K. Sankhe, A. Gritsenko, J. Dy, K. Chowdhury, and S. Ioannidis, “Deep Learning for RF Fingerprinting: A Massive Experimental Study,” *IEEE Internet Things Mag.*, vol. 3, no. 1, pp. 50–57, 2020, doi: 10.1109/IOTM.0001.1900065.

[135] A. Ng and M. Jordan, “On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes,” in *Adv. Neural Inf. Process. Syst.*, vol. 14. MIT Press, 2002, pp. 841–848, doi: 10.5555/2980539.2980648.

[136] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, ser. Springer Series in Statistics. Springer New York, pp. 133-134, 2009, doi: 10.1007/978-0-387-84858-7.

[137] A. Temko, E. Thomas, W. Marnane, G. Lightbody, and G. Boylan, “EEG-based neonatal seizure detection with Support Vector Machines,” *Clin. Neurophysiol.*, vol. 122, no. 3, pp. 464–473, 2011, doi: 10.1016/j.clinph.2010.06.034.

[138] T. Yiu, “Understanding Random Forest.” [Online]. Available: https://towardsdatascience.com/understanding-random-forest-58381e0602d2

[139] A. Singh, N. Thakur, and A. Sharma, “A review of supervised machine learning algorithms,” in *Int. Conf. Comput. Sustain. Glob. Dev., Mar.*, 2016, pp. 1310–1315.

[140] V. M. Suresh, R. Sidhu, P. Karkare, A. Patil, Z. Lei, and A. Basu, “Powering the IoT through embedded machine learning and LoRa,” in *IEEE World Forum Internet Things (WF-IoT), Feb.*, 2018, pp. 349–354, doi: 10.1109/WF-IoT.2018.8355177.

[141] C. Leech, Y. P. Raykov, E. Ozer, and G. V. Merrett, “Real-time room occupancy estimation with Bayesian machine learning using a single PIR sensor and microcontroller,” in *IEEE Sensors Appl. Symp., Mar.*, 2017, pp. 1–6, doi: 10.1109/SAS.2017.7894091.

[142] DIGI, “XBee ZigBee Mesh Kit.” [Online]. Available: https://www.digi.com/resources/library/data-sheets/ds-xbee3-zigbee-mesh-kit

[143] G. D. O’Mahony, P. J. Harris, and C. C. Murphy, “Developing Low-Cost Testbeds for Enhancing Security Techniques in Wireless Sensor Network Protocols,” in *30th IEEE Irish Signals Syst. Conf., Jun.*, 2019, pp. 1–6, doi: 10.1109/ISSC.2019.8904967.

[144] J. T. Curran, C. Fernandez-Prades, A. Morrison, and M. Bavaro, “The Continued Evolution of Software-Defined Radio for GNSS,” *GPS World*, no. 29, pp. 43–49, 2018.

[145] Nooelec, “NESDR SMArTee v2 SDR.” [Online]. Available: https://www.nooelec.com/store/nesdr-smartee-sdr.html

[146] Osmocom, “rtl-sdr.” [Online]. Available: https://osmocom.org/projects/rtl-sdr/wiki/Rtl-sdr

[147] M. Quigley, S. Gleason, and P. Abbeel, “fastgps,” 2013. [Online]. Available: https://sourceforge.net/projects/fastgps/

[148] Siretta, “Antenna Stubby, Bluetooth, WiFi, WLAN, Zigbee, 2.4 GHz to 2.5 GHz, 2 dBi, SMA.” [Online]. Available: http://ie.farnell.com/siretta/delta15-smam-ra-rp-11/stubby-antenna-2-4-2-5ghz-sma/dp/2717669?st=ZigBeeAntenna

[149] Ouyang, Chenxi, “Design and Implementation of a Wireless Zigbee Mesh Network,” *Masters Thesis, VAMK - Vaasa University of Applied Sciences*, 2014.

[150] S. G. Nikhade and A. A. Agashe, “Wireless sensor network communication terminal based on embedded Linux and Xbee,” in *Int. Conf. Circuits, Power Comput. Technol., Mar.*, 2014, pp. 1468–1473, doi: 10.1109/ICCPCT.2014.7055026.

[151] M. Tao, X. Hong, C. Qu, J. Zhang, and W. Wei, “Fast access for ZigBee-enabled IoT Devices Using Raspberry Pi,” *Proc. Chinese Control Decis. Conf. (CCDC), Jun.*, pp. 4281–4285, 2018, doi: 10.1109/CCDC.2018.8407868.

[152] S. R. Akbar, W. Kurniawan, M. H. H. Ichsan, I. Arwani, and M. T. Handono, “Pervasive device and service discovery protocol in XBee sensor network,” in *Int. Conf. Adv. Comput. Sci. Inf. Syst. (ICACSIS), Oct.* IEEE, 2016, pp. 79–84, doi: 10.1109/ICACSIS.2016.7872763.

[153] S. G. Nikhade, “Wireless sensor network system using Raspberry Pi and zigbee for environmental monitoring applications,” *Int. Conf. Smart Technol. Manag. Comput. Commun. Control. Energy Mater. (ICSTM), May*, pp. 376–381, 2015, doi: 10.1109/ICSTM.2015.7225445.

[154] S. Ferdoush and X. Li, “Wireless sensor network system design using Raspberry Pi and Arduino for environmental monitoring applications,” *Procedia Comput. Sci.*, vol. 34, pp. 103–110, 2014, doi: 10.1016/j.procs.2014.07.059.

[155] T. Suzuki, “GNSS-Radar,” 2014. [Online]. Available: http://www.taroz.net/GNSS-Radar.html

[156] Raspberry Pi, “Raspberry Pi 3 B+.” [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/?resellerType=home

[157] L. Breiman, “Bagging Predictors,” *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996, doi: 10.1023/A:1018054314350.

[158] R. E. Schapire, “The Boosting Approach to Machine Learning: An Overview,” in *Denison D.D., Hansen M.H., Holmes C.C., Mallick B., Yu B. (eds) Nonlinear Estim. Classif. Lect. Notes Stat. vol 171. Springer, New York, NY.*, 2003, pp. 149–171, doi: 10.1007/978-0-387-21579-2_9.

[159] Y. Freund and R. E. Schapire, “Experiments with a New Boosting Algorithm,” in *Proc. Thirteen. Int. Conf. Int. Conf. Mach. Learn.*, ser. ICML’96. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1996, pp. 148–156, doi: 10.5555/3091696.3091715.

[160] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., Aug.*, pp. 785–794, 2016, doi: 10.1145/2939672.2939785.

[161] J. H. Friedman, “Stochastic gradient boosting,” *Comput. Stat. Data Analysis*, vol. 38, no. 4, pp. 367–378, 2002, doi: 10.1016/S0167-9473(01)00065-2.

[162] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” in *Int. Conf. Learn. Represent., Dec.*, 2015, pp. 1–13. [Online]. Available: http://arxiv.org/abs/1412.6980

[163] B. Hanin, “Which Neural Net Architectures Give Rise to Exploding and Vanishing Gradients?” in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, ser. NIPS’18. Red Hook, NY, USA: Curran Associates Inc., 2018, pp. 580–589, arXiv: 1801.03744.

[164] Gps.gov, “GPS Spectrum and Interference Issues,” 2020. [Online]. Available: https://www.gps.gov/spectrum/

[165] G. D. O’Mahony, P. J. Harris, and C. C. Murphy, “Investigating the Prevalent Security Techniques in Wireless Sensor Network Protocols,” in *30th IEEE Irish Signals Syst. Conf., Jun.*, 2019, pp. 1–6, doi: 10.1109/ISSC.2019.8904934.

[166] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, “A theory of learning from different domains,” *Mach. Learn.*, vol. 79, no. 1-2, pp. 151–175, 2010, doi: 10.1007/s10994-009-5152-4.

[167] S. Sun, H. Shi, and Y. Wu, “A survey of multi-source domain adaptation,” *Inf. Fusion*, vol. 24, pp. 84–92, 2015, doi: 10.1016/j.inffus.2014.12.003.

[168] P. Colangelo, N. Nasiri, E. Nurvitadhi, A. Mishra, M. Margala, and K. Nealis, “Exploration of Low Numeric Precision Deep Learning Inference Using Intel® FPGAs,” in *2018 IEEE 26th Annu. Int. Symp. Field-Programmable Cust. Comput. Mach. (FCCM), Apr.*, 2018, pp. 73–80, doi: 10.1109/FCCM.2018.00020.