CORA Cork Open Research Archive
Cartlann Taighde Oscailte Chorcaí

Title	Investigating supervised machine learning techniques for channel identification in wireless sensor networks
Authors	O'Mahony, George D.;Harris, Philip J.;Murphy, Colin C.
Publication date	2020-06-01
Original Citation	O'Mahony, G. D., Harris, P. J. and Murphy, C. C. (2020) 'Investigating Supervised Machine Learning Techniques for Channel Identification in Wireless Sensor Networks', 31st Irish Signals and Systems Conference (ISSC), LetterKenny, Ireland, 11-12 June, pp. 1-6. doi: 10.1109/ISSC49989.2020.9180209
Type of publication	Conference item
Link to publisher's version	https://ieeexplore.ieee.org/abstract/document/9180209 - 10.1109/ISSC49989.2020.9180209
Rights	© 2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
Download date	2026-11-16 14:46:46
Item downloaded from	https://hdl.handle.net/10468/11184

UCC
University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

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

George D. O'Mahony
Dept. of Electrical and Electronic Engineering, University College Cork
Cork, Ireland
george.omahony@umail.ucc.ie

Philip J. Harris
Raytheon Technologies Research Center
Cork,
Ireland
harrispj@rtx.com

Colin C. Murphy
Dept. of Electrical and Electronic Engineering, University College Cork
Cork, Ireland
colinmurphy@ucc.ie

***Abstract*—Knowledge of the wireless channel is pivotal for wireless communication links but varies for multiple reasons. The radio spectrum changes due to the number of connected devices, demand, packet size or services in operation, while fading levels, obstacles, path losses, and spurious (non-)malicious interference fluctuate in the physical environment. Typically, these channels are applicable to the time series class of data science problems, as the primary data points are measured over a period. In the case of wireless sensor networks, which regularly provide the device to access point communication links in Internet of Things applications, determining the wireless channel in operation permits channel access. Generally, a clear channel assessment is performed to determine whether a wireless transmission can be executed, which is an approach containing limitations. In this study, received in-phase (I) and quadrature-phase (Q) samples are collected from the wireless channel using a software-defined radio (SDR) based procedure and directly analyzed using python and Matlab. Features are extracted from the probability density function and statistical analysis of the received I/Q samples and used as the training data for the two chosen machine learning methods. Data is collected and produced over wires, to avoid interfering with other networks, using SDRs and Raspberry Pi embedded devices, which utilize available open-source libraries. Data is examined for the signal-free (noise), legitimate signal (ZigBee) and jamming signal (continuous wave) cases in a live laboratory environment. Support vector machine and Random Forest models are each designed and compared as channel identifiers for these signal types.**

***Index Terms*—Classification, IoT, Machine Learning, Random Forest, SVM, WSN and ZigBee.**

I. Introduction

Wireless Sensor Networks (WSNs) are frequently integrated into safety-critical applications, as a result of over a decade of research and development. An example is the emerging area of the Internet of Things (IoT) [1], which is rapidly changing the wireless landscape. IoT advances are directly affecting (or creating) broadly accepted models such as smart cities/homes [2], edge/cloud computing and big data analytics, amongst others. Typically, WSN protocols enable the communication links between the IoT sensing/actuating device and the internet access-point, as this utilization reduces node firmware complexity and supports a larger number of connected devices and higher range, compared to wireless local area networks (WLANs). An extensive WSN application range exists and includes precision agriculture, environmental monitoring, health care (remote patient monitoring [3]), wireless body area networks and space-based WSNs, amongst others [4]. These critical applications, combined with forecasts from Gartner that over 25 billion devices will be connected by 2021, emphasizes WSN importance, usefulness and service load. Embracing WSNs will, likely, continue in the modern cost-centered age, due to their enabling easier design, installation and maintenance procedures, while simultaneously providing new deployment opportunities.

Therefore, strict operational and availability requirements can be imposed on computationally constrained devices, using similar physical (PHY) and medium access control (MAC) designs [4], which hinder the use of computationally intensive security protocols. WSNs encompass security challenges due to the open-access nature of the wireless channel, which grants access to unintentional and malicious interference and attacks. As IoT applications expand and WSNs become more interlinked with critical applications, the incentive to attack, intentionally disrupt or compromise WSNs escalates. The changeable abundant attack approaches derive from known WSN vulnerabilities [5] and, so, to ensure safety and privacy in many IoT applications, WSNs must be secure. Due to these 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 paper focuses on identifying the wireless channel (signal) in operation by statistically analyzing received samples and employing a machine learning (ML) classification approach. Here, the channels of interest include noise, ZigBee and continuous wave (CW) jamming. These channels are accessible and can be transmitted in laboratory environments using a designed software defined radio (SDR) approach, which utilizes python code and an Analog Pluto SDR on a Raspberry Pi embedded platform. The purpose is to develop a methodology for classifying channels in specific wireless environments based entirely on received in-phase (I) and quadrature-phase (Q) samples. In data science, this problem can be described as a time series classification as the required output is categorical, being of a specific channel type. As wireless channels vary for numerous reasons, each wireless channel will, generally, be unique and, so, the ability to identify specific channels/signals is beneficial to model development and channel access. For this study, a support vector machine and a Random Forest decision tree structure are compared in terms of received I/Q sample-based channel identifiers.

The remainder of this paper is organized as follows. Section

II discusses related work in the area. Section III describes the link between WSNs and the IoT and establishes the significance of the wireless channel. Section IV describes the designed data collection and data transmission processes. Section V explores, depicts and discusses the main results, including the applied ML algorithms, while section VI concludes this paper and provides future work progression.

II. Related Work

This section summarizes previous work using ML for wireless signal classification and improvements in clear channel assessment (CCA) techniques. Typically, IEEE802.15.4 based protocols use carrier sense multiple access with collision avoidance (CSMA/CA) to access the channel, which applies a CCA prior to transmitting a packet to check channel availability. Decisions are based on either energy detection (power threshold) or carrier sense (IEEE 802.15.4 signal) [4]. Previous improvements include splitting one eight-symbol CCA decision into two four-symbol CCA decisions [6], which allows the end of a packet transmission to be distinguished from a busy channel. In [7] and [8] an interference aware adaptive CCA mechanism was introduced, which used packet loss information to change the CCA mode in use to improve ZigBee performance under WiFi interference. In terms of signal identification, O'Shea et al. [9], have outlined the compelling possibility of using deep learning techniques for radio signal identification and provide methods for real-world adoption. Other literature exists, but, the above provides a brief overview of the context of this investigation. In contrast, this study focuses on using relatively low-cost hardware, supervised ML techniques (rather than deep learning) and received I/Q samples. The desired deployment is on embedded edge devices which can adapt to the classified channel.

III. WSN, IoT & Wireless Channel Discussion

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. Additionally, critical WSN applications and their transmitted data may incentivize malicious attackers to intentionally disrupt or compromise network operation by emitting malicious signals. For a wireless communications system, a channel refers to a logical connection over a multiplexed medium such as a radio channel, which is used to convey all information signals, typically, digital bit streams, from one or several transmitters to one or several receivers. Each channel has a certain 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 $R < C$, a coding technique exists which allows 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 aides in the transmission procedure and can provide insights for packet rates. In this investigation, the

TABLE I
IEEE 802.15.4 (ZigBee) PHY Parameters

Parameter:	2.4 GHz PHY Value:	
Number of Channels | Access	16	CSMA/CA
Channel Width | Spacing	2 MHz	5 MHz
Data Rate | Symbol Rate	250 kb/s	62.5 ksymbols/s
Data Byte Spreading | Chip Rate	DSSS	2 Mchips/s
Modulation Scheme	OQPSK	
Pulse Shaping	Half Sine/Normal Raised Cosine	
Maximum Packet Length	133 bytes	

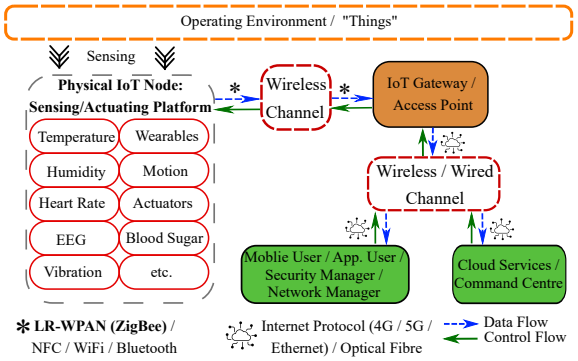


Fig. 1. An example IoT Architecture showing the potential utilization of WSNs and the potential security vulnerability of the wireless channel

ZigBee WSN protocol is the chosen legitimate signal model and operates on the wireless channels in (1), where F_c is the ZigBee center frequency and i is the channel number. Each channel operates in the unlicensed 2.4 GHz industrial, scientific and medical (ISM) radio frequency (RF) band and the channel can be categorized by the signals in operation.

$$F_c = 2405 + 5(i-11)MHz, \ \ for \ i = 11, 12, ...26 \quad (1)$$

ZigBee is based on IEEE 802.15.4 and is the de-facto standard for WSNs, as almost all available commercial and research sensor nodes are equipped with a ZigBee transceiver chip [10]. Table I provides the relevant PHY and MAC layer specifications, while Fig. 2 visualizes the signal. Every byte is split into two 4-bit symbols, which are each spread to a unique predefined 32-bit pseudo-noise (PN) sequence, as part of the direct sequence spread spectrum (DSSS) process. Offset quadrature-phase shift keying (OQPSK) modulation mutually offsets the I and Q components by half a symbol duration to ensure bit transmissions occur at different time instants. Pulse shaping, as per Table I, is applied before transmission, which ideally achieves zero inter-symbol-interference at the maximum effect points. The CSMA/CA protocol provides access to the channel and uses a CCA to determine whether a channel is free, or busy, before packet transmission.

ZigBee is interlinked with IoT applications, as WSNs regularly provide the sensing/actuating device to internet access point communication link, as specified in Fig. 1, which allows a larger number of connected devices and longer range, compared to WLANs. Fig. 1 depicts the importance of the wireless channel and specifies where interference, both malicious and unintentional, can be emitted to cause WSN disruption or

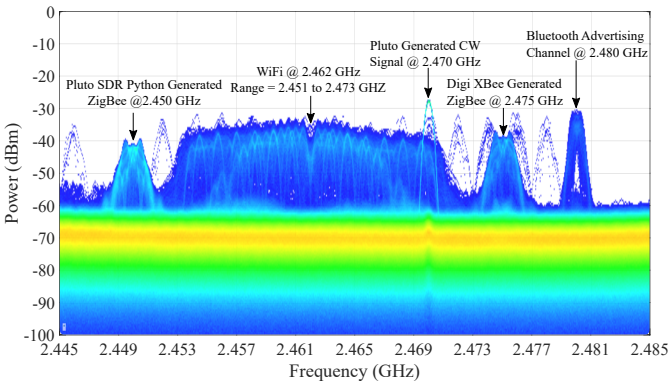


Fig. 2. DPX visualization of noise, ZigBee (commercial node), ZigBee (Pluto SDR), CW (Jammer) and coexistence with Bluetooth and WiFi

compromise. For WSNs, security can, generally, be described in terms of four interlinked distinct components; requirements, vulnerabilities, attacks and defenses. Example requirements include confidentiality, data integrity and origin authenticity. However, guaranteeing requirements are met can be difficult as WSNs have known security vulnerabilities [4], which include the open interface of the wireless channel, (unavoidably) publicly known WSN protocols and node deployments in unattended remote locations where devices can be physically available to potential attackers. The low processing power, memory and speed of WSN devices, coupled with a finite energy source, impedes using conventional security protocols, while WSN attack types are various and can occur across the entire communication protocol stack. Attacks 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, for example, cryptography, DSSS, frame check sequences and intrusion detection systems. Here, security is coupled with classification in terms of identifying legitimate signals.

IV. Data Collection

As part of this investigation, a wireless channel data logger was designed, which incorporated a Raspberry Pi 3 B, an Analog Pluto SDR and a 2.4 GHz antenna. The open-access python3 "pyadi-iio" and Analog Devices "libiio" libraries were implemented on the Raspberry Pi embedded platform to provide control and configuration of the Pluto SDR. Initially, an operating baseline channel that nodes transmit through and receive from was required, which corresponded to the noise channel for the operating environment (laboratory here). By identifying the noise channel characteristics, other channels could be analyzed and compared. As this work focused on WSNs, the sixteen ZigBee channels, provided in (1), and a 4 MHz sampling rate, were the chosen receiving specifications. From here on, noise data is annotated as a "signal-free" channel, ZigBee data is annotated as a " legitimate channel", while CW data corresponds to a "jammed channel".

For the data collection experiments, specific time intervals and data lengths were applied to collect data over a one hour

TABLE II
Data Collection: 16 ZigBee Channel center frequencies

Channel Type	Data: Length / No. Frame	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

period for each channel and center frequency. Initially, a Rohde & Schwarz Analogue Signal Generator supplied CW signals, which were transmitted through a DC block and a power combiner connected to the Pluto SDR, as shown in Fig. 3. However, this approach proved to be time consuming and inefficient as all required changes had to be applied using the available hardware interface. Therefore, a second Raspberry Pi/Pluto SDR combination was developed to create a low-cost signal generator. This Pluto configuration allowed for parameters like center frequency, signal type, sampling rate, etc. to be configured in code rather than using any hardware interface. Both 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 Table I and previous simulations [5], were generated. A second Raspberry Pi device was required 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. This experimental setup is shown in Fig. 4 and the output signals are visualized in Fig. 2, where the Pluto generated ZigBee signal is compared with a ZigBee signal from a commercial node, the DIGI XBee. The Pluto SDR signals correlate well with the commercial XBee ZigBee transmissions and, so, are considered acceptable for use in this investigation. In these experiments, both the SDR and Rohde & Schwarz setups were connected 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 overall data collection method can be summarized in Fig. 5, where the spectrum is visually inspected for unwanted signals using a SDR and the SDRConsole software package, or similar, and data is collected using either Fig. 3 or Fig. 4. The results of the above data collection process are summarized in Table II. In each case, the collected data was split into 70% for training, 20% for validation and 10% for testing. This data split was implemented for each of the specific cases, including noise, noise with spurious interference, CW and ZigBee.

V. Results: Machine Learning

This investigation aims to design an approach that can provide additional benefits compared to the traditional CCA methods and classify received samples, while using relatively low-cost hardware and data available to all potential edge devices. The overall data collection methodology is specified in Fig. 5, which comprises of both an experimental stage and a post-processing stage. The data produced probability density functions (PDF) for each of the received channels, where the results were averaged over the data sets and frequencies, as

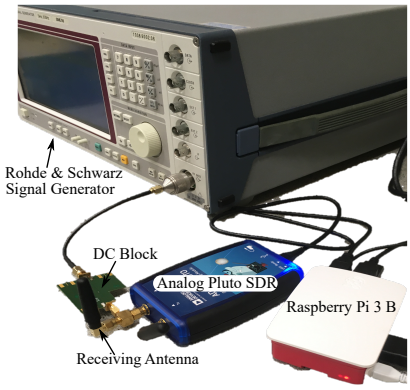


Fig. 3. Initial experimental setup for signal addition, which minimizes impact on surrounding services using the $2.4 \rightarrow 2.5GHz$ RF band.

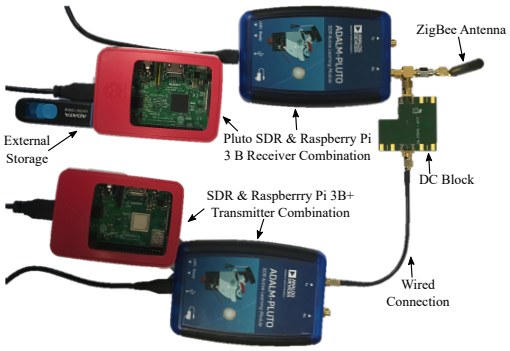


Fig. 4. SDR experimental setup for signal addition, which minimizes impact on surrounding services using the $2.4 \rightarrow 2.5GHz$ RF band

shown in Fig. 6, where the ZigBee signal matches the pattern shown in Fig. 2, as the center bins are smaller than the two outermost bins. The data was post-processed to down-convert sample values to the range [-1,1] using corresponding Pluto specifications and outputs. The wired method implemented in this data analysis contains disadvantages, namely, the lack of path loss, wireless fading, etc. and, in terms of the Analog Pluto, an intermittent CW wave in between successive ZigBee transmissions (Fig. 7(a)). Thus, the transceiver becomes saturated with bursts of SDR transmitted signals and the extracted data needs to be investigated and post-processed to ensure the correct I/Q samples are used in each analysis window. This post processing involves determining the beginning and end of the specific transmissions and extracting features from

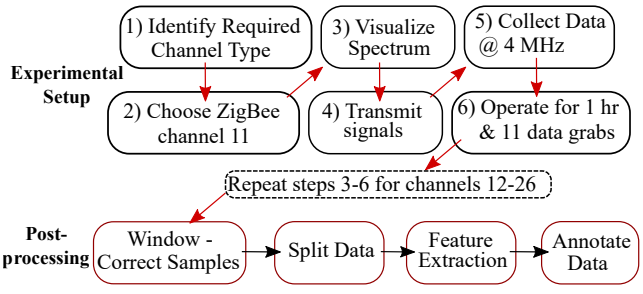


Fig. 5. Overall methodology for data collection and verification, sample identification and feature extraction

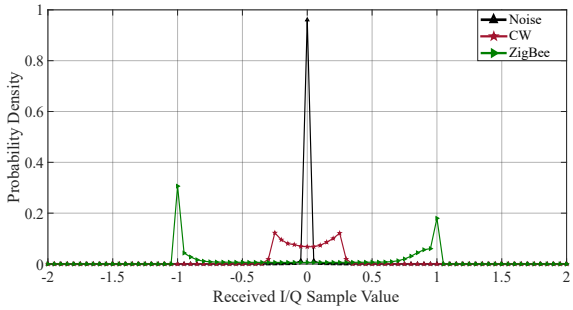


Fig. 6. Received I/Q sample PDFs for Noise, ZigBee and CW channels

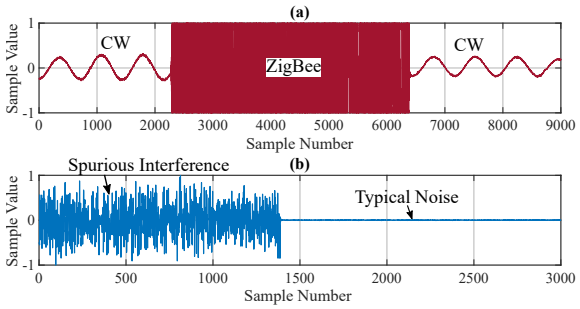


Fig. 7. Example of (a) Pluto operation between successive transmissions, (b) random spurious interference in the data collection

the windowed data. The uncontrollable spurious interference signals in the environment, shown in Fig. 7(b), which are evident in the noise data at random intervals and power levels, had to be identified separately as non-ideal noise. Several features were extracted from the PDF including the area in the center bins, value at the center bin, number of non-zero elements and maximum peak. Statistical analysis of the I/Q samples produced the features of variance, entropy, absolute maximum and mean. These features are summarized in Table III, where the signal differences and similarities are indicated.

In this study, two distinct machine learning (ML) approaches were investigated as potential channel identifiers using features based entirely on received I/Q samples from relatively low-cost hardware. The Support Vector Machine (SVM) and Random Forest approaches were applied to this classification problem, where the required output is an identified channel type. Each approach was based entirely on a feature set, (x), comprised from analyzing received I/Q samples from a SDR using a sampling rate of $4\ MHz$, which is data that can be available on IoT edge devices. In these supervised learning approaches, the algorithm is employed to learn the mapping function from the input variable (x) to the output variable (y); that is $y = f(X)$. Each algorithm attempts to estimate the mapping function (f) from the input variables (x) to discrete or categorical output variables (y).

A SVM, Fig. 8(a), is a supervised binary classification algorithm, which aims to find the optimal hyperplane that linearly separates the data points into two components by maximizing the margin. 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 like outlier detection, for example. A good separation (or margin)

TABLE III
EXTRACTED FEATURES: NOISE, NON-IDEAL NOISE, CW & ZIGBEE

Channel	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	2.9135	0.9797	4.834e-05	1.4183e-05	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-06	-0.0022	0.0085	0.6020
Non-ideal Noise	Average	0.9398	0.0	0.6947	25.7360	0.6948	0.0079	-9.1642e-05	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-04	-0.0031	0.1102	0.9843
CW	Average	0.8377	0.0334	0.0677	13.2801	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	7.3351e-05	-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

TABLE IV
SVM SIGNAL CLASSIFICATION GENERALIZATION ERROR RESULTS

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	0.49	1.2376	0.12
Noise/CW	0.0	0.0	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	0.0	0.0
Non-Ideal/CW	0.0	0.0	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	0.0	0.0
CW/ZigBee	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

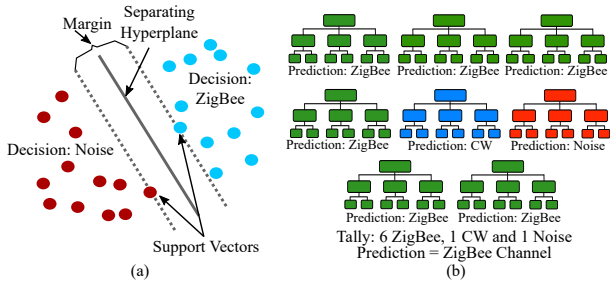


Fig. 8. Example visual representation of (a) a SVM in operation (b) a Random Forest model making a prediction

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 generalization error of the classifier, which assigns new data points to one of the given categories. Typically, the number of support vectors is much smaller than the total number of elements in the training dataset. Hence, training a SVM can be resource intensive, but the actual classification algorithm can be lightweight, which is an important concept for implementing channel classification models on an edge device. In practice, SVM algorithms are implemented using a kernel, which is a function that maps data to a higher dimension where data are separable and converts non-linear separable problems to linear separable problems by adding more dimensions. Here, the kernel was determined by using the available validation and test data and a ten-fold cross-validation approach, where the SVM was used as a binary classifier for the six separate combinations of noise, non-ideal noise, CW and ZigBee. As there is sufficient separation between the signals of interest, as shown in Fig. 6, the achievable margin is high in all cases and so the generalization error is low (approx. zero) for each kernel chosen, except for the noise/non-ideal noise case ($1 - 1.5\%$) as more data is required to reduce the error. The summarized SVM results are provided in Table IV, 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.

Random Forest [11] is a supervised ML approach, which consists of a large number of individual uncorrelated weak learners (decision trees), that operate as an ensemble. This 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. Here, each tree uses a deconstructed observed input to construct a series of binary intermediate nodes, that successively choose the attribute and associated threshold that provides the best split into groups, that are as different from each other as possible but contain members as similar as possible. The ensemble decision making concept depends on a diverse group rather than a homogeneous approach, as each individual tree needs to be unique. This majority voting scheme is visualized in Fig. 8(b) and depends on having a low correlation between individual trees, as this protects each tree from their individual error [12] and is ensured by two methods: bagging (bootstrap aggregating) and feature randomness. The former exploits the high sensitivity to the training data used and the latter ensures each tree can only pick from a random subset of available features. Random forest models unique trees by applying replacement, which allows each weak learner to be constructed from a random subset of the training samples and maintain the sample size by repeating previously used samples. Thus, each sample-set is randomly chosen from the total training set and each corresponding decision tree contains a different variation of the original classification data, which reduces variance and helps to avoid over-fitting. Once a set of decision trees has been computed, a new sample can be classified using a majority voting scheme, as visualized in Fig. 8(b).

Here, a multi-class algorithm was developed and optimal metrics identified using the available validation and testing data. The classes included noise, non-ideal noise, CW and ZigBee. The initial results are provided in Fig. 9(a), where the number of decision trees and predictor lengths, which is the size of the subset of features available for selection at random

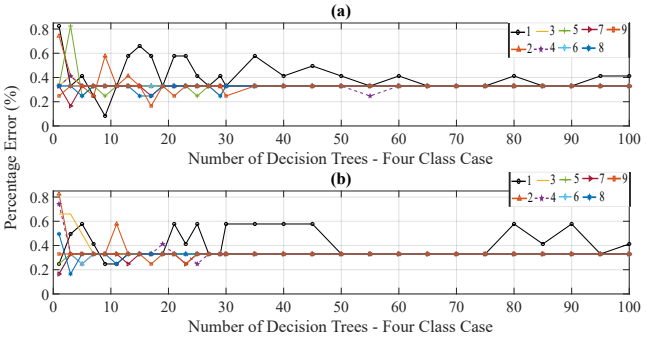


Fig. 9. Random Forest generalization error investigation using available validation and test data for combinations of predictor length (1-9) and the number of decision trees. (a) First Test, (b) Second Test

for each tree, were varied to find the optimized set of parameters. The main cause of error relates to the uncontrollable spurious interference signals, annotated as non-ideal noise, in the environment, provided in Fig. 7(b). This required non-ideal noise in the training, validation and testing data, thus, the $70-20-10$ data split occurred for the four data classes. Fig. 9 shows that the error can be reduced to below 1%. Initially, the single predictor model is the optimal (0.0825%) approach with nine decision trees 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 III are not as suitable as others and the overall process can be optimized. This is seen when the optimization is repeated in Fig. 9(b) as the overall error reduces to 0.33%, but again the single predictor is the most erratic. The tree structure and the "wisdom of crowds" of the random forest model are the key elements in the classification process and any future work will require the identification and removal of obsolete features to gain a more accurate feature representation of the signals. However, the results outline the usefulness of using the Random Forest approach for signal/channel classification and a single model applies to 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 providing a training time of 1.25 seconds and an average prediction time of 294 milliseconds.

The experimental results, based on data transmitted over wires, prove the usefulness of using ML methods and received I/Q samples. The results also help validate previous simulation based features focused on I/Q samples in [13], as live wired signals are used in conjunction with a wireless signal received from the channel. However, the need for wireless data collection has become more apparent due to a lack of environmental and spectral changes in the wired transmissions. Nonetheless, this study has provided an insight into how ML can be applied to wireless signal classification when the algorithms are based on received I/Q samples.

VI. Conclusion & Future Work

This study focused on using received I/Q samples and ML techniques to classify wireless channels. A SVM and a Random Forest approach were designed, using features extracted from the statistical analysis and the PDF of received I/Q samples, to identify SDR wired transmitted signals in a laboratory environment. The Random Forest approach can implement multi-class classification, allowing numerous signals to be identified using a single model, while, typically, multiple SVMs are required. The results indicate that ML, using I/Q samples, can classify received signals with a small generalization error. This study focused on signal classification, while the next stage needs to optimize extracted features and incorporate previous simulated work on interference detection [13], which also focused on using I/Q samples. Due to the limitations of wired signals and simulations, future work needs to focus on using wirelessly received I/Q data, while leveraging the previous simulated work and the results of this study, to produce a WSN interference detection system.

Acknowledgments

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

References

[1] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things : A survey," *Comput. Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[2] 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.

[3] S. Tennina, M. Santos, A. Mesodiakaki, et al., "WSN4QoL: WSNs for remote patient monitoring in e-Health applications," in *2016 IEEE Int. Conf. Commun.*, May 2016, pp. 1–6.

[4] G. D. O'Mahony, P. J. Harris, and C. C. Murphy, "Investigating the Prevalent Security Techniques in Wireless Sensor Network Protocols," in *Proc. 30th IEEE Irish Signals Syst. Conf. (ISSC)*, 2019, pp. 1–6.

[5] ——, "Analyzing the Vulnerability of Wireless Sensor Networks to a Malicious Matched Protocol Attack," in *2018 Int. Carnahan Conf. Secur. Technol. (ICCST)*, Montr. QC., 2018, pp. 1–5.

[6] 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," *2015 Int. SoC Des. Conf. (ISOCC)*, Gyungju, 2015, pp. 251–252.

[7] 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*, pp. 765–768, 2013.

[8] 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*, pp. 336–343, 2013.

[9] 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.

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

[11] L. Breiman, "Random Forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct 2001.

[12] T. Yiu, "Understanding Random Forest." [Online]. Available: https://towardsdatascience.com/understanding-random-forest-58381e0602d2

[13] G. D. O'Mahony, P. J. Harris, and C. C. Murphy, "Detecting Interference in Wireless Sensor Network Received Samples : A Machine Learning Approach," in *IEEE Virtual World Forum on Internet of Things (WF-IoT V2020), Virtual*, 2020, pp. 1–6.