

University of Southern Queensland
Faculty of Health, Engineering & Sciences

**Unique Identification of Bluetooth Transmitters Through
RF Fingerprinting**

A dissertation submitted by

E. Priest



in fulfilment of the requirements of

ENG4111 and ENG4112 Research Project

towards the degree of

Bachelor of Electrical & Electronic Engineering

Submitted: October, 2021

Abstract

Physical layer radio frequency (RF) fingerprinting has been used in military and civilian applications to identify RF transmitters for spectrum management purposes, and has been considered as a mechanism to improve assurance that a transmitter is not an impostor. It relies on the presence of observable device-specific variations to expected signal output, even between transmitters of the same type. Minor fluctuation in component values during transmitter assembly—and even placement of those components—can result in minor variances to frequency synthesis systems, modulator subsystems, and RF amplifiers, all which can be observed and used to characterise the transmitter. The complexity of the variations makes these characteristics inherently difficult to reproduce, and technically difficult to obscure.

RF fingerprinting of Bluetooth devices has been explored in the literature, but there is not sufficient information to reproduce the transient extraction stage used to produce the high-results of others. Additionally, there has been little reported work on the effects of expected environmental variables (temperature, motion, low signal to noise ratio) on classification success. This dissertation expands the existing literature by investigating the implementation and performance of a physical layer RF fingerprinting system, and the effect of real-world environmental conditions on system performance.

A downconverter was constructed to shift the entire Bluetooth band (2400–2480 MHz) down to 20–100 MHz, allowing acquisition of the entire band with low-cost acquisition hardware (i.e. a PicoScope 5444B). An RF fingerprinting system, specifically the transient detection sub-system and feature extraction sub-system, is implemented in MATLAB®. Energy Criterion is confirmed as an excellent method for detecting the start of a transient portion. Additionally, a new method for detecting the end of the transient is introduced, based on the settling time of the envelope. These two methods successfully extracted

the transients from several waveforms reliably; however, some transmitter types were observed to produce waveforms with significant ripple to the steady-state envelope, causing unreliable operation of the transient detection system.

To support classification a feature extraction system was implemented in MATLAB®. Features are extracted from the energy envelope and the time-frequency-energy distribution (TFED) of the signal. A link is identified between inconsistent transient length detection and inconsistent features. Classifiers were implemented using MATLAB®'s Classification Learner app, with the optimum classifier found to be a Support Vector Machine, which confirms existing literature.

A new dataset of turn-on transients was acquired for 17 devices using the constructed downconverter and acquisition system. This dataset, and an existing reference dataset, were used to assess the transient detection, feature extraction, and classifier sub-systems, and the results compared. After optimisation, classifiers were able to correctly attribute waveforms from the reference dataset to a specific device with an accuracy of 32.6%, while correct attribution when using the acquired dataset was 69.9%. When the classifiers were used to attribute waveforms to a device-type, as opposed to specific device, prediction success increased to 92.6%. This research was unable to reproduce the extremely high results (over 99% success) reported in the literature. Further work in the field, specifically improvement to the transient detection stage, is required to make RF fingerprint classification of Bluetooth devices more viable.

University of Southern Queensland
Faculty of Health, Engineering & Sciences

ENG4111/2 <i>Research Project</i>
--

Limitations of Use

The Council of the University of Southern Queensland, its Faculty of Health, Engineering & Sciences, and the staff of the University of Southern Queensland, do not accept any responsibility for the truth, accuracy or completeness of material contained within or associated with this dissertation.

Persons using all or any part of this material do so at their own risk, and not at the risk of the Council of the University of Southern Queensland, its Faculty of Health, Engineering & Sciences or the staff of the University of Southern Queensland.

This dissertation reports an educational exercise and has no purpose or validity beyond this exercise. The sole purpose of the course pair entitled “Research Project” is to contribute to the overall education within the student’s chosen degree program. This document, the associated hardware, software, drawings, and other material set out in the associated appendices should not be used for any other purpose: if they are so used, it is entirely at the risk of the user.

Dean

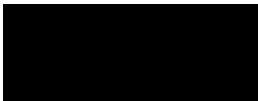
Faculty of Health, Engineering & Sciences

Certification of Dissertation

I certify that the ideas, designs and experimental work, results, analyses and conclusions set out in this dissertation are entirely my own effort, except where otherwise indicated and acknowledged.

I further certify that the work is original and has not been previously submitted for assessment in any other course or institution, except where specifically stated.

E. PRIEST



Acknowledgments

I would like to thank the following people:

my project supervisor, John Leis, for his valuable support, encouragement and guidance throughout this project;

my friend and occasional competitor, Aidan Galt, for continually driving me to achieve better outcomes for less effort;

and finally, and most importantly, my wife Kathleen whose support of me and my studies never wavered, despite the challenges encountered along the way.

E. PRIEST

Contents

Abstract	i
Acknowledgments	vii
List of Figures	xv
List of Tables	xix
Nomenclature	xxi
Chapter 1 Introduction	1
1.1 Introduction	1
1.2 Definition of the Problem	2
1.3 Research Objectives	3
1.4 Consequential Effects	4
1.4.1 Accurately Identifying Devices	4
1.4.2 Reduced Barrier to Entry for Future Researchers	4
1.4.3 Ethical Issues	5
1.5 Risk Assessment	6

1.6	Project Timeline	6
1.7	Resource Requirements	7
Chapter 2 Literature Review		9
2.1	Introduction	9
2.2	RF Fingerprinting	9
2.3	RF Fingerprinting of Bluetooth Devices	12
2.4	Transient Detection	13
2.5	Classifier	14
2.6	Literature Gap	15
2.7	Conclusion	16
Chapter 3 Methodology		17
3.1	Introduction	17
3.2	Proposed Objectives	17
3.3	RF Downconverter	18
3.3.1	RF Downconverter Construction	18
3.3.2	RF Downconverter Verification	20
3.4	Acquisition System	21
3.4.1	Acquisition System Construction	21
3.4.2	Acquisition System Verification	21
3.5	Transient Detection Software	22

3.5.1	Transient Detection Software Construction	22
3.5.2	Transient Detection Software Verification	22
3.6	Feature Extraction System	23
3.6.1	Feature Extraction System Construction	23
3.6.2	Feature Extraction System Verification	24
3.7	Classifier	25
3.7.1	Classifier Construction	25
3.8	Experiment Design	26
3.8.1	Experiment design	26
3.8.2	Dataset collection	27
3.8.3	Interpretation of results	30
3.9	Project plan	31
3.10	Chapter summary	31
 Chapter 4 System Design		33
4.1	Introduction	33
4.2	System Model	33
4.3	RF Downconverter	35
4.4	Acquisition System	39
4.5	File processing system	40
4.6	Transient Detection System	41
4.7	Feature Extraction System	43

4.8	Classifier	46
4.9	Chapter Summary	46
Chapter 5 Data Collection		47
5.1	Introduction	47
5.2	Existing Reference Dataset	47
5.3	Collected Dataset	49
5.3.1	High SNR Dataset (Dataset A)	50
5.3.2	Temperature Variance Dataset (Dataset B)	51
5.3.3	Doppler Shift Dataset (Dataset C)	51
5.3.4	Lower SNR Dataset (Dataset D)	51
5.4	Chapter Summary	52
Chapter 6 Results and Discussion		53
6.1	Introduction	53
6.2	Downconverter	53
6.2.1	Downconverter Using Mini Circuits ZX05-63LH-S+ Mixer	53
6.2.2	Downconverter Using HMC175-based Mixer	55
6.2.3	Comparison of Mixers	57
6.3	Acquisition System	58
6.4	Transient Detection	58
6.5	Feature Extraction	61

6.6	Classifier Performance	65
6.6.1	Reference Dataset, Device Attribution Classification Results	65
6.6.2	Dataset A, Device Attribution Classification Results	67
6.6.3	Classifying Based on Device-type	67
6.7	Discussion	68
6.8	Chapter Summary	70
Chapter 7 Conclusions and Further Work		71
7.1	Recommendations for Further Research	71
7.1.1	Collection of Data	71
7.1.2	Improve Transient Detection Algorithm	72
7.1.3	Initial Correlation of Turn-on Transients	72
7.1.4	Improve Downconverter	73
7.2	Conclusions	73
References		77
Appendix A Project specification		83
Appendix B Risk assessment		85
Appendix C Project timeline		89
Appendix D Project resources		91
Appendix E MATLAB® code		93

E.1	Overview of code blocks	94
E.2	RF fingerprinting system	95
E.2.1	main.m	95
E.2.2	traverse_folders.m	98
E.2.3	loadDir.m	100
E.2.4	EC.m	105
E.2.5	extractFeaturesEMD.m	106
E.2.6	class2vec.m	109
E.2.7	importFromRef.m	110
E.2.8	filter_250.m	112
E.2.9	filter_500.m	113
E.3	File conversion system	114
E.3.1	processFiles.m	114
E.3.2	convertFile.m	118
E.3.3	isTransient.m	120

List of Figures

2.1	Unique waveforms captured from four Wi-Fi radios (Ureten & Serinken 2007).	10
2.2	Example of an optimal hyperplane formation using Support Vector Machine algorithm with a linear kernel function for binary classification of two classes.	15
3.1	Block diagram of RF acquisition system, highlighting the major components of the downconverter.	20
3.2	Predicted Doppler shift of Bluetooth devices at expected human speeds. Calculated by application of Equation 3.6 for Bluetooth frequency range and velocities of up to 30 m s^{-1}	29
3.3	Overhead plot showing radial walk testing paths for Doppler testing (Bowen, 2017).	30
4.1	Block diagram showing the major components of the constructed system.	34
4.2	Use of a mixer to downconvert RF signals. Shown are first-order component output of the downconverter when mixing Bluetooth signals with two local oscillator frequencies. Local oscillator of $f_{LO} = 2500 \text{ MHz}$ (top) results in spectral inversion of the lower side band, but $f_{LO} = 2380 \text{ MHz}$ (bottom) does not result in spectral inversion.	36

4.3	Photograph of Voltage Controlled Oscillator used to produce continuous wave frequencies to test the performance of the downconverter system. . .	37
4.4	Photograph of the constructed downconverter system (top layer) showing 5 V linear regulator and tuning circuit for the voltage controlled oscillator. . .	38
4.5	Photograph of the constructed downconverter system (bottom layer) showing the low-noise amplifier, bandpass filter, mixer, local oscillator, and low-pass filter.	38
4.6	The acquisition system, showing RF downconverter connected to the PicoScope 5444B.	40
4.7	Example of automated transient detection system operation. The signal envelope is extracted, then Energy Criterion is used to find the transient start and settling time is used to find the steady-state start/transient end.	43
5.1	Photograph of the data acquisition system deployed in an RF-shielded room to collect Bluetooth turn-on transients. The laptop was placed in flight-mode during acquisition.	50
6.1	Bluetooth turn-on transient of Samsung Galaxy S8 as output by the downconverter using the ZX05-63LH-S+ mixer, shown in the time domain. . .	54
6.2	Bluetooth turn-on transient of Samsung Galaxy S8 as output by the downconverter using the HMC175 mixer, shown in the frequency domain. . . .	55
6.3	Bluetooth turn-on transient of Samsung Galaxy S8 as output by the downconverter using the HMC175 mixer, shown in the time domain.	56
6.4	Bluetooth turn-on transient of Samsung Galaxy S8 as output by the downconverter using the HMC175 mixer, shown in the frequency domain. . . .	56
6.5	Comparison of frequency components in output of downconverter using ZX05-63LH-S+ and HMC175-based mixers for single-frequency unmodulated input, shown in the frequency domain.	57

6.6	Boxplot of feature f_1 , transient length, for the reference dataset.	59
6.7	Boxplot of feature f_1 , transient length, for Dataset A.	60
6.8	Boxplot of feature f_3 , total energy of transient energy envelope, for the reference dataset.	62
6.9	Boxplot of feature f_{12} , maximum of sum of transient energy distribution along frequency axis, for the reference dataset.	62
6.10	Boxplot of feature f_4 , variance of transient energy envelope, for the reference dataset.	63
6.11	Scatter plot of features f_4 and f_{12} , for the reference dataset.	63
6.12	Boxplot of feature f_5 , StD of transient energy envelope, for the reference dataset.	64
6.13	Boxplot of feature f_6 , entropy of instantaneous phase, for the reference dataset.	64
6.14	Confusion matrix of Reference Dataset without device-type grouping, using SVM classifier after optimisation (32.6% classification success).	66
6.15	Confusion matrix of Dataset A without device-type grouping, using SVM classifier after optimisation (69.6% classification success).	67
6.16	Confusion matrix of Dataset A with device-type grouping, using SVM classifier (92.6% classification success).	68
E.1	Overview of code blocks used in RF fingerprinting system.	94

List of Tables

- 3.1 Overview of the thirteen features to be extracted from each record for classification. 24

- 4.1 Overview of method for calculating features from transients. 45

- 5.1 Summary of the devices included in the existing reference dataset. 48
- 5.2 Summary of the devices included in the collected datasets. 49

- D.1 Project resources. 92

Nomenclature

Bluetooth Classic	A short-range personal area network, defined by the Bluetooth Special Interest Group.
BLE	<i>Bluetooth Low Energy</i> , a low-energy variant of Bluetooth Classic (BR/EDR) defined by the Bluetooth Special Interest Group.
EC	<i>Energy Criterion</i> , a method for detecting signal transients by variation of signal energy content.
HHT	<i>Hilbert-Huang Transform</i> , a method for decomposing a signal into intrinsic mode functions to obtain instantaneous energy and frequency.
MAC	<i>Media Access Control</i> , the unique address assigned to a network interface.
RF	<i>Radio Frequency</i> .
SDR	<i>Software Defined Radio</i> , a radio in which some or all of the subcomponents are implemented in software instead of hardware.
SNR	<i>Signal to Noise Ratio</i> , the ratio of desired signal to background noise.
SVM	<i>Support Vector Machine</i> , a supervised learning model for classification.
TFED	<i>Time-Frequency-Energy Distribution</i> , the three-dimensional representation of signal energy in the time and frequency domain.

VFDTD

Variance Fractal Dimension Threshold Detection, a method for detecting signal transients by using the fractal dimension calculated from the variance of signal amplitude.

Chapter 1

Introduction

This chapter provides an overview and background of the dissertation and supporting research project, and a summary of the implications resulting from the outcomes of the dissertation.

1.1 Introduction

Media Access Control (MAC) addresses are unique addresses used to identify a particular node in a network. The MAC address is intended to be a static identifier for the interface, unique to that device. In a wireless network, it is necessary for at least one node to broadcast its presence, along with their MAC address, to allow other devices to connect and begin communications.

MAC addresses were intended to be static identifiers per interface; however, in the realm of personal devices, the use of a static MAC address can be used to identify and track people based on the devices they carry. This is especially applicable to wireless systems, where devices need to announce their presence to begin communications with others. To address this privacy concern, vendors implement MAC address randomisation. Under this scheme, the MAC address is frequently changed to a random address to thwart passive tracking, with mechanisms present to allow legitimate and authenticated communications to continue (Martin, Mayberry, Donahue, Foppe, Brown, Riggins, Rye & Brown 2017).

Despite the intention of MAC randomisation, it is not wholly sufficient for true device

anonymisation. MAC addresses are a convenient identifier for the device, but they are not the only unique identifier. Like human biometrics, transmitters possess characteristic traits that can be used to identify or infer the identity of a transmitter. These traits could occur at the physical layer, the data-link layer, or the application layer. The identification of devices based on these characteristics is known as RF fingerprinting.

Physical layer RF fingerprinting has been used in military and civilian applications to identify RF transmitters for spectrum management purposes and has been considered as a mechanism to improve assurance that a transmitter is not an impostor (Ureten & Serinken 2007, Frederick 1995, Elmrbet, Arjoune, el Ghazi, Majd & Kaabouch 2018). It relies on the presence of observable device-specific variations to expected signal output, even between transmitters of the same type. Minor fluctuation in component values during transmitter assembly—and even placement of those components—can result in minor variances to frequency synthesis systems, modulator subsystems, and RF amplifiers, all which can be observed and used to characterise the transmitter. The complexity of the variations makes these characteristics inherently difficult to reproduce, and technically difficult to obscure (Polak, Dolatshahi & Goeckel 2011, Ureten & Serinken 2007). This dissertation describes the construction and implementation of a physical layer RF fingerprinting system for Bluetooth devices, based on observation of their turn-on transient, and documents the performance.

1.2 Definition of the Problem

In their paper, Ali, Uzundurukan & Kara (2019) describe the realisation of a system for physical layer RF fingerprinting of Bluetooth devices, and report classification success of over 99%. Despite the detail provided within the publication, there is no indication that the system or high success rates have been reproduced by the broader academic community. There is also insufficient information in the literature to allow reproduction of the transient extraction stage used to produce these high success rates.

In addition to reproduction of the work of others, there has been little research reported on the effects of expected environmental variables (temperature, motion, low signal to noise ratio) on classification success. Several real-world effects are known to induce minor changes in transmitter signals, and received signals, and therefore could cause variance

in the effectiveness of such a system. The output frequency of an oscillator is often a function of its temperature, so it is expected that changes in temperature could cause minor drift in output frequency, resulting in clock and frequency error (Helluy-Lafont, Boe, Grimaud & Hauspie 2020). Additionally, changes in the relative distance between a transmitter and receiver during transmission can introduce an apparent frequency and phase shift due to the Doppler effect. Existing methods for fingerprinting are known to use instantaneous phase and frequency information as features for classifying devices (Ali et al. 2019, Aghnaiya, Ali & Kara 2019), yet it is not known how these minor fluctuations in frequency affect the accuracy of RF fingerprinting systems. Given the application of this system is the identification of non-cooperative transmitters, it is insufficient to assume signatures are provided in perfect conditions. There is currently no literature that explains how real-life environmental conditions these effects could affect the reliability or accuracy of RF fingerprinting systems. There are also no known datasets that allow exploration into how the physical environment (temperature, movement, background RF noise) affect classification.

1.3 Research Objectives

This dissertation expands the existing literature by investigating the implementation and performance of a physical layer RF fingerprinting system, and the effect of real-world environmental conditions on system performance. The RF fingerprinting system uses a downconverter to facilitate acquisition with equipment capable of sampling at 500 MS s^{-1} . The downconverter, transient detection sub-system, and feature extraction sub-system are implemented based on the findings of a number of related works (Ali, Uzundurukan & Kara 2017, Ali et al. 2019, Uzundurukan, Dalveren & Kara 2020b, Uzundurukan, Ali, Dalveren & Kara 2020).

The specific goals of the dissertation are:

1. Design and build an RF front-end downconverter to allow acquisition of Bluetooth signals using equipment with low sampling rates (i.e. 500 MS s^{-1}).
2. Research and implement a system to automatically extract the ‘turn-on’ transient from a captured Bluetooth signal at various sample rates.

3. Acquire a collection of ‘real-world’ fingerprint data.
4. Assess the accuracy of the classifier to correctly identify a device for a given turn-on transient using an existing dataset and acquired dataset, and compare these results to those in the literature.

1.4 Consequential Effects

1.4.1 Accurately Identifying Devices

As RF fingerprints are believed to be immutable, and difficult to forge, the outcomes of this research can benefit a range of use-cases. For network security uses, RF fingerprinting could be used as an additional authentication measure, providing a higher-degree of assurance that a wireless network node is genuine. For policing uses, RF fingerprinting could be applied to attribute transmissions or interest to a particular transmitter (and therefore party), which could be helpful when combatting identity theft or distribution of child exploitation material. For spectrum management uses, RF fingerprinting could be used to attribute illegal transmissions or interference to a particular transmitter, even before the transmission location is identified.

This research also has applicability within government and enterprise settings, where it is common to have restrictions on the specific electronic devices allowed in particular areas. For example, the Australian Government’s Information Security Manual requires sites implement security measures to detect unauthorised RF devices in areas processing Secret or Top Secret information (Australian Cyber Security Centre 2020); unfortunately MAC randomisation makes it difficult for site owners to passively monitor for authorised devices, let alone non-cooperative unauthorised devices. The requirement to automatically discriminate authorised and unauthorised transmitters without user participation could also have applicability to schools, sensitive research areas, and corrections institutions.

1.4.2 Reduced Barrier to Entry for Future Researchers

Prior research from others was reliant on equipment capable of direct sampling at rates of up to 20 GS/s, resulting in high equipment costs, and high computation costs to process

the data. These restrictions imposed a barrier to entry to RF fingerprinting. One of the elements of this research is to implement the RF fingerprinting system using equipment with low sampling rates. This makes RF fingerprinting more accessible, by lowering the cost of establishing a system. While this could result in more systems being used for positive purposes, it also makes the system more accessible for nefarious purposes.

1.4.3 Ethical Issues

The ethics behind security and vulnerability research is inherently complex. Despite the promising applications, this research improves ability to attribute of devices employing privacy-enhancing features, which ultimately translates to a reversal of the ‘anti-tracking’ features of modern Bluetooth specifications; this therefore re-introduces the risk of tracking people by correlating the wireless devices they carry. Depending on the circumstances, this may not seem a significant risk; however, there remains an enduring potential to compromise personal privacy.

Utilitarian ethics theory requires actions be evaluated to determine how much good they do. The evaluation considers only the consequence of the action, not the original intention (Russell, Hogan & Junker-Kenny 2012). The aim is to create the greatest good for the greatest number. Under this theory, researchers aim to maximise happiness. It could be argued that publication of vulnerability research benefits those with ill-intent, and thus increases danger to the public. When vulnerabilities are already well known, however, continuing the research is unlikely to cause further harm. Rather, it raises vendor and public awareness of the vulnerabilities, so mitigations can be developed. RF fingerprinting is already a well-known and researched method for enhancing security of wireless network links, so this proposed research is unlikely to increase the danger or harm to the public.

Deontological ethics theory contrasts Utilitarianism, requiring people to follow a set of moral rules, act from duty, and possess a good intent, rather than a good outcome (Russell et al. 2012). Applying this framework, researching and disclosing vulnerabilities in privacy, which to allow the public to make educated decisions about the privacy or security offered by their products, is a sound decision. Bringing further attention to RF fingerprinting techniques allows the public to be better informed on limitations of privacy-enhancing features within their devices, and allows them to make more informed choices about trusting the claims of such put forward by vendors of this technology.

After the application of Utilitarianism and Deontological frameworks, the research is still ethically justified.

1.5 Risk Assessment

To ensure this research was undertaken safely, hazards were identified, the risks assessed and managed to ensure any residual risk was as low as reasonably practicable. The *USQ Safety Risk Management System* was used as the framework for assessing and documenting risks. Where a risk control measure is detailed in the *USQ Laboratory & Workshop Safety Manual v2.2*, this control is applied to control the risk.

The risks identified in the Risk Management Plan are fairly standard for working in a laboratory environment. The RF equipment being tested are standard consumer Bluetooth devices, so there was no significant risk of dangerous RF. The only notable risks identified related to potential entrapment within the RF shielded room, and workstation ergonomics due to the prolonged amount of computer-based work.

All identified risks were assessed as Low after basic control were applied.

A copy of the complete risk assessment can be found in Appendix B.

1.6 Project Timeline

To ensure the research project was completed within the required time-frame, and the objectives of the project specification were met, a project timeline was developed. The project timeline includes sequencing, and approximate start and end dates, for each major event.

A Gantt chart of the project timeline can be found in Appendix C.

1.7 Resource Requirements

A number of resources were required to complete the research project. These are broadly categorised as:

- RF downconverter;
- low-cost acquisition hardware;
- computer with MATLAB[®] for feature extraction and classification;
- sample Bluetooth devices; and
- environment suitable for conducting controlled testing.

A complete list of equipment required for the research project can be found in Appendix D.

Chapter 2

Literature Review

2.1 Introduction

To undertake any further research into RF fingerprinting of Bluetooth devices, and the potential implications when using samples acquired under non-ideal conditions, it is necessary to review the existing body of literature. This chapter provides a review of the literature relevant to RF fingerprinting of Bluetooth devices, with a focus on the use of turn-on transients for this purpose.

2.2 RF Fingerprinting

The unique fingerprinting of transmitters may be of interest to nation-states for defence reasons, so the body of work may be larger than has been discovered.

RF fingerprinting methods can be broadly categorised as physical layer techniques, or higher layer techniques; within the physical layer techniques, methods can be further categorised as transient methods (Ureten, Serinken et al. 1999, Ellis & Serinken 2001, Hall 2006, Ureten & Serinken 2007, Mohamed, Dalveren & Kara 2020), or steady-state methods (Brik, Banerjee, Gruteser & Oh 2008, Candore, Kocabas & Koushanfar 2009, Polak et al. 2011, Nguyen, Zheng, Han & Zheng 2011).

Early attempts at RF fingerprinting focussed on physical layer characterisation of the

transient ‘turn on’ stage of the transmitter (see Figure 2.1). Hippenstiel & Payal (1996) reported success in identifying push-to-talk radio transmitters by applying wavelet transform and Euclidean distance algorithm; however, the scale of the testing was small and the conditions unknown. Ureten & Serinken (2007) applied RF fingerprinting techniques to the start-up transient of IEEE 802.11b Wi-Fi signals. The Hilbert Transform was employed to yield instantaneous frequency and amplitude of the transient, and a probabilistic neural network used for classification. When tested in a controlled environment, the system was able to classify transmitters with an error rate of 2%.

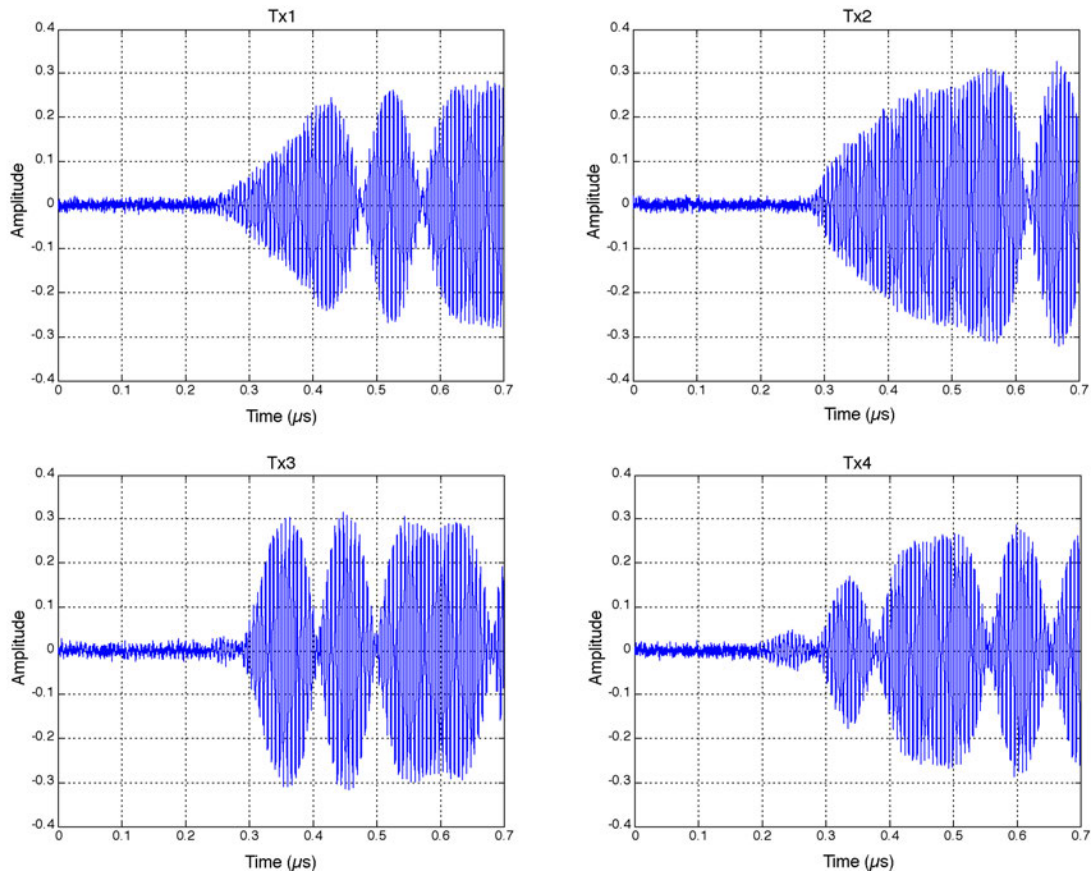


Figure 2.1: Unique waveforms captured from four Wi-Fi radios (Ureten & Serinken 2007).

The effort on RF fingerprinting of IEEE 802.11 devices increased in response to vendors introducing MAC address randomisation. Although earlier work focused on physical layer fingerprinting, research began to focus on higher layer methods—specifically the data link layer (or MAC layer). Some argued that this was because physical layer fingerprinting requires prior knowledge about the transmitter to create a trained dataset (Robyns, Bonn e, Quax & Lamotte 2017). Others noted the plethora of vendor-specific implementation idiosyncrasies which lend themselves to unique identification, even with MAC address randomisation (Becker, Li & Starobinski 2019, Celosia & Cunche 2020, Martin

et al. 2017, Oliveira, Schneider, Souza & Shen 2019). The fundamental issue with data link layer fingerprinting is that it is not immutable. While the current variance of implementation makes this a rich area for identification, the drawback is that fingerprint characteristics are implemented in software and thus can be modified by the vendor, or the device owner. This means researchers need to keep their signature library updated with all patches for all devices, but also makes it hard to prove ownership as the characteristics could be changed at any time.

Polak et al. (2011) argued many higher layer characteristics could be modified in software, and would therefore not be admissible in court; they instead focused on steady-state errors in the digital to analogue converters (DACs) and non-linearity in the power amplifiers to realise immutable physical layer fingerprinting.

Others have researched steady-state physical layer fingerprinting techniques, though this work largely focuses on the IEEE 802.11 WiFi domain. Brik et al. (2008) developed PARADIS, a system which compared frames with the ‘ideal’ version of itself in the modulation domain, allowing identification of frequency error, SYNC correlation, I/Q offset, phase error and magnitude error to be extracted and converted to multi-dimension vectors, which could then be processed in a support vector machine (SVM) classifier; this study reported identification of 130 devices with 99% accuracy, with robustness for mobility, ambient noise and transmitter ageing. Desmond, Yuan, Pheng & Lee (2008) attempted to infer clock-skew based on probe requests but discovered inconsistent performance, as timing between probe requests are highly dependent on internal device processes which cannot be inferred remotely. Wenhao, Zhi, Kui, Bocheng & Sixu (2015) attempted to use time-domain and frequency-domain characteristics to fingerprint devices and explore the effects of environmental changes. They argued physical layer fingerprinting is prone to misclassification in real-world application; however, it is noteworthy that their fingerprinting extraction was based on time-domain and frequency-domain analysis. Despite this, many have reported success performing RF fingerprinting using wavelet transform (Hippenstiel & Payal 1996, Hall, Barbeau & Kranakis 2006, Klein, Temple & Mendenhall 2009, Danev, Zanetti & Capkun 2012, Xu, Zheng, Saad & Han 2015) and discrete Gabor transform (Lukacs, Collins & Temple 2015) to extract useful characteristics from signal transients.

The viability of using low-cost software defined radio (SDR) to complete RF fingerprinting on Bluetooth devices has also been investigated. In research by Helluy-Lafont et al. (2020), three features were used: preamble duration, hopping clock skew, and carrier

clock skew. The authors argue clock skew is highly affected by variances in temperature, suggesting further research could be completed in this area. However, the SDR used had limited bandwidth (28 MHz) and was unable to cover the entire Bluetooth band. Rather than increasing SDR bandwidth, others designed modular RF front-end systems to downconvert Bluetooth signals to the band 20–100 MHz (Uzundurukan, Ali & Kara 2017). These systems have been shown to be effective at reproducing Bluetooth signals at lower frequencies (Uzundurukan, Ali, Dalveren & Kara 2020); the benefit of this method is that the entire Bluetooth band can be sampled with a sampling rate as low as approximately 200 MS s^{-1} .

2.3 RF Fingerprinting of Bluetooth Devices

Recent research indicates Bluetooth devices can be reliably identified through their characteristic turn-on transient. The Hilbert-Huang transform (HHT) has been used to generate time-frequency-energy distributions (TFEDs), which can be used to extract features which support identification, even under high SNR (Ureten & Serinken 2007, Ali et al. 2019, Uzundurukan, Ali, Dalveren & Kara 2020). Using TFEDs, Ali et al. (2019) identified 13 features which can be used to classify Bluetooth turn-on transients, and report greater than 98.95% accuracy when using the Linear Support Vector Machine (L-SVM) classifier, with Complex Tree and Linear Discriminant Analysis (LDA) classifiers performing slightly worse.

Aghnaiya et al. (2019) investigated the viability of Variational Mode Decomposition (VMD) as an alternative to the Empirical Mode Decomposition function underpinning the Hilbert-Huang Transform. Fingerprint signals were decomposed using VMD, and higher order statistical features (variance, skewness and kurtosis) are calculated from the instantaneous amplitude, frequency and phase, resulting in nine features per record. When compared to existing feature extraction using the HHT (Ali et al. 2019), the VMD technique showed increased classification accuracy of 8% with an SVM classifier.

2.4 Transient Detection

The literature indicates that the accurate detection of the transient portion of a signal is a major challenge (Uzundurukan, Ali, Dalveren & Kara 2020).

There are a number of potential methods for detecting the start of the transient portion of an RF signal. Variance fractal dimension threshold detection (VFDTD) can work well for signals with high SNR where there is an abrupt change in amplitude at the beginning of the transient, though it can be challenging to determine appropriate thresholds for a range of conditions (Hall, Barbeau & Kranakis 2003). Bayesian step change detection (BSCD) remains popular in the literature, and has been employed to automatically detect transmission transients from VHF two-radio radios (Ureten et al. 1999). Hall et al. (2003) also developed a novel transient detection algorithm which analysed the phase of the signal, and relied on the slope of the phase being linear throughout the transient; this method was shown to be superior to VFDTD and BSCD methods at detecting cellular-phone radios.

A comprehensive review by Mohamed et al. (2020) investigated the accuracy and computational complexity of several popular transient detection algorithms. The study investigated the performance of VFDTD and BSCD, mean change point detection (MCPD), and the phase detection (PD) algorithms developed by Hall et al. (2003). Additionally, two implementations of the Energy Criterion algorithm used in the power field were also investigated (Wagenaars, Wouters, Van der Wielen & Steennis 2008). The experiment showed EC- α exhibited superior accuracy while also requiring the least elapsed-time to detect the transient.

While Energy Criterion can be used to reliably detect the start-point of a transient, detecting the end-point (that is, the location where the signal moves from transient to steady-state) is also challenging. Bluetooth signals exhibit multiple local energy maxima, both within the transient and steady-state portion, making envelope detection challenging. Successful implementations of transient detection identify the steady-state by its gradient, find the length of the transient for each record in a class, and then calculate the median transient length of multiple records within a class to improve accuracy (Ali et al. 2019).

2.5 Classifier

The performance of classifiers in RF fingerprinting applications has not been extensively studied. The most prominent review was completed by Ali et al. (2019), who compared Complex Decision Tree, Linear Support Vector Machine (L-SVM), and Linear Discriminant Analysis. Their results show CDT is least suited to accurate classification, followed by LDA, with L-SVM performing the best overall. This is also supported by results from Aghnaiya et al. (2019), which show good performance of L-SVM classifiers when applied to features extracted by Variational Mode Decomposition.

Uzundurukan, Ali, Dalveren & Kara (2020) compared the performance of non-linear Support Vector Machine (SVM) and Neural Network classifiers on RF Fingerprinting of Bluetooth devices, while Helluy-Lafont et al. (2020) compared logistic regression, multi-layer perception, random-forests, and SVM. In each analysis, SVM with a linear kernel function (L-SVM) showed to have superior classification accuracy compared to other classifiers. Some researchers indicate multi-class SVM classifiers achieve accuracy of more than 99.8% for sets containing more than twenty records (Helluy-Lafont et al. 2020).

SVM classifiers work by mapping all input data records into n -dimension hyperspace. A kernel function is used to turn the input data into a point in the hyperspace, reducing computational effort and time required to classify records. Once the kernel function is applied, a hyperplane is drawn between the records to separate the classes. The hyperplane becomes the decision point for binary classification—everything on one side of the plane is one classification, everything on the other side is the other. The hyperplane is chosen to maximise the margin between the classes, as shown in Figure 2.2. SVM algorithms were originally developed for binary classification problems, though they can be extended for multi-class classification through ‘one-to-one’ or ‘one-to-rest’ approaches (Hsu & Lin 2002).

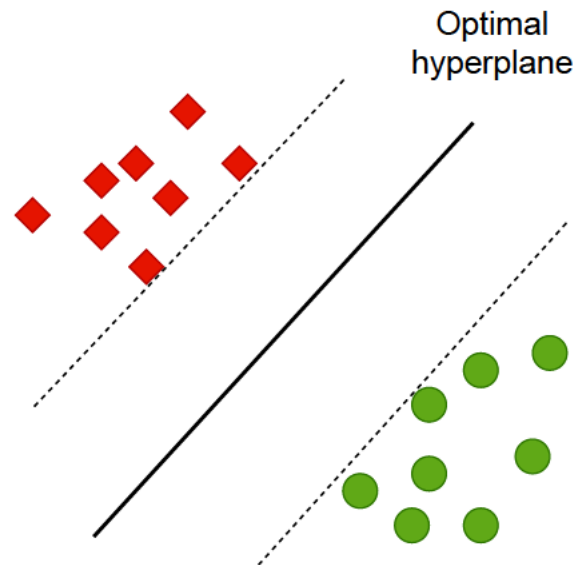


Figure 2.2: Example of an optimal hyperplane formation using Support Vector Machine algorithm with a linear kernel function for binary classification of two classes.

2.6 Literature Gap

The RF fingerprinting method for Bluetooth described by Ali et al. (2019) has, to the authors knowledge, only been deployed in laboratory testing under ideal conditions.

Numerous real-world effects are known to affect the received signal and could affect reliability of RF fingerprinting systems. While others have attempted to model and identify the environmental effects on physical layer RF fingerprinting, studies have been limited to different communications protocols, fingerprinting, and classification systems (Wenhao et al. 2015).

The temperature of a crystal oscillator is known to cause shift in output frequency, which manifests in clock and frequency error (Helluy-Lafont et al. 2020). As another example, the movement of a transmitter toward or away from the receiver can introduce a Doppler shift, which is an apparent frequency shift due to the motion during transmission. It is not known how these frequency errors may affect RF fingerprinting, nor how immune the system described by Ali et al. (2019) is to real-world effects.

2.7 Conclusion

This chapter has presented existing research relating to RF fingerprinting systems, with particular focus on identification of WiFi and Bluetooth transmitters. Methods for extracting critical identifying features from the turn-on transient portion of a waveform are explored, with Empirical Mode Decomposition and Variable Mode Decomposition showing promise as methods to transform the waveform into a time-frequency-energy distribution. Potential solutions for transient detection are explored, with Energy Criterion highlighted as the prominent method for detecting the transient start. Additionally, the selection of classifiers for similar systems is explored, with Support Vector Machine classifiers showing the best performance for classification of similar features. An opportunity for further work is identified, specifically in quantifying how environmental variables affect RF fingerprinting systems.

Chapter 3

Methodology

3.1 Introduction

This chapter presents the methodology used to implement a physical-layer RF fingerprinting system for Bluetooth devices, and methodology to verify system performance. The processes for construction/implementation and verification of each major subsystem is also detailed.

3.2 Proposed Objectives

The proposed objectives to be addressed by this research are outlined below.

1. Design and build an RF front-end downconverter to allow acquisition of Bluetooth signals using equipment with low sampling rates (i.e. 500 MS s^{-1}).
2. Research and implement a system to automatically extract the ‘turn-on’ transient from a captured Bluetooth signal at various sample rates.
3. Assess the accuracy of the classifier to correctly identify a device for a given turn-on transient for an existing dataset, and compare these results to those in the literature.
4. Using the RF downconverter built earlier, acquire a collection of ‘real-world’ fingerprint data under controlled conditions, specifically:

- (a) under a range of (transmitter) temperatures;
 - (b) while the transmitter is moving toward or away from the receiver; and
 - (c) in the presence of ambient (background) noise (i.e. lower SNR).
5. Assess the accuracy of classifiers to correctly identify a device for a given turn-on transient when presented with ‘real-world’ fingerprint data. Compare and contrast these results against the performance of classifier when using controlled fingerprint data.

The realisation of these objectives provides valuable insight into the viability of implementing the RF fingerprinting system using lower-cost hardware to detect authorised and unauthorised Bluetooth transmitters, even when the transmitters deploy advertising address or MAC address randomisation.

3.3 RF Downconverter

This section provides a high-level plan for construction of the RF downconverter, and the methodology for verifying its performance. The actual construction and verification testing results are detailed in Chapter 4.

3.3.1 RF Downconverter Construction

Both Bluetooth Classic and Bluetooth Low Energy (BLE) operate in the 2.4 GHz ISM band, with all channels (including frequency guard bands) occupying less than 80 MHz of contiguous bandwidth. The Bluetooth Classic system operates on 79 RF channels. RF channels are 1 MHz wide, and spaced every 1 MHz, such that channel centre frequency can be defined by:

$$f = 2402 + k \text{ MHz, where } k = 0, \dots, 78 \quad (3.1)$$

Similarly, the BLE system operates at the same frequency ranges, and relies on 1 MHz RF channels, but increases the spacing between channels to 2 MHz, which reduces the number of RF channels to 40. Three of these RF channels (channels 37, 38, and 39) are

used as primary advertising channels. Channel centre frequency in BLE can be defined by:

$$f_{BLE} = 2402 + 2k \text{ MHz, where } k = 0, \dots, 39 \quad (3.2)$$

Note that the BLE channel ordering is not sequential, so k in Equation 3.2 does not directly map to channel number.

To prevent aliasing introduced by insufficient sampling of the signal, it is necessary to sample faster than the Nyquist rate (Leis 2011). The formula for Nyquist rate is shown in Equation 3.3:

$$f_s \geq 2f_{max} \quad (3.3)$$

where f_s is the sampling frequency and f_{max} is the highest frequency components of the signal being sampled.

The Nyquist rate required to sample the highest-frequency Bluetooth signals (assuming direct sampling) is $f_s \geq 4.961 \text{ GHz}$; additionally, this is the minimum sampling rate, and in practice a higher rate is used. This will necessitate the use of expensive sampling equipment.

To reduce the sampling rate, a downconverter is used to shift the entire Bluetooth spectrum to a lower band. When two signals are mixed together through fundamental mixing, the output can be expressed by the general form:

$$f_{out} = m \cdot f_1 \pm n \cdot f_2 \quad (3.4)$$

where m and n are integer values. This will generate both an upper sideband (USB) at $f_1 + f_2$, a lower sideband (LSB) at $f_1 - f_2$, and a number of intermodulation components due to mixing of integer harmonics of the inputs frequencies. In downconversion two signals are mixed with the intention of using the generated LSB as the output. It is often convenient to use the simplified use case formula:

$$f_{IF} = |f_{LO} - f_{RF}| \quad (3.5)$$

By tuning the VCO to generate a 2380 MHz signal, the downconverter can shift the Bluetooth signals into the band 20–100 MHz. This makes the Nyquist rate 200 MHz, which is trivial to sample using lower cost equipment.

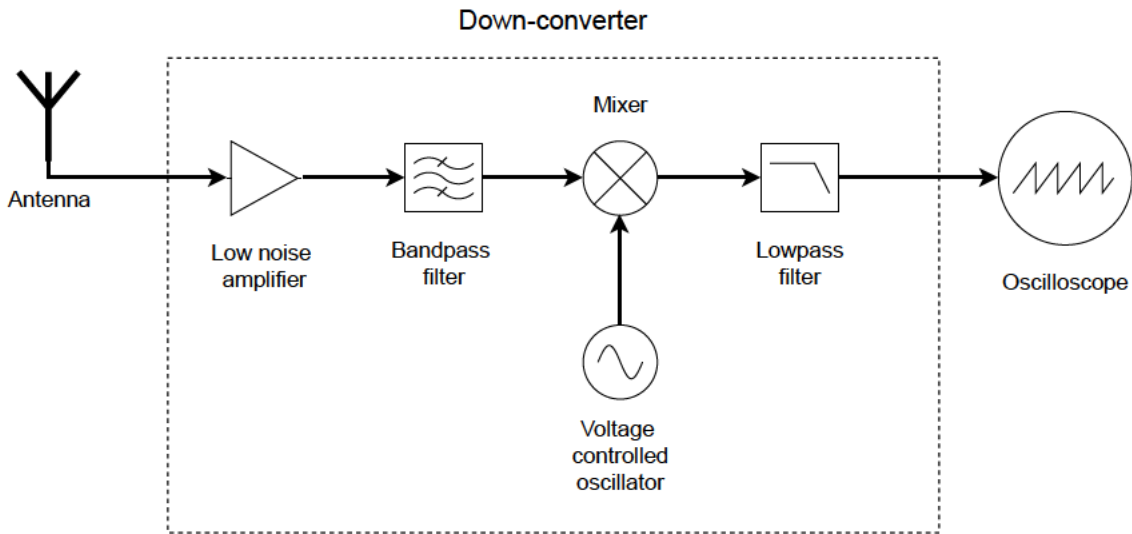


Figure 3.1: Block diagram of RF acquisition system, highlighting the major components of the downconverter.

A low noise amplifier (LNA), tuned to the area of interest, allows the amplification of Bluetooth signals prior to filtering, which removes noise from adjacent bands. After mixing, a low-pass filter removes higher frequency components that may have been introduced. A diagram of the downconverter is shown in Figure 3.1.

This system has the added benefit of wide bandwidth, sufficient to acquire the entire Bluetooth band using a single low-cost acquisition system.

3.3.2 RF Downconverter Verification

The performance of the RF downconverter system can be verified through two methods.

The first test involves applying a narrow-band single-tone frequency (f_{RF}) to the downconverter RF input, and observing the downconverter output (f_{IF}). The frequency components of the downconverter output is then observed using either the PicoScope 5444B, or a spectrum analyser (Rohde & Schwarz FSH6). This test shows the frequencies and power levels of the downconverter output. The Total Harmonic Distortion (THD) of the system can then be calculated over the entire sampling frequency range (DC–250 MHz).

The second test involves using the downconverter to shift Bluetooth signals into the sampling range of the downconverter, and observing the frequency components and overall shape of Bluetooth waveforms. This is performed to confirm the downconverter does not

introduce any observable errors.

3.4 Acquisition System

This section provides a high-level plan for implementing the acquisition system, and the methodology for verifying its performance. The actual implementation is detailed in Chapter 4, and verification testing results are detailed in Chapter 6.

3.4.1 Acquisition System Construction

Earlier works by Uzundurukan, Dalveren & Kara (2020*b*) used two different methods for collection of Bluetooth signals. The first method used direct sampling of the spectrum using an oscilloscope with high-sampling rates (5 GS s^{-1} , 10 GS s^{-1} , 20 GS s^{-1}). The second method, which is reproduced in this project, uses a modular RF downconverter to shift the frequencies of interest to a range that can be sampled using lower-grade equipment.

The acquisition system in this research will use a PicoScope 5444B PC oscilloscope for Bluetooth signal acquisition (Pico Technology 2016). The 5444B has 200 MHz analogue bandwidth, so can sample the entire Bluetooth band after downconversion. It can sample a single channel at rates up to 500 MS s^{-1} in 12-bit resolution, or up to 1 GS s^{-1} at 8-bit resolution; this exceeds the Nyquist rate requirement of 200 MS s^{-1} .

3.4.2 Acquisition System Verification

The acquisition system verification is identical to the RF downconversion verification testing. The first test involves applying a known frequency input (f_{RF}) to the downconverter RF input, and passing the output of the downconverter to Channel A of the PicoScope 5444B. The PicoScope software is set to display the incoming signal in both the time-domain (scope view) and frequency-domain (spectrum analyser view). Triggers should be not placed on the PicoScope at this stage, to ensure triggers do not inadvertently block signals being detected. The signal is observed in the time-domain to ensure there is no distortion or superposition of the VCO test signal. The signal is also observed in the frequency

domain to confirm the acquisition system is able to cleanly detect the entire frequency range of interest.

Once the VCO test signal is observed and verified, experimentation can be conducted using real Bluetooth signals. The objective of these experiments is identify the trigger settings that allow the acquisition system to reliably detect Bluetooth signals.

3.5 Transient Detection Software

This section provides a high-level plan for implementing the transient detection system, and the methodology for verifying its performance. The actual implementation is detailed in Chapter 4, and verification testing results are detailed in Chapter 6.

3.5.1 Transient Detection Software Construction

The transient-detection system is implemented in MATLAB[®]. A single waveform vector is passed into the function, and is parsed to identify the start and end of the transient period. The start and end instants are returned.

Based on work by Mohamed et al. (2020), the Energy Criterion method is the most promising candidate for detecting the transient start point. Ali et al. (2019) published the pseudocode for an algorithm that estimates transient end point based on local energy maxima after the transient start point, and refines the estimate based on averaging a number records from within a given class. The published pseudocode includes six unknown parameters, which need to be determine experimentally as part of the construction.

3.5.2 Transient Detection Software Verification

The transient detection system is verified through two separate processes. The first stage is to evaluate the placement of the transient start and end points. The transient detection system is applied to a number of Bluetooth waveforms. For each waveform, and the detected transient start and end points are plotted and inspected. The start and end points should be consistently located at the intuitive location for each waveform.

Once the start and end points appear to be placed consistently in the correct location, the second stage can be completed. The transient extraction system is applied to a dataset of Bluetooth waveforms, and the results are inspected in a box-plot. Given transient length remains constant for a given device, an effective transient extraction system should show tight grouping of transient length for each device. This can be displayed as a box-plot.

This sub-system can be evaluated using a dataset of Bluetooth turn-on transients assembled by others (Uzundurukan, Dalveren & Kara 2020*b*).

3.6 Feature Extraction System

3.6.1 Feature Extraction System Construction

The feature-extraction system is implemented in MATLAB[®], based on the work completed by Ali et al. (2019) and verified by others (Uzundurukan, Dalveren & Kara 2020*b*). Thirteen features are extracted for each record for dimensionality reduction. The features extracted are shown in Table 3.1.

Table 3.1: Overview of the thirteen features to be extracted from each record for classification.

Feature group	Feature name	Feature label
Transient signal and energy envelope	Duration of transient	f_1
	Total energy of transient energy	f_2
	Total energy of transient energy envelope	f_3
	Variance of transient energy envelope	f_4
	StD of instantaneous phase of transient signal	f_5
TFED of the transient signal along time axis	Entropy of instantaneous phase of transient signal	f_6
	Length of transient energy distribution	f_7
	Slope of transient energy distribution	f_8
	Variance of sum of transient energy distribution	f_9
	Maximum of sum of transient energy distribution	f_{10}
	Third order polynomial fitting coefficient of sum of transient energy distribution	f_{11}
TFED of the transient signal along frequency axis	Maximum of sum of transient energy distribution	f_{12}
	Variance of sum of transient energy distribution	f_{13}

3.6.2 Feature Extraction System Verification

The feature extraction sub-system is responsible for performing dimensionality reduction—that is, analysing each record (waveform) and extracting the smallest number of dimensions that account for the observed properties of that record. Dimensionality reduction is an important step, as the reduction of information being used by the classifier reduces the risk of over-fitting the model, decreases training time, and improves accuracy.

Because dimensionality reduction is necessarily an abstraction of the waveform data, it is difficult to intuitively inspect results and confirm their accuracy. Additionally, the

feature extraction performance cannot be easily verified, as there is no known reference implementation or dataset. However, it is possible to confirm each feature within a class remains consistent; this can be achieved by viewing each feature in a box-plot, grouped by class (device).

This sub-system can be evaluated using a dataset of Bluetooth turn-on transients assembled by others (Uzundurukan, Dalveren & Kara 2020*b*).

3.7 Classifier

3.7.1 Classifier Construction

The process for developing classification models, and then assessing the performance of those models, is identical for each dataset and predicted value permutation. The general process is:

- Store all features calculated by the feature extraction stage in a table.
- Create a (*DeviceID* column, and store the device identified against each record.
- Create a (*DeviceModel* column, and store the device type (e.g. 'iPhone7') against each record.
- Ingest the table to the MATLAB Classification Learner app, selecting the correct predictive value (*DeviceID* for classification to a particular device, *DeviceModel* for classification to a particular model of device), and use cross-fold validation ($k = 5$). Take care not to include the unused predictive value as a feature.
- Using default values, generate models for all classifier types.
- Using default values, create an optimised version of the best-performing classifier from the last run.

By using either the *DeviceID* or *DeviceModel* as predictive values, it is possible to build classifiers to attribute turn-on transients to a particular transmitter (e.g. this particular Apple iPhone 7), and to attribute turn-on transients to a particular *type* or transmitter (e.g. any Apple iPhone 7).

3.8 Experiment Design

This section describes the design of the proposed experiment to assess the efficacy of the RF fingerprinting system when input datasets are collected under conditions that are not optimal.

3.8.1 Experiment design

The experiment requires collection of Bluetooth turn-on transients under a range of controlled conditions. Due to the nature of RF fingerprinting, it is not necessary that the Bluetooth devices are of the same type, but it is equally not necessary that they be different. Within this research, turn-on transients are captured from 17 devices representing nine unique device models.

The intention of the research was to vary one independent variable at a time, creating datasets that allow the effect of the variation to be determined. Measurements were to be collected for the following variables:

- high signal to noise ratio reference (labelled **Dataset A**);
- transmitter temperature variance (labelled **Dataset C**);
- movement/velocity variance (labelled **Dataset B**); and
- lower signal to noise ratio (labelled **Dataset D**).

Due to the permutations involved in capturing multiple samples per variable value for multiple devices, many discrete samples are required.

Once the datasets are captured, two separate models are built: the first model is built from the reference dataset, which was captured under extremely controlled conditions; the second model is built based on the remaining datasets, introducing variation in the measurements.

Once the two classifier models are built, datasets are created for each value of each variable being tested. For example, to test the effect of temperature, a dataset is built to include features from all devices for a temperature of 5 °C; a second dataset would be produced

for 10 °C, and so on. This process is repeated for each variable type and variable value recorded. Each dataset can then be applied to each model, allowing the performance of the classifier to be evaluated for each value of each variable.

3.8.2 Dataset collection

Unfortunately, COVID-19 lockdowns affected ability to access equipment and locations required to collect this data within the project time-frame. As a result, this research only presents the high SNR reference dataset, **Dataset A**.

The data collection methodology is based on those described within Uzundurukan, Dalveren & Kara (2020*b*) and Uzundurukan, Ali, Dalveren & Kara (2020).

High SNR reference samples are collected for each device. The transmitter and receiver will be placed in a low-noise environment, such as an RF shielded room. Both the transmitter and receiver will operate at room temperature. The transmitter is positioned approximately 30 cm from the receiver antenna. Bluetooth will be activated, and the turn-on transient captured. This process will be completed for all transmitters. This dataset is labelled **Dataset A**.

For the temperature variance testing, the transmitter and receiver are placed in a low-noise environment. The transmitter is positioned approximately 30 cm from the receiver antenna. The transmitter is heated or cooled to within 1 °C of the target temperature, taking care to ensure the receiver equipment remains at room temperature. Bluetooth is activated, and the turn-on transient captured. This process completed for all transmitters, across the range 5–35 °C (with a step-size 5 °C steps). These ranges are consistent with stated manufacturer ranges for smart-phone devices. This dataset is labelled **Dataset B**.

The radial walk testing process is adapted from recommended methods for characterising intrusion detection sensors (Bowen, Sohinki, Potter & Vaughn 2017) (see Figure 3.3). The transmitter and receiver will be placed in a relatively low noise, but real-world, environment. Due to the distances required, it is not possible to eliminate background noise. Both the transmitter and receiver will operate at room temperature. The transmitter is located a specified distance from the receiving antenna. The maximum distance between the transmitter and receiver must be close enough to allow detection of the turn-on tran-

sient. The transmitter will then be moved toward the receiver at a given velocity; while moving, Bluetooth is activated, and the turn-on transient captured. This capture should include a minor Doppler shift in the received waveform. This process is completed for all transmitters, across the velocity range 0.5 m/s, 1 m/s, 2 m/s. This dataset is labelled **Dataset C**, and includes subsets for each velocity under test.

Doppler shift in an observed signal arises when the distance between the transmitter and observer is not constant, and can be given by:

$$f' = f \times \left(\frac{1}{1 + \frac{v}{c}} \right) \quad (3.6)$$

where:

f' is the observed frequency resulting from the Doppler shift

f is the transmit frequency

v is the velocity of the observer in the direction away from the transmitter

c is the velocity of the wave in the given medium.

The Bluetooth Core Specifications (Bluetooth Special Interest Group 2019) includes tolerance for initial channel frequency error (up to ± 75 kHz from the centre frequency), plus tolerance for frequency drift within a packet (up to ± 40 kHz). A Bluetooth transmitter moving at 30 m/s (108 km/h) with respect to the observer would have a Doppler shift of less than 250 Hz. Given the magnitude of the allowable drift compared to the magnitude of the frequency change caused by Doppler shift at expected human speeds, it is highly likely that a transmitter will be accurate enough for a sub-300 Hz frequency shift to be detected. Figure 3.2 shows the expected frequency shift for Bluetooth devices moving for a range of speeds.

Given the relatively low speeds being employed, it is expected the Doppler shift introduced on the Bluetooth signals will be imperceptible to the acquisition system, and will therefore have no effect on the accuracy of the classifier.

For the lower SNR testing, the transmitter and receiver will be placed in an environment where other WiFi and Bluetooth devices are operating. Both the transmitter and receiver will operate at room temperature. The transmitter will be located 30 cm from the receiver antenna. Bluetooth will be activated, and the turn-on transient captured. This process will be completed for all transmitters. This dataset will be labelled **Dataset D**.

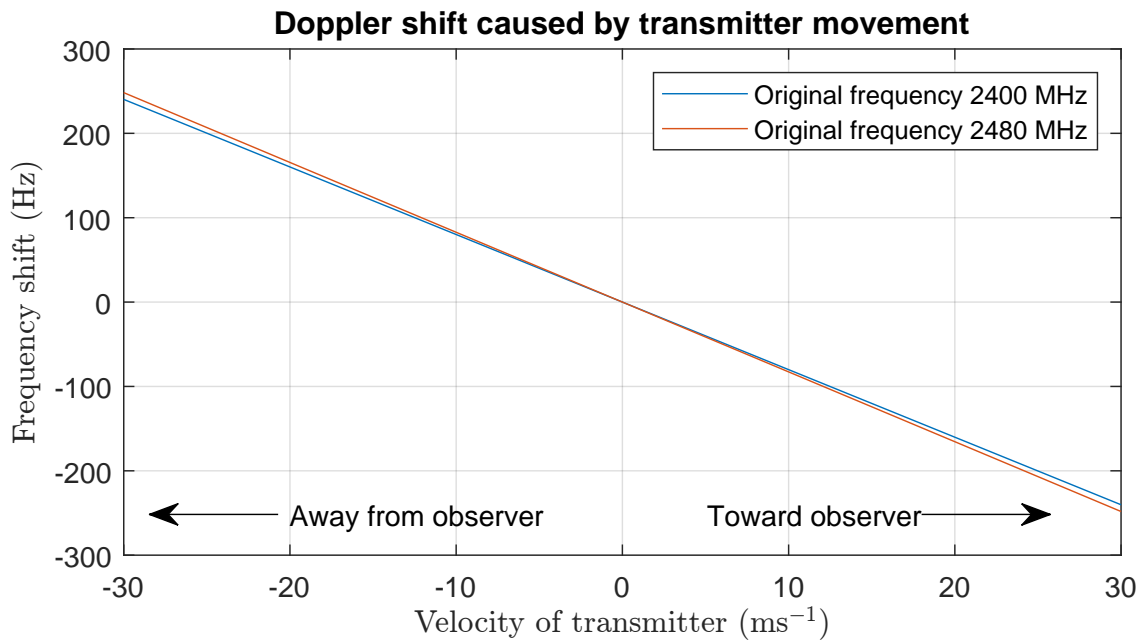


Figure 3.2: Predicted Doppler shift of Bluetooth devices at expected human speeds. Calculated by application of Equation 3.6 for Bluetooth frequency range and velocities of up to 30 m s^{-1} .

The support vector machine training process splits the dataset for tuning purposes. Brik et al. (2008) report seven records (minimum) are required per transmitter to develop an effective training set. This means the following number of samples must be acquired, per device, to facilitate an effective model:

- Reference (ideal conditions): seven samples per device.
- Temperature: one samples per temperature per device (seven samples per device).
- Velocity: Three samples per velocity per device (nine samples per device).

Based on the above, each device must produce 23 unique samples. It is expected that the Bluetooth devices will transmit very rapidly, so many transients can be acquired during a single test.

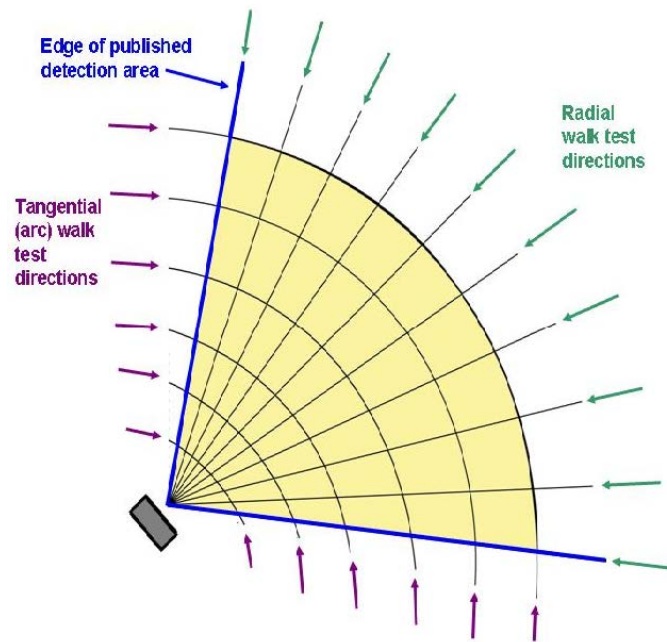


Figure 3.3: Overhead plot showing radial walk testing paths for Doppler testing (Bowen, 2017).

3.8.3 Interpretation of results

The results from a classifier are expressed as a confusion matrix, which summarises predictions the classifier made, and highlights how many were correct or not. Confusion matrices are good for understanding the classifiers performance on the dataset, but not for intuitively comparing classifiers. However, classifiers can also be assigned an accuracy score, expressed as a percentage, which describes how often the classifier is correct overall. The classifier accuracy score is better for intuitively comparing classifiers.

The experiment aimed to identify:

- the effect of transmitter temperature variance on RF fingerprinting;
- the effect of transmitter movement on RF fingerprinting; and
- the effect of ‘real-world’ input signals RF fingerprinting.

Each dataset was to be statistically analysed to determine if there is a link between that variable and the success of the RF fingerprinting system. However, due to COVID-19 lockdowns, these datasets could not be acquired, and the results could not be compared.

3.9 Project plan

The research project is broken down into six major phases:

- Collection and setup
- Construction of acquisition system
- Acquisition of datasets under different conditions
- Classification of acquired RF fingerprints
- Analysis of results
- Dissertation writing

A Gantt chart showing the project timeline is presented in Appendix C.

3.10 Chapter summary

This chapter has described the high-level methodology for construction of physical-layer RF fingerprinting system for Bluetooth devices, including a brief overview of the project plan to realise this. Methodology describing the realisation of critical subsystems, and methods for verifying performance of those subsystems, are also provided. Finally, methodology for the collection of samples and interpretation of the classification results is described.

Chapter 4

System Design

4.1 Introduction

This chapter describes the system design used to develop a Radio Frequency (RF) fingerprinting system, consistent with that described in Chapter 3. This chapter proposes the overall system design, and details implementation of the RF downconverter, acquisition system, transient detection system, and feature extraction.

4.2 System Model

A system flow chart that addresses the major components of the RF fingerprinting system is presented in Figure 4.1. Red blocks indicate a hardware component or subsystem. Blue blocks indicate the PicoScope software. Green blocks indicate code implemented in MATLAB®.

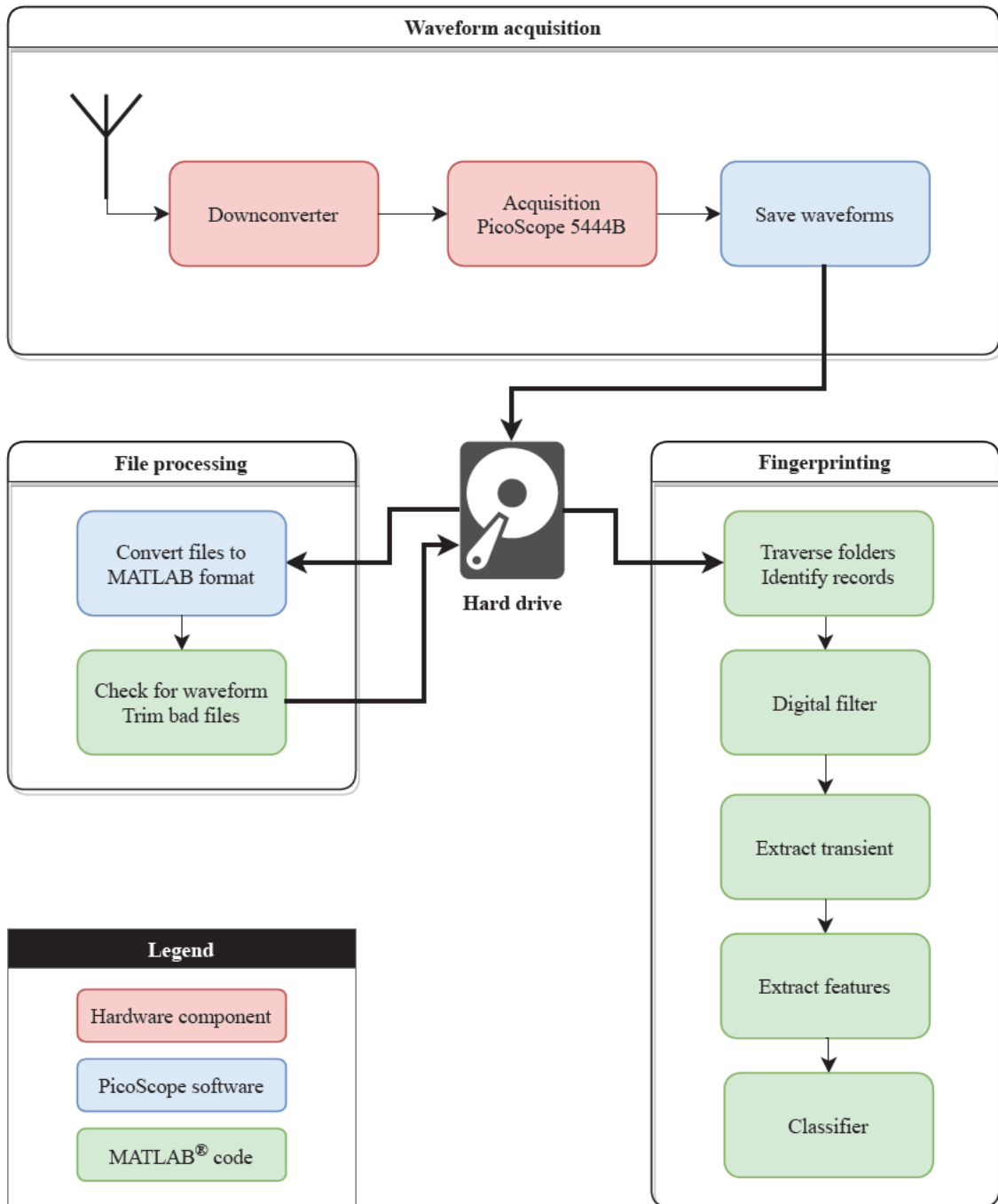


Figure 4.1: Block diagram showing the major components of the constructed system.

4.3 RF Downconverter

A modular downconverter unit was created, based on the earlier work by Uzundurukan et al. (2017) and Uzundurukan, Ali, Dalveren & Kara (2020). Components were selected for modularity, suitability, and accessibility. The LNA used was a Mini Circuits ZQL-2700MLNW+, as it exhibits 25 dB gain across the range 2200–2700 MHz, has a very low noise figure (less than 1.5 dB), and good gain flatness. The bandpass filter used was a Mini Circuits VBF-2435+, which has a passband of 2340–2530 MHz. The lowpass filter used a Mini Circuits VLFX-105, as it has passband from DC to 105 MHz, which is just wider than the expected bandwidth output from the mixing stage.

The original intention was to a VCO to produce the local oscillator frequency source of 2500 MHz, consistent with the downconverter used by others. However, when applied to the Bluetooth frequencies of interest (2400–2480 MHz), this approach results in spectral inversion.

Spectral inversion, also known as frequency inversion, occurs when a mixing operation with high-side local oscillator (LO) injection is performed (cases where $f_{LO} > f_{RF}$), and the lower sideband (LSB) is used as the output. The general formula for mixer output, shown in Equation 3.4 is reproduced below.

$$f_{out} = f_1 \pm f_2 \quad (4.1)$$

Take f_1 to be the input RF frequency f_{RF} , and f_2 to be the local oscillator frequency f_{LO} . For downconversion through high-side LO injection, $f_{RF} - f_{LO}$ results in a negative frequency, which is reflected around the DC point to yield the absolute (positive) frequency. For cases where f_{LO} remains fixed and f_{RF} varies, the reflection means that f_{IF} decreases as f_{RF} increases. This is illustrated in Figure 4.2. To overcome spectral inversion, the local oscillator frequency of 2380 MHz was selected.

The local oscillator generator was a VCO, built using the Crystek Microwave CVC055CC-2380-2380 which provides a narrow tuning range centred around 2380 MHz, with very small adjustment (approximately ± 2.5 MHz). An LM7805 linear regulator was used to supply power to the VCO, and to provide the tuning voltage upper rail. To provide the tuning voltage, a small circuit was created using a multi-turn trimpot as voltage divider. Because of the high-impedance on the VCO tuning input, loading effects were negligible,

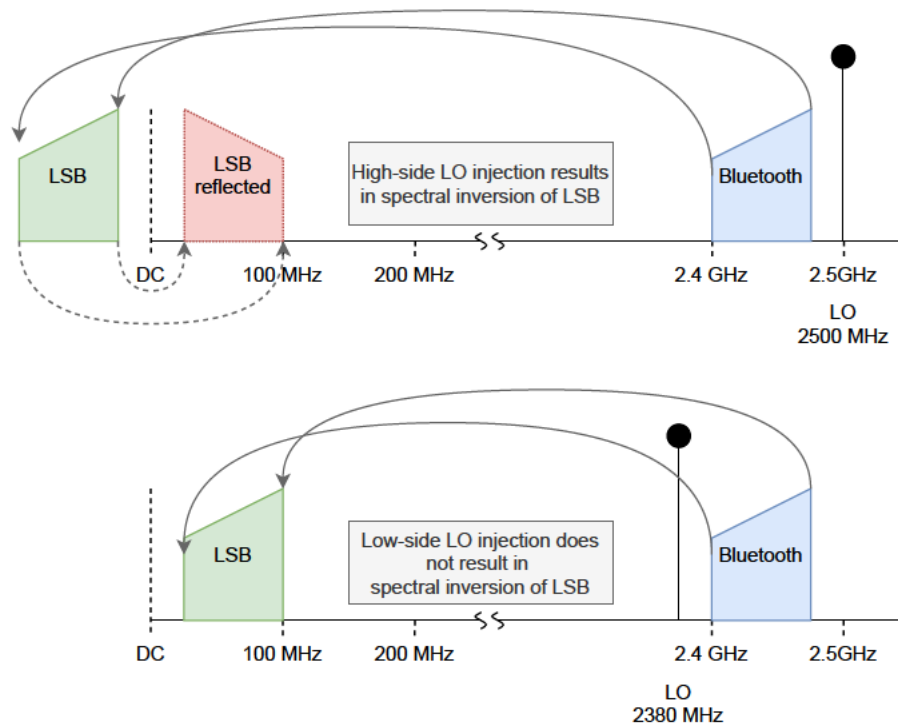


Figure 4.2: Use of a mixer to downconvert RF signals. Shown are first-order component output of the downconverter when mixing Bluetooth signals with two local oscillator frequencies. Local oscillator of $f_{LO} = 2500$ MHz (top) results in spectral inversion of the lower side band, but $f_{LO} = 2380$ MHz (bottom) does not result in spectral inversion.

and a buffer circuit was not required. The output of local oscillator circuit was validated using a Rohde & Schwarz FSH6 spectrum analyser, and confirmed to produced the desired 2380 MHz to within 40 kHz.

All components are mounted on aluminium plates, and placed within a diecast aluminium enclosure to limit effects of background noise, and to improve thermal stability. The built downconverter system can be seen in Figure 4.4 and Figure 4.5.

To confirm correct operation of the complete downconverter system, a test VCO was used to generate narrow-band continuous wave (CW) test frequencies within the frequency range of interest (2400–2380 MHz), and the frequency components of the output signals were inspected. The test VCO, shown in Figure 4.3, is based on Crystek the CVCO55CC-2380-2580, and has a tunable range of 2380–2580 MHz with a nominal output power of +5 dBm. The frequency components of the downconverter output is then observed using using either the PicoScope 5444B, or a spectrum analyser (Rohde & Schwarz FSH6). This test shows the frequencies and power levels of the downconverter output. The Total Harmonic Distortion (THD) of the system can then be calculated over the entire sampling

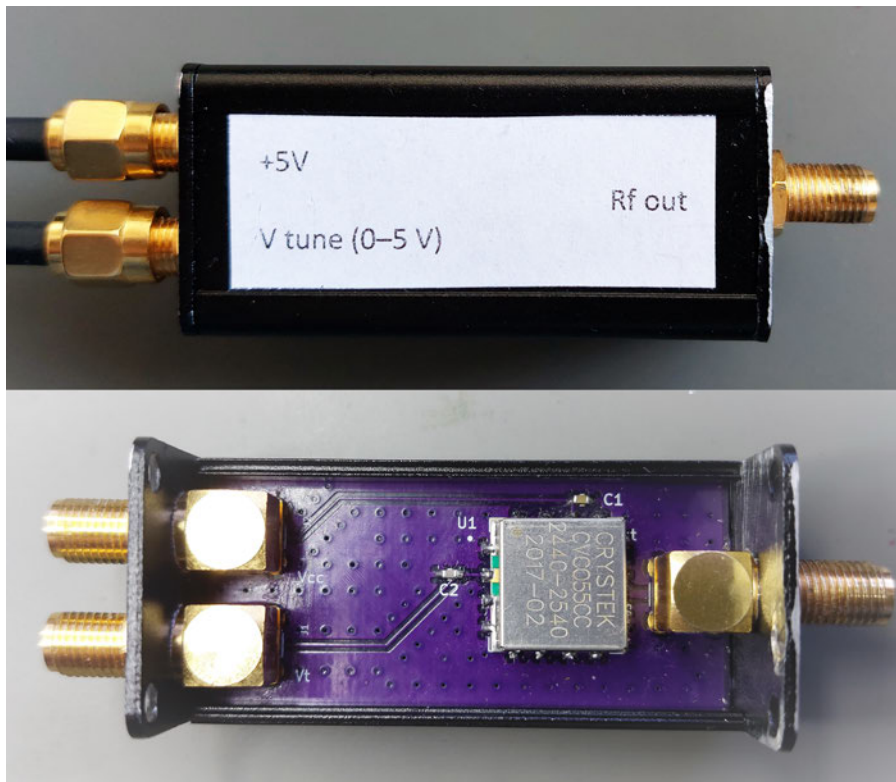


Figure 4.3: Photograph of Voltage Controlled Oscillator used to produce continuous wave frequencies to test the performance of the downconverter system.

frequency range (DC–250 MHz).

The second test involves using the downconverter to shift Bluetooth signals into the sampling range of the downconverter, and observing the frequency components and overall shape of Bluetooth waveforms. This is performed to confirm the downconverter does not introduce any observable errors.

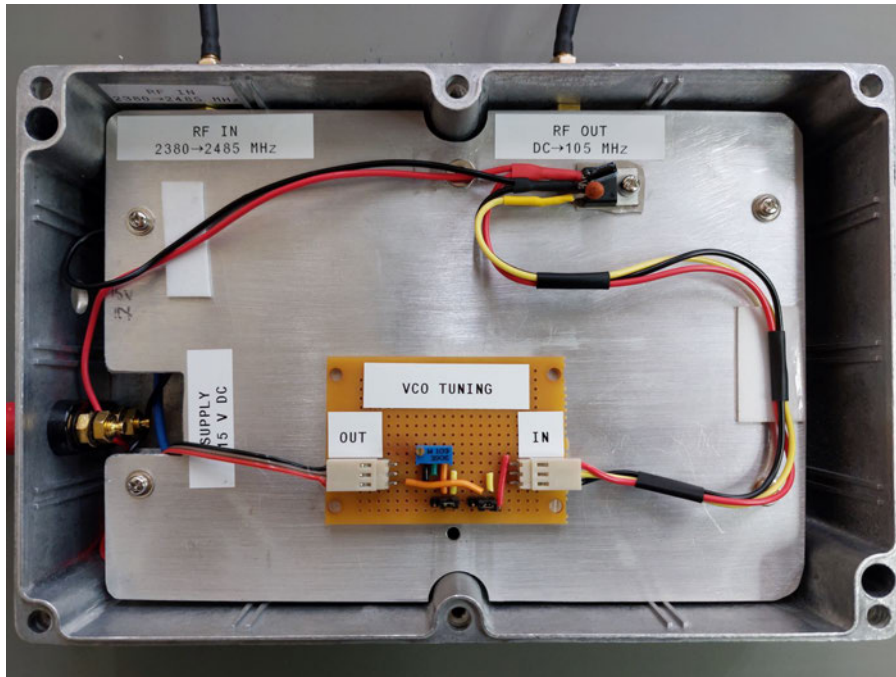


Figure 4.4: Photograph of the constructed downconverter system (top layer) showing 5 V linear regulator and tuning circuit for the voltage controlled oscillator.

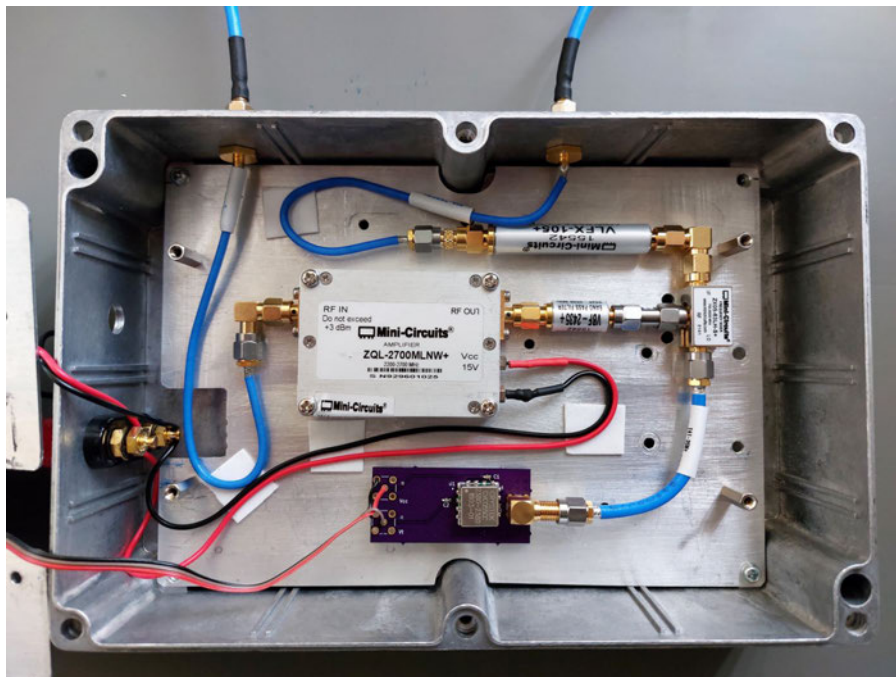


Figure 4.5: Photograph of the constructed downconverter system (bottom layer) showing the low-noise amplifier, bandpass filter, mixer, local oscillator, and low-pass filter.

4.4 Acquisition System

The PicoScope 5444B was selected as the acquisition system because it met the exceeded the minimum sampling rate and resolution (200 MS s^{-1} at 8-bits), and supported rapid automatic ingestion by a PC. While other hardware-based acquisition tools also meet these criteria, the PicoScope 5444B was also readily available to the author. It also has the benefit of being USB-powered when using two or less channels, and can be externally powered by a 5 V DC supply; these features make it suitable for field deployment where mains power is not available, which was required for some of the testing.

The PicoScope was set to sample at 500 MS s^{-1} , with a hardware resolution of 12 bits. The downconverter was connected to Channel A, and AC coupling was selected. Initially, a sample window size of 2500 samples ($5 \mu\text{s}$) was used with the trigger-point at 50% of the capture. A photograph of the complete acquisition system is shown in Figure 4.6.

The rapid triggering mode was used, to ensure all Bluetooth transients were collected. Without rapid triggering, there is a hold-off time after a transient is detected. A number of Bluetooth Low Energy devices were observed to transmit advertising packets on Channels 37, 38 and 39 in rapid succession. When using trigger modes other than rapid, the first transient is detected correctly, but the others are masked by the hold-off time of the PicoScope.

The PicoScope software allows macros and alarms to be used to automate tasks. To prevent having to manually save capture data to disk when the buffer was full, the 'Alarms' functionality was used. This was achieved by enabling alarm actions for the **Buffers Full** event, creating two alarm actions:

- Save All Buffers
- Restart Capture



Figure 4.6: The acquisition system, showing RF downconverter connected to the PicoScope 5444B.

4.5 File processing system

The PicoScope acquisition system saves multiple waveform buffers within a single proprietary PicoScope `.psdata` file. To enable processing by MATLAB[®], the PicoScope command line application is to convert the waveforms in each buffer to `.mat` format. Because the metadata of the original file is not carried over into the new file, output files are named based on the file's modified date metadata in ISO8601 format, and are appended with the buffer number of the file. For example, the tenth buffer from a capture file saved on 1 March 2000, 15:47:17 would be named `20000301T154717_010.mat`. There is also a collision detection system to prevent overwrites in case multiple `.psdata` files have the same timestamp.

Once the files are converted, they are parsed to determine if there is a transient present. The script looks for a basic transient shape (low amplitude for the first 20% of the waveform, high-amplitude for the last 30%). If a transient is detected, the file is moved to the final output location. If a transient is not detected, the file is deleted.

A MATLAB[®] script was created to automate the actions of the file processing system.

4.6 Transient Detection System

Reliable detection of the transient portion of the waveform is thought to be critical to extracting consistent and meaningful features for classification. Indeed, feature f_1 is the transient length.

The transient detection algorithm can be broken down into six steps:

- Apply bandpass filter to waveform (passband of 20–100 MHz).
- Perform DC-offset correction and amplitude normalisation.
- Calculate the signal envelope via the Hilbert transform.
- Smooth the envelope using a median filter.
- Detect start of transient portion of the signal.
- Detect the end of the transient portion of the signal.

After applying the bandpass filter to each waveform, any DC-offset within the signal is corrected, and the amplitude is normalised. The envelope of the waveform is calculated by applying MATLAB[®]'s `hilbert()` function. This results in a bi-level signal envelope that closely resembles a step-response. However, the envelope can contain noise, which frustrates the transient detection algorithms. To remove the noise, a one-dimensional median filter is applied.

Once the signal envelope is smoothed, the transient start and end instants can be detected. The literature indicates this is a well-known problem, but solutions have the potential to be computationally-expensive.

The Energy Criterion (EC) algorithm is used to detect the transient start point. EC has low computational cost, and has been successful in estimate the time of arrival of partial discharge pulses within the power field (Wagenaars et al. 2008). More recently, this technique has been evaluated for its ability to detect turn-on transients in RF waveforms (Mohamed et al. 2020). EC uses the energy content of the signal to determine transient starting points. It combines the energy of the signal (a cumulative sum of energies) with a negative trend, resulting in a global minimum at the point where the transient starts.

The EC algorithm can be expressed as

$$E'_i = \sum_{k=0}^i (x_k^2 - i\delta), \quad i = 1, \dots, N \quad (4.2)$$

where N is the discrete signal length, and δ is a negative trend expressed by

$$\delta = \frac{E_N}{N\vartheta} \quad (4.3)$$

and ϑ is a factor which reduces the delaying effect of δ .

The EC method has been compared experimentally to other common methods for RF signal transient detection, including variance fractal dimension threshold detection, Bayesian step change detection, phase detection, and mean change point detection; EC is superior in terms of computational speed and detection accuracy at higher signal to noise ratio levels (Mohamed et al. 2020). Based on this finding, EC is used with $\vartheta = 35$ to reliably detect the start of transients in the dataset.

Detecting the end of the transient is also challenging. In their earlier research on this issue, Ali et al. (2019) published pseudocode for an algorithm that estimates transient end point based on local energy maxima, and refines the estimate based on averaging a number records from within a given class. Despite the reportedly promising results, this algorithm could not be reproduced based on the information present in the literature. There are six parameters needed by the algorithm, which are not reported. Additionally, it is unclear how multiple received waveforms are attributed to a single transmitter to allow averaging to occur; without this critical step, it is unclear how averaging occurs without prior knowledge of the true class.

Within this project, the transient end point is detected by treating the signal envelope as a bi-level step-response. Detection of the end of the transient portion using settling time is a novel technique, which is not believed to have been implemented by others. The transient end instant can be calculated as the sample when the signal envelope enters and remains within a given percentage of the final steady-state. MATLAB[®]'s `settlingtime()` and `midpoint()` functions can be used to achieve this. An example waveform with transient detection markers is shown at Figure 4.7.

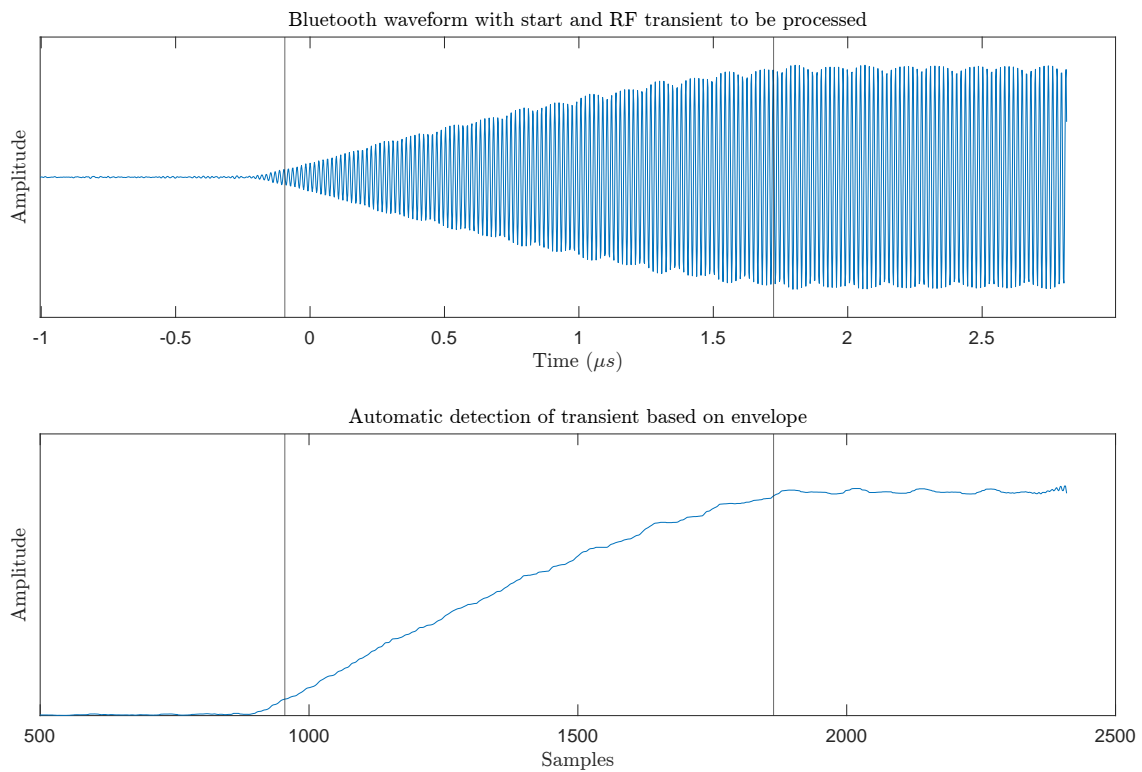


Figure 4.7: Example of automated transient detection system operation. The signal envelope is extracted, then Energy Criterion is used to find the transient start and settling time is used to find the steady-state start/transient end.

4.7 Feature Extraction System

There is significant research into potential features which can be used to classify RF signals, but this project implements a feature extraction consistent with Ali et al. (2019). The general process used is:

- apply a digital band-pass filter (FIR) to waveform record;
- detect the transient portion of the record;
- calculate the energy envelope of the transient;
- calculate the time-frequency-energy distribution (TFED) of the transient, through empirical mode decomposition and Hilbert-Huang Transform; and
- calculate 13 features described in Table 3.1.

Once the transient start location and transient length are calculated, it is possible to extract features from the records. The time-frequency-energy distribution (TFED) is

calculated through empirical mode decomposition, and the Hilbert-Huang Transform; the MATLAB[®]'s `emd()` and `hht()` functions, respectively, are used to achieve this.

Table 4.1 provides an overview of the method used to extract features from each transient. Note that the following terms are used in the overview:

- `envelope` refers to the waveform signal envelope of the transient, as calculated by the `hilbert()` function.
- `imfinse` refers to the instantaneous energies of the intrinsic mode functions, as calculated by the `hht()` function.
- `x_step` refers to the distance (time) between samples.

A complete copy of the MATLAB[®] code used to parse waveform files, detect turn-on transients, and extract these features is included in Appendix E.

Table 4.1: Overview of method for calculating features from transients.

Feature label	Feature name	Calculation method
f_1	Duration of transient	<code>endIndex - startIndex</code>
f_2	Total energy of transient energy	<code>sum(sum(imfinse))</code>
f_3	Total energy of transient energy envelope	<code>sum(envelope.^2)</code>
f_4	Variance of transient energy envelope	<code>var(envelope)</code>
f_5	StD of instantaneous phase of transient signal	<code>std(atan(imag(envelope) ./ real(envelope)))</code>
f_6	Entropy of instantaneous phase of transient signal	<code>entropy(atan(imag(envelope) ./ real(envelope)))</code>
f_7	Length of transient energy distribution	<code>y_step = diff(imfinse(:,1)) distance = sum(sqrt(x_step^2 + y_step.^2))</code>
f_8	Slope of transient energy distribution	<code>slope(polyfit(x_step, sum(imfinse', 1), 1))</code>
f_9	Variance of sum of transient energy distribution	<code>var(sum(imfinse', 1))</code>
f_{10}	Maximum of sum of transient energy distribution	<code>max(sum(imfinse', 1))</code>
f_{11}	Third order polynomial fitting coefficient of sum of transient energy distribution	<code>[p3, ~, ~] = polyfit(x_step, sum(imfinse', 1), 3); p3(1)</code>
f_{12}	Maximum of sum of transient energy distribution	<code>max(sum(imfinse))</code>
f_{13}	Variance of sum of transient energy distribution	<code>var(sum(imfinse))</code>

4.8 Classifier

The classifier stage is implemented using MATLAB[®]'s Classification Learner app.

The entire dataset is ingested in the Classification Learner app, where the app then uses k -fold cross-validation ($k = 5$) to split the dataset into training and validation sets. This is acceptable, as the features of each waveform are calculated completely independently of each other, so there is no possibility of introducing bias to the dataset.

The Classification Learner app trains classification models for all model types, and reports the success rates for the initial pass. It is noted that others have reported SVM models to be the most effective for classifying Bluetooth devices based on their turn-on transients (Ali et al. 2019). Once the models have been trained, the most successful model type is then tuned for hyper-parameter optimisation.

4.9 Chapter Summary

This chapter has presented a proposed design for a system to identify Bluetooth transmitters through RF fingerprinting. The major components and subsystems are based on the methodology outlined in Chapter 3.

Chapter 5

Data Collection

5.1 Introduction

This chapter describes the process followed for collection of signals containing Bluetooth turn-on transients, and details the devices signals were collected from. Details on the reference datasets provided by Uzundurukan, Dalveren & Kara (2020*a*), which are used to benchmark and validate the constructed system, are also provided.

5.2 Existing Reference Dataset

Uzundurukan, Dalveren & Kara (2020*a*) collected a dataset of Bluetooth turn-on transients from 27 smartphones, which reflects 16 different smartphone models over four manufacturers. The dataset includes waveforms sampled via direct sampling at 5 GS s^{-1} , 10 GS s^{-1} and 20 GS s^{-1} ; a fourth dataset includes waveforms sampled at 250 MS s^{-1} after downconversion using a modular RF front-end. Note that not all devices are sampled at each sampling rate. Where a device is sampled, 150 unique records are provided. The devices included in the reference datasets are described in Table 5.1.

This project aims to build an RF fingerprinting system with a relatively lower sampling rate (500 MS s^{-1}). Therefore, the 10 GS s^{-1} and 20 GS s^{-1} reference datasets are not used within this project. The 500 MS s^{-1} and 5 GS s^{-1} reference datasets are used to allow the MATLAB[®] code to be tested prior to acquisition subsystem being completed.

Table 5.1: Summary of the devices included in the existing reference dataset.

Brand	Model	Unique device count	
		250 MS/s	5 GS/s
Apple	iPhone 4S	2	-
Apple	iPhone 5	2	2
Apple	iPhone 5S	2	2
Apple	iPhone 6	2	2
Apple	iPhone 6S	3	3
Apple	iPhone 7	2	-
Apple	iPhone 7 plus	2	-
Apple	G4	2	2
Apple	V20	2	-
Samsung	J7	2	-
Samsung	Note 2	2	-
Samsung	Note 3	2	2
Samsung	S5	2	2
Samsung	S7 Edge	2	-
Sony	Xperia M5	2	2
Xiaomi	Mi6	2	-

5.3 Collected Dataset

The downconverter and PicoScope 5444B system were used to capture a number of Bluetooth turn-on transients under controlled conditions. Samples were collected from 16 unique devices, representing nine different models over seven device manufacturers. The devices included in the collected datasets are described in Table 5.2.

It was noted that the devices transmit advertising information rapidly after Bluetooth is enabled. As an indicative example, when the Bluetooth was enabled on an Apple iPhone 7, the sampling system was triggered 2176 times within one minute (approximately 36 triggers per second). It is noted that this occurred just after Bluetooth was enabled so the device would have been scanning for known devices. Additionally, some captures would be false-triggers. Nevertheless, it gives an idea of the rapid nature and ubiquity of Bluetooth advertising transmissions.

When a dataset was collected, several thousand Bluetooth turn-on transients were collected per device. However, to reduce the processing overhead, each dataset is trimmed to 200 turn-on transients per device.

Table 5.2: Summary of the devices included in the collected datasets.

Brand	Model	Unique device count
Apple	iPhone 7	2
Apple	iPhone X	1
Essager	BT001 Bluetooth receiver	2
Fitbit	Charge 2	1
Tile	Mate	1
Samsung	Galaxy A51	1
Samsung	Galaxy S8	1
Unihertz	Jelly Pro	1
Unbranded	BLS-TX3 Bluetooth transmitter	6

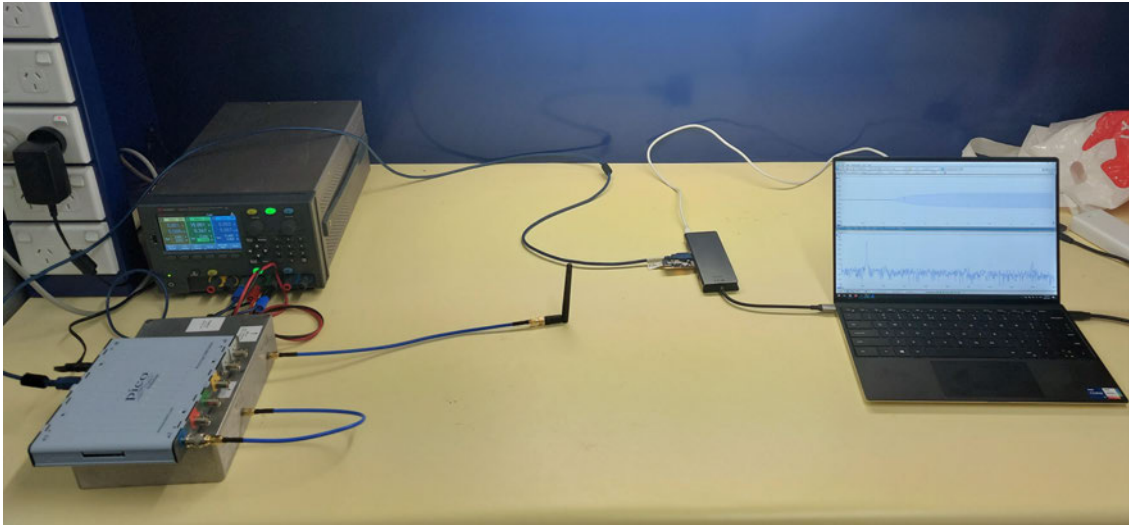


Figure 5.1: Photograph of the data acquisition system deployed in an RF-shielded room to collect Bluetooth turn-on transients. The laptop was placed in flight-mode during acquisition.

5.3.1 High SNR Dataset (Dataset A)

Samples were collected in an RF shielded room, which provided an environment free of background RF noise. All samples were collected at room temperature 25°C , and equipment was given at least 15 minutes to warm up and stabilise.

A rubber-duck style antenna, resonant at 2.4 GHz with a gain of 2.2 dBi, was used to couple Bluetooth signals into the downconverter. A Keysight E36313A linear power supply was used to provide regulated power to the RF downconverter. As the PicoScope 5444B was only sampling on Channel A, it was possible to power it from the USB port; however, the supplied 5 V power supply was connected to lessen the likelihood of power-related issues. The PicoScope was set to trigger on Bluetooth turn-on transients, buffer them in local storage, and then upload them to a laptop via USB3 when the buffer filled. The PicoScope software running on the laptop automatically saved the file to disk, cleared the PicoScope buffer, and restarted the capture.

All devices were kept outside the RF shielded room, with Bluetooth disabled (if possible). This ensured only the Bluetooth device under evaluation could be sampled. One by one, each device was placed approximately 30 cm from the antenna, and then Bluetooth was enabled; the data was captured, Bluetooth was disabled, and the device was placed outside the RF shielded room.

This dataset is labelled **Dataset A**.

5.3.2 Temperature Variance Dataset (Dataset B)

The original intention was to collect a dataset for all devices where transmitter temperature was controlled, to determine if variance in temperature had any influence in the classification of devices through Bluetooth RF fingerprinting. Unfortunately, COVID-19 lockdowns affected ability to access the equipment required to realise this testing.

5.3.3 Doppler Shift Dataset (Dataset C)

The original intention was to collect a dataset for all devices where transmitter was moving at a controlled velocity, to determine if minor variation in frequency due to Doppler shift had any influence in the classification of devices through Bluetooth RF fingerprinting. Unfortunately, COVID-19 lockdowns affected ability to access sites suitable for conducting this testing.

Additionally, the calculations performed in Section 3.8.2 predict Bluetooth devices will experience less than 300 Hz frequency shift at speeds less than 108 km/h, but the Bluetooth transmitters are allowed up ± 40 kHz drift within a single packet. Given the relatively small contribution of Doppler shift, it is unlikely to have any noticeable effect on the classification of devices.

5.3.4 Lower SNR Dataset (Dataset D)

The original intention was to collect a dataset for all devices where higher levels of background noise are present, to determine how the presence of other Bluetooth and WiFi transmitters (lower SNR) influences the classification of devices through Bluetooth RF fingerprinting.

This was very quickly abandoned, as it was not possible to discriminate turn-on transients of background Bluetooth devices from the device under test, making it impossible to train a classifier or curate a dataset for validation. Thought was given to artificially reducing the SNR through addition of additive white Gaussian noise (AWGN), however time constraints did not permit this.

5.4 Chapter Summary

This chapter has described the provenance of the reference dataset used to develop transient extraction and feature extraction systems, and to evaluate the classifier performance. Additionally, the process that was used collect **Dataset A** (the high-SNR dataset) at 500 MS/s is described. Unfortunately due to time and resource constraints, it was not possible to collect the other datasets (**Datasets B, C and D**) under different environmental conditions.

Chapter 6

Results and Discussion

6.1 Introduction

This chapter presents the results of the hardware analysis of the downconverter and acquisition subsystem, the software testing of the transient detection and feature extraction system, and the performance of the classifier to existing and acquired datasets.

6.2 Downconverter

This section describes the results of the hardware verification of the hardware downconverter. The downconverter is evaluated with two different double-balanced mixer options, and the results compared.

6.2.1 Downconverter Using Mini Circuits ZX05-63LH-S+ Mixer

Two mixers were evaluated in the downconverter design. A passive double-balanced mixer with bandwidth of 750–6000 MHz (Mini Circuits ZX05-63LH-S+) was investigated first.

The test VCO was used to apply a known frequency to the input of the downconverter, and the output was inspected using the PicoScope 5444B. Inspection of the output signal showed the downconverter stage functioned as intended, and successfully shifted the 2.4 GHz signals down to the <100 MHz range. When viewing the output signal in the

time domain, the waveform appeared to be sinusoidal with no observable DC-offset. Figure 6.1 shows a plot of the downconverter output in the time domain, and Figure 6.2 shows the same signal in the frequency domain.

It is highly desirable for the downconverter to shift the frequencies of interest to a lower band without introducing additional artefacts or biases. When applying a testing signal from the VCO, it was evident that the downconverter output contained intermodulation components within the area of interest (20–100 MHz). These intermodulation components were most prominent when the difference between f_{IF} and f_{LO} was at a minima (that is, when $f_{IF} = 2400$ MHz). However, when compared to the magnitude of the fundamental carrier, the intermodulation products were relatively small (-24.75 dBc). The Total Harmonic Distortion (THD) of the system (for an input of $f_{IF} = 2400$ MHz) was calculated as -23.24 dBc. Figure 6.5 shows the intermodulation products observed during this test.

Despite the presence of intermodulation products during the previous test, when a real waveform was applied the output appeared to be an accurate reproduction of the input waveform. The THD of the system for a real Bluetooth waveform input ($f_{RF} = 2402$ MHz) was observed to be -43.86 dBc. Inspection of the data captured using the PicoScope and the RF front-end show the downconverter stage is working well, and has successfully shifted the entire Bluetooth band (2400–2480 MHz) down to 20–100 MHz.

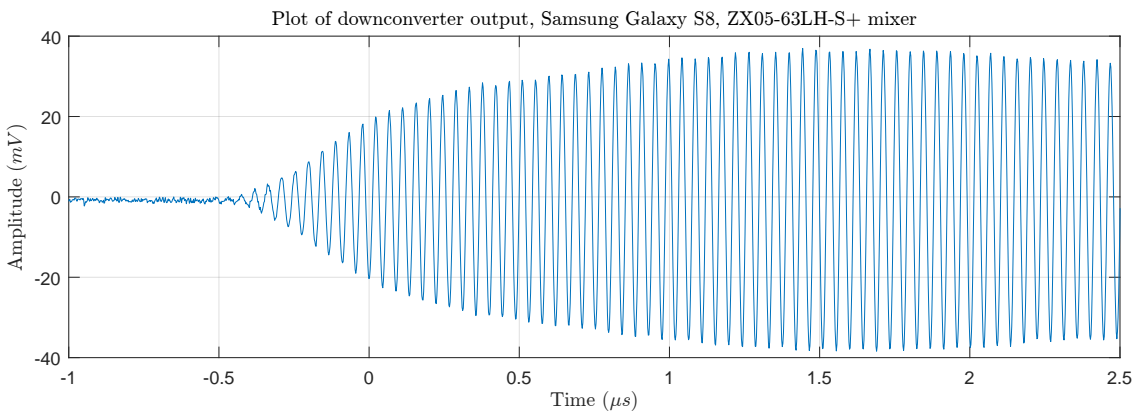


Figure 6.1: Bluetooth turn-on transient of Samsung Galaxy S8 as output by the downconverter using the ZX05-63LH-S+ mixer, shown in the time domain.

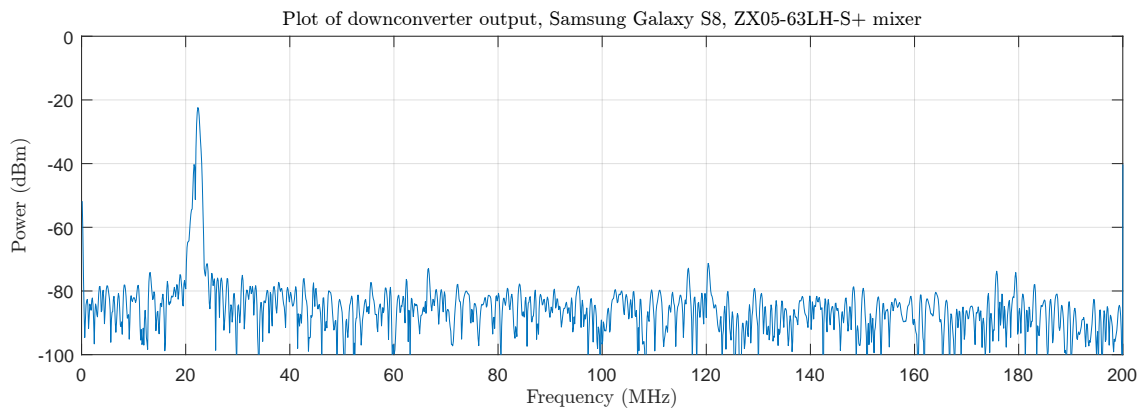


Figure 6.2: Bluetooth turn-on transient of Samsung Galaxy S8 as output by the downconverter using the HMC175 mixer, shown in the frequency domain.

6.2.2 Downconverter Using HMC175-based Mixer

In an attempt to further reduce some intermodulation components observed with the ZX05-63LH-S+ mixer, a second mixer was evaluated in the downconverter design. The second mixer was a passive double-balanced mixer based on the Analog Devices HMC175 integrated circuit. Testing was performed in the same manner as previously described. Inspection of the output signal showed the downconverter stage functioned as intended, and successfully shifted the 2.4 GHz signals down to the <100 MHz range. When viewing the output signal in the time domain, the waveform appeared to be sinusoidal with no observable DC-offset.

In the frequency domain, the intermodulation components were attenuated. Figure 6.5 shows the comparison between the ZX05-63LH-S+ mixer and the HMC175-based mixer for a narrow-band input test frequency.

This mixer caused a reduction in the fundamental output power (by approximately 2 dBm), but also significantly reduced the THD under the same conditions to -39.23 dBc. Figure 6.5 shows the intermodulation products observed during this test.

When subjected to real waveform testing, however, obvious distortion of the waveform was observed. It appeared that the output was the superposition of the downconverted RF signal plus a slow-moving step offset voltage, which varied proportionally with the power of the signal. This resulted in the expected RF waveform following an inverse step-response type shape. The amount of DC-offset only varied during the transient stage of the signal, when the power increased. Because the transient portion of the waveform is the

area of interest, this is problematic. Applying a digital bandpass filter (with a passband of 20–100 MHz) was able to remove the slow-moving distortion at higher frequencies, but was not effective at lower frequency signals. The filtered signal remained asymmetrical.

Additionally, when a real Bluetooth waveform was processed by the downconverter, harmonic components not identified during the initial testing were visible. Figure 6.4 shows the frequency components of a real Bluetooth waveform input ($f_{RF} = 2402$ MHz). Relative to the power of the fundamental, these components appeared to be much larger than those observed in waveforms collected by the system reliant on the ZX05-63LH-S+ mixer. The THD of the system using this mixer increased to -5.99 dBc.

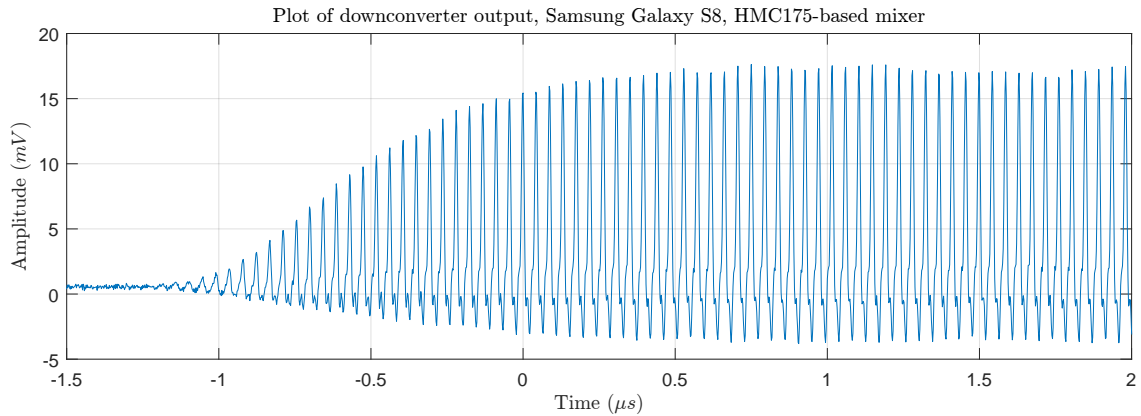


Figure 6.3: Bluetooth turn-on transient of Samsung Galaxy S8 as output by the downconverter using the HMC175 mixer, shown in the time domain.

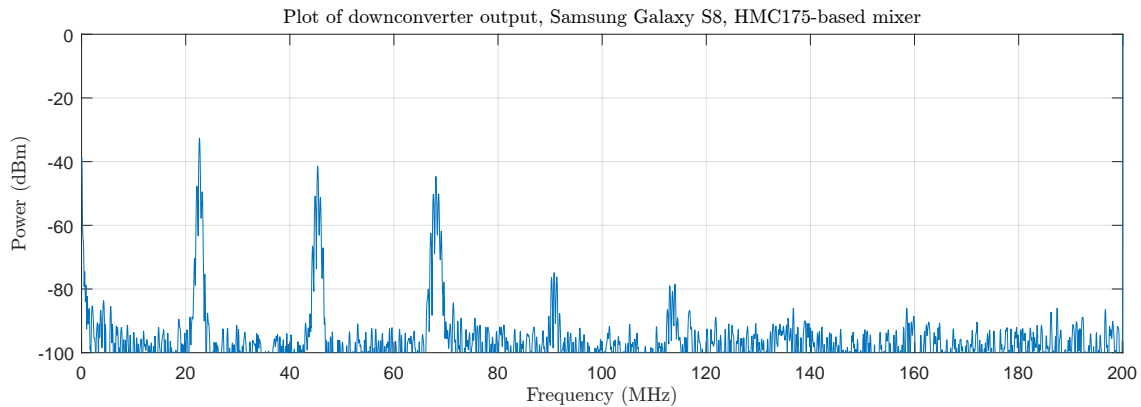


Figure 6.4: Bluetooth turn-on transient of Samsung Galaxy S8 as output by the downconverter using the HMC175 mixer, shown in the frequency domain.

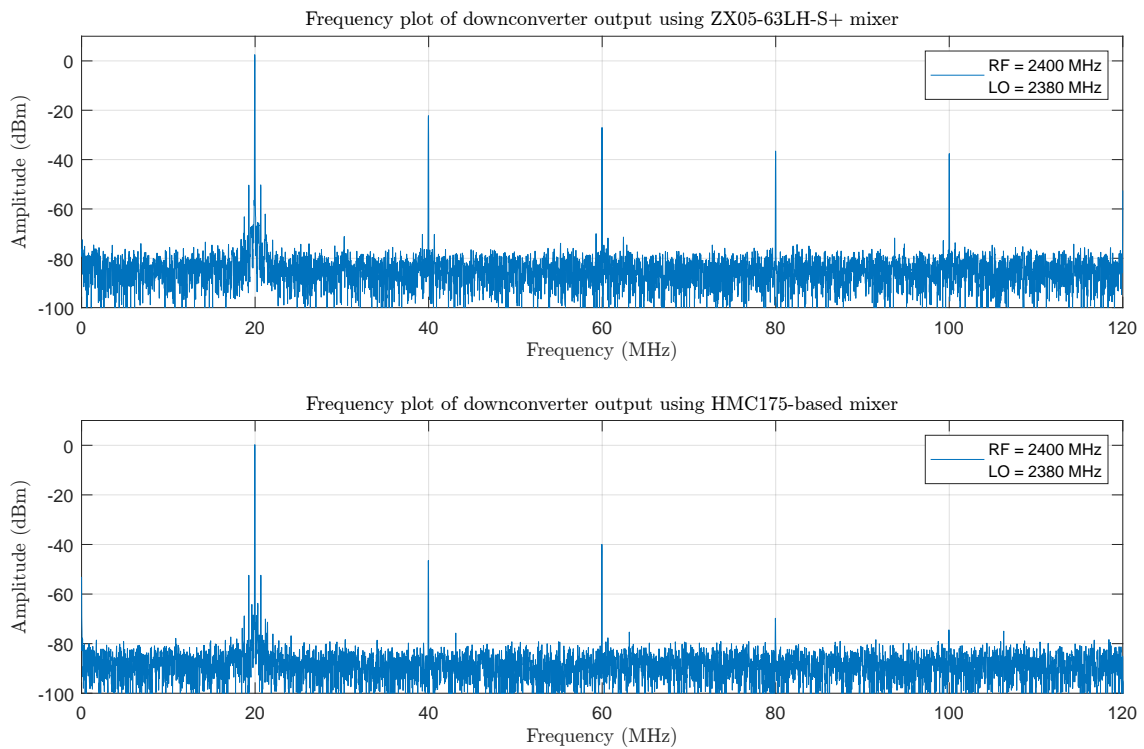


Figure 6.5: Comparison of frequency components in output of downconverter using ZX05-63LH-S+ and HMC175-based mixers for single-frequency unmodulated input, shown in the frequency domain.

6.2.3 Comparison of Mixers

While the HMC-175 mixer appeared superior under simulated testing with the VCO, obvious distortions in the output were observed when lower-power Bluetooth signals were used as an input; the waveform exhibited a distortion based on its power, the frequencies of interest were attenuated, and total harmonic distortion increased across the output frequency range increased.

Conversely, the ZX05-63LH-S+ mixer performed worse under simulated testing with the VCO, but ultimately performed better when lower-power Bluetooth signals were used; no distortion of the waveform was observed, frequencies of interest had a higher power, and total harmonic distortion was reduced.

6.3 Acquisition System

The acquisition system worked as expected, with no major concerns identified. The downconverter appeared to function as expected, shifting Bluetooth signals down to the 20–100 MHz, allowing sampling by the PicoScope 5444B.

It was noted that the trigger settings used in the PicoScope could be improved. There was a general difficulty in finding a single trigger setting that would reliably detect Bluetooth transients from all channels. As the device transmits on different channels, the frequency of the downconverted waveform varies, as expected; however this can pose challenges for triggers that include window time or dwell time. Care must be taken to ensure trigger settings do not inadvertently block transient being detected dependant on their channel (frequency).

Additionally, after capturing it was discovered that some Bluetooth devices have noisy steady-state waveforms, hampering effective identification of the transient end point in the transient detection stage. This would normally be improved by filtering, but the waveforms captured are limited by the amount of steady-state samples acquired. The acquired signals were 5 μ s in length, with the trigger point located halfway into the buffer. To overcome this, the sample length should be increased to at least 10 μ s (5000 samples per trigger), and the trigger-point should be moved to 30% of the buffer, increasing the steady-state waveform collected.

6.4 Transient Detection

Accurate transient detection remains a challenge for RF Fingerprinting techniques.

Experimentation showed the EC function could be applied to the envelope of the signal, as opposed to the waveform itself; this allowed the delaying parameter to be set much higher ($\vartheta = 120$) than the recommended 35 required when EC is applied directly to the waveform.

A one-dimension median filter (equivalent to 1200 ns, or 600 samples at 500 MS/s) was applied to each waveform envelope to remove noise, and enhance the ability of the software to locate the beginning of the steady-state, which was defined as the point at which the

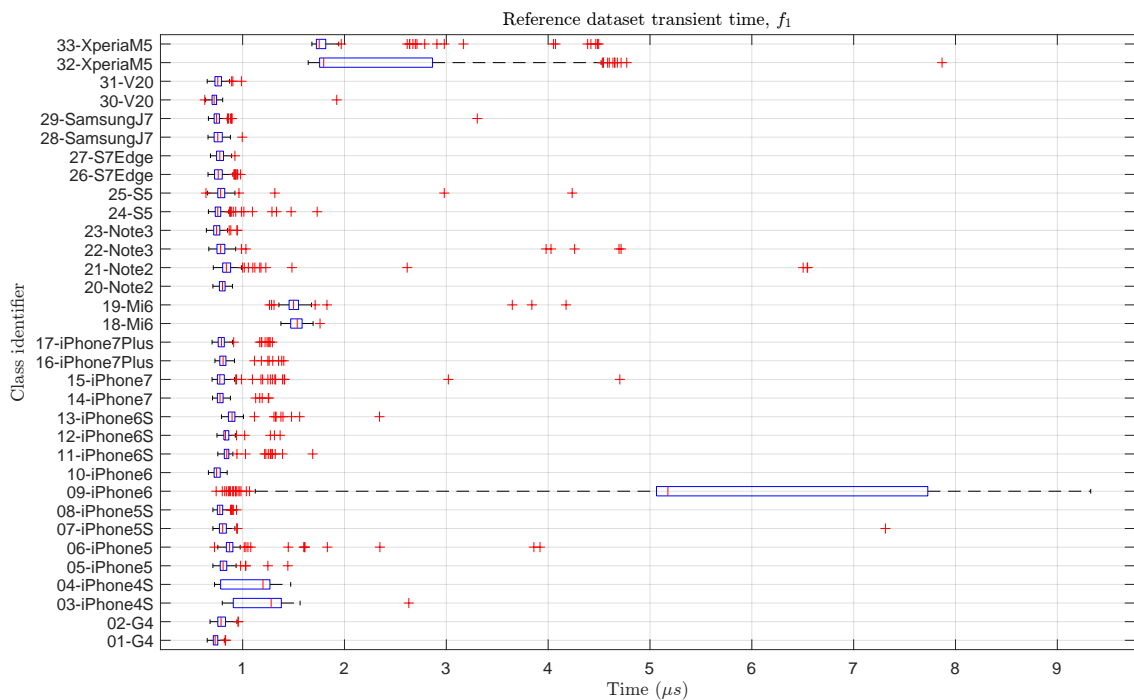


Figure 6.6: Boxplot of feature f_1 , transient length, for the reference dataset.

signal entered and stayed within 3.5% of the steady-state value.

If a settling point cannot be determined, the value is set to `NaN` and the record is excluded from the dataset. There are numerous reasons why a settling point cannot be found, but the majority encountered seem to result from unstable steady-state; the envelope of the signal appears to exhibit ripple which, in some devices, exceeds the tolerance used to identify steady-state.

When applied to **Dataset A** the transient detection algorithm was able to find a transient in 94.0% of the waveform records. Of those records where a transient could not be identified, the iPhone X was the most difficult to process, with 121 of 400 records (30.25%) failing. The transient length was calculated for all records over all classes, and the results were visualised in a boxplot. The boxplot can be seen in Figure 6.7. There are a number of outliers, but considering there are 200 records per class the number of outliers is comparatively low.

When applied to the **Reference Dataset** the transient detection algorithm was able to find a transient in 98.24% of the waveform records. In this dataset, the most difficult device to reliably detect transient length was an Apple iPhone 6. A transient was detected in every case, but there was significant variance in the length of the transient across the

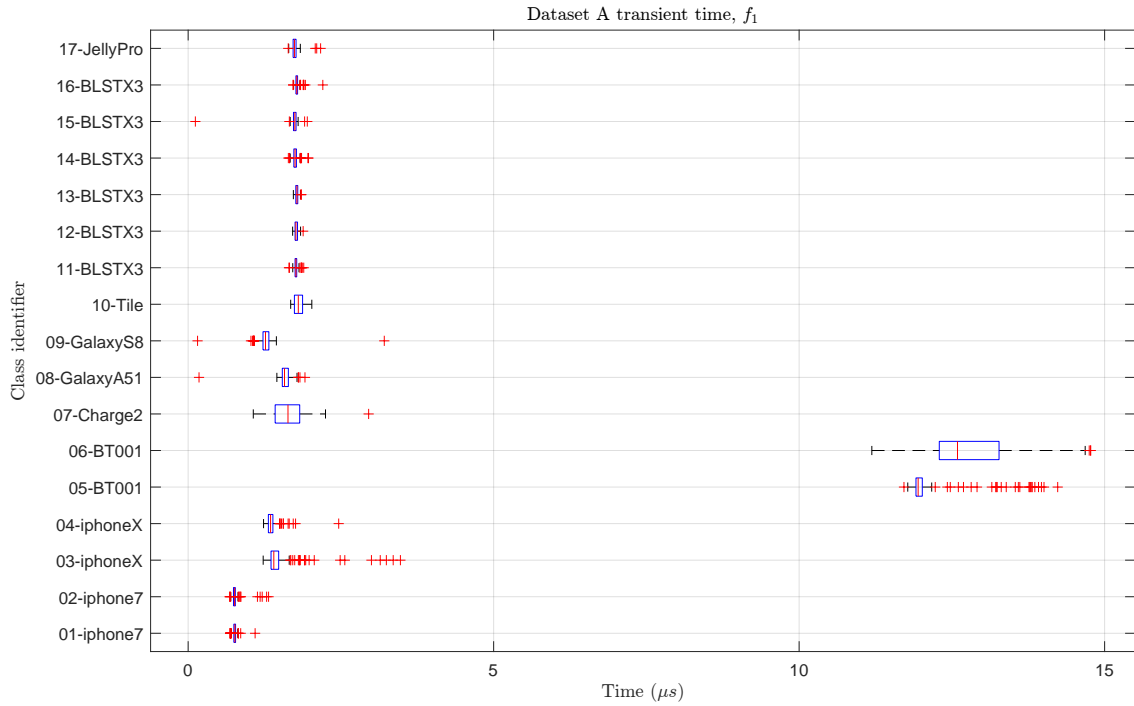


Figure 6.7: Boxplot of feature f_1 , transient length, for Dataset A.

records. The transient length was calculated for all records over all classes, and the results were visualised in a boxplot. The boxplot can be seen in Figure 6.6. As with the results for **Dataset A**, there are still a number of outliers, but considering there are 150 records per class the number of outliers is comparatively low.

The Apple iPhone 7 is the only device type present in both **Dataset A** and the **Reference Dataset**. By inspection of the boxplots in Figure 6.6 and Figure 6.7, it can be seen that the transient length extractor works reliably on this device type, and has assigned all iPhone 7 devices with very similar transient lengths. The median transient length for Phone 7 records in the **Reference Dataset** are 776 ns and 778 ns (less than one sample resolution), whereas those in **Dataset A** are 764 ns and 760 ns (within two samples). This finding shows the transient length extractor produces constant results, despite differences in sampling rate.

However, many devices in both datasets have large variances in transient length. As this occurs across both datasets, it indicates this is not an artefact or error introduced by the sampling system; rather, it indicates there is a problem in the transient detection algorithm (or its parameters). Given the reliance of all features on this critical algorithm, refinement in this space may increase performance of the classification stage.

6.5 Feature Extraction

By analysing the distribution of some features, it can be seen that some appear to have similar distributions, indicating they may be duplicated dimensions; for example, the distributions for features f_3 and f_{12} , shown in Figure 6.8 and Figure 6.9 respectively, appear substantially similar. However, there is noticeable difference between feature f_4 and f_{12} , shown in Figure 6.10 and Figure 6.9 respectively, indicating they are probably good candidates as dimensions use in the classification stage. Figure 6.11 shows a scatter plot of f_4 and f_{12} for the **Reference Dataset**. The clustering of points by class indicates these two features are stronger features for classification, however the overlapping of devices in the centre-left of the plot (attributed to other devices of the same type) indicate the current implementation of this features extraction is insufficient for device-specific classification.

Classes f_5 and f_6 are derived from the instantaneous phase characteristics of the transient, and are reported to be two of the most robust features for classification (Ali et al. 2019). While these may be effective when calculated using waveforms captured at high sampling rates, they appear less meaningful when calculated using waveforms captured at or close to the Nyquist rate. The **Reference Dataset** is captured at 250 MS s^{-1} , which was close to the Nyquist rate of 200 MS s^{-1} . By inspection of the distributions for f_5 and f_6 (see Figure 6.12 and Figure 6.13 respectively), it is clearly seen that all classes have wide, overlapping distributions. In isolation, these features do not appear to be good candidates as features for classification. At higher sampling rates these feature may be useable, but they appear unhelpful at 250 MS s^{-1} .

Inspection of the boxplots indicate the feature extraction system is highly dependant on, and influenced by, the results of the transient extraction system. From Figure 6.6, it is clear that 'Class 09 iPhone 6' has an abnormally large variance of feature f_1 transient length. By reviewing the other boxplots from the same dataset, it can be seen that 'Class 09 iPhone 6' displays abnormality in many other features (see feature f_3 in Figure 6.8, feature f_{12} in Figure 6.9, and feature f_4 in Figure 6.10).

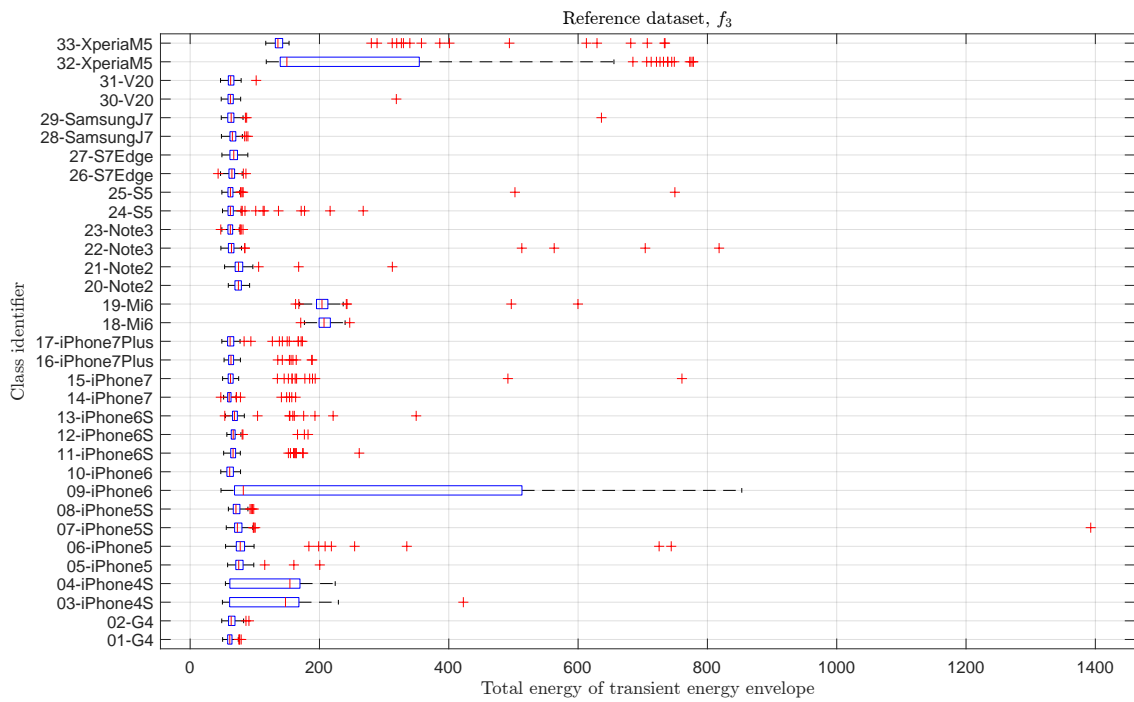


Figure 6.8: Boxplot of feature f_3 , total energy of transient energy envelope, for the reference dataset.

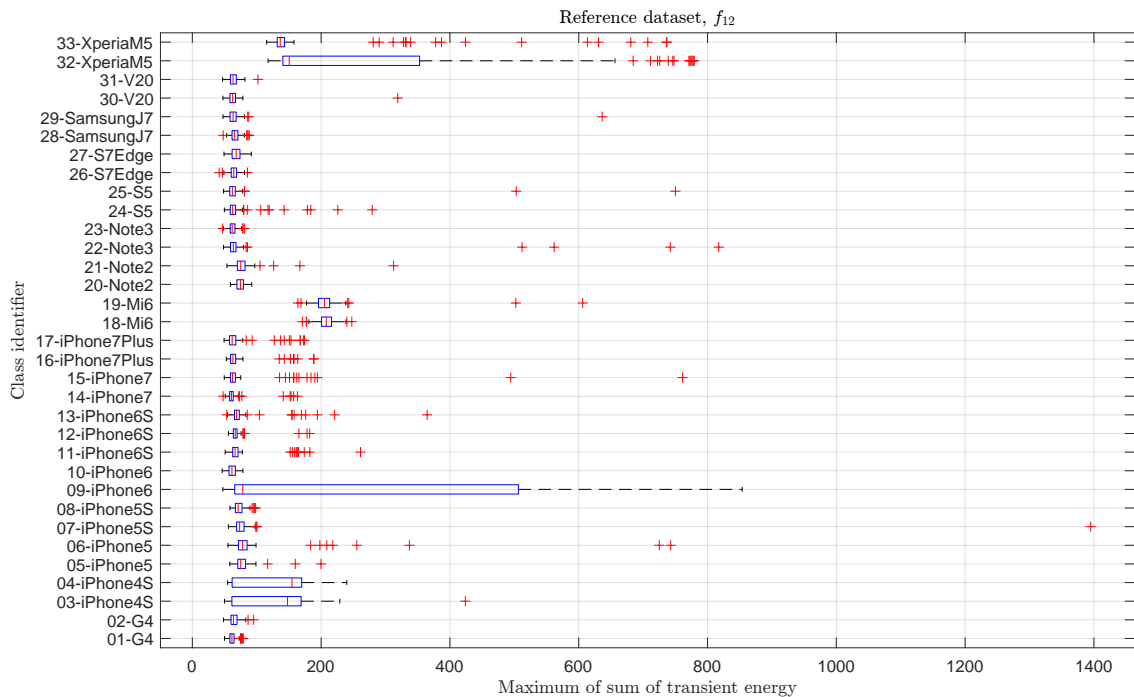


Figure 6.9: Boxplot of feature f_{12} , maximum of sum of transient energy distribution along frequency axis, for the reference dataset.

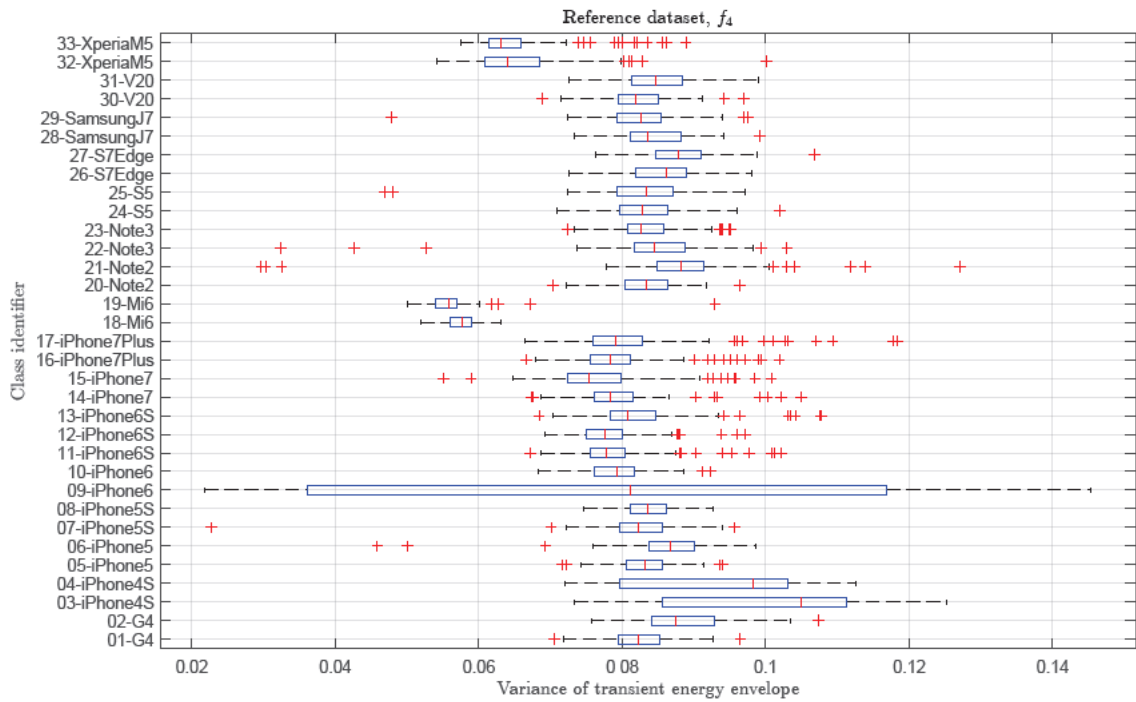


Figure 6.10: Boxplot of feature f_4 , variance of transient energy envelope, for the reference dataset.

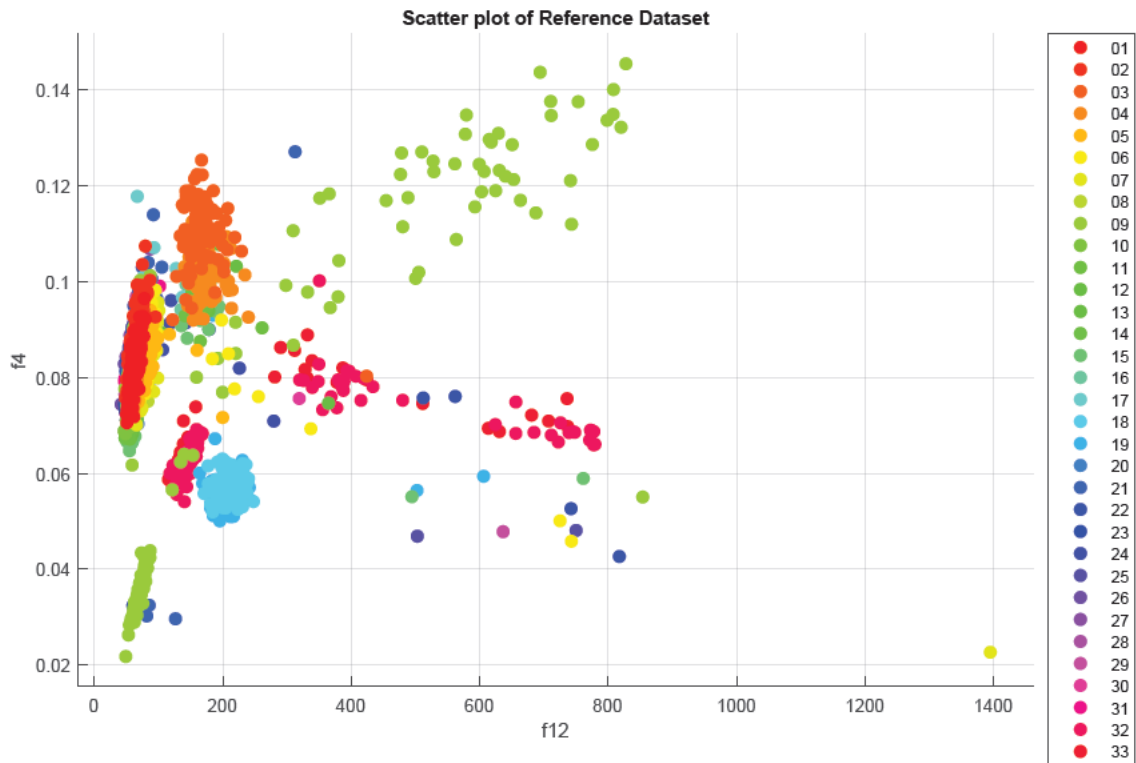


Figure 6.11: Scatter plot of features f_4 and f_{12} , for the reference dataset.

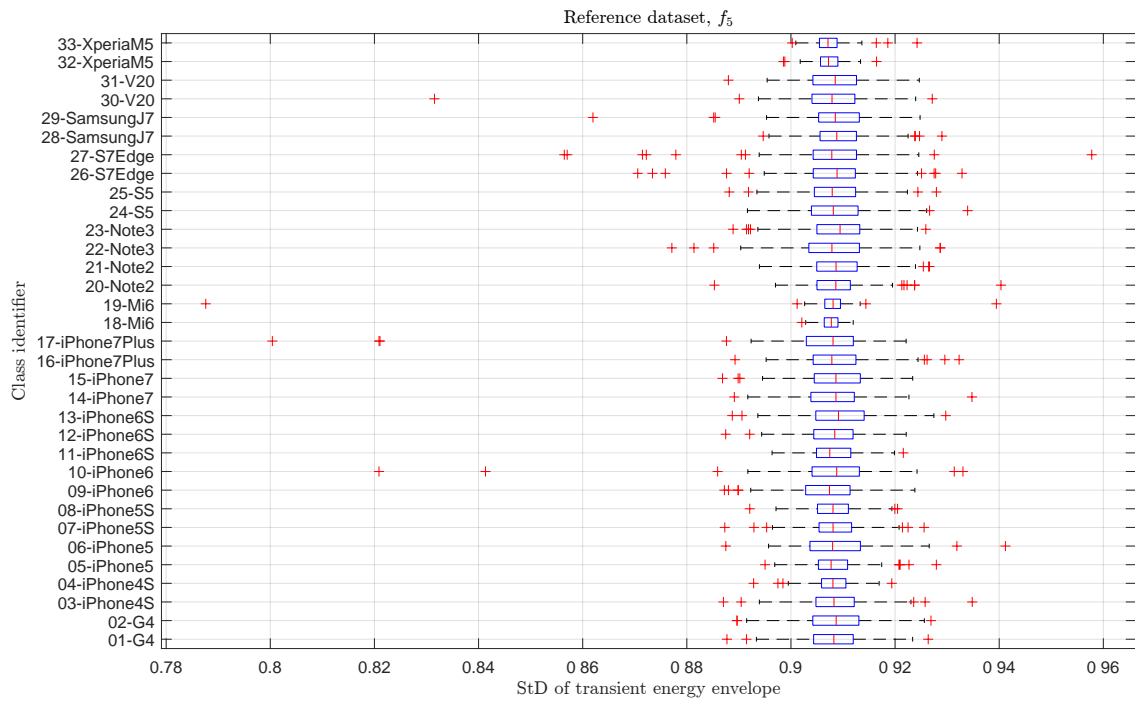


Figure 6.12: Boxplot of feature f_5 , StD of transient energy envelope, for the reference dataset.

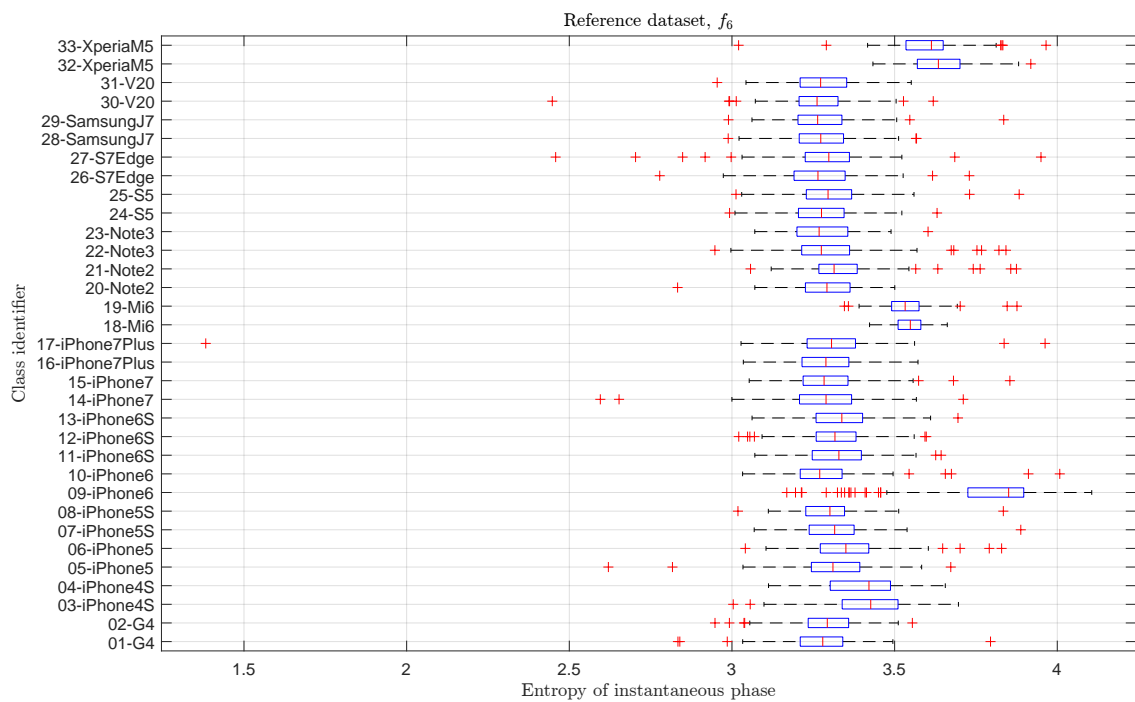


Figure 6.13: Boxplot of feature f_6 , entropy of instantaneous phase, for the reference dataset.

6.6 Classifier Performance

By using either the *DeviceID* or *DeviceModel* as predictive values, it is possible to build classifiers to attribute turn-on transients to a particular transmitter (e.g. this particular Apple iPhone 7), and to attribute turn-on transients to a particular *type* or transmitter (e.g. any Apple iPhone 7).

6.6.1 Reference Dataset, Device Attribution Classification Results

When applied to the **Reference Dataset**, the classifier performed poorly. It was able to successfully classify a turn-on transient to a particular device only 32.6% of the time. Figure 6.14 shows the confusion matrix. The classifier did perform better than random chance—for a multi-class classification problem of n classes, the probability of guessing correctly is $P = \frac{1}{n}$. In this case, there are $n = 33$ classes, so it is expected that only 3.03% of random guesses would be correct, on average.

6.6.2 Dataset A, Device Attribution Classification Results

When applied to **Dataset A**, the classifier performed better than for the **Reference Dataset**, but still lower than others have reported. It was able to successfully classify a turn-on transient to a particular device 69.6% of the time. Figure 6.15 shows the confusion matrix.

It is unclear why these results are markedly different that those for the **Reference Dataset**, though it could be due to the increased sampling rate. Some of the features used during extraction rely on instantaneous phase information, which becomes less meaningful as the sampling rate approaches the Nyquist rate. Another possible reason is the difference in classes; the **Reference Dataset** includes 33 classes, whereas **Dataset A** only includes 17.

Dataset A confusion matrix (SVM)

01	40.8%	28.3%					0.5%										
02	18.0%	70.7%		0.5%													
03	20.0%		67.6%	16.0%	1.0%		9.5%	1.0%									0.5%
04	5.4%	0.5%	20.7%	78.2%			3.0%		2.2%								
05					97.0%	1.8%											
06	8.7%				1.5%	98.2%											
07	2.0%		9.9%	1.1%			84.9%	2.9%	2.2%								0.5%
08		0.5%		0.5%			1.0%	94.6%									1.1%
09	4.8%		1.8%	3.7%	0.5%		0.5%		95.6%								
10										100.0%							
11										49.3%	8.4%	6.1%	5.6%	9.8%	13.4%	1.6%	
12										9.1%	41.4%	17.9%	12.3%	11.4%	5.6%	1.1%	
13										5.5%	17.7%	54.2%	8.9%	8.8%	5.6%	3.8%	
14								0.5%		7.3%	14.9%	7.8%	48.6%	16.6%	6.9%	1.1%	
15								0.5%		8.2%	13.5%	6.1%	19.0%	46.6%	4.7%	3.3%	
16								0.5%		15.1%	2.8%	5.0%	3.4%	2.1%	59.5%	1.6%	
17	0.3%						0.5%			5.5%	1.4%	2.8%	2.2%	4.1%	4.3%	85.7%	
PPV	40.8%	70.7%	67.6%	78.2%	97.0%	98.2%	84.9%	94.6%	95.6%	100.0%	49.3%	41.4%	54.2%	48.6%	46.6%	59.5%	85.7%
FDR	59.2%	29.3%	32.4%	21.8%	3.0%	1.8%	15.1%	5.4%	4.4%		50.7%	58.6%	45.8%	51.4%	53.4%	40.5%	14.3%
	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17

Predicted Class

Figure 6.15: Confusion matrix of Dataset A without device-type grouping, using SVM classifier after optimisation (69.6% classification success).

6.6.3 Classifying Based on Device-type

Inspection of the confusion matrices in Section 6.6.1 and Section 6.6.2 show the classifier struggles to correctly predict a class when there is more than one device of the same

Dataset A confusion matrix (SVM, optimised, grouped)

True Class	BLSTX3	88.9%				0.5%	10.4%			
	BT001	1.8%	100.0%							
	Charge2	0.5%		88.9%	2.5%	0.5%			5.6%	
	GalaxyA51			2.6%	94.0%	0.5%			1.8%	
	GalaxyS8	0.2%		0.5%		93.5%			2.6%	
	JellyPro	4.0%		1.1%			88.4%			
	Tile						0.6%	100.0%		
	iPhone7					0.5%			99.5%	0.3%
	iPhoneX	4.6%		6.9%	3.5%	4.5%	0.6%		0.5%	89.8%
	PPV	88.9%	100.0%	88.9%	94.0%	93.5%	88.4%	100.0%	99.5%	89.8%
	FDR	11.1%		11.1%	6.0%	6.5%	11.6%		0.5%	10.2%
		BLSTX3	BT001	Charge2	GalaxyA51	GalaxyS8	JellyPro	Tile	iPhone7	iPhoneX
		Predicted Class								

Figure 6.16: Confusion matrix of Dataset A with device-type grouping, using SVM classifier (92.6% classification success).

type. However, where there is only one device of a given type (as is the case for the Tile Mate, class 10 in Figure 6.15) the classifier predicts that class quite well. To investigate this further, a new classifier was built to attribute turn-on transients to a given device type; that is, it is sufficient to classify a transient as being generated from an iPhone 7, as opposed to a specific iPhone 7. To achieve this the same datasets were used, but the predictive value was changed to *DeviceModel*. In this scenario, the classification problem becomes simplified.

The SVM classifier with a quadratic kernel function (QSVM) performed the best prior to optimisation, with 92.6% of turn-on transients correctly attributed to the device type. This classifier is significant more accurate compared to the per-device attribution classifiers, the results of which are shown in Section 6.6.2. Figure 6.16 shows the confusion matrix for the classifier with classification based on device-type.

6.7 Discussion

This project has recreated a functional Bluetooth RF fingerprinting system capable of attributing turn-on transients to a given device, or device-type. However, it has not been

able to replicate the success of others, nor was it possible to fully explore how the physical environment (temperature, movement, background RF noise) affects classification. Despite not acquiring turn-on transients under these conditions, the calculations in Section 3.8.2 provide a compelling case that minor shift in frequency due to movement of a device during transmission is unlikely to be discernible, at least for speeds a human is likely to be travelling.

The constructed downconverter and acquisition system performed as expected. Investigation showed the Mini Circuits' ZX05-63LH-S+ mixer was better at reproducing low-power RF signals compared to a mixer based on the HMC156 IC. The triggering settings in the PicoScope could be improved, to improve detection while lowering false triggers; alternatively, an external triggering device could be created to trigger the PicoScope 5444B.

Accurate and consistent detection of the transient start and end point remains a challenge. Some of the waveforms captured in **Dataset A** did not include enough steady-state after the transient, which hampered identification of the end of the transient. During this project, such short waveforms were simply discarded, but future researcher should endeavour to capture enough steady-state to allow detection of the transient end, even after filtering of the envelope. The author believes a sample length of 10 μs , with the transient start trigger-point at approximately 30% of the buffer, should be sufficient. Given the reliance of all features on accurate transient detection, refinement in this space may increase performance of the classification stage.

In all cases, the classifiers built to attribute devices based on their Bluetooth turn-on transients differed markedly from the results reported by others, who claim success rates of greater than 99% with the **Reference Dataset** (Uzundurukan, Dalveren & Kara 2020*b*). It is noted that these published results do not include sufficient information to allow transient detection and feature extraction to be implemented in an identical manner, so the observed discrepancy in results could be attributed to difference in implementation.

Additionally, in their implementation of a similar system, Ali et al. (2019) briefly explain a mechanism that uses multiple samples to determine the start and end transient more accurately, and then extract the features. This approach is claimed to result in a classifier success rate of over 99%; however, there was insufficient information to determine how the authors were able to attribute multiple unknown transients to a single transmitter prior to successful classification. The implementation of such a system would reduce the

variance in transient length, which has been shown to greatly influence the extraction of features for classification.

6.8 Chapter Summary

This chapter has reviewed the performance of the acquisition system, the transient extraction system, and the feature extraction system; final performance of the end-to-end system has been reviewed through the results of the classifier. Results are presented for **Reference Dataset** and the acquired **Dataset A**, and include per-device-attribution and device-type attribution tests.

Chapter 7

Conclusions and Further Work

This chapter provides a summary of the results achieved, mapped against the original project specifications. A number of potential future research areas that could build on this work are explored.

7.1 Recommendations for Further Research

This section explores a number of potential further research areas that could build on this work.

7.1.1 Collection of Data

The original intention was to collect a number of datasets under varied environmental conditions, which could be used by future researchers to better understand the efficacy of RF fingerprinting systems outside of ideal laboratory conditions. As has been noted earlier, existing research on the topic focuses exclusively on ideal laboratory conditions; there are no publicly known datasets of Bluetooth turn-on transients that have been acquired under such conditions.

Such datasets are advantageous to researchers, as they allow evaluate and verification of RF fingerprinting systems in real-world environments, without having to reproduce the testing. Due to lockdowns relating to the COVID-19 pandemic it was not possible

to access the necessary facilities and equipment, and consequently those datasets were not acquired. However, the collection of such datasets would be advantageous to future researchers.

7.1.2 Improve Transient Detection Algorithm

Accurate and consistent detection of the transient start and end point remains a challenge, especially for waveforms that exhibit steady-state amplitude variance or those that grow to steady-state very slowly. Both of these features have caused problems for reliably identifying the start of the steady-state of the waveform (end of the transient).

7.1.3 Initial Correlation of Turn-on Transients

Some researchers have overcome some of the challenges in transient detection by averaging the detected transient length from multiple turn-on transients from a single device (Ali et al. 2019). However, the literature does not sufficiently explain how the researchers were able to attribute multiple unknown transients to a single device in order to perform the averaging. Further research in this area could allow the transient extraction stage to be made much more robust.

One potential method for initial correlation of turn-on transients is to estimate the physical distance between the transmitter and receiver, on the assumption that this will not vary substantially over a short period of time. This issue has application in other domains, and so is explored within the literature (Castillo-Cara, Lovón-Melgarejo, Bravo-Rocca, Orozco-Barbosa & García-Varea 2017, Giuliano, Cardarilli, Cesarini, Di Nunzio, Fallucchi, Fazzolari, Mazzenga, Re & Vizzarri 2020).

A second potential method is to perform temporal correlation of transient signals based on a shared Bluetooth Low Energy (BLE) advertising address. The BLE advertising address includes features to prevent tracking and surveillance—the address can be generated randomly and changed frequently. However, for practical purposes the address remains static for a period of time. If advertising addresses can be decoded and attributed to a captured turn-on transient, it may be possible to attribute transients to a single but unknown device, allowing averaging of the transient length to be completed.

An extension of this second method is to correlate turn-on transients to a single device despite an advertising address change. Assuming the area around the receiver is relatively quiet (in terms of Bluetooth traffic) and the number of devices is not in constant flux, it may be possible to track a Bluetooth device after address change by correlating the presence of a new address with the disappearance of another. This method has been built on in the *address-carryover algorithm* developed by Becker et al. (2019), though additional tokens were extracted from advertising data to make the process more robust.

7.1.4 Improve Downconverter

The modular RF downconverter has been shown to be an effective at shifting the entire Bluetooth band (2400–2480 MHz) down to 20–100 MHz, allowing the entire Bluetooth band to be sampled using low-cost equipment (compared to the cost of equipment required for direct sampling of the RF). The modular design was chosen by researchers because the components are accessible, reusable, and relatively low-cost. However, the system could be refined through the design of a fit-for-purpose downconverter with the following improvements:

- reduce size and mass to simplify deployment;
- ability to be powered from less than 15 V;
- reduce current consumption;
- include circuitry to prevent introduction of distortion due to power supply ripple or instability;
- reduce intermodulation distortion and harmonics at output; and
- improve stability of local oscillator to temperature variance.

7.2 Conclusions

This project has described the design and implementation of an RF fingerprinting system based on Bluetooth turn-on transients. The project specification, shown in Appendix A,

includes a number of primary goals. A review of the progress made to these goals is included below.

A downconverter, which allows the Bluetooth band (2400–2480 MHz) to be shifted to 20–100 MHz, has been described and constructed. The downconverter facilitated sampling of the entire Bluetooth band using a PicoScope 5444B; sampling was completed at 500 MS/s at 12-bits of resolution, exceeding the minimum sampling rate and resolution required.

A system for automatically extracting the turn-on transient from a sampled waveform was implemented in MATLAB[®]. Energy Criterion was confirmed as an excellent method for detecting the start of a transient portion. Additionally, a new method for detecting the end of the transient, based on the settling time of the envelope. These two methods successfully extracted the transients from a number of waveforms reliably; however, some device types generate waveforms that cause unreliable operation of the transient detection system.

To support classification of the Bluetooth transients, a feature extraction system was implemented in MATLAB[®], based on the thirteen features described by Ali et al. (2019). Inspection of those features as boxplots shows a link between inconsistency in the transient detection stage (realised as a wide spread of values) and inconsistency in the features. Classifiers were implemented using MATLAB[®]'s Classification Learner app. In each case, the optimum classifier was a Support Vector Machine, confirming the results of others (Ali et al. 2019, Uzundurukan, Ali, Dalveren & Kara 2020, Helluy-Lafont et al. 2020).

Due to lockdowns relating to the COVID-19 pandemic it was not possible to access the necessary facilities and equipment required to collect datasets under controlled conditions, and thus this original goal has not been achieved. As these datasets have not been acquired, it is not possible to fully explore how the physical environment (temperature, movement, background RF noise) affects classification. Despite not acquiring turn-on transients under these conditions, the calculations in Section 5 provide a compelling case that minor shift in frequency due to movement of a device during transmission is unlikely to be discernible, at least for speeds a human is likely to be travelling.

The performance of the classifier was assessed using a **Reference Dataset** provided by Uzundurukan, Dalveren & Kara (2020*a*), and **Dataset A** acquired using the downconverter and acquisition system described earlier. After optimisation, the classifier was able

to correctly attribute waveforms from the **Reference Dataset** to a specific device with an accuracy of 32.6%. When attributing waveforms from the acquired dataset, **Dataset A**, device-specific attribution accuracy was 69.9%; the reason for the performance improvement is not yet understood, but could be due to the increased sampling rate, as some features rely on instantaneous phase information that becomes less meaningful when sampled close to the Nyquist rate. Compared to the results reported in the literature, this research was unable to reproduce the extremely high results (over 99% success) reported by others (Ali et al. 2019, Aghnaiya et al. 2019, Uzundurukan, Ali, Dalveren & Kara 2020). However, when the classifier was used to attribute waveforms to a device-type, as opposed to specific device, prediction success increased to 92.6%.

This project has implemented a functional and successful RF fingerprinting system for classifying transmitters based on their Bluetooth turn-on transient, however the results of others could not be reproduced. Nevertheless, the results show that RF fingerprinting based on turn-on transient shows promise as a method for transmitter attribution; additionally, attribution is greatly simplified if classifying to a device model, instead of a specific device.

References

- Aghnaiya, A., Ali, A. M. & Kara, A. (2019), ‘Variational mode decomposition-based radio frequency fingerprinting of bluetooth devices’, *IEEE Access* **7**, 144054–144058. <<https://ieeexplore.ieee.org/document/8854788>>.
- Ali, A. M., Uzundurukan, E. & Kara, A. (2017), Improvements on transient signal detection for RF fingerprinting, *in* ‘2017 IEEE Signal Processing and Communications Applications (SIU)’, IEEE, pp. 1–4. <<https://ieeexplore.ieee.org/document/7960417>>.
- Ali, A. M., Uzundurukan, E. & Kara, A. (2019), ‘Assessment of features and classifiers for Bluetooth RF fingerprinting’, *IEEE Access* **7**, 50524–50535. <<https://ieeexplore.ieee.org/document/8691737/>>.
- Australian Cyber Security Centre (2020), *Australian Government Information Security Manual*, Australian Government, chapter Guidelines for physical security. <<https://www.cyber.gov.au/acsc/view-all-content/ism>>.
- Becker, J. K., Li, D. & Starobinski, D. (2019), Tracking anonymized Bluetooth devices, *in* ‘Proceedings on Privacy Enhancing Technologies’, Vol. 2019, pp. 50–65. <https://www.researchgate.net/publication/334590931_Tracking_Anonymized_Bluetooth_Devices>.
- Bluetooth Special Interest Group (2019), ‘Bluetooth core specification version 5.2’. <<https://www.bluetooth.com/bluetooth-resources/bluetooth-core-specification-version-5-2-feature-overview/>>.
- Bowen, J., Sohinki, S., Potter, E. & Vaughn, J. (2017), ‘Intrusion detection systems and subsystems: Technical information for NRC licensees’. <<https://www.nrc.gov/reading-rm/doc-collections/nuregs/staff/sr1959>>.

- Brik, V., Banerjee, S., Gruteser, M. & Oh, S. (2008), Wireless device identification with radiometric signatures, *in* ‘Proceedings of the 14th ACM international conference on Mobile computing and networking’, pp. 116–127. <<https://dl.acm.org/doi/10.1145/1409944.1409959>>.
- Candore, A., Kocabas, O. & Koushanfar, F. (2009), Robust stable radiometric fingerprinting for wireless devices, *in* ‘2009 IEEE International Workshop on Hardware-Oriented Security and Trust’, pp. 43–49. <<https://ieeexplore.ieee.org/document/5224969>>.
- Castillo-Cara, M., Lovón-Melgarejo, J., Bravo-Rocca, G., Orozco-Barbosa, L. & García-Varea, I. (2017), ‘An empirical study of the transmission power setting for bluetooth-based indoor localization mechanisms’, *Sensors* **17**(6), 1318. <<https://www.mdpi.com/1424-8220/17/6/1318/pdf>>.
- Celosia, G. & Cunche, M. (2020), Discontinued privacy: Personal data leaks in Apple Bluetooth-Low-Energy continuity protocols, *in* ‘Proceedings on Privacy Enhancing Technologies’, Vol. 2020, pp. 26–46. <<https://content.sciendo.com/downloadpdf/journals/popets/2020/1/article-p26.xml>>.
- Daney, B., Zanetti, D. & Capkun, S. (2012), ‘On physical-layer identification of wireless devices’, *ACM Computing Surveys (CSUR)* **45**(1), 1–29. <<https://www.research-collection.ethz.ch/bitstream/handle/20.500.11850/72822/eth-2868-02.pdf>>.
- Desmond, L. C. C., Yuan, C. C., Pheng, T. C. & Lee, R. S. (2008), Identifying unique devices through wireless fingerprinting, *in* ‘Proceedings of the first ACM conference on Wireless network security’, ACM Press, pp. 46–55. <<https://dl.acm.org/doi/10.1145/1352533.1352542>>.
- Ellis, K. & Serinken, N. (2001), ‘Characteristics of radio transmitter fingerprints’, *Radio Science* **36**(4), 585–597. <<https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2000RS002345>>.
- Elmrabet, Z., Arjoune, Y., el Ghazi, H., Majd, B. A. E. & Kaabouch, N. (2018), ‘Primary user emulation attacks: A detection technique based on Kalman filter’, *Journal of Sensor and Actuator Networks* **7**, 26. <<https://arxiv.org/ftp/arxiv/papers/1903/1903.03684.pdf>>.
- Frederick, M. B. (1995), ‘Cellular telephone anti-fraud system’, US Patent

- US5448760. <<https://patentscope.wipo.int/search/en/detail.jsf?docId=US38511221>>.
- Giuliano, R., Cardarilli, G. C., Cesarini, C., Di Nunzio, L., Fallucchi, F., Fazzolari, R., Mazzenga, F., Re, M. & Vizzarri, A. (2020), ‘Indoor localization system based on bluetooth low energy for museum applications’, *Electronics* **9**(6), 1055. <<https://www.mdpi.com/2079-9292/9/6/1055/pdf>>.
- Hall, J. (2006), Detection of rogue devices in wireless networks, PhD thesis, Carleton University. <https://curve.carleton.ca/system/files/etd/ff61fd85-2da8-492b-ba9d-24cc66946040/etd_pdf/252431ae07b3a61dddcee7d2df686f2e/hall-detectionofroguedevicesinwirelessnetworks.pdf>.
- Hall, J., Barbeau, M. & Kranakis, E. (2003), ‘Detection of transient in radio frequency fingerprinting using signal phase’, *Wireless and Optical Communications* pp. 13–18. <<http://people.scs.carleton.ca/~kranakis/Papers/RFFPaper3.pdf>>.
- Hall, J., Barbeau, M. & Kranakis, E. (2006), Detecting rogue devices in Bluetooth networks using radio frequency fingerprinting, in ‘Proceedings of the Third IASTED International Conference on Communications and Computer Networks, CCN 2006’, pp. 108–113. <https://www.researchgate.net/publication/220964646_Detecting_rogue_devices_in_bluetooth_networks_using_radio_frequency_fingerprinting>.
- Helluy-Lafont, E., Boe, A., Grimaud, G. & Hauspie, M. (2020), Bluetooth devices fingerprinting using low cost SDR, in ‘2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC)’, IEEE, pp. 289–294. <<https://ieeexplore.ieee.org/document/9144756>>.
- Hippenstiel, R. D. & Payal, Y. (1996), Wavelet based approach to transmitter identification, in ‘Fourth International Symposium on Signal Processing and Its Applications’, Vol. 2, pp. 740–742. <<https://ieeexplore-ieee-org.ezproxy.usq.edu.au/document/615150>>.
- Hsu, C.-W. & Lin, C.-J. (2002), ‘A comparison of methods for multiclass support vector machines’, *IEEE transactions on Neural Networks* **13**(2), 415–425. <<https://www.csie.ntu.edu.tw/~cjlin/papers/multisvm.pdf>>.

- Klein, R., Temple, M. & Mendenhall, M. (2009), 'Application of wavelet-based RF fingerprinting to enhance wireless network security', *Journal of Communications and Networks* **11**(6), 544–555. <<https://ieeexplore.ieee.org/abstract/document/6388408>>.
- Leis, J. W. (2011), *Digital signal processing using MATLAB for students and researchers*, John Wiley and Sons, Inc., Hoboken, NJ, USA, chapter Sampled signals and digital processing, pp. 45–102.
- Lukacs, M., Collins, P. & Temple, M. (2015), 'Classification performance using 'rf-dna' fingerprinting of ultra-wideband noise waveforms', *Electronics Letters* **51**(10), 787–789. <<https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/el.2015.0051>>.
- Martin, J., Mayberry, T., Donahue, C., Foppe, L., Brown, L., Riggins, C., Rye, E. C. & Brown, D. (2017), 'A study of MAC address randomization in mobile devices and when it fails', *Proceedings on Privacy Enhancing Technologies* **2017**(4), 365–383. <https://www.researchgate.net/publication/314361145_A_Study_of_MAC_Address_Randomization_in_Mobile_Devices_and_When_it_Fails>.
- Mohamed, I. S., Dalveren, Y. & Kara, A. (2020), 'Performance assessment of transient signal detection methods and superiority of Energy Criterion (EC) method', *IEEE Access* **8**, 115613–115620. <<https://ieeexplore.ieee.org/document/9123350>>.
- Nguyen, N. T., Zheng, G., Han, Z. & Zheng, R. (2011), Device fingerprinting to enhance wireless security using nonparametric bayesian method, in '2011 Proceedings IEEE INFOCOM', pp. 1404–1412. <<https://ieeexplore.ieee.org/document/5934926>>.
- Oliveira, L., Schneider, D., Souza, J. D. & Shen, W. (2019), 'Mobile device detection through WiFi probe request analysis', *IEEE Access* **7**, 98579–98588. <<https://ieeexplore.ieee.org/document/8747391>>.
- Pico Technology (2016), 'Picoscope 5000 series datasheet'. <<https://www.picotech.com/download/datasheets/MM040.en-8.pdf>>.
- Polak, A. C., Dolatshahi, S. & Goeckel, D. (2011), 'Identifying wireless users via transmitter imperfections', *IEEE Journal on selected areas in communications* **29**(7), 1469–1479. <<https://ieeexplore.ieee.org/document/5963165>>.
- Robyns, P., Bonn e, B., Quax, P. & Lamotte, W. (2017), 'Noncooperative 802.11 MAC layer fingerprinting and tracking of mobile devices', *Security and Communication Networks* . <<https://www.hindawi.com/journals/scn/2017/6235484/>>.

- Russell, C., Hogan, L. & Junker-Kenny, M. (2012), *Ethics for Graduate Researchers: A Cross-disciplinary Approach*, Newnes.
- Ureten, O. & Serinken, N. (2007), ‘Wireless security through RF fingerprinting’, *Canadian Journal of Electrical and Computer Engineering* **32**(1), 27–33. <<https://ieeexplore.ieee.org/document/4211360>>.
- Ureten, O., Serinken, N. et al. (1999), Bayesian detection of radio transmitter turn-on transients., in ‘NSIP’, pp. 830–834. <<https://www.eurasip.org/Proceedings/Ext/NSIP99/Nsip99/papers/178.pdf>>.
- Uzundurukan, E., Ali, A. M., Dalveren, Y. & Kara, A. (2020), ‘Performance analysis of modular RF front end for RF fingerprinting of Bluetooth devices’, *Wireless Personal Communications* **112**, 2519–2531. <<https://link.springer.com/article/10.1007/s11277-020-07162-z>>.
- Uzundurukan, E., Ali, A. M. & Kara, A. (2017), Design of low-cost modular RF front end for RF fingerprinting of Bluetooth signals, in ‘2017 IEEE Signal Processing and Communications Applications (SIU)’, IEEE. <<https://ieeexplore.ieee.org/document/7960367>>.
- Uzundurukan, E., Dalveren, Y. & Kara, A. (2020a), ‘Bluetooth dataset’. <<https://zenodo.org/record/3876140>>.
- Uzundurukan, E., Dalveren, Y. & Kara, A. (2020b), ‘A database for the radio frequency fingerprinting of Bluetooth devices’, *Data* **5**(55), 55. <<https://www.mdpi.com/2306-5729/5/2/55>>.
- Wagenaars, P., Wouters, P., Van der Wielen, P. & Steennis, E. (2008), Algorithms for arrival time estimation of partial discharge pulses in cable systems, in ‘Conference Record of the 2008 IEEE International Symposium on Electrical Insulation’, pp. 694–697. <https://www.researchgate.net/publication/224320902_Algorithms_for_Arrival_Time_Estimation_of_Partial_Discharge_Pulses_in_Cable_Systems>.
- Wenhao, W., Zhi, S., Kui, R., Bocheng, Z. & Sixu, P. (2015), ‘Wireless physical-layer identification: Modeling and validation’. <<https://ieeexplore.ieee.org/document/7448935>>.
- Xu, Q., Zheng, R., Saad, W. & Han, Z. (2015), ‘Device fingerprinting in wireless net-

works: Challenges and opportunities', *IEEE Communications Surveys and Tutorials*
18(1), 94–104. <<https://ieeexplore.ieee.org/document/7239531>>.

Appendix A

Project specification

ENG 4111/2 Research Project

Project Specification

For: **Eli Priest**
Topic: Unique Identification of Bluetooth Transmitters Through RF Fingerprinting
Supervisors: J. Leis
Sponsorship: Faculty of Health, Engineering & Sciences
Project Aim: To investigate the performance of a physical-layer RF fingerprinting system for signals acquired under 'real-world' (noisy) environmental conditions.

Program: Version 2, 26 August 2021

1. Design and build an RF front-end to allow acquisition of Bluetooth signals using equipment with low sampling rates (i.e. 250 MS/s).
2. Research and implement a system to automatically extract the 'turn-on' transient from a captured Bluetooth signal at various sample rates.
3. Research and build a machine learning system to automatically classify Bluetooth turn-on transients.
4. Assess the accuracy of the classifier to correctly identify a device for a given turn-on transient, using a dataset provided by others, and compare these results to those in the literature.
5. Using the RF front-end built earlier, acquire a collection of 'real-world' fingerprint data under controlled conditions, specifically:
 - (a) under a range of (transmitter) temperatures;
 - (b) while the transmitter is moving toward or away from the receiver; and
 - (c) in the presence of ambient (background) noise (i.e. lower SNR).
6. Assess the accuracy of the classifier to correctly identify a device for a given turn-on transient when presented with 'real-world' fingerprint data. Compare and contrast these results against the performance of classifier when using controlled fingerprint data.

As time and resources permit:


1. Automate acquisition and ingestion of Bluetooth turn-on transients.
2. Automate attribution of Bluetooth turn-on transients using transmitted identification address to assist non-cooperative identification.

Agreed:

Student Name: Eli Priest
Date: 26 August 2021
Supervisor Name: John Leis
Date: 26 August 2021

Appendix B

Risk assessment



University of Southern Queensland

USQ Safety Risk Management System

Read Only View

Develop as new RMP

Version 2.0

Safety Risk Management Plan

Risk Management Plan ID: RMP_2021_5544	Status: Approve	Current User:	Author:	Supervisor:	Approver:
Assessment Title: Collection of radiofrequency data samples under controlled conditions		Assessment Date: 18/05/2021			
Workplace (Division/Faculty/Section): 204070 - School of Mechanical and Electrical Engineering		Review Date: 18/05/2022 (3 years maximum)			
Approver: John Let's			Supervisor: (for notification of Risk Assessment only) John Let's		

Context

DESCRIPTION:

What is the task/event/purchase/project/procedure?
The collection of radiofrequency signals (consumer Bluetooth) under ideal and varied but controlled conditions. The ideal signals will be captured within an RF shielded room. The varied but controlled signals will be captured within a large temperature-controlled shed with specific variations (such as movement distance device-temperature) introduced to vary the signals.

Why is it being conducted?
To generate a database of Bluetooth turn-on transient signals in support of Honours research project.

Where is it being conducted?
AGD test facility Meajura ACT Australia

Course code (if applicable) _____ **Chemical Name (if applicable)** _____

WHAT ARE THE NOMINAL CONDITIONS?

Personnel involved
Eli Priest

Equipment
PC radio-frequency test equipment antenna mobile phones RF shielded room

Environment
open-area shed RF shielded room

Other

Briefly explain the procedure/process

- Assemble RF test equipment PC and the mobile phones (devices) in the RF shielded room.
- Turn on the Bluetooth on each device and use the RF test equipment and PC to capture the 'turn-on' transient for that device. Shutdown the device and repeat for all other devices.
- Use the environmental chamber to heat/cool each device. For each proscribed temperature turn on the Bluetooth on each device and use the RF test equipment and PC to capture the 'turn-on' transient for that device. Shutdown the device and repeat for all other devices.
- Move the RF test equipment antenna to the open-area shed. Move all devices to given distances from the antenna (e.g. 3m 10m etc.). For each proscribed distance turn on the Bluetooth on each device and use the RF test equipment and PC to capture the 'turn-on' transient for that device. Shutdown the device and repeat for all other devices.
- Using a metronome and 1 m markings on the shed floor walk at a given velocity while carrying each mobile device. While at the proscribed velocity turn on the Bluetooth on each device and use the RF test equipment and PC to capture the 'turn-on' transient for that device. Shutdown the device and repeat for all other devices.

Assessment Team - who is conducting the assessment?

Assessor[s]:
Eli Priest

Others consulted: (eg elected health and safety representative other personnel exposed to risks)
Site Health and Safety Representative

Risk Matrix

Probability	Consequence				
	Insignificant <small>No Injury 0-\$5K</small>	Minor <small>First Aid \$5K-\$50K</small>	Moderate <small>Med Treatment \$50K-\$100K</small>	Major <small>Serious Injury \$100K-\$250K</small>	Catastrophic <small>Death More than \$250K</small>
Almost Certain <small>1 in 2</small>	M	H	E	E	E
Likely <small>1 in 100</small>	M	H	H	E	E
Possible <small>1 in 1 000</small>	L	M	H	H	H
Unlikely <small>1 in 10 000</small>	L	L	M	M	M
Rare <small>1 in 1 000 000</small>	L	L	L	L	L

Recommended Action Guide

Extreme:	E Extreme Risk – Task MUST NOT proceed
High:	H High Risk – Special Procedures Required (Contact USQSafe) Approval by VC only
Medium:	M Medium Risk - A Risk Management Plan/Safe Work Method Statement is required
Low:	L Low Risk - Manage by routine procedures.

Risk Register and Analysis													
Step 1	Step 2	Step 2a	Step 2b	Step 3			Step 4						
Hazards: From step 1 or more if identified	The Risk: What can happen if exposed to the hazard w/ fault or ifing controls in place?	Consequence: What is the harm that can be caused by the hazard w/ fault or ifing controls in place?	Existing Controls: What are the existing controls that are already in place?	Risk Assessment: Consequence x Probability = Risk Level			Additional Controls: Enter additional controls if required to reduce the risk level		Risk assessment with additional controls: Has the consequence or probability changed?				
				Probability	Risk Level	ALARP				Consequence	Probability	Risk Level	ALARP
<i>Example</i>													
Use of environmental chamber	Injury due to burning Damage to device leading to fire	Minor	Limit maximum temperature to 40 degrees (rated device operating temperature). Double-check range on environmental chamber is within normal range. Wear gloves if necessary.	Rare	Low		Use IR thermometer to measure the temperature of any object in or near the environmental chamber prior to touching.	Insignificant	Rare	Low			
Slip trips falls	Injury or damage to equipment	Minor	Keep work area clean and tidy. Ensure power-cords are routed appropriately and secured where necessary.	Unlikely	Low								
Electrical hazard due to faulty equipment	Burns, fire or electrocution causing injury or death	Catastroph	All equipment to be tested and tagged. Use circuits with RCD. Familiarise self with power isolation procedures prior. Ensure dry-chem and CO2 fire extinguishers are placed close-by. Ensure first-aid kit and trained first-aiders are nearby.	Rare	Low								
Entrapment in RF shielded room	Accidentally locked in RF shielded room with no mechanism to escape	Insignificant	Processes require second person on-site to be aware of use of the room. Ensure emergency alarm system activates before use. Keep doorway clear. Check escape mechanism functions correctly.	Rare	Low								
Poor posture for long periods of	Injury due to poor posture over-extension	Minor		Possible	Med.		Ensure work area is setup in accordance with ergonomic advice. Use keyboard mouse and monitor in favour of laptop.	Minor	Rare	Low			

Step 5 - Action Plan (for controls not already in place)				
Additional Controls:	Exclude from Action Plan: (repeated control)	Resources:	Persons Responsible:	Proposed Implementation Date:
Use IR thermometer to measure the temperature of any object in or near the environmental chamber prior to touching.		IR thermometer	Eli Priest	18/05/2021
Ensure work area is setup in accordance with ergonomic advice. Use keyboard mouse and monitor in favour of laptop.		Keyboard monitor mouse	Eli Priest	18/05/2021

Supporting Attachments	
View Attachments	Click here to attach a file

Step 6 – Request Approval	
Drafters Name: Eli Priest	Draft Date: 18/05/2021
Drafters Comments: Discussed with the site's Health and Safety Representative.	
Assessment Approval: All risks are marked as ALARP	
Maximum Residual Risk Level: Low - Manager/Supervisor Approval Required	
Document Status: Approve	

Step 6 – Approval	
Approver's Name: John Leit	Approver's Position Title: project advisor
Approver's Comments: Previously discussed and minor points amended. Appears to cover the requirements. Recommend a PDF of this be included for ENG4111/2 report.	
I am satisfied that the risks are as low as reasonably practicable and that the resources required will be provided.	
Approval Decision: Approve	Approve / Reject Date: 19/05/2021
Document Status: Approve	

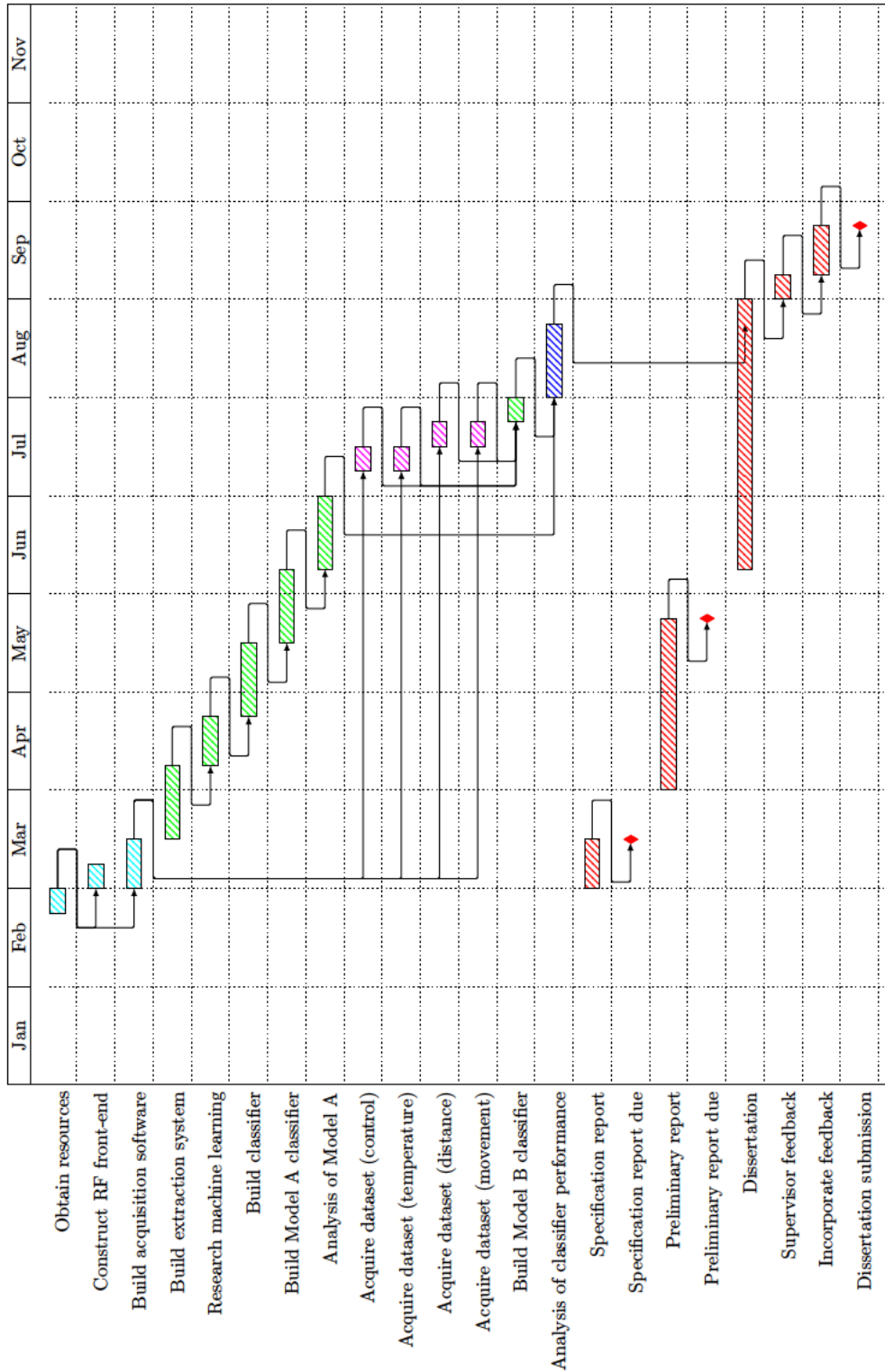
Appendix C

Project timeline

ENG 4111/2 Research Project

Project timeline

Eli Priest, 0061082889



Appendix D

Project resources

ENG 4111/2 Research Project

Project resources

Eli Priest, u1082889

The resources required for this project are broadly categorised as:

- RF downconversion front-end;
- low-cost RF acquisition hardware;
- computer equipment and software, for processing;
- mobile devices with Bluetooth for testing; and
- environment suitable for conducting controlled testing.

A list of specific equipment to complete the project can be found in Table D.1.

Table D.1: Project resources.

Qty	Item	Cost	Source/supplier
1	ISM2400 band 2.2 dBi omnidirectional antenna (WRL-00145 or similar)	\$12.33	Core electronics
1	Voltage controlled oscillator module, 2500 MHz (ZX95-2650-S+ or similar)	~\$60.00	Student
1	Low noise amplifier module, 2500–2700 MHz (ZQL-2700MLNW+ or similar)	\$128.34	cseonline.com.au
1	Mixer module, 750–6000 MHz (ZX05-63LH-S+ or similar)	\$87.98	cseonline.com.au
1	Low pass filter, 0–105 MHz (VLFX-105+ or similar)	\$73.30	cseonline.com.au
1	Band pass filter, 2340–2530 MHz (VBF-2435+)	\$64.05	cseonline.com.au
-	Various SMA / banana adaptors and cables	~\$40	cseonline.com.au
1	Linear power supply, triple output (Keysight E36313A or similar)	Nil	Student
1	Dataset of RF fingerprints acquired by others	Nil	Uzundurukan, Dalveren & Kara 2020
1	Digital oscilloscope, >250 MHz (PicoScope 5444B or similar)	Nil	Employer
1	PC running MATLAB	Nil	Student
10	Mobile devices (phones) with Bluetooth	Nil	Student's colleagues
1	Environmental climate chamber (5–35 °C)	Nil	Employer
1	RF shielded room	Nil	Employer
1	Large open-spaced warehouse/shed	Nil	Employer
1	Thermometer (resolution and accuracy at leave 0.1 °C)	Nil	Employer

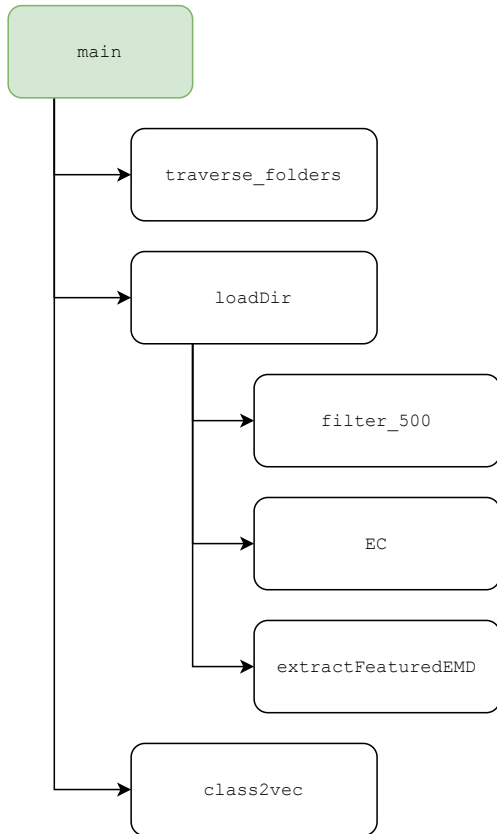
Appendix E

MATLAB[®] code

E.1 Overview of code blocks

RF fingerprinting

This column shows the hierarchy of code files required by the RF fingerprinting system. This includes code to automatically traverse dataset folders, perform digital filtering, extract the transient, calculate features, and store results in a table.



File conversion

This column shows the hierarchy of code files required to convert PicoScope files into MATLAB readable files. This includes code to automatically traverse dataset folders, perform file conversion, and determine if a transient shape is present - if it is, then keep the file; otherwise,

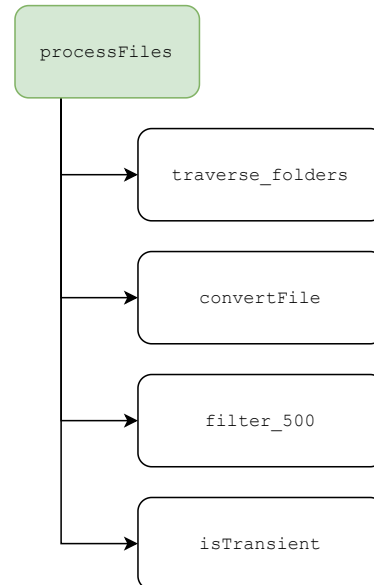


Figure E.1: Overview of code blocks used in RF fingerprinting system.

E.2 RF fingerprinting system

E.2.1 main.m

```
1 % Author: Eli Priest
2 % Student ID 61082889
3 % University of Southern Queensland
4 % Last revision: 2021-09-01
5 clear;
6 clc;
7 close all;
8
9 %% Definitions and constants
10 % Specify the base folder holding the datasets
11 base_folder = 'D:\fingerprinting\DatasetA';
12
13 % Sampling rate of the acquired samples (Hz)
14 fs = 500e6;
15
16 % Specify the constants and setting relevant to transient detection.
17 % Vartheta, for start of transient detection
18 extractorParams(1).EC_param = 120;
19 % Settling state tolerance (percent)
20 extractorParams(1).settleTol = 4;
21 % Window size of the 1-d median filter
22 extractorParams(1).tWindSize = 1200e-9;
23 extractorParams(1).mfWindSize = round(extractorParams(1).tWindSize *
    fs);
24 % Upper limit for settling time (in seconds)
25 extractorParams(1).MaxSettleTime = 5e-6;
26
27 % -----
28 %   Settings for bypassing or forcing recanning of the files or data
29 % -----
30 % Force re-extraction of classification data for each waveform?
31 %   true    = force force reading of waveforms from files, and create
    a new save file for retrieval later.
32 %   false   = do not force re-reading, but do it if a save file does
    not exist.
33 extractorParams(1).rescan = true;
34
35 % Force re-extraction of classification data for each waveform?
36 %   true    = force re-extraction of classification data, and create a
    new save file for retrieval later.
37 %   false   = do not force re-extraction, but do it if a save file
    does not exist.
38 extractorParams(1).reextract = true;
39
```

```

40 %% Identify number of unique classes , and file path of saved records
41 % Finds all folders , referenced from the base_folder , which are stored
    in the form <device>\<record_file>
42 % Each device folder represents a class
43
44 % Generate a complete list of directories that might hold waveform
    files by traversing from a given location .
45 arrayOfDeviceFiles = ...
46     traverse_folders(base_folder);
47
48 % Run loadDir() to scan all of the directories and extract the
    filtered and unfiltered waveforms
49 char_data = {};
50 for n = 1:length(arrayOfDeviceFiles)
51     % Open the files , and extract the filtered and unfiltered
        waveforms
52     devicePath = arrayOfDeviceFiles(n).path;
53     fprintf('Found record folder: %s\n', devicePath);
54     [~, class(n), char] = loadDir(devicePath, extractorParams(1), fs);
55
56     % Place data in a cell array , not an array , as it is not certain
        that all classes include the same number of records .
57     char_data{end+1} = char;
58 end
59
60 % Prepare the array for ingestion into the Classification Learner
61 all_device_features = [];
62 for nClass = 1:length(class)
63     % Iterate through each device_record in the class struct . Convert
        the f1 to f13 parameters to a matrix , with each row
        corresponding to one record . Append a device index to the
        beginning of the row , so that it takes the form:
64     % device_array = [dev f1 f2 f3 f4 f5 f6 f7 f8 f9 f10 f11 f12 f13];
65     dev_features = class2vec(class(nClass));
66     dev_id = ones(length(dev_features), 1) * nClass;
67     device_array = [dev_id, dev_features];
68     all_device_features = [all_device_features; device_array];
69 end
70
71 % Report the number of waveforms with a NaN
72 fprintf('Failed scanning %d waveforms (%.2f%%)\n', max(sum(isnan(
    all_device_features))), (max(sum(isnan(all_device_features)))*100/
    length(all_device_features)));
73
74 % Place all data to be processed by the classifier into a table
75 T = array2table(all_device_features);
76 T.Properties.VariableNames(1:14) = {'Class_number', 'f1', 'f2', 'f3', 'f4',
    'f5', 'f6', 'f7', 'f8', 'f9', 'f10', 'f11', 'f12', 'f13'};
77 T.Label = [arrayOfDeviceFiles(T.Class_number).device]';

```



```
78 |
79 | % Get the device type from the folder name. This allows the classifier
    |     to either check device-to-device comparisons (class=DeviceID) or
    |     model-to-model comparisons (class=DeviceModel).
80 | % Note: folders must be labelled in the format <DevID>-<DeviceModel>
81 | % For example "01-iPhone7" refers to device ID '01' with Device Model
    |     'iPhone7 '.
82 | splitLabel = split(T.Label, '-');
83 | T.DeviceID = splitLabel(:,1);
84 | T.DeviceModel = splitLabel(:,2);
85 | % Remove the unused device class identifier assigned by the loadDir
    |     stage
86 | T.Class_number = [];
87 |
88 | disp('End of program. ');
89 | %[EOF]
```



```
42 %                               datenum
43
44     files = dir(folder);
45     dirMask = [files.isdir] & ~strcmp({files.name}, '.') & ~strcmp({
46         files.name}, '..');
47     subFolders = files(dirMask);
47 end
48 %[EOF]
```

E.2.3 loadDir.m

```
1 % Author: Eli Priest
2 % Student ID 61082889
3 % University of Southern Queensland
4 % Last revision: 2021-09-01
5
6 function [waveform_data, features, char_data] = loadDir(folder, params
    , fs)
7 % This function scans a given folder for .txt and .csv files (which
    are assumed to contain valid waveform files) and parses them. The
    waveforms are filtered using the low-pass filter suitable for the
    sampling rate, and the hilbert transform is applied to extract the
    envelope.
8 %
9 % A struct is created to hold data from each record (file) being
    processed.
10 % folder      = the folder of the record
11 % name        = the filename of the record
12 % Fs          = the sampling frequency
13 % wave        = the unfiltered wave, normalised, but otherwise as
    read from the file
14 % filtWave    = wave, after low-pass filtering
15 % hilbertWave = the complex Hilbert Transform of filtWave
16 % startIndex  = the index of hilbertWave representing the start of
    the transient
17 % endIndex    = the index of hilbertWave representing the end of
    the transient, as detected in the single forward-pass
18 %
19 % Inputs:
20 % folder      the path (absolute or relative) of the folder
    holding the class records
21 % params      the extractorParams structure
22 % Outputs:
23 % waveform_data the struct array holding the waveforms and
    transforms of each record
24 % features     the struct holding details of the features
    extracted from the waveform
25
26 % -----
27 % record_data - should be stored in data.mat
28     anything_changed_flag = false;
29
30     % Check if this folder has already been processed and saved to
        file. If so, return that. Otherwise, do a new scan and save to
        file.
31     recordFileLocation = fullfile(folder, 'data.mat');
32
33     % Check if there is already a record data file in the folder
```

```

34     if isfile(recordFileLocation) && params(1).rescan==false
35         % If the file exists open the file
36         fprintf('\tOpening records file\n');
37         waveform_data = load(recordFileLocation, '-mat').waveform_data
38         ;
39         % Return here
40     else
41         % If the file does not exist, do a full scan
42         anything_changed_flag = true;
43         fprintf('\tParsing records\n');
44
45         % Get a list of all suitable files in the specified folder
46         waveformFiles1 = dir(fullfile(folder, '*.txt'));
47         waveformFiles2 = dir(fullfile(folder, '*.mat'));
48         % Remove the data files used to store critical information
49         % from the list, as they cannot be parsed as a waveform.
50         toRemove = ismember({waveformFiles2.name}, {'class.mat', 'data
51         .mat', 'char.mat'});
52         waveformFiles2(toRemove) = [];
53
54         filesInCurrentDir = [waveformFiles1; waveformFiles2];
55
56         % Iterate through all files in the list fileInCirrectDir
57         for n = 1:height(filesInCurrentDir)
58             currentFile = filesInCurrentDir(n);
59             currentFilePath = fullfile(currentFile.folder, currentFile
60             .name);
61
62             % Import the file contents using the correct decoder for
63             % the type.
64             [filepath, name, ext] = fileparts(currentFilePath);
65             if ext == '.txt'
66                 % When importing from textfile there is no time data,
67                 % so the sampling frequency fs must be specified.
68                 [time, wave] = importFromRef(currentFilePath, fs);
69             elseif ext == '.csv'
70                 [time, wave] = importFromPico(currentFilePath);
71             elseif ext == '.mat'
72                 % MATLAB files from the PicoScope do not include a
73                 % time series, but they do include information that
74                 % allows one to be calculated
75                 A = load(currentFilePath, '-mat');
76                 wave = A.A;
77                 time = ((0:A.Length - 1) * A.Tinterval) + A.Tstart;
78             end
79
80             % Calculate the sample frequency based on the timestamps
81             Fs = 1/mean(diff(time));

```

```
75     Fs_MHz = round(Fs/1e6);
76
77     % Filter the wave using the correct object for the given
       sampling rate, Fs_MHz
78     if Fs_MHz == 500
79         grpDelay = round(mean(grpdelay(filter_500)));
80         filtWave = filter(filter_500, wave);
81         filtWave = filtWave(grpDelay:end);
82     elseif Fs_MHz == 250
83         grpDelay = round(mean(grpdelay(filter_250)));
84         filtWave = filter(filter_250, wave);
85         filtWave = filtWave(grpDelay:end);
86     else
87         fprintf('No filter available for Fs = %d MHz.\n',
           Fs_MHz);
88         filtWave = wave;
89     end
90
91     % Normalise the waveform – remove DC offset, and rescale
       to [-1, 1]
92     filtWave = filtWave - mean(filtWave);
93     filtWave = filtWave / max(abs(filtWave));
94
95     % Calculate the envelope using the Hilbert transform
96     hilbertWave = hilbert(filtWave);
97     env = abs(hilbertWave);
98
99     % Apply a 1-d median filter to the signal envelope. Use
       the 'truncate' option to allow variable-length window
       padding at the edges, lessening underestimation of the
       signal.
100    envFiltered = medfilt1(env, params.mfWindSize, 'truncate')
       ;
101
102    % Find the start of the transient
103    %nTransStart = EC(filtWave, params.EC_param);
104    nTransStart = EC(envFiltered, params.EC_param);
105    waveform_data(n).startIndex = nTransStart;
106
107    % Find the end of the transient based on the settling
       point. Settling point is referenced from the midpoint,
       so this also needs to be calculated.
108    nMidPoint = ceil(midcross(envFiltered));
109    nSettleWind = length(envFiltered) - nMidPoint - 1;
110    % Set the maximum length the transient can be found in. It
       's either the end of the waveform (referenced from
       nMidPoint) or the maximum length of time stored in
       MaxSettleTime.
111    if nSettleWind > (Fs * params.MaxSettleTime)
```

```

112         nSettleWind = round(Fs * params.MaxSettleTime);
113     end
114     % Interpolate the instant of the settling time, in samples
      . Note that it is interpolated, so might not be an
      integer.
115     [~,~,SINST] = settlingtime(envFiltered, nSettleWind, '
      tolerance', params.settleTol);
116     nTransEnd = round(SINST);
117
118     % If a settling point could not be found, then the
      transient cannot be used. Set the badDataFlag.
      Otehrwise, report the start and end indexes.
119     if isnan(SINST)
120         waveform_data(n).endIndex = NaN; %Placeholder to
      overcome errors
121         waveform_data(n).badDataFlag = true;
122     else
123         waveform_data(n).startIndex = nTransStart;
124         waveform_data(n).endIndex = nTransEnd;
125         waveform_data(n).folder = currentFile.folder;
126         waveform_data(n).name = currentFile.name;
127         waveform_data(n).Fs = Fs;
128         waveform_data(n).transWave = filtWave(waveform_data(n)
      .startIndex:waveform_data(n).endIndex);
129         waveform_data(n).hilbertWave = hilbertWave(
      waveform_data(n).startIndex:waveform_data(n).
      endIndex);
130         waveform_data(n).badDataFlag = false;
131     end
132 end
133 end
134 % -----
135 % class_data - should be stored in class.mat in the device folder
136 classFileLocation = fullfile(folder, 'class.mat');
137 if isfile(classFileLocation) && params(1).rescan==false
138     % If the file exists open the file
139     fprintf('\tOpening class file\n', classFileLocation);
140     features = load(classFileLocation, '-mat').features;
141     % Return here
142 else
143     fprintf('\tCalculating class data\n', classFileLocation);
144     anything_changed_flag = true;
145     % If the file does not exist, process the class information
146     features(1).folder = folder;
147     features(1).fileList = filesInCurrentDir;
148 end
149
150 % -----
151 % char_data - should be stored in char.mat

```

```
152 charFileLocation = fullfile(folder, 'char.mat');
153 if isfile(charFileLocation) && anything_changed_flag == false &&
    params(1).reextract == false
154     % If the file exists open the file
155     fprintf('\tOpening features file\n', charFileLocation);
156     char_data = load(charFileLocation, '-mat').char_data;
157     % Return here
158 else
159     % Calculate the other features
160     fprintf('\tCalculating features\n');
161     for n = 1:length(waveform_data)
162         char_data(n) = extractFeaturesEMD(waveform_data(n));
163     end
164
165     % Create vectors for the individual characteristics (f1 to f13
        ) and save those vectors in class_data.
166     features(1).f1 = [char_data(:).f1];
167     features(1).f2 = [char_data(:).f2];
168     features(1).f3 = [char_data(:).f3];
169     features(1).f4 = [char_data(:).f4];
170     features(1).f5 = [char_data(:).f5];
171     features(1).f6 = [char_data(:).f6];
172     features(1).f7 = [char_data(:).f7];
173     features(1).f8 = [char_data(:).f8];
174     features(1).f9 = [char_data(:).f9];
175     features(1).f10 = [char_data(:).f10];
176     features(1).f11 = [char_data(:).f11];
177     features(1).f12 = [char_data(:).f12];
178     features(1).f13 = [char_data(:).f13];
179
180     % Save the resulting file for future retrieval
181     save(classFileLocation, 'features', '-mat');
182     save(recordFileLocation, 'waveform_data', '-mat');
183     save(charFileLocation, 'char_data', '-mat');
184 end
185 end
186 %[EOF]
```


E.2.4 EC.m

```
1 % Author: Eli Priest
2 % Student ID 61082889
3 % University of Southern Queensland
4 % Last revision: 2021-04-12
5
6 function [index] = EC(x, v)
7     % This function calculate the Energy Criterion of a signal.
8     % Implemented based on formulae from 'Performance Assessment of
9         Transient Signal Detection Methods and Superiority of Energy
10        Criterion (EC) Method' by I. S. MOHAMED, Y. DALVEREN, and A.
11        KARA (2020)
12
13     % Input:
14     % x          the time-varying signal (amplitude)
15     % v          the factor used to reduce the delay effect of
16        delta (other report success with default of 30)
17
18     % Output:
19     % index      the index of the lowest energy point (should
20        map the transient starting point).
21
22     if nargin < 2
23         fprintf(2, '[Error]\tno paramater passed. Using default v=35\n
24         ');
25         v = 35;
26     end
27
28     N = length(x);
29     Ei = cumsum(x.^2);
30     delta = Ei(end) / (v * N);
31     Ei_noise = [0:N-1]'*delta;
32     Ei = Ei - (Ei_noise);
33
34     [~, index] = min(Ei);
35
36 end
37 %[EOF]
```

E.2.5 extractFeaturesEMD.m

```

1 % Author: Eli Priest
2 % Student ID 61082889
3 % University of Southern Queensland
4 % Last revision: 2021-09-01
5
6 function class = extractFeaturesEMD(device_record)
7 % This takes a device_record struct, and calculates thirteen features
   to be used in the classifier. The features are as defined in '
   Assessment of Features and Classifiers for Bluetooth RF
   Fingerprinting' by Ali, Aysha M. / Uzundurukan, Emre / Kara, Ali
   (2019)
8 %
9 % Input:
10 % The device_record struct must include the following variables:
11 %   transWave   the 1d vector holding the amplitude data of the
   waveform, only within the transient stage, and after bandpass
   filtering has been applied.
12 %   startIndex  the index representing the start of the transient in
   filtWave
13 %   endIndex    the index representing the end of the transient in
   filtWave
14 % Output:
15 %   class struct holding features f1 to f13. Each feature is a single
   number.
16
17 % Takes in a device_record, and determines the characteristics
18   if device_record.badDataFlag == true
19       str = sprintf('\tBad data flag found: %s\n', device_record.
   name);
20       fprintf(str);
21       class(1).f1 = NaN;
22       class(1).f2 = NaN;
23       class(1).f3 = NaN;
24       class(1).f4 = NaN;
25       class(1).f5 = NaN;
26       class(1).f6 = NaN;
27       class(1).f7 = NaN;
28       class(1).f8 = NaN;
29       class(1).f9 = NaN;
30       class(1).f10 = NaN;
31       class(1).f11 = NaN;
32       class(1).f12 = NaN;
33       class(1).f13 = NaN;
34   else
35       % Empirical mode decomposition, EMD. Since the signal is not
   smooth, specify 'pchip' as the interpolation method.
36       imf = emd(device_record.transWave, 'Interpolation', 'pchip');

```

```

37
38     % Calculate the time–frequency–energy distribution for the
        waveform using the Hilbert–Huang transform (HHT). Use
        normalised frequency.
39     [hs, f, t, imfinsf, imfinse] = hht(imf);
40
41     % f1, transient duration (in samples)
42     class(1).f1 = device_record endIndex – device_record.
        startIndex;
43
44     % f2, total energy of transient energy
45     % Calculated by summing the instantaneous energy calculated by
        HHT
46     % Normalised by transient length
47     class(1).f2 = sum(sum(imfinse)) / class(1).f1;
48
49     % f3, total energy of transient energy envelope
50     % Normalise by transient length to remove bias
51     class(1).f3 = sum(device_record.transWave.^2) / class(1).f1;
52
53     % f4, Variance of transient energy envelope
54     class(1).f4 = var(abs(device_record.hilbertWave));
55
56     % Instantaneous phase of transient signal
57     inst_phase = atan(imag(device_record.hilbertWave)./real(
        device_record.hilbertWave));
58
59     % f5, StD of instantaneous phase of transient signal
60     class(1).f5 = std(inst_phase);
61
62     % f6, Entropy of inst. phase of transient signal
63     class(1).f6 = entropy(inst_phase);
64
65     % f7, Length of transient energy distribution
66     % Calculate the distance between the last point and the next
        using pythagoras, and take the sum of the vector.
67     % Normalise by transient length to remove bias
68     d = diff(imfinse(:,1));
69     distance = sqrt(mean(diff(t)).^2 + d.^2);
70     class(1).f7 = sum(distance) / class(1).f1;
71
72     % Sum of transient energy distribution, time axis
73     sum_time = sum(imfinse', 1);
74
75     % f8, Slope of transient energy distribution
76     p = polyfit(t, sum_time, 1);
77     slope = p(1);
78     class(1).f8 = slope;
79

```

```
80     % f9 , Variance of sum of transient energy distribution , time
      axis
81     % Normalise by transient length to remove bias from sum_time
82     class(1).f9 = var(sum_time) / class(1).f1;
83
84     % f10 , Maximum of sum of transient energy distribution , time
      axis
85     % Normalise by transient length to remove bias from sum_time
86     class(1).f10 = max(sum_time) / class(1).f1;
87
88     % f11 , Third order polynomial fitting coefficient of sum of
      transient energy distribution .
89     % Uses least-squares method to estimate third-order
      coefficient
90     % No requirement to normalise this by transient length
91     [p3, ~, ~] = polyfit(t, sum_time, 3);
92     f11 = p3(1);
93     class(1).f11 = f11;
94
95     % Sum of transient energy distribution , frequency axis
96     sum_freq = sum(imfinse);
97
98     % f12 , Maximum of sum of transient energy distribution ,
      frequency axis
99     class(1).f12 = max(sum_freq);
100
101     % f13 , Variance of sum of transient energy distribution ,
      frequency axis
102     class(1).f13 = var(sum_freq);
103     end
104 end
105 %[EOF]
```

E.2.6 class2vec.m

```
1 % Author: Eli Priest
2 % Student ID 61082889
3 % University of Southern Queensland
4 % Last revision: 2021-09-01
5
6 function [out] = class2vec(classStructure)
7 % This function converts a structure of Class Data into a vector of
   class data.
8 % The classStructure is expected to include fields f1 to f13.
9
10 % Get the size of the arrays in the class Structure
11 class_len = length(classStructure.f1);
12
13 % Create an array for output
14 out = zeros(class_len , 13);
15
16 out(:,1) = classStructure.f1;
17 out(:,2) = classStructure.f2;
18 out(:,3) = classStructure.f3;
19 out(:,4) = classStructure.f4;
20 out(:,5) = classStructure.f5;
21 out(:,6) = classStructure.f6;
22 out(:,7) = classStructure.f7;
23 out(:,8) = classStructure.f8;
24 out(:,9) = classStructure.f9;
25 out(:,10) = classStructure.f10;
26 out(:,11) = classStructure.f11;
27 out(:,12) = classStructure.f12;
28 out(:,13) = classStructure.f13;
29 end
30 %[EOF]
```

E.2.7 importFromRef.m

```
1 % Author: Eli Priest
2 % Student ID 61082889
3 % University of Southern Queensland
4 % Last revision: 2021-03-27
5
6 function [time, wave] = importFromRef(filename, Fs, dataLines)
7 % This function imports a waveform from a text file of ASCII-encoded
8 % float amplitude values, where each value occupies a row.
9 % There is no time information, so a time-series is generated by the
10 % function based on the sampling frequency Fs (given in seconds).
11 %
12 % Input:
13 % filename      full-path of text file to open, including
14 %               extension
15 % Fs            sampling frequency (in seconds)
16 % dataLines     two-element array of the first and last rows to be
17 %               read from the file. If blank, the function defaults to [1, Inf]
18 % Output:
19 % time          vector of time-series (in seconds)
20 % wave          vector of wave values
21
22 % If dataLines is not specified, define defaults - all rows of the
23 % file
24 if nargin < 3
25     dataLines = [1, Inf];
26 end
27
28 % If Fs is not specified, define defaults - 500 MS/s
29 if nargin < 2
30     Fs = 500e6;
31 end
32
33 opts = delimitedTextImportOptions("NumVariables", 1);
34 % Specify range and delimiter
35 opts.DataLines = dataLines;
36 opts.Delimiter = "";
37
38 % Specify column names and types
39 opts.VariableNames = "wave";
40 opts.VariableTypes = "double";
41
42 % Specify file level properties
43 opts.ExtraColumnsRule = "ignore";
44 opts.EmptyLineRule = "read";
45 opts.ConsecutiveDelimitersRule = "join";
46
47 % Import the data
```

```
43 tbl = readtable(filename , opts);
44
45 % Return the time series and the wave data
46 wave = tbl.wave;
47 N = length(wave);           % Number of samples
48 Duration = N/Fs;           % Signal Duration
49 time = linspace(0,Duration ,N); % Time vector
50 end
51 %[EOF]
```

E.2.8 filter_250.m

```
1 function Hd = filter_250
2 % Returns a discrete-time filter object.
3 % Generated by MATLAB(R) 9.9 and Signal Processing Toolbox 8.5.
4 % Equiripple Lowpass filter designed using the FIRPM function.
5 % All frequency values are in MHz.
6
7 Fs = 250; % Sampling Frequency
8
9 Fpass = 100; % Passband Frequency
10 Fstop = 105; % Stopband Frequency
11 Dpass = 0.057501127785; % Passband Ripple
12 Dstop = 0.0001; % Stopband Attenuation
13 dens = 20; % Density Factor
14
15 % Calculate the order from the parameters using FIRPMORD.
16 [N, Fo, Ao, W] = firpmord([Fpass, Fstop]/(Fs/2), [1 0], [Dpass, Dstop
17     ]);
18 % Calculate the coefficients using the FIRPM function.
19 b = firpm(N, Fo, Ao, W, {dens});
20 Hd = dfilt.dffir(b);
21 % [EOF]
```


E.2.9 filter_500.m

```
1 function Hd = filter_500
2 % Returns a discrete-time filter object.
3 % Generated by MATLAB(R) 9.9 and DSP System Toolbox 9.11.
4 % Chebyshev Type II Bandpass filter designed using FDESIGN.BANDPASS.
5 % All frequency values are in MHz.
6
7 Fs = 500; % Sampling Frequency
8
9 Fstop1 = 15; % First Stopband Frequency
10 Fpass1 = 20; % First Passband Frequency
11 Fpass2 = 102; % Second Passband Frequency
12 Fstop2 = 105; % Second Stopband Frequency
13 Astop1 = 90; % First Stopband Attenuation (dB)
14 Apass = 1; % Passband Ripple (dB)
15 Astop2 = 90; % Second Stopband Attenuation (dB)
16 match = 'stopband'; % Band to match exactly
17
18 % Construct an FDESIGN object and call its CHEBY2 method.
19 h = fdesign.bandpass(Fstop1, Fpass1, Fpass2, Fstop2, Astop1, Apass, ...
20 Astop2, Fs);
21 Hd = design(h, 'cheby2', 'MatchExactly', match);
22 % [EOF]
```

E.3 File conversion system

E.3.1 processFiles.m

```
1 % Author: Eli Priest
2 % Student ID 61082889
3 % University of Southern Queensland
4 % Last revision: 2021-08-31
5
6 close all;
7 clear;
8
9 %% File conversion
10 % Convert from Picoscope capture files to MATLAB files. The timestamp
    for the capture is used in the filename, for future use. Check the
    waveform file to confirm a transient is present, using the
    isTransient() function. If there is no transient, the file is
    deleted.
11
12 inputBasePath = 'C:\fingerprinting\captures_5Aug\unprocessed';
13 outputPath = 'C:\fingerprinting\captures_5Aug\processef_2';
14
15 folderList = traverse_folders(inputBasePath);
16 numFolders = length(folderList);
17
18 fprintf('Found %d folders\n', numFolders);
19
20 for a = 1:numFolders
21     fprintf('Current folder is %s\n', folderList(a).path);
22
23     % Get the number of valid files in that folder
24     fileList = getpsdataFiles(folderList(a).path);
25     numFiles = height(fileList);
26     fprintf('Found %d files\n', numFiles);
27
28     % Determine output folder. Create the folder, if required
29     currentFolderStructure = strsplit(folderList(a).path, filesep);
30     currentSubFolder = currentFolderStructure{end};
31     outputPath = fullfile(outputBasePath, currentSubFolder);
32     fprintf('Using output path %s\n', outputPath);
33     if ~exist(outputPath)
34         fprintf('Device folder does not exist. Creating...');
35         status = mkdir(outputPath);
36         if status == true
37             fprintf('Success\n');
38         else
39             fprintf('Failed\n');
40         end
    end
end
```

```
41     end
42
43     for b = 1:numFiles
44         % Do the file conversion
45         currentFile = fullfile(fileList(b).folder, fileList(b).name);
46         fprintf('%d/%d\t%d/%d : %s\n', a, numFolders, b, numFiles,
47             currentFile);
48         [s, subFolderName] = convertFile(currentFile, outputPath);
49
50         % PicoScope places all of the output files in a subfolder,
51         % which I do not want. I'll move them in a second. For now,
52         % check if the files actually contain a transient. If not,
53         % delete them.
54         filesToCheck = getWaveformFiles(fullfile(outputPath,
55             subFolderName));
56         numFilesToCheck = length(filesToCheck);
57         for c = 1:numFilesToCheck
58             % Open the waveform from the file
59             fileToParse = fullfile(filesToCheck(c).folder,
60                 filesToCheck(c).name);
61             A = load(fileToParse, '-mat', 'A');
62             A = A.A;
63
64             % Check if there is bad data (Inf, -Inf), which indicates
65             % the waveform was overrange for the PicoScope and data
66             % was clipped. Waveforms with bad data cannot be checked
67             % for a transient. If a waveform has bad data, it will be
68             % discarded.
69             badData = any(isinf(A));
70
71             % If the data is not bad, then check for a transient.
72             % Apply the filter first, to ensure any slow-moving
73             % superposition offsets are corrected.
74             if badData == false
75                 filtWave = filter(filter_500, A);
76                 transientStatus = isTransient(filtWave);
77             else
78                 transientStatus = false;
79             end
80
81             % If there is a valid transient (which also means the data
82             % is good) then keep the file, otherwise delete the file
83             .
84             if transientStatus == false
85                 delete(fileToParse);
86             else
87                 fprintf('Transient found: %s\n', filesToCheck(c).name)
88                 ;
89             end
90         end
91     end
```

```

75     end
76
77     % PicoScope creates subfolders for each waveform file , which I
       do not want. Move all of the '.mat' files one level up (
       where I want them) and delete the subfolder.
78     filesToMove = getWaveformFiles(fullfile(outputPath,
       subFolderName));
79     for d = 1:length(filesToMove)
80         existingFile = fullfile(filesToMove(d).folder, filesToMove
       (d).name);
81         proposedFile = fullfile(outputPath, filesToMove(d).name);
82
83         counter = 1;
84         while exist(proposedFile, 'file')
85             % The file exists – need to modify the name
86             [fPath, fName, fExt] = fileparts(proposedFile);
87             fName = [fName, '-', num2str(counter)];
88             proposedFile = fullfile(fPath, [fName, fExt]);
89             fprintf('File collision. Using name %s\n', [fName,
       fExt]);
90             counter = counter + 1;
91         end
92
93         movefile(existingFile, proposedFile);
94     end
95
96     pathToDelete = fullfile(outputPath, subFolderName);
97     removeStatus = rmdir(pathToDelete);
98     if removeStatus == false
99         fprintf('Error deleting folder %s\n', pathToDelete);
100    end
101
102    end
103 end
104
105 %% getWaveformFiles function
106 function [waveformFiles] = getWaveformFiles(folder)
107 % Reads out the contents of a folder , and returns a struct for each .
       mat found. Removes links to current and parent directory ( '.' and
       '..').
108 %
109 % Input:
110 %   folder           the path to the folder being scanned
111 %
112 % Output:
113 %   waveformFiles   a vector of struct which describes the subfolders
       within
114 %
       the given folder. Each struct within the vector
       has the

```

```
115 %             form :
116 %             name
117 %             folder
118 %             date
119 %             bytes
120 %             isdir
121 %             datenum
122
123     waveformFiles = dir(fullfile(folder , '*.mat'));
124 end
125
126 %% getpsdataFiles function
127 function [fileList] = getpsdataFiles(folder)
128 % Reads out the contents of a folder , and returns a struct for each .
129 % psdata found. Removes links to current and parent directory ( '.'
130 % and '..').
131 %
132 % Input :
133 % folder      the path to the folder being scanned
134 %
135 % Output :
136 % waveformFiles  a vector of struct which describes the subfolders
137 %                within
138 %                the given folder. Each struct within the vector
139 %                has the
140 %                form :
141 %                name
142 %                folder
143 %                date
144 %                bytes
145 %                isdir
146 %                datenum
147
148     fileList = dir(fullfile(folder , '*.psdata'));
149 end
150 %[EOF]
```

E.3.2 convertFile.m

```
1 % Author: Eli Priest
2 % Student ID 61082889
3 % University of Southern Queensland
4 % Last revision: 2021-07-21
5
6 function [status, subFolderName] = convertFile(inFile, outPath)
7 % This function converts a given .psdata file to Matlab file. The
   filename is generated with the date-time stamp of the file creation
   date (in ISO 8601 format). File conversion is handled by the
   picoscope executable.
8 %
9 % Input:
10 %   inFile           the full file-path of the .psdata file to be
   parsed
11 %   outPath          the full path to the output directory
12 %
13 % Output:
14 %   status           the status passed back from the picoscope system
15 %                   call
16
17 % Destination format. csv, txt, png, bmp, gif, a[nimated]gif, psdata,
   pssettings, mat. This is a mandatory argument.
18 format = 'mat';
19
20 % [<n>[:<m>]]|all    Waveform number n, waveform range n to m or all
   buffers
21 % Default is current waveform.
22 buffer = 'all';
23
24 % View to convert. Default is current view.
25 viewport = 'IF';
26
27 % The date-time string format to use in Matlab's datestr() function. I
   use ISO8601, which stores 01-Mar-2000 15:45:17 as '20000301T154517
   '.
28 datestrfmt = 30;
29
30 % Generate the date-time stamp string from metadata of inFullPath
31 FileInfo = dir(inFile);
32 v = FileInfo.datenum;
33 timedatestring = datestr(v, datestrfmt);
34 subFolderName = timedatestring;
35
36 % Destination file, including the timestamp
37 destFile = fullfile(outPath, [timedatestring, '.', format]);
38
39 % Use PicoScope software to complete the conversion
```

```
40 cmd = ['picoscope', ...
41       '/c "', inFile, '"', ... % Files to convert
42       '/d "', destFile, '"', ... % Full-name of output files ,
         including extension
43       '/f ', format, ... % Output format
44       '/q ', ... % Quite mode – no prompting
45       '/b ', buffer, ... % Buffers to convert
46       '/v ', viewport, ... % Viewport to convert
47       ''];
48 fprintf('Calling: %s\n', cmd);
49 status = system(cmd);
50 end
51 %[EOF]
```

E.3.3 isTransient.m

```

1 % Author: Eli Priest
2 % Student ID 61082889
3 % University of Southern Queensland
4 % Last revision: 2021-07-21
5
6 function [isTransientBool] = isTransient(inWave)
7 % This function determines if a given waveform vector contains a clean
   % turn-on transient.
8 %
9 % Input:
10 %   inWave           the time-varying signal (amplitude)
11 %
12 % Output:
13 %   isTransientBool the boolean result.
14 %       true = transient detected
15 %       false = transient not detected (or not clean)
16
17 % The envelope of a turn-on transient has the following basic shape:
18 %
19 %           -----
20 %           /
21 %          /
22 % ----- /
23 %
24 % It can be assumed that the start of the envelope will be below some
   % threshold, and the end of the envelope will be above some other
   % threshold. The waveform is corrected for DC offset, normalised, and
   % the peak envelope is extracted.
25
26 filtersize = 24; % Size of smoothing filter used in envelope
   % extraction
27 DEBUG = false; % Toggles the plotting of the waveform and the
   % critical thresholds
28
29 % The portion of the signal that forms the 'start' and 'end'. These
   % values are normalised by signal length.
30 lowerBound = 0.2;
31 upperBound = 0.7;
32 % The normalised thresholds for the maximum allowable magnitude during
   % the signal 'start' and signal 'end' periods.
33 lowerThreshold = 0.25;
34 upperThreshold = 0.7;
35
36 inWaveMean = mean(inWave);
37 if abs(inWaveMean) > 1e-6
38     % The waveform hasn't been corrected for DC offset, so correct it.
39     inWave = inWave - inWaveMean;

```



```
40 end
41
42 inWaveMax = max(abs(inWave));
43 if inWaveMax ~= 1
44     % The waveform hasn't been normalised, so normalise it.
45     inWave = inWave / inWaveMax;
46 end
47
48 % Calculate the envelope of the downsampled signal
49 env = envelope(inWave, filtersize, 'peak');
50
51 % Test to determine if first waveform follows the correct waveform
52 n = length(inWave);
53 lowerBoundSample = floor(n * lowerBound);
54 upperBoundSample = floor(n * upperBound);
55 expectedStart = max(abs(env(1:lowerBoundSample))) <= lowerThreshold;
56 expectedEnd = min(abs(env(upperBoundSample:end))) >= upperThreshold;
57
58 if DEBUG==true
59     lowest_y = min([0, min(env)]);
60     highest_y = max([1, max(env)]);
61     xv_low = [lowerBoundSample lowerBoundSample 1 1 lowerBoundSample];
62     yv_low = [lowest_y, lowerThreshold, lowerThreshold, lowest_y,
63             lowest_y];
64     xv_hi = [length(env) upperBoundSample upperBoundSample length(env)
65            length(env)];
66     yv_hi = [upperThreshold upperThreshold highest_y highest_y
67            upperThreshold];
68     xv = [xv_low NaN xv_hi];
69     yv = [yv_low NaN yv_hi];
70
71     plot(env);
72     hold on;
73     plot(xv, yv, 'k—');
74     hold off;
75
76     title('Plot of waveform peak envelope');
77     xlabel('Sample');
78     ylabel('Normalised amplitude');
79 end
80
81 % If both conditions are satisfied, we probably have a transient
82 waveform!
83 isTransientBool = expectedStart && expectedEnd;
84 %[EOF]
```