



University of Southern Queensland
School of Engineering

**WiFi Based Passive Vermin Sensing, Proof
of Concept and Feasibility Analysis**

Dissertation submitted by
Benjamin Balanzategui

In fulfilment of the requirements of
Course ENG4111/4112 – Research Project

Abstract

Vermin are a significant pest in Australia and throughout the world. When vermin ingress into a dwelling they are usually only detected after infestation has occurred. It is hypothesised that by analysing channel state information extracted from a WiFi network, it will be possible to detect and identify vermin when they are within the signal path of a typical WiFi network. A sensing utility function could be embedded into a WiFi system that detects vermin and alerts to their presence. This project determines if this concept is feasible by collecting channel state information from a WiFi system constructed using only commodity components.

Current standards defining the protocols that WiFi devices use require the determination of channel state information. Channel state information provides a rich representation of the propagation of individual components of the signal used in a WiFi system. Significant changes are observed when a physical object obstructs the signal path between transceivers. These changes can be analysed and categorised to identify the event occurring in the signal path, enabling passive sensing. Previous studies have investigated a variety of potential applications of WiFi sensing with an emphasis on health and wellbeing applications. The concept of using WiFi sensing to detect vermin is novel and has not previously been investigated.

To determine if WiFi based passive vermin sensing is feasible a WiFi network consisting of a single pair of transceivers was used to generate channel state information when a mouse is within the signal path. The collected channel state information was then analysed in comparison to channel state information collected from a control signal path, containing the same static objects but without a mouse present. The mouse caused conspicuous fluctuations in the magnitude measurements of the channel state information data and was able to be reliably identified by a Neural Network. The WiFi network utilised for testing was not modified in a way that inhibited normal communication functions. The findings of this project demonstrate that it is feasible to embed a sensing utility function into a typical WiFi network and vermin can be detected by a WiFi sensing system.

University of Southern Queensland
School of Engineering
ENG4111/ENG4112 Research Project

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.

University of Southern Queensland
School of Engineering
ENG4111/ENG4112 Research Project

CERTIFICATION OF DISSERTATION


I Benjamin Balanzategui, 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.

Benjamin Balanzategui



15 October 2023



Acknowledgements

I would like to give a special thanks to Professor John Leis for supervising this research project and his ongoing support throughout all phases of the project. My family, for their support both in maintaining my garden while I was busy working on the project and for putting up with the WiFi being turned off regularly while testing and verifying equipment.

I would also like to thank the Townsville City Council for allowing me the time off to complete my studies and this project while allowing me to maintain my role and employment, Professor John Leis and Dr. Andrew Maxwell who's courses inspired me to complete a research project in this field and Solex Electronics for sourcing some of the obscure equipment used in this project and the support they provided.

Table of Contents

Abstract	2
Acknowledgements	5
List of Tables.....	9
List of Figures	10
Abbreviations, Nomenclature and Acronyms	12
1. Introduction.....	13
1.1 Increasing the Functionality of Modern Electronics	13
1.2 Idea Development	14
1.3 Why Pests and Vermin?	14
1.4 Project Aims.....	15
2. Literature Review.....	16
2.1 Background to Wireless Communications Technologies	16
2.1.1 Orthogonal Frequency-Division Multiplexing and Multiple-Input Multiple-Output	16
2.2 Channel State Information	17
2.2.1 Channel State Information for Sensing	19
2.3 Survey and Analysis of WiFi Sensing Research.....	19
2.3.1 Challenges Derived from Using WiFi.....	20
2.3.2 How the Wireless Spectrum is Managed and Potential Impacts on Sensing	21
2.3.3 Detection Modelling and Analysis.....	22
2.4 Significance of Previous of Work to a Vermin Detection System.....	24
2.5 Limitations of using CSI for Sensing	25
2.6 Constructing a Sensing System with WiFi.....	26
2.6.1 Survey of WiFi Sensing Research Hardware Utilisation	28
2.6.2 The Beamforming Matrix for Sensing	30
2.6.3 The Nexmon CSI Extraction Tools.....	31
3. Methodology	33
3.1 Planning	33
3.1.1 Animal Testing and Ethics	33
3.1.2 Consequences.....	34

3.1.3 Risk Management	34
3.1.4 Project Resources.....	35
3.2 Design of WiFi Network to Capture CSI.....	39
3.2.1 Setting up Raspberry Pi for CSI Extraction	41
3.2.2 Operating the Extraction Tools	43
3.2.3 CSI Capture Operating Instructions and Example	46
3.2.4 Parsing CSI Data for Analysis	51
3.2.5 Matlab Code, Ingetsing CSI:.....	52
3.2.6 Matlab Code, Plotting Raw CSI:.....	53
3.3 Verifying the Test Equipment	57
3.3.1 Filtering Functions	57
3.3.2 Controlling Bandwidth and Channel.....	59
3.4 Testing Arrangement and Configuration of Test Network	63
3.4.1 Testing Procedure	66
4. Results and Analysis	69
4.1 Removing Outliers and Unwanted Subcarriers.....	70
4.1.1 Hampel Filtering for General Outlier Removal	70
4.1.2 Implicit Removal of Unwanted CSI Captures	72
4.1.3 Pre-processing Methodology and MATLAB Code	73
4.2 Utilising CSI Data for Sensing.....	76
4.3 Features Created in CSI from Vermin in Signal Path	78
4.3.1 Observations of the Mouse in Filmed Testing	80
4.3.2 Variance in Power Distribution.....	80
4.3.3 Changes between Subcarrier Measurements.....	82
4.4 Constructing a Sensing System.....	84
4.4.1 Block by Block Analysis and Feature Extraction	84
4.4.2 MATLAB code for block by block Processing and Feature Extraction	85
4.4.3 Sensing via Machine Learning.....	86
4.4.2 The MALAB code to Implement Neural Network	87
4.5 Components of the WiFi Sensing System.....	90

5. Conclusion.....	91
5.1 Reflection and Achievement of Objectives.....	91
5.1.1 Project Objectives	91
5.2 Further Work.....	92
6. References	95
Appendix A Project Specification.....	100
Appendix B Risk Management Plan	101
Appendix C Ethics Approval	103
Appendix D Gannt Chart	108
Appendix E MATLAB Function Ingesting CSI	109

List of Tables

Table 1. Common statsical features of CSI used in detection models (Zhang et al. 2020).....	23
Table 2. Hardware survey of WiFi sensing research.....	28
Table 3. Devices compatible with the Nexmon CSI extraction tools.....	31
Table 3. Project resource requirements	35
Table 4. Arguments for the <i>mcp</i> Command	45
Table 5. Arguments for the <i>ping</i> Command (incomplete list relevant commands only)	48
Table 6. Format of CSI Samples	50
Table 7. Number of subcarriers for each Channel Width in IEEE 802.11ac	52
Table 8. List of data carrying subcarriers IEEE 802.11ac.....	55
Table 9. Testing the CSI filtering functions	57
Table 10. Router testing and verification procedure	62
Table 11. Survey of signal path length used in CSI sensing research.....	65
Table 12. Testing Procedure.....	67
Table 13. Null and pilot subcarriers for channel bandwidths available on Pi 4.....	70

List of Figures

Figure 1. The Black Rat, <i>Rattus Rattus</i>	15
Figure 2. Block diagram of frequency division multiplexing	16
Figure 3. MIMO communication channel.....	17
Figure 4. Channel description of MIMO system.....	18
Figure 5. The challenge of isolating the signal refracted through a fruit	20
Figure 6. Possible channel usage in IEEE802.11ac from 5170MHz to 5835MHz	21
Figure 7. Subcarrier arrangement for 802.11ac (and similar 802.11 variants).....	22
Figure 8. The outlier removal algorithm used by Schäfer et al. (2021)	24
Figure 9. Diagram of signal fading between WiFi devices used for height estimation	25
Figure 10. Model of a typical human activity WiFi sensing system	26
Figure 11. Intel WiFi Link 5300 Wireless Network Interface Card.....	27
Figure 12. Intel WiFi Link 5300 Ultimate N Wireless Card Half Mini Pcie 802.11n	28
Figure 13. Example of the effect of the beamforming (steering) matrix.....	31
Figure 14. Schematic of Broadcom BCM43455 (Left) and Raspberry Pi 3B+ with WiFi components circled (Right)	33
Figure 15. Testing equipment arrangement.....	40
Figure 16. Flow chart of WiFi NIC firmware for CSI extraction	41
Figure 16. The Raspberry Pi configuration tool.....	42
Figure 17. Accessing the Pi and checking USB drive mounting	43
Figure 18. Raspberry Pi 3B+ with required connections for CSI capture.....	44
Figure 19. Help view of the Nexmon utility	46
Figure 20. TP-Link webserver – disabling the 2.4GHz IEEE 802.11b/g/n network.....	47
Figure 21. TP-Link webserver, the 5GHz IEEE 802.11ac settings for test extraction.....	47
Figure 22. Pinging the TP-Link router via the Raspberry 4 with 10ms ping interval	48
Figure 23. Remote Terminal of Pi 3B+ During CSI Capture.....	50
Figure 24. Wireshark captured of CSI Data	51
Figure 25. Test capture CSI data ingested, <i>csi_raw</i> Variable	53
Figure 26. Plot of CSI data from test capture: Phase and Amplitude vs Subcarrier Index	54
Figure 27. Plot of CSI data from test capture, null subcarriers removed	55

Figure 28. Test capture of four CSI samples from the Atheros CSI Tool.....	56
Figure 29. Neatgear configuration webserver	59
Figure 30. Scan of WiFi network in range of the Pi 4.....	60
Figure 31. Overlapping WiFi channels of varying width.....	61
Figure 32. Plot to identify channel width	61
Figure 33. The mouse positioned for testing (left) and close up of the mouse used during testing	63
Figure 34. Singal path used for testing.....	64
Figure 35. Close up of the TP-Link router and Pi 4 in the test network	64
Figure 36. CSI data from test 37	72
Figure 37. CSI data from test 37, all raw data before pre-processing	75
Figure 38. CSI data from test 37, processed data.....	75
Figure 39. CSI frame components and Fourier analysis of amplitude and magnitude for frame 500 from test 37	77
Figure 40. Comparison of CSI data, left control test, right test with mouse	78
Figure 41. Power per frame comparison of Tests 37 and 38.....	79
Figure 42. Results from filmed test 73, Power vs Approximate Test Duration	80
Figure 43. Variance of power per frame throughout 20 Tests	81
Figure 44. Blank test data containing split magnitude grouping.....	82
Figure 45. Subcarrier analysis, Mouse vs Blank Control	83
Figure 46. Confusion matrix from Neural Network test	88
Figure 47. Analysis of misidentified CSI data from mouse	89
Figure 48. Correctly identified blank control CSI block (right) and misidentified block (left)	89
Figure 49. Components of sensing system.....	90

Abbreviations, Nomenclature and Acronyms

ACMA - Australian Communications and Media Authority

AP – Access Point, wireless networking devices that sets up and connects a wireless network to other networks

CSI – Channel State Information

EMI – Electromagnetic Interference

E-waste – Electronic Waste

IEEE – the Institute of Electrical and Electronics Engineers

IP – Internet Protocol – where relevant version 4 is the sole version referred to within this dissertation

KVM – Keyboard, Video, Mouse

LOS – Line of Sight

MAC – Media Access Control

MAD – median absolute deviation

MIMO – Multiple-Input Multiple-Output

NIC – Network Interface Controller

OFDM - Orthogonal Frequency-Division Multiplexing

Pi – Raspberry Pi miniature computer

PoE – Power over Ethernet

RF – Radio Frequency

Rx – Receive

RSSI – Receive Signal Strength Indicator

SDR – Software Defined Radio

Tx – Transmit

UDP – User Datagram Protocol

VHT – Very High Throughput

WiFi – wireless fidelity, devices used for wireless networking

1. Introduction

This project aims to determine if it is feasible to create a system that could detect vermin using only inputs that would be generated by a functioning WiFi network that complies with the modern suite of Institute of Electrical and Electronics Engineers (IEEE) 802.11 standards. It is hypothesized that in a typical domestic dwelling with a WiFi network, the channel state information (CSI) generated by the WiFi network would contain the required data to detect vermin. If this concept is correct, it may be possible that tracking algorithms could be refined and optimised until they are suitably robust, generic and reliable to create a software alert system that could be deployed in conjunction with a WiFi network to monitor the dwelling for vermin and alert when any vermin are detected so that the occupants can act prior to infestation. Such a system would be readily accessible as WiFi usage in homes is already ubiquitous and it would create an additional useful utility to existing WiFi devices that would help protect homes and property against vermin.

Testing conducted as part of the project aims to demonstrate that vermin sensing, and detection is possible. Also, a review of detection techniques employed in similar studies will be used to identify and analyse the key issues that would need to be overcome to create a system that is capable enough to be useful.

Electronic devices such as smartphones are becoming increasingly essential to everyday life and after decades of advances in computer sciences, integrated circuit manufacture and other related fields, an electronic device is available for almost every conceivable application, from delivering pizzas to complex keyhole surgery. The world is also becoming more digitally interconnected than ever. In 2021 91% of Australian adults had a home internet connection and this proportion is growing in Australia and the world each year (ACMA 2021a).

1.1 Increasing the Functionality of Modern Electronics

While this technological and information revolution has led to significant improvements to quality of life and a varied range of improved outcomes, there are also challenges. In 2018 the total generation of E-waste throughout the world was approximately 49.8 million metric tons, and the generation of E-waste is growing at three times the rate of other streams of waste (Islam, Dias & Huda 2020). One of the ways innovators are working to abate the generation of E-waste is to increase the functionality of devices maximising their potential use and reducing the need for discrete singular purpose devices. Smartphones are perhaps the best example of this as they now contain cameras, GPS mapping, calculators, along with a wide range of other app and platform-based functions, and new functions and applications are being considered and investigated continuously.

The adoption of the current Institute of Electrical and Electronics Engineers (IEEE) suite of standards IEEE802.11, that describes the protocols utilised by wireless networking equipment, has unintentionally provided an opportunity to expand the functionality of typical WiFi devices beyond networking applications and into sensing applications. Prior to IEEE802.11n wireless networking devices, only determined received signal

strength indication (RSSI). RSSI will reveal characteristics of the signal propagation between transceivers but is a coarse metric in terms of sensing capability and is only based on the power in the baseband signal.

IEEE802.11n (and all newer iterations of the standard) compliant WiFi devices determine and utilise CSI, which is a granular representation of the power and phase of each subcarrier component in the signal (IEEE 2013). By using similar principles to the way radar technology has been used in aviation for decades, it is possible to use radio frequency (RF) signals to discern quite a lot of information about the surrounding environment (Zhou et al. 2015). Passive sensing systems based on the measurement of received RF signals within the frequency spectrum used by WiFi have been developed that show potential to detect the presence of humans, count, localise and track people and recognise gestures and activities (E. Cianca 2017).

1.2 Idea Development

The project concept of using CSI from wireless networks for detection and monitoring was proposed by Professor John Leis. A significant focus of the previous studies examining potential sensing applications of CSI from WiFi networks relate to health and wellbeing. A recent project conducted by Wang, R. et al. (2022) demonstrated that respiration rate could be measured somewhat reliably by using information captured from a WiFi network. Damodaran et al. (2020) and Wang, C. et al. (2022) were able to create somewhat effective fall detection systems that could potentially assist carers in monitoring the elderly or vulnerable.

Due to the important nature of healthcare, devices used for medical applications are often designed using a best practice approach. While perhaps useful in supplementing the existing tools available for such health and wellbeing applications, CSI from WiFi networks is not designed for or optimised for sensing activities (Zhang et al. 2022). A key barrier impeding the development of WiFi sensing systems is the reluctance of wireless chip vendors and manufacturers to expose CSI and make available control of the features of WiFi devices that would enable sensing (Schäfer et al. 2021). If applications of WiFi sensing that provide useful utility value without any serious consequence in the case of error are proven feasible it may motivate vendors and manufacturers to assist in the development of WiFi sensing due to marketability.

1.3 Why Pests and Vermin?

Rats are often not seen in dwellings unless they are present in large numbers (*Vermin-Managing Rats in Your Home* 2019). A close friend recently suffered from an unexpected and severe vermin infestation. His Townsville, North Queensland, home was infested with Black Rats. If this friend had realised there were vermin in his home before sighting a Rat, the damage caused, and cost of eradication could have been significantly reduced. Vermin can procreate extremely quickly. Mice in particular are prolific breeders and can give birth to a litter of up to 10 young every 20 days (CSIRO 2021). Australia is also one of only 2 countries that experience mice plagues, so management technologies are of particular interest to Australians (CSIRO 2021). When residing in a dwelling vermin can cause harm in many ways including carry disease such as leptospirosis and

typhus, gnawing electrical cables and structures and carry harmful parasites (SAHealth 2022). Sydney is currently experiencing an alarming increase in the rat population, with estimates the population was between 500 million and 1 billion in 2021(Sydney Struggles to Get Rat Problem Under Control 2021) . A system that could be deployed into preexisting WiFi networks alerting to the ingress of vermin would allow for eradication and control measures to be implemented at an earlier stage, significantly reducing the impact of vermin and providing valuable data to help monitor the vermin population in WiFi dense areas.



Figure 1. The Black Rat, *Rattus Rattus*

(AustralianMuseum 2022)

1.4 Project Aims

1. Survey, review and analyse previous WiFi sensing research and experimentation. Conduct initial background research into using radio frequency signals, specifically microwaves for sensing and the operational aspects of WiFi networks which will affect sensing.
2. Procure and configure WiFi hardware that will facilitate the extraction of CSI and parse the CSI data into a suitable software application e.g. MATLAB that can perform statistical analysis and implement detection algorithms.
3. Design a test apparatus that simulates a WiFi network, where stimuli can be placed in the signal path including vermin (mice) to capturing and log CSI.
4. Gather data from testing that can be used to examine the feasibility of using WiFi sensing as a vermin detection system.

5. Determine if it is possible to detect vermin via CSI what limitations and constraints may impede the development of a system intended to be used as an additional utility function in a typical WiFi communication network.

2. Literature Review

2.1 Background to Wireless Communications Technologies

Two of the most important technologies that have enabled modern wireless communication networks to facilitate high data rates are Orthogonal Frequency-Division Multiplexing (OFDM) and Multiple-Input Multiple-Output (MIMO) (Ma 2019). The suite of IEEE standards that are used to define modern WiFi equipment, typically 802.11 a/b/g/n/ac, require the implementation of OFDM-MIMO technology.

2.1.1 Orthogonal Frequency-Division Multiplexing and Multiple-Input Multiple-Output

OFDM is a modulation technique that encodes data streams in multiple channels across multiple frequencies, enabling high bandwidth transmissions (Leis 2018). Each channel encodes a bit stream mixed with a unique carrier frequency. The final transmitted signal is a summation of each channel, so contains many different frequency components (Leis 2018). Figure 2. depicts three channel frequency division multiplexing. An OFDM scheme is similar, but each channel would contain an additional orthogonal frequency component also i.e. a sine and cosine component.

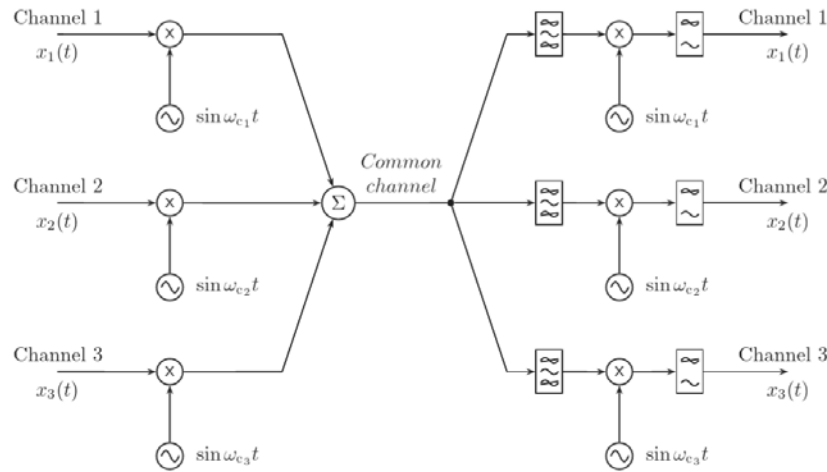


Figure 2. Block diagram of frequency division multiplexing

(Leis 2018)

MIMO utilises multiple transmit and receive antennas in the same communication session. Using multiple antennas allows for a higher number of individual channels to be used, increasing the data rate (Ma 2019). Reliability can also be increased by MIMO as each signal path between a single transmit-receive antenna pair will be unique. Some paths will be impacted less by signal fading caused by obstructions within the signal path that create reflections, refractions, scattering or other affects that attenuate RF signals, increasing the chance of

one path transferring the data successfully (Paul & Ogunfunmi 2008). MIMO systems are also capable of overcoming and in fact leveraging off of multipathing, a phenomenon where reflections, refractions and scattering caused by objects within the signal path cause the transmitted signal to arrive at the receiver multiple times independently.

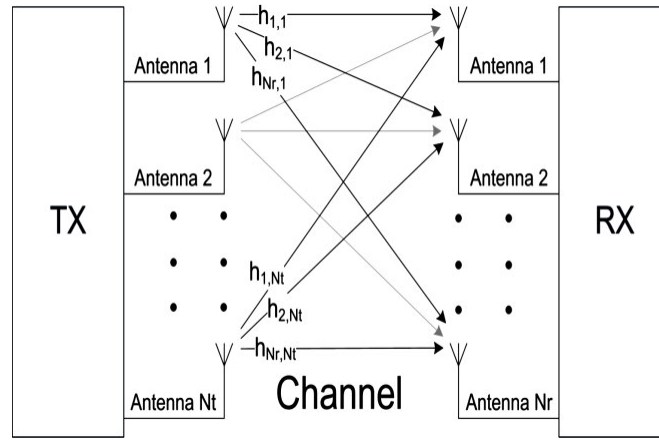


Figure 3. MIMO communication channel

(Abbas 2016)

Using multiple antennas in an RF transmission will not necessarily provide any benefit. MIMO becomes beneficial when signal processing techniques are used in each device to take advantage of multipathing and channel capacity (Jansons & Dorins 2012).

2.2 Channel State Information

An OFDM-MIMO system will transmit multiple frequencies using multiple signal paths. Channel State Information (CSI) describes how each frequency component of the transmission will propagate via each signal path. It provides a very granular assessment of the transmission as each CSI component describes an individual carrier frequencies attenuation and response as it traverses a specific signal path between a transmit and receive antenna pair.

The received signal in an MIMO system can be described by:

$$Y(f, t) = H(f, t) \cdot X(f, t) + N$$

Where $Y(f, t)$ and $X(f, t)$ are the received and transmitted signals respectively in the frequency domain, with carrier frequency f , measured at time t (Viswanathan 2014). The convolution of the transmitted signal with the CSI, $H(f, t)$, including the addition of some noise N , results in the received signal.

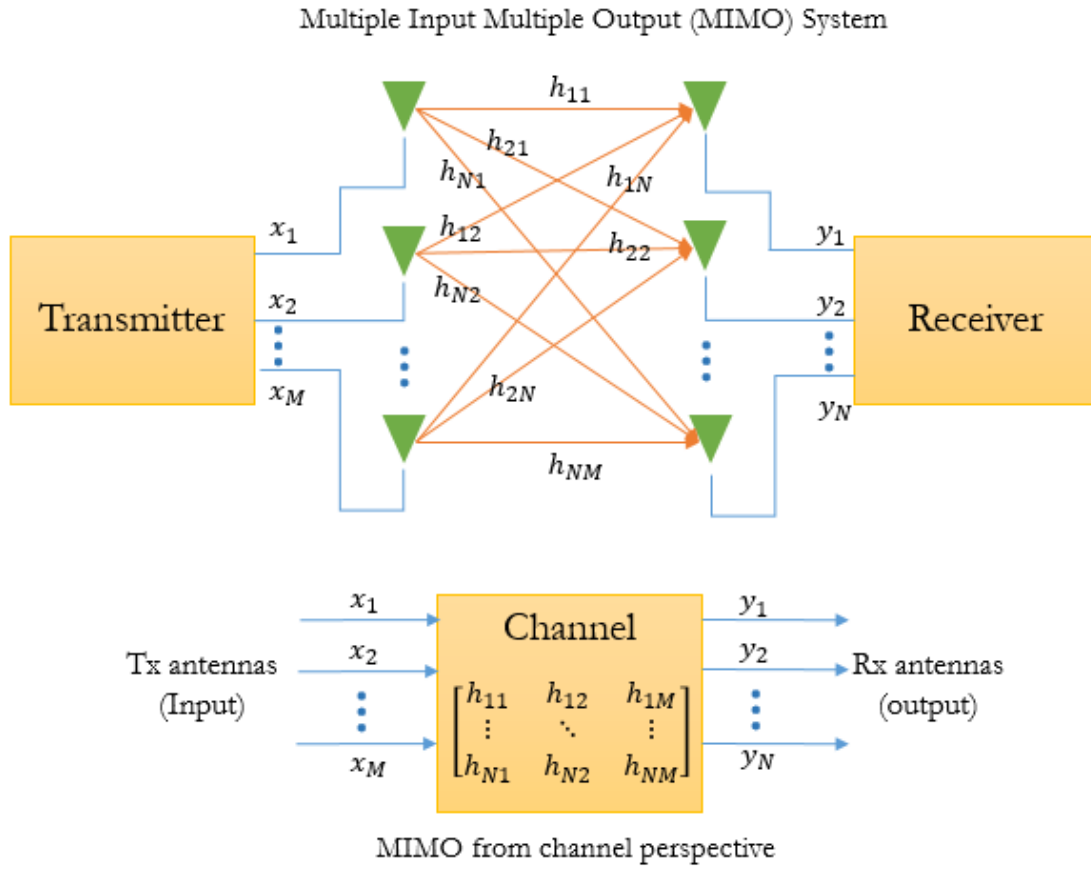


Figure 4. Channel description of MIMO system

(Viswanathan 2014)

In Figure 4, the H matrix in the Channel depiction is the CSI matrix. When a MIMO system also utilizes OFDM each transmission within the common channel will be comprised of several individual signals modulated using a carrier frequency then mixed into a common signal to be transmitted across the channel.

In a WiFi system, CSI will describe how an individual subcarrier propagates through a signal path between the antennas of two adapters (Wang et al. 2021). CSI is a set of values that will correspond to an OFDM subcarrier:

$$CSI_i = |CSI_i|e^{j(\angle CSI_i)}$$

(Wang et al. 2021)

Where CSI_i is the CSI of the i th OFDM subcarrier.

$|CSI_i|$ represents the gain, $\angle CSI_i$ represents the phase.

The signal received by an OFDM-MIMO system can be described in terms of the subcarrier index:

$$y_i = H_i x_i + n_i$$

(He et al. 2020)

Where i is the subcarrier index of the OFDM scheme, $x_i \in \mathbb{R}^{N_T}$ and $y_i \in \mathbb{R}^{N_R}$ are the transmitted and received signal respectively. N_T and N_R are the number of transmit and receive antennas respectively and n_i is a vector representing the noise (He et al. 2020). H_i represents the CSI matrix for the i th OFDM subcarrier.

$$\mathbf{H}_i = \begin{bmatrix} \text{CSI}_i^{11} & \text{CSI}_i^{12} & \dots & \text{CSI}_i^{N_T} \\ \text{CSI}_i^{21} & \text{CSI}_i^{22} & \dots & \text{CSI}_i^{2N_T} \\ \vdots & \vdots & \ddots & \vdots \\ \text{CSI}_i^{N_R1} & \text{CSI}_i^{N_R2} & \dots & \text{CSI}_i^{N_T N_R} \end{bmatrix}$$

(He et al. 2020)

2.2.1 Channel State Information for Sensing

The capability of using high frequency radio systems (UHF, SHF and EHF) for sensing and localisation applications has been utilised for decades and is well understood. RSSI is determined almost universally in wireless communications systems and has been used as the input to sensing systems successfully. However, RSSI is much coarser metric than CSI. RSSI's utility value for sensing is reduced dramatically in complex situations (such as indoors with multiple objects in the signal path) due to multipath fading and temporal dynamics (Yang, Zhou & Liu 2013). RSSI would not be a suitable input for a system capable of detecting vermin reliably indoors and would likely be unsuitable for most other potential utility sensing functions of a WiFi network.

The intent of determining CSI is to allow the WiFi devices to optimise communication by avoiding signal fading. When a physical object impacts the signal path there will be distinct changes in the CSI values (Wang et al. 2021). By recognising the pattern in which these changes occur when the signal path is impacted allows the event causing the change to be categorised and recognised, enabling passive sensing.

Each antenna-to-antenna signal path within an MIMO-OFDM WiFi system will traverse the objects within the signal path in different ways and differing areas of the frequency spectrum are not uniformly impacted by the size and composition of physical objects. This makes it possible to discern information about the physical environment by analysing the signal (He et al. 2020).

2.3 Survey and Analysis of WiFi Sensing Research

WiFi devices that determine and utilise CSI have now been commonplace for over a decade and with WiFi networking becoming increasingly ubiquitous, the potential of using CSI (and WiFi systems in general) to create passive sensing systems has attracted the attention of researchers. Earlier projects focused on detection of humans and some basic characterising of actives such as being stationary or moving (Xiao et al. 2012; Wu et al. 2015; Zhou et al. 2015; Palipana, Agrawal & Pesch 2016). Once the sensing potential was proven to be feasible, WiFi based intruder detection systems were noted as a potential application and investigated by (Gong et al. 2015; Tian et al. 2018; Lin et al. 2020). A more recent focus has been the potential of using WiFi sensing in healthcare applications in part driven by the emphasis the COVID19 pandemic placed on the importance of improving healthcare technology (Ge et al. 2022). Two of the most common aims are to accurately sense vital signs (He et al. 2020; Wang, Yang & Mao 2020; Kanda et al. 2022), and create fall detection systems (Wang, Yang & Mao 2017; Damodaran et al. 2020).

The targets and aims of WiFi sensing testing and experimentation are broad. Some novel studies examined whether fruit ripeness can be measured (Tan, Zhang & Yang 2018) and more recently if fire can be reliably detected using commodity WiFi devices (Li et al. 2021).

In Tan, Zhang & Yang's (2018) work, it was assumed that the relative permittivity of fruit would change as the fruit ripens due to physiological changes. The change in relative permittivity ϵ , would then cause a change in the attenuation factor α , which would be detectable by analysis of CSI due to changes in refraction.

Complex relative permittivity:

$$\epsilon^* = \epsilon' - j\epsilon''$$

Attenuation factor:

$$\alpha = \frac{2\pi}{\lambda_0} \left[\frac{1}{2} \epsilon' \left(\sqrt{1 + \left(\frac{\epsilon''}{\epsilon'} \right)^2} - 1 \right) \right]^{\frac{1}{2}}$$

Where λ_0 is the wavelength of the WiFi signal, α is the attenuation factor and ϵ , is the relative permittivity.

(Tan, Zhang & Yang 2018)

Due to multipathing, it proved difficult to isolate the signal within the WiFi system that travelled directly through the fruit being tested for ripeness but by examining the power delay over a wide range of subcarriers it was able to be isolated and the system developed was able to detect the ripeness of kiwi fruit and avocados with 90% accuracy (Tan, Zhang & Yang 2018).

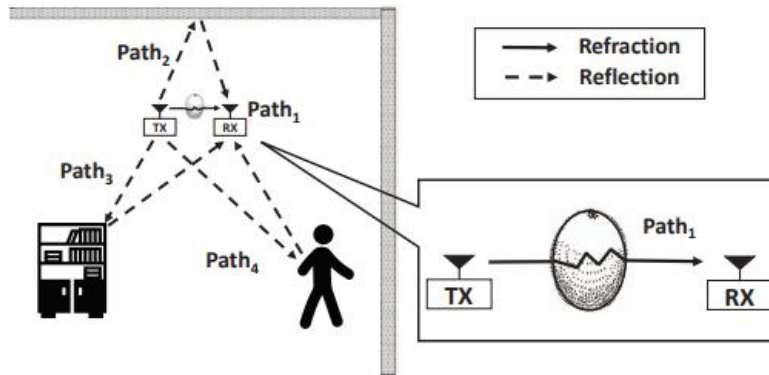


Figure 5. The challenge of isolating the signal refracted through a fruit

(Tan, Zhang & Yang 2018)

2.3.1 Challenges Derived from Using WiFi

The problem which Tan, Zhang & Yang (2018) encountered was exacerbated by two factors that would not be present in a purpose-built RF sensing system, for example an aircraft radar system. An MIMO-OFDM WiFi system creates very complex and less predictable multipathing that is not optimal for interpreting and thus using

a simple sensing algorithm due to the number of antennas and frequencies used. Tan, Zhang & Yang (2018) also intended to use signals spread across 600MHz of bandwidth to interpret which signal passed through the fruit with sufficient accuracy. The radio hardware used in a modern WiFi device is capable of performing this sweep, however local regulations and the WiFi devices configuration may prohibit using 600MHz of contiguous bandwidth to transmit signals with uniform spacing as was required.

2.3.2 How the Wireless Spectrum is Managed and Potential Impacts on Sensing

Within Australia the Australian Communications and Media Authority (ACMA) determines which spectrum areas are available for different services and purposes (ACMA 2022). The ACMA also govern the use of each spectrum area by imposing rules such as the maximum transmission power at certain frequencies and whether a license must be held to use certain spectrum areas. This includes which frequencies are available for use in local area wireless broadband communication services including WiFi. Other regions throughout the world have similar governing bodies and while the spectrum areas used for WiFi in other regions are similar due to commonality in the IEEE802.11 standards, there are differences and the complete spectrum utilised by the standard may not be accessible and what is accessible may not be contiguous.

The spectrum utilised is portioned and assigned channels. IEEE802.11ac, often referred to as 5GHz WiFi 5, channels are referenced by the centre frequency and are 5MHz wide (IEEE Standards Association 2013). Standard bandwidths of 20MHz, 40MHz, 80MHz and 160MHz are used and are referenced by the centre channel i.e. channel 36 at 20MHz bandwidth will use 5170MHz – 5190MHz.

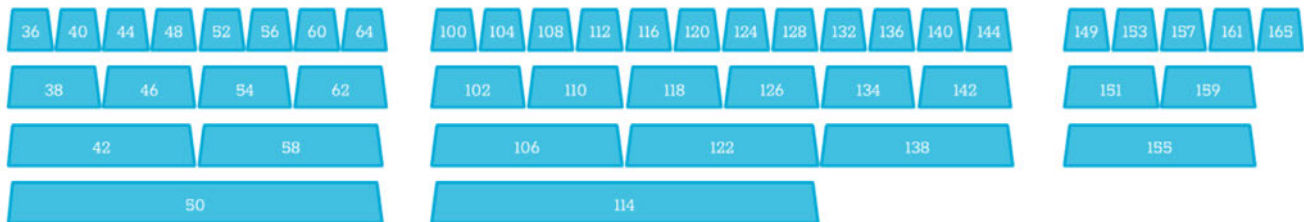


Figure 6. Possible channel usage in IEEE802.11ac from 5170MHz to 5835MHz

(IEEE Standards Association 2013)

In Australia channel 32, centre frequency 5160MHz to channel 48 centre frequency 5240MHz can be used indoors with up to 200mW of transmit power without restrictions (ACMA 2021b). Many other areas of the spectrum are available for use in WiFi, but this highlights the difficulty of performing a bandwidth sweep in order to collect CSI for sensing. Within channels data is encoded via OFDM on subcarriers with a default spacing of 312.5kHz (IEEE Standards Association 2013). There are also special purpose subcarriers that do not transmit data. These null and pilot subcarriers are designed to make decoding data less complex as well as for error and integrity checking (Gast 2013). These subcarriers, in particular the null subcarriers, which are only used as a DC offset will create outliers in the CSI samples that need to be handled or disregarded in analysis.

Figure 7. depicts the subcarrier arrangement for IEEE802.11ac with pilot subcarriers denoted by a zero horizontal axis value.

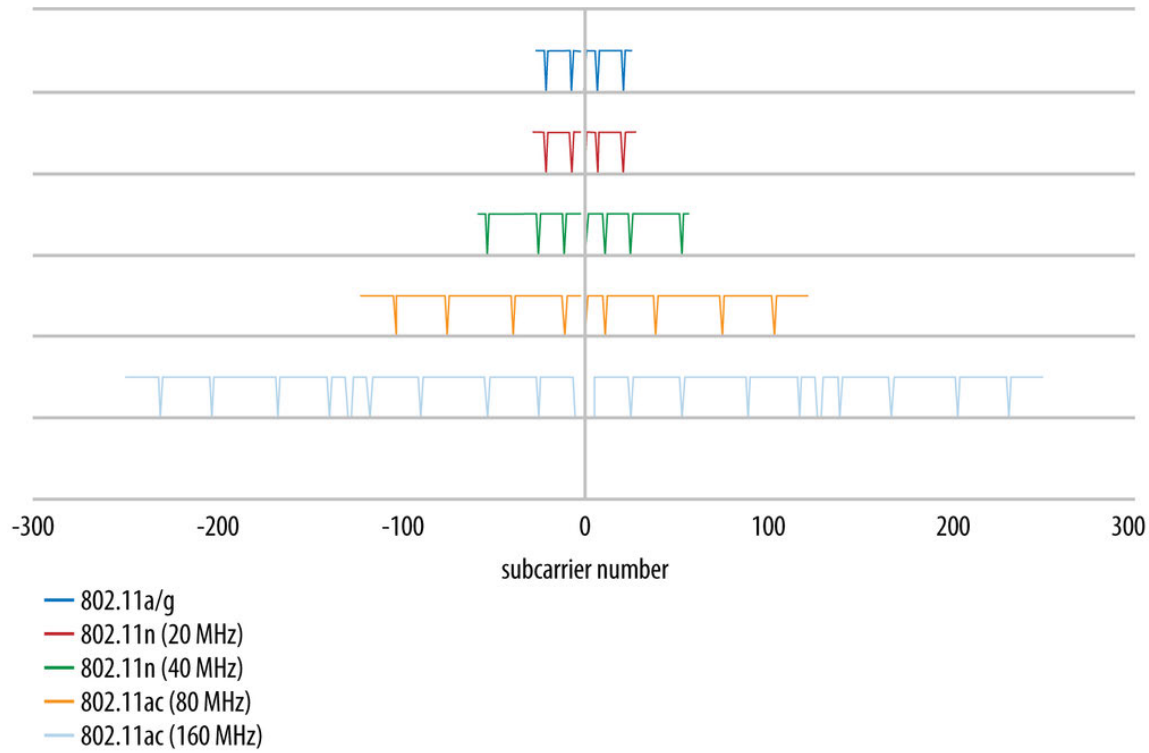


Figure 7. Subcarrier arrangement for 802.11ac (and similar 802.11 variants)

(Gast 2013)

2.3.3 Detection Modelling and Analysis

The choice of modelling techniques in experimental sensing applications varies considerably. Simple models comparing the variance of CSI amplitudes captured in a static environment with those captured when motion is occurring in the signal path have been shown to reliably detect the presence of humans (Tian et al. 2018). Palipana et. Al (2016) used a nonlinear model utilising kernel principal component analysis that yielded high accuracy in detecting humans and was tested successfully in gesture recognition experiments also. Kanda et al. (2022) used a similar method to analyse the magnitude of specific frequency components that are impacted by the chest fluctuations occurring during breathing. Testing demonstrated that respiratory rate could be estimated with an error rate of only 3.5 breaths per minute (Kanda et al. 2022).

Table 1. Common statistical features of CSI used in detection models (Zhang et al. 2020)

Features	Formula and Description
Standard Deviation	$\sigma = \frac{1}{L} \sum_{i=1}^L (CSI_i - \mu)^2$
Mean	$\mu = \frac{1}{L} \sum_{i=1}^L CSI_i$
Max	Maximum Value of CSI_i
Min	Minimum Value of CSI_i
Median	The “middle” value
Range	$\text{range} = CSI_{\max} - CSI_{\min}$
Interquartile Range	$IQR = Q_3 - Q_1$
Skewness	$\text{Skewness} = \frac{\sum_{i=1}^L \frac{(H_i - \mu)^3}{L}}{\sigma^3}$
Kurtosis	$\text{Kurtosis} = \frac{\sum_{i=1}^L \frac{(H_i - \mu)^4}{L}}{\sigma^4}$
Normalised Entropy	$\text{Entropy} = - \sum_{i=1}^L p_i \log_2 p_i$

(Zhang et al. 2020)

Schäfer et al. (2021) constructed multiple CSI extraction systems using different hardware which inputted CSI data into two machine learning algorithms, SVM and LTSM. The objective was to classify human activities such as lying, sitting, walking, falling etc. within the signal path. All the conducted experiments produced classification systems with high accuracy. One of the key challenges faced was pre-processing CSI data before inputting into learning algorithms to remove anomalies. The sources of these anomalies were assigned to either, null or pilot subcarriers and, noise and ambiguities caused by hardware and firmware including automatic gain control, adaptive loading and carrier frequency offset nonlinearity (Schäfer et al. 2021).

Schäfer et al. (2021) used three algorithms to pre-process CSI. The first simply removed CSI from any null or pilot subcarriers as well as any CSI samples that remained unchanged. The second removed CSI samples with significant magnitude outliers using a Hampel filter and the third smoothed noise using a discrete wavelet transform (Schäfer et al. 2021).

Algorithm 2 OUTLIER REMOVAL

Input: $CSI_{med} \leftarrow$ local median of current window ($W_{len} = 3$) of $CSI_{mag}(i)$

Output: Outliers removed as CSI_{ham}

- 1: Compare the current sample (i) with $n_\sigma \times \sigma_i$
 - 2: **if** $CSI_{mag}(i) - CSI_{med}(i) > n_\sigma \times \sigma_i$ **then**
 - 3: $CSI_{mag}(i) = CSI_{med}(i)$
 - 4: **end if**
 - 5: $CSI_{ham} \leftarrow CSI_{mag}(i)$
-

Figure 8. The outlier removal algorithm used by Schäfer et al. (2021)

(Schäfer et al. 2021)

A variety of statistical analysis methods and machine learning techniques seem to be suitable for sensing with CSI as the input and all techniques tested by Schäfer et al.'s (2021) provide capable of detecting the target stimuli with reasonable accuracy (Schäfer et al. 2021). It is highly likely that the analysis technique or sensing algorithm used can be somewhat abstracted from the other parts of a CSI sensing system. Therefore, a CSI sensing system could leverage of any number of the myriad of generalised automated machine learning and data analysis tools available. However, effective pre-processing requires a deeper and specific understating of the nature of CSI and the operation of WiFi equipment and signal processing in general and thus is a more important focus area when developing a sensing system.

2.4 Significance of Previous of Work to a Vermin Detection System

It is reasonably clear that even a very minor physical change to the signal path of a WiFi system can be detected via CSI analysis, validating the potential that vermin can be detected. Although the capability to distinguish between a static signal path and a signal path containing vermin may not ever yield a useful system. The detection would need to be specific enough that vermin could be identified in a non-static environment and most common stimuli like human presence and movement would not trigger a false alarm. One possible solution is to estimate the height of the obstruction of the signal path. Lin et al. (2020) explored this technique in order to prevent false alarms caused by pets in a WiFi based intruder detection system. Estimating height with only a single pair of WiFi devices is difficult but by examining the geometry of the signal fading with a known transmitter and receiver height a technique was developed with reasonable accuracy.

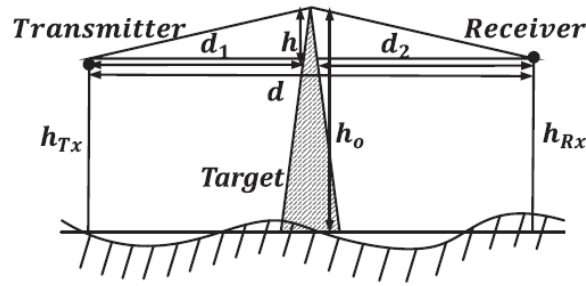


Figure 9. Diagram of signal fading between WiFi devices used for height estimation

(Lin et al. 2020)

Another potential method to distinguish between vermin and humans is to estimate respiratory rate. There is a significant difference between the typical respiratory rate of humans and vermin. A typical respiratory rate for rats is 85 breaths per minute (Ades 2018). The model Kanda et al. (2022) used for human respiration could be able to estimate vermin's respiration rate and earlier work by Wang et al. (2017) and Wang et al. (2020) could provide alternate models that could also be potentially adapted to sense vermin's vital signs.

This project only attempts to determine if passive WiFi sensing can determine when a mouse is in the LOS signal path within a static indoor environment with static antenna placement for the purpose of determining if the concept of passive WiFi vermin sensing is feasible. This is not sufficient to determine if a system could be constructed that could detect vermin in a more useful and generalised sensing system that could be deployed with a typical WiFi network in a realistic scenario. Integrating different sensing models and developing the algorithms to facilitate automated model training as well as addressing challenges like dynamic antenna placement and obstructions outside of the LOS signal path will be beyond the scope of this project.

2.5 Limitations of using CSI for Sensing

While a degree of sensing potential of WiFi systems is proven, all previous works encountered several practical challenges that either limit the capability of the sensing system or must be overcome to make the system function. Some of the key limitations are summarised by Zhang et al. (2022): accessibility, sampling frequency, unsynchronized transceivers, distortion and non-distributed data collection.

Issues with accessibility of CSI data arise because WiFi devices are not designed to make CSI available externally (Zhang et al. 2022). CSI is only handled in low level functions of the device where software and firmware applications are proprietary.

Issues with distortion, sampling and non-distributed data collection can be somewhat overcome in testing by controlling the WiFi network in a manner suitable for sensing. However, these issues are significant challenges

if the eventual goal is to design a system that is abstracted from specific hardware and can operate in parallel with normal communication functions.

2.6 Constructing a Sensing System with WiFi

Regardless of the exact purpose, the general framework and components of most WiFi sensing systems are similar. A system can be deconstructed into three main elements, data collection, signal pre-processing and modelling that then infers the sensing results (Wang et al. 2021).

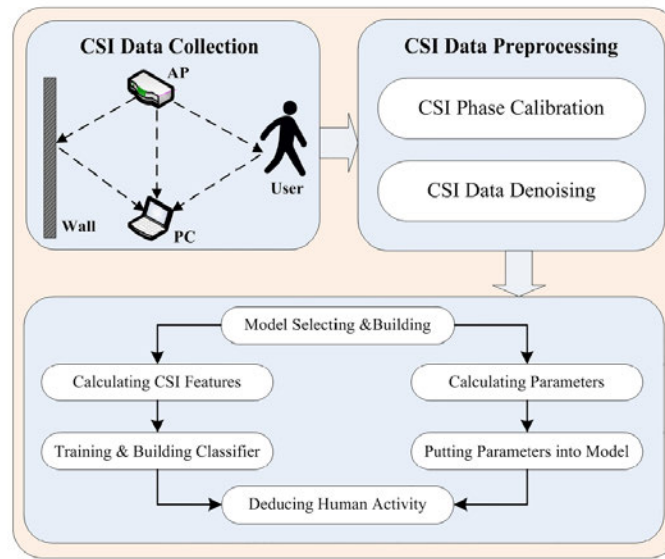


Figure 10. Model of a typical human activity WiFi sensing system

(Wang et al. 2021)

One of the most critical and difficult aspects of the project was constructing a WiFi network capable of determining and logging CSI samples. To be assured that CSI from any readily available, modern, commodity WiFi device can be accessed and logged the software and firmware vendors would need to offer information and assistance. Seeking assistance from vendors and manufacturers is not practical for the project, and in any case willingness to help researches extract CSI parameters is extremely unlikely, as much of the required information may be considered sensitive and generally the systems that would need to be modified are proprietary (Kanda et al. 2022; Yadav et al. 2022).

Open-source tools that extract the CSI parameters from legacy WiFi chipsets are available. The tool documented by Halperin et al. (2011) is the oldest known and has been widely used in research projects. It is implemented via a Linux operating system and can only be used to extract CSI measurements from an Intel WiFi Link 5300 wireless network interface card (NIC).

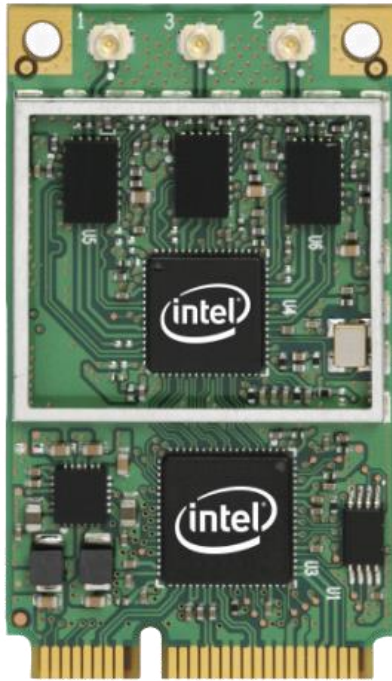


Figure 11. Intel WiFi Link 5300 Wireless Network Interface Card

(Halperin et al. 2011)

Obtaining and installing an Intel 5300 NIC the NIC on a suitable computer that can then be used for testing is within the resources available to the project but could prove challenging. Both the hardware and the operating system needed to implement the tool are legacy, and thus they are becoming more difficult to source and may not interact expectedly with modern equipment. While investigating hardware to procure for the project it was noted that a more modern miniaturisation of the Intel 5300 was all that was readily available for purchase. It could not be verified whether the miniaturised version is compatible with Halperin et al.'s (2011) tools. The seller's notes on several online retailers also stated that the NIC "would not work" with IBM, Lenovo, ThinkPad and Hewlett Packard laptops. Although unverified and non-specific these notes likely indicate challenges in using the NIC with a modern computer.



Figure 12. Intel WiFi Link 5300 Ultimate N Wireless Card Half Mini Pcie 802.11n

Source: <<https://www.amazon.com/Intel-Wifi-Ultimate-Wireless-802-11n/dp/B00B0RRIJ8>>

2.6.1 Survey of WiFi Sensing Research Hardware Utilisation

Many of the published works detailing CSI sensing, testing and experimentation do not provide a detailed specification of all hardware and software used to extract CSI. Since time and funding were significant restrictions to the project, it was not practical to procure enough hardware as well as dedicate time to investigating different options using hardware and trials. A single choice had to be made as to which hardware could be used to design a system capable of logging CSI for sensing. To aide in the selection of hardware, a survey of 20 projects that undertook CSI sensing was conducted. The abbreviated findings are shown in Table 2.

Table 2. Hardware survey of WiFi sensing research

Title	Author	CSI Extraction System Hardware
FIMD: Fine-grained Device-free Motion Detection	(Xiao et al. 2012)	Intel WiFi Link 5300 Wireless Network Interface Card – and generic 802.11m AP
From RSSI to CSI: Indoor Localization via Channel Response	(Yang, Zhou & Liu 2013)	Intel WiFi Link 5300 Wireless Network Interface Card
WiFi-Based Real-Time Calibration-Free Passive Human Motion Detection	(Gong et al. 2015)	Intel WiFi Link 5300 Wireless Network Interface Card - Halperin et al. (2011) tools
Non-Invasive Detection of Moving and Stationary Human With WiFi	(Wu et al. 2015)	Intel WiFi Link 5300 Wireless Network Interface Card - Halperin et al. (2011) tools
Channel State Information Based Human Presence	(Palipana, Agrawal & Pesch 2016)	Intel WiFi Link 5300 Wireless Network Interface Card - Halperin et al. (2011) tools

Detection using Non-linear Techniques		
WiFi-Based Adaptive Indoor Passive Intrusion Detection	(Tian et al. 2018)	Intel WiFi Link 5300 Wireless Network Interface Card - Halperin et al. (2011) tools – implemented with ProBox23 MS-B083 mini PCs
Complex Motion Detection Based on Channel State Information and LSTM-RNN	(Zhang et al. 2020)	Intel WiFi Link 5300 Wireless Network Interface Card - Halperin et al. (2011) tools
Device free human activity and fall recognition using WiFi channel state information (CSI)	(Damodaran et al. 2020)	Intel WiFi Link 5300 Wireless Network Interface Card - Halperin et al. (2011) tools – two Lenovo laptops, Ubuntu version 14.04
Revisiting Indoor Intrusion Detection With WiFi Signals: Do Not Panic Over a Pet!	(Lin et al. 2020)	Intel WiFi Link 5300 Wireless Network Interface Card - Halperin et al. (2011) tools
Recognition, and Detection With Commodity MIMO OFDM WiFi	(He et al. 2020)	Intel WiFi Link 5300 Wireless Network Interface Card - Halperin et al. (2011) tools
WiFi-based Human Activity Recognition using Raspberry Pi	(Forbes, Massie & Craw 2020)	Raspberry Pi 4 , Nexmon Firmware Gringoli et al. (2019) tools
On CSI-Based Vital Sign Monitoring Using Commodity WiFi	(Wang, Yang & Mao 2020)	Intel WiFi Link 5300 Wireless Network Interface Card - Halperin et al. (2011) tools
Passive WiFi Radar for Human Sensing Using a Stand-Alone AP	(Li et al. 2020)	Intel WiFi Link 5300 Wireless Network Interface Card
Eliminating the Barriers: Demystifying Wi-Fi Baseband Design and Introducing the PicoScenes Wi-Fi Sensing Platform	(Jiang et al. 2021)	Qualcomm Atheros AR9300 and Intel WiFi Link 5300 – PicoScenes software – significant low level control of WiFi is utilised, turning adapters it adapters into SDR's
Fire Detection Using Commodity WiFi Devices	(Li et al. 2021)	Raspberry Pi 4 - Nexmon Firmware Gringoli et al. (2019) tools

Human Activity Recognition Using CSI Information with Nexmon	(Schäfer et al. 2021)	Raspberry Pi 3B+ and 4, Asus RT-AC86U. - Nexmon Firmware Gringoli et al. (2019) tools and 802.11ac AP
Vehicle In-Cabin Contactless WiFi Human Sensing	(Ibrahim & Brown 2021)	Raspberry Pi 4, Nexmon Firmware Gringoli et al. (2019) tools – PoE used to power and control Pi's
A Subcarrier Selection Method for Wi-Fi-based Respiration Monitoring using IEEE 802.11ac/ax Protocols	(Wang, R. et al. 2022)	PicoScenes CSI-toolbox – hardware not specified
Indoor Human Fall Detection Algorithm Based on Wireless Sensing	(Wang, C. et al. 2022)	Intel WiFi Link 5300 Wireless Network Interface Card - Halperin et al. (2011) tools
Respiratory Rate Estimation Based on WiFi Frame Capture	(Kanda et al. 2022)	WXR-5700AX7S AP and Intel AX200 – no modified firmware used; sensing model input was the beamforming matrix

Prior to 2020, the Intel WiFi Link 5300 and the tools detailed by Halperin et al. (2011) appear to be the only open-source tool that were available to researchers. More recently other options have emerged that are more suitable for the project. Mainly the Nexmon CSI tools detailed by Gringoli et al. (2019) and the use of the beamforming matrix by Kanda et al. (2022), both of which were considered for use in this project.

2.6.2 The Beamforming Matrix for Sensing

An alternate approach to WiFi sensing that removes the requirement to access CSI is to use the beamforming matrix as the input to the sensing system. The beamforming matrix is transmitted between WiFi devices unencrypted so can be captured and extracted by motoring the traffic in the WiFi network. Kanda et al. (2022) built a successful sensing system that estimates respiratory rate (in humans) using only the beamforming matrix as an input. A few drawbacks are noted though. Significant effort would still be required to create a tool that would extract and parse the beamforming matrix as no premade tools were accessible to the project. However, the required information would be available in the IEEE802.11ac standard where beamforming was first introduced to WiFi (Gast 2013).

It is also very unlikely the beamforming matrix would be as effective as CSI as an input to sensing algorithms. The beamforming matrix is an indirect measure of the environment within the signal path as opposed to CSI which effectively is measuring directly (Kanda et al. 2022). The beamforming matrix is designed to steer the power of a WiFi transmission in the most optimal direction (Gast 2013). Though physically static, the antenna array in MIMO WiFi devices can direct transmissions by controlling the current delivered to each antenna

within the array to create an effect analogous to physically moving a directional antenna such as a Yagi. CSI is used to determine the beamforming matrix, for example, if CSI reveals a significant amplitude decay in the transmission in one direction but not another, the beamforming matrix can be constructed to direct more energy into the more effective signal path (Gast 2013). In Figure 13. Q denotes the beamforming matrix. Future work could involve implementing a sensing model using the beamforming matrix and not requiring the use of modified firmware for WiFi devices, but this project will focus on CSI.

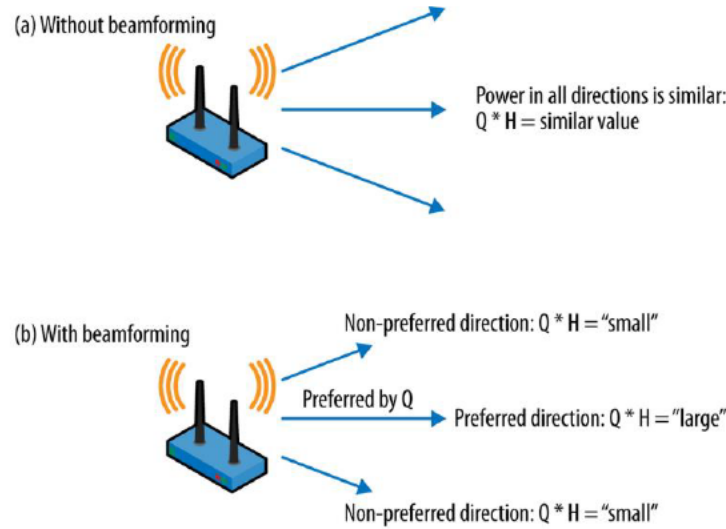


Figure 13. Example of the effect of the beamforming (steering) matrix

(Gast 2013)

2.6.3 The Nexmon CSI Extraction Tools

Four of the more recent projects surveyed utilised a more modern CSI extraction system. In 2019 Gringoli et al. (2019) released a suite of open-source tools that can extract CSI measurements from modern Cypress, Broadcom WiFi NICs. The tools were created utilizing earlier work from Schulz, Wegemer and Hollick (2016) who created a platform that can be used to create C-based firmware modifications to a wide range of common WiFi chipsets. Gringoli et al. (2019) tool's provide access to CSI via several common devices including Nexus smartphones, Raspberry Pi miniature computers and Asus routers.

Table 3. Devices compatible with the Nexmon CSI extraction tools

WiFi Chip	Firmware Version	Used in
bcm4339	6_37_34_43	Nexus 5
bcm43455c0	7_45_189	Raspberry Pi B3+/B4
bcm4358	7_112_300_14_sta	Nexus 6P
bcm4366c0	10_10_122_20	Asus RT-AC86U

(Schulz, Wegemer & Hollick 2016; Gringoli et al. 2019)

CSI is extracted Cartesian form:

$$\mathbf{H}_p = \begin{bmatrix} \text{CSI}_{si}^{11} + \text{CSI}_{si}^{11} & \text{CSI}_{si}^{12} + \text{CSI}_{si}^{12} & \dots & \text{CSI}_{si}^{1N} + \text{CSI}_{si}^{1N} \\ \text{CSI}_{si}^{21} + \text{CSI}_{si}^{21} & \text{CSI}_{si}^{22} + \text{CSI}_{si}^{22} & \dots & \text{CSI}_{si}^{2N} + \text{CSI}_{si}^{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \text{CSI}_{si}^{N1} + \text{CSI}_{si}^{N1} & \text{CSI}_{si}^{N2} + \text{CSI}_{si}^{N2} & \dots & \text{CSI}_{si}^{NN} + \text{CSI}_{si}^{NN} \end{bmatrix}$$

Where si is the index of the subcarrier, \mathbf{H} is the CSI matrix, P is the captured packet number and N is the number of antennas.

The tools also enable CSI to be extracted from VHT 80MHz IEEE802.11ac bandwidth channels and provide greater resolution (10-14 bit as opposed to 8 bit) than the tools created by Halperin et al. (2011) designed for use in the Intel 5300 NIC (Halperin et al. 2011; Gringoli et al. 2019). As demonstrated by Tan, Zhang & Yang (2018) wider channel bandwidth can be an extremely important element to sensing systems and the increased resolution of the Nexmon tools in comparison to the earlier tools will also provide better granularity of changes in the signal path. The WiFi NICs Gringoli et al.'s (2019) Nexmon tools operate with are also modern and utilise the WiFi the IEEE802.11ac that is typical of many currently in use commodity WiFi devices as opposed to IEEE802.11n which is now a legacy standard.

Importantly procuring and operating Raspberry Pi miniature computers was within the scope of time and funding available to the project. Raspberry Pi models 3B+ and 4B provide a complete hardware platform with the Broadcom bcm43455c0 WiFi NIC as a standard inclusion. The standard hardware platform makes implementing an operating system compatible with the Nexmon firmware that can extract CSI straightforward forward, as all available standard operating systems and Linux kernel versions can be downloaded as disc images from Raspberry Pi's website.

One potential limitation of using a Raspberry Pi and Broadcom bcm43455c0 as the input to a CSI based sensing system is that the WiFi chip only contains a single antenna and only a single spatial stream is handled at a time (Gringoli et al. 2019). It is assumed that this will limit the sensing capability in comparison to a device with multiple antennas. The extracted CSI will be in the form of vector as the dimensions of \mathbf{H} , reduce proportional to the number of antennas. The WiFi components of the Raspberry Pi 3B+ and 4 including the resonant cavity antenna are located within a metallised can stamped with the Raspberry Pi insignia as shown in Figure 14.

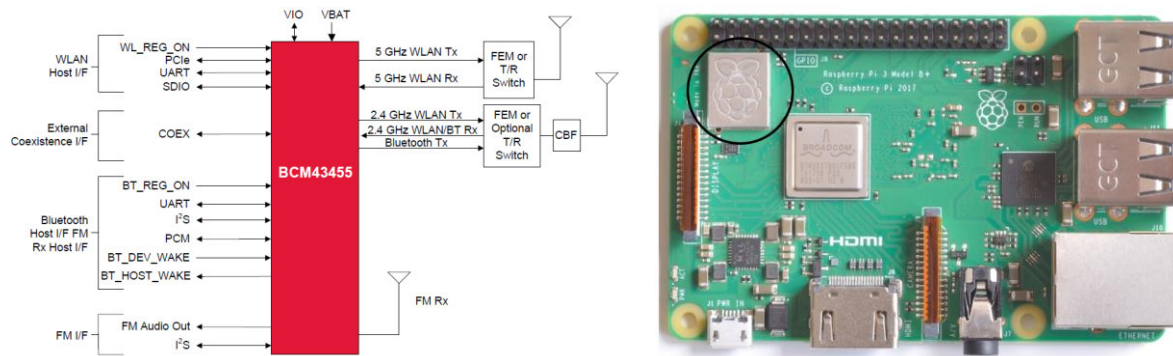


Figure 14. Schematic of Broadcom BCM43455 (Left) and Raspberry Pi 3B+ with WiFi components circled (Right)

(Preliminary Data Sheet BCM43455 2016; Raspberry Pi Documentation 2023)

3. Methodology

3.1 Planning

The most meaningful way to test and develop a sensing system was to test with live vermin. Testing with mice was the most practical as a supporter of the project owns domesticated pet mice which can be used as stimuli. The use of animal testing in research is permissible but must be undertaken in accordance with the Australian Code for the Care and use of Animals for Scientific Purposes 8th Edition 2013 to ensure any use of animals is ethical, humane and the animals are treated responsibly. Adherence to the University of Southern Queensland's Animal Wellbeing and Ethics Policy is also a requirement and ensures obligations under Australian code are met.

3.1.1 Animal Testing and Ethics

The University of Southern Queensland's Animal Wellbeing and Ethics Policy requires researchers to seek approval prior to any project that involves animals (UniSQ 2022). There are two mechanisms to seek approval, completion of the Animal Ethics Research Application Form and seeking a formal exemption from the Animal Ethics committee. A formal exemption was requested as the project could be completed in accordance with two key criteria:

- no interference with animals
- no abnormal disruption of habitat

(UniSQ 2022)

Testing required a mouse was located in the signal path between a pair of transceivers in the WiFi network which CSI was being extracted from. The WiFi devices are small and portable so were positioned as required

while the mouse remained within its usual habitat. Since the mouse is a pet, it is considered to be within its habitat provided it is not placed in unfamiliar enclosure or moved to a different location for the purpose of testing. As the mouse is kept indoors in a typical (for North Queensland, Australia) domestic dwelling and is accustomed to being in close proximity to people and everyday household objects such as WiFi devices this was achievable without interfering with the mouse and without causing abnormal disruption of its habitat. On this basis, application ETH2023-0118 was submitted to the University of Southern Queensland Animal Ethics Committee in March 2023 and was approved based on negligible or risk exempt status. A copy of the application and approval are included in Appendix C.

3.1.2 Consequences

The project is intended to assess the feasibility of, as well as provide some groundwork to the development of a system that can provide an additional utility function to a typical WiFi network. It is hoped that the results will determine if vermin can be detected by CSI and that the concept of detecting vermin passively in a system utilising WiFi equipment is potentially feasible. By detailing the configuration and arrangement of testing equipment, the testing process and collecting CSI data, it is expected that the project will provide insight for other researchers to continue WiFi sensing testing and inspire further idea generation for the targets of WiFi sensing systems.

The benefits of additional utility functions being deployed into WiFi networks include improved quality of life via convenience and a potential reduction in E-waste and the manufacture of single purpose devices. It is also possible that the specific sensing function being examined in this project may help contribute to the control measures already used against vermin (further detail of these consequences can be found in Section 1).

3.1.3 Risk Management

Most of the works that contributed to the project required programming and word processing undertaken on a desktop computer. While this is an extremely low risk activity the considerable amount of time spent on these activities did justify the employment of control measures such as:

- Monitors located at arm's length and eye level.
- Keyboard and mouse on flat surface at least 10cm from the edge of the desk.
- Wireless mouse and keyboard that can be repositioned for best possible ergonomics based the monitor in use.
- Adjustable ergonomic chair with footrest adjusted so that hips and knees are level.
- Timer to remind of posture change every 30 minutes as well as breaks.

(WorkSafe 2020)

Configuring and operating the WiFi equipment during CSI capture was the highest risk activity undertaken during the project. Tasks requiring manual handling and working with electrical equipment were undertaken


which required control measures be implemented to mitigate risk. Control measures included substituting low voltage electrical equipment where possible and inspecting electrical equipment before use when substitution was not possible and wearing non-slip footwear as well as housekeeping around the test area to prevent slips, trips and falls. Risk Management Plan – 2122, was submitted to University of Southern Queensland in March 2023 and will be employed during the CSI capture phase of the project. It Is located in Appendix B.



3.1.4 Project Resources

The key components needed to complete the project are the equipment required to capture and log CSI during testing and software tools capable of performing the CSI processing required as well as testing and implementing the detection method. MATLAB is a powerful generic tool that was used for analysis. MATLAB was already known to the student working on the project, is ubiquitous and has been employed in similar projects successfully including the work by Zhang et al. (2020) and Wang R. et al. (2022).

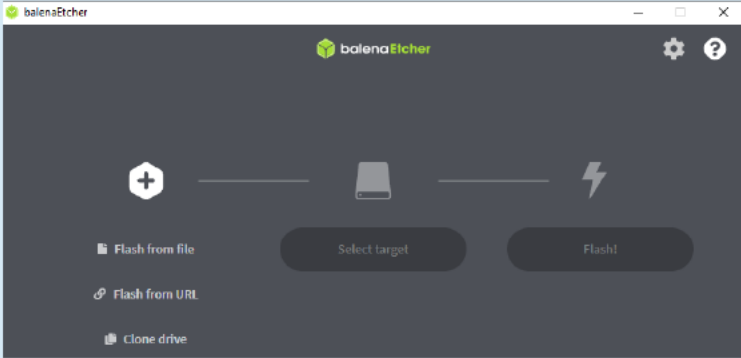
Based on the survey of hardware utilisation in CSI sensing research (see Table 2.) and in particular the issues noted with utilising an Intel 5300 WiFi NIC, Raspberry Pi miniature computers were used to extract CSI using the firmware and tools created by Gringoli et al. (2019). A detailed list of all equipment and software used in the project is included in Table 3. below.

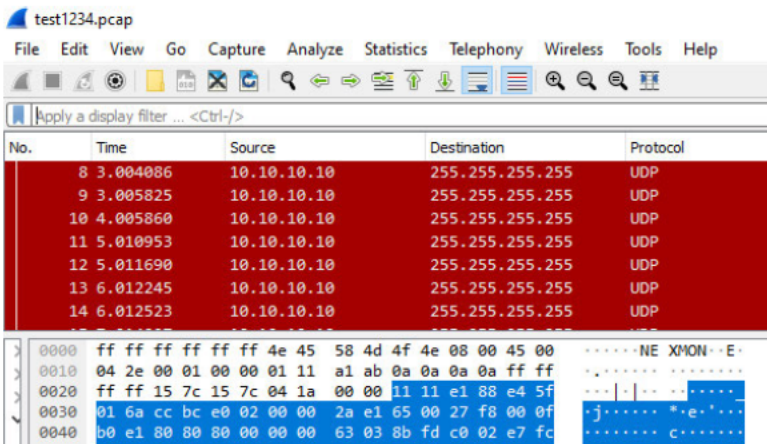
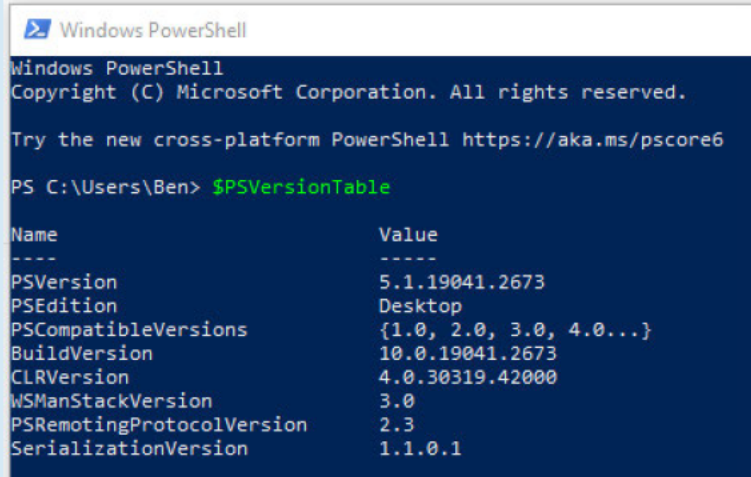
Table 3. Project resource requirements

Resource	Source	Purpose	Comments and Images
Raspberry Pi 3 model 3B+	Purchased online – second hand	CSI Extraction	<p>Contains Cypress Broadcom BCM43455c0 WiFi NIC and is compatible with the CSI extraction tools.</p> <p>Purchased with case to prevent damage when being positioned to set up signal path for testing.</p> 

Raspberry Pi 4 model B	Purchased online	CSI Extraction	<p>Contains Cypress Broadcom BCM43455c0 WiFi NIC and is compatible with the CSI extraction tools.</p> <p>Variant B07TC2BK1X – 4GB RAM, WiFi</p> <p>Purchased with case to prevent damage when being positioned to set up signal path for testing.</p> 
Generic IEEE802.11ac WiFi Access Point - TP-Link AC750 Wireless Router Dual Band	Purchased	Create special purpose WiFi network for CSI extraction	<p>Feature rich legacy model which allows high degree of control and configuration via in-built webserver, same WiFi generation (802.11ac) as Raspberry Pi's. TPLink model chosen specifically as only vendor tested that allowed for control of frequency hopping.</p> 
Micro SD Cards and USB A Adapters	Purchased	Hard drive for Pi's – Imaged with operating system	<p>Raspberry Pi and ScanDisk Ultra type – Scan Disk Ultra recommended as cost effective micro SD cards suitable for use in Pi's (Fromaget 2020).</p>

			
Sundry Hardware Items for Operating Test Equipment	Miscellaneous	Interfacing with equipment and transferring CSI data	<p>Sundry items needed to interface with and position CSI extraction equipment as well as transfer CSI data from Pi to computer running MATLAB, Including: USB thumb drive, mini HDMI adapter, desk and stands, ethernet patch leads, power supplies mouse, keyboard, monitor etc.</p> 
MATLAB Software	University of Southern Queensland License	CSI analysis and testing and implementation of sensing algorithms	<p>Version 2022b build 9.13.0.2193358, licence 40904778, Neural Network Toolbox used in sensing system, all native functions used for handling CSI are built-in and require no additional licencing</p> <p>Download URL: https://au.mathworks.com/products/matlab/student </p> 
BelnaEtcher Software	License Free Version	Flashing micro SD cards with	<p>Version 1.18.4</p> <p>Download URL:</p>

		Raspberry Pi operating system	https://etcher.balena.io/ 
Raspberry Pi Operating System Images	Available License Free from Raspberry Pi	Operating system for Raspberry Pi	<p>For use with CSI Extraction Tools: raspbios_lite_armhf-2022-01-28 Linux Kernel Version: 5.10.92-v7+</p> <p>For use as Tx WiFi device: raspbios_armhf-2023-05-03 Linux Kernel Version: 6.1.21-v8+</p> <p>Download URL: https://www.raspberrypi.com/software/operating-systems/</p>
CSI Extraction Tools	Open Source Tools	Special Purpose Firmware to Extract CSI Data	<p>CSI tools developed by Gringoli et al. (2019) using firmware patching tools created by Schulz, Wegemer and Hollick (2016).</p> <p>The precompiled installation version for Linux kernel version: 5.10.92-v7+ is used, along with the script to ingest CSI data into MATLAB, the original published article as well as the support guides and troubleshooting forums are referred to.</p> <p>Download URL: https://github.com/nexmonster/nexmon_csi/tree/master</p>
Wireshark Software	Free License	Examine Raw CSI Samples and Troubleshoot	<p>Version: 4.0.4 (v4.0.4-0-gea14d468d9ca).</p> <p>Network Analyser tool that can interpret .pcap files (the format CSI is captured in).</p> <p>Raw packets captured from the CSI extraction can be examined using Wireshark to verify or troubleshoot parsing to MATLAB.</p>

			<p>Download URL:</p> <p>https://www.wireshark.org/download.html</p> 
Windows PowerShell Software	Standard Application on Windows	Remote control of the CSI extraction Pi	<p>Remove the need to have a monitor mouse and keyboard connected to the PI during testing.</p> 

3.2 Design of WiFi Network to Capture CSI

To complete testing and generate CSI data for analysis a WiFi network had to be constructed that can capture and log CSI data, be isolated from other WiFi networks and unwanted sources of EMI and controlled to allow the capture of CSI with known parameters. The optimisation of the design was limited by the time, resources and skillset of the undergraduate student researcher working on the project, so some aspects of the design were only optimised and implemented as far as was reasonably practicable within the scope of the project.

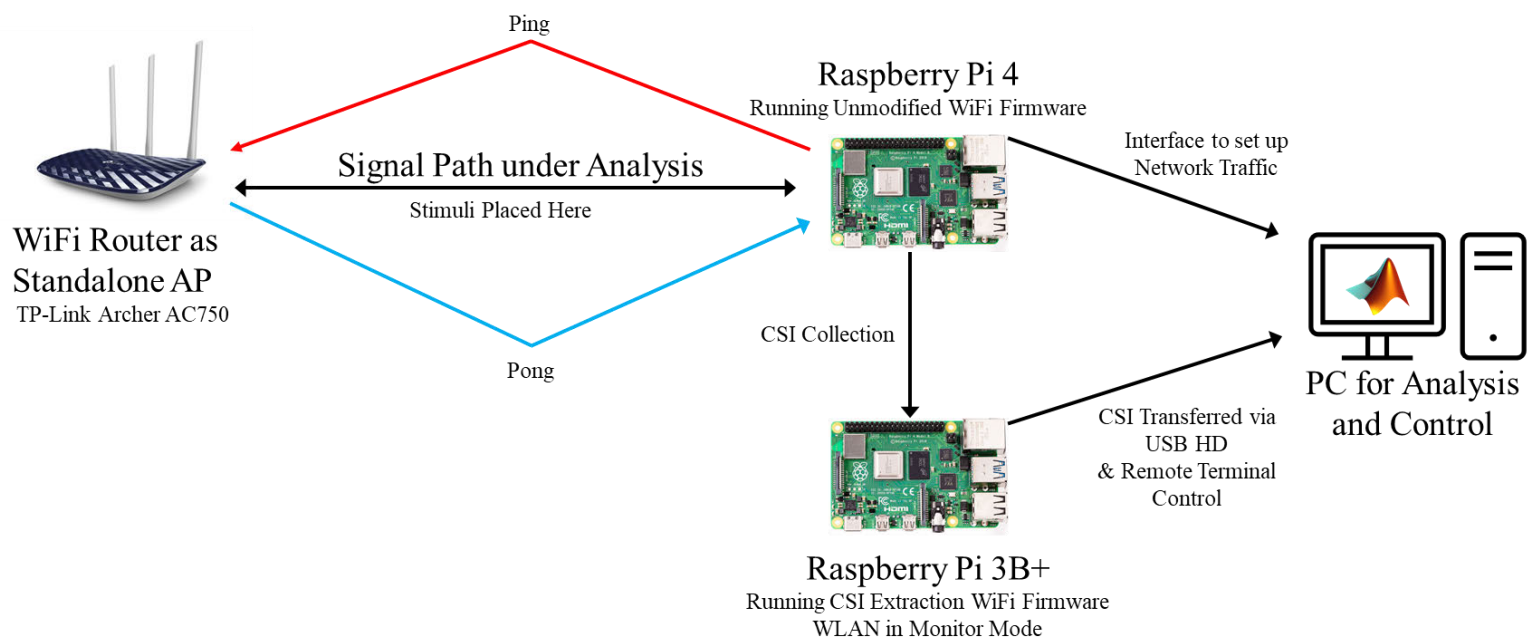


Figure 15. Testing equipment arrangement

The design and arrangement of the testing equipment drew on elements of the tests documented by Forbes, Massie & Craw (2020), Ibrahim & Brown (2021), Li et al. (2021) and Schäfer et al. (2021). A standalone WiFi network is created by configuring the router to act as an AP but not connect to any other networks. The Raspberry Pi 4 then joins this network hosted by the AP and traffic is generated between the Pi and the router. While tools exist that can enable frame injection with a high degree of control and that can turn the WiFi NIC in the Pi into a primitive form of SDR, such tools are complex to implement and not realistic simulations of network traffic that would be present in typical WiFi network. Simply pinging the router proved sufficient in order to generate traffic for proof-of-concept testing. If the system was ever to operate in parallel with normal communication functions, traffic generated solely for sensing needs to be sparse enough to not overwhelm communications traffic.

The Raspberry Pi 3B+ operates the Nexmon CSI extraction tools created by Schulz, Wegemer & Hollick (2016) and Gringoli et al. (2019). The WiFi NIC is placed into monitor mode by the tools, so will capture all traffic accessible and determine the CSI of each packet based on the method implemented by Gringoli et al. (2019). Figure 16. depicts a high-level flowchart of the WiFi NIC firmware with the modifications made to extract CSI shown in bold typeface. The tools also contain filtering functions to ensure that only desired traffic from the Pi 4 has the CSI logged for analysis. While it would be possible to extract CSI from the router as well, there is some ambiguity in relation to how CSI is resolved when the number of antennas, cores and spatial streams are not aligned between the transmit devices and the CSI extraction device (Link et al. 2019). By extracting CSI from another Pi with identical WiFi hardware any ambiguity caused by transceiver mismatch is avoided with the exception that it is not possible to determine which antenna on the router the Pi 4's antenna is paired with. The

filtered CSI can then be captured and stored to USB drive to be transferred to the computer where the analysis will be undertaken.

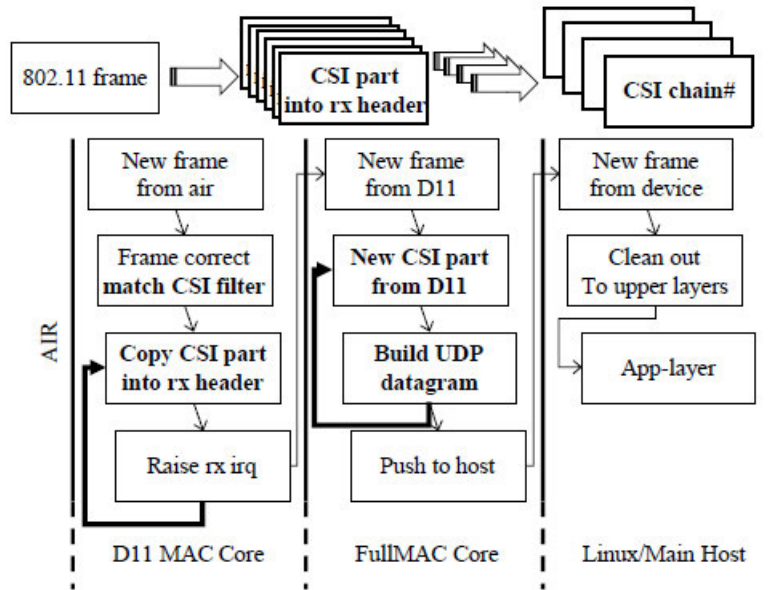


Figure 16. Flow chart of WiFi NIC firmware for CSI extraction

(Gringoli et al. 2019)

3.2.1 Setting up Raspberry Pi for CSI Extraction

The lite version of the Raspberry Pi operating system version was selected for use with the modified firmware patches to extract CSI as the minimal operating system is less likely to perform automated functions without deliberate user input which may interfere with the CSI extraction. There are also far fewer instances of installation and usage issues reported as opposed to when using the full Raspberry Pi operating system (Link et al. 2019).

An installation bundle with precompiled binaries can be used to make installation of the firmware more straightforward. The installation bundle used in the project requires Linux kernel version 5.10.92 which is the kernel version of the January 2022 release of the Raspberry Pi lite operating system. The disk image of the operating system is downloaded from the Raspberry Pi website (refer Table 3.) and flashed to a 16GB micro-SD card using BelnaEtcher. It is not recommended to use Raspberry Pi's official flashing tool as it doesn't allow selection of specific legacy operating systems.

Once the SD-card is inserted in the Pi it can be powered on and then must be connected to the internet. Since there is no graphical user interface there is little value in connecting a monitor to the Pi during CSI extraction, but a monitor is required for initial set up. Using the *raspi-config* command in the root group (root group via: *sudo su*) the software configuration tool can be accessed and used to set the time zone and local area network region as well as the configuration needed to establish a connection to the internet i.e. SSID and password. To

control the Pi from a remote computer and remove the need for a monitor, mouse and keyboard to be connected during testing it is also necessary to enable secure shell in the interface options.

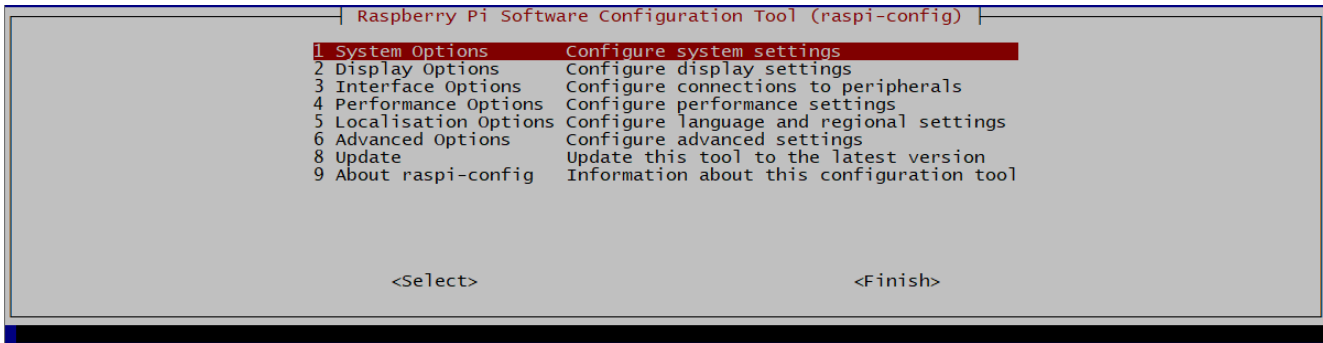


Figure 16. The Raspberry Pi configuration tool

Once the secure shell interface is enabled, the Pi can be interfaced with via a remote windows computer using PowerShell. An Ethernet connection and valid network settings must be established between the windows computer and the Pi. If the username and IP settings are left as default in the Pi, a Linux terminal can be accessed in the Pi via *ssh pi@169.254.217.1* in PowerShell. The address of the Pi can be confirmed via *hostname -I* before attempting to connect via PowerShell if required. A wireless LAN connection cannot be used as the network connection for the remote terminal as the CSI extraction tools will disable normal wireless LAN operation in the Pi and attempting to connect to WiFi network will interfere with the CSI extraction tools.

To store and transfer the CSI data captured to another computer and parse into MATLAB a USB thumb drive must also be mounted to the Pi. The lite operating system will not do this automatically once a drive is inserted. A directory must be created to mount the drive using the *mkdir /mnt/* command. Then, once the drives partition ID is known it can be configured to mount automatically by parsing the drives details as arguments to the *sudo nano /etc/fstab* command. After the drive is mounted automatically, it is important to note that the USB drive must be inserted every time the Pi is booted. The computer used for analysis is utilising a windows operating system so the USB drive must be formatted using a file system common to both Windows and Linux. NTFS was used in the project.

Figure 17. shows a connection via PowerShell to the Pi being established and some basic checks. The *df -h* command is run to verify the USB is mounted. The 8GB ScanDisk USB is mounted at *"/dev/sda1"* and labelled *"usbhdd"*. The USB directory is then opened and all files are listed using the *ls -a* command showing four .pcap files *"output1"*, *"output"*, *"test1234"* and *"test2504"* which contain CSI captures performed during testing and design of the CSI extraction system used in the project.


```
pi@raspberrypi: ~
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Ben> ssh pi@169.254.217.1
pi@169.254.217.1's password:
Linux raspberrypi 5.10.92-v7+ #1514 SMP Mon Jan 17 17:36:39 GMT 2022 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Apr 12 20:55:49 2023 from 169.254.18.120

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set a new password.

pi@raspberrypi:~$ sudo su
root@raspberrypi:/home/pi# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root        15G   1.6G   13G  12% /
devtmpfs         333M    0   333M    0% /dev
tmpfs            462M    0   462M    0% /dev/shm
tmpfs            185M   684K   184M    1% /run
tmpfs            5.0M   4.0K   5.0M    1% /run/lock
/dev/mmcblk0p1   253M   49M   204M   20% /boot
/dev/sdal        7.5G   24M   7.5G    1% /mnt/usbhdd
tmpfs            93M    0    93M    0% /run/user/1000
root@raspberrypi:/home/pi# cd /mnt/usbhdd
root@raspberrypi:/mnt/usbhdd# ls -la
.      ..      output1.pcap  output.pcap  'System Volume Information'  test1234.pcap  test2504.pcap
root@raspberrypi:/mnt/usbhdd#
```

Figure 17. Accessing the Pi and checking USB drive mounting

It is important to complete all set-up tasks, in particular the tasks requiring connection to the internet before attempting to install the CSI extraction tools. Connecting to the internet is not possible once the tools are running and it is important to never update any of the Pi operating system components, even if prompted, as the CSI tools are version sensitive. To install the tools enter the root environment via *sudo su* and input: *curl -fsSL https://raw.githubusercontent.com/nexmonster/nexmon_csi_bin/main/install.sh | sudo bash* into the terminal.

The tools take several minutes to install, and it is important to observe the terminal for any error messages during installation. Once installed avoid making any changes to the Pi's wireless LAN interface to ensure the tools remain operational.

3.2.2 Operating the Extraction Tools

The CSI tools are operated via the Pi 3B+'s remote command line. The Pi can be positioned as required for testing and is reasonably robust and portable while within a case but requires connections to the power supply (right Figure 18.), the USB HD (top left Figure 18.) used to transfer CSI data and the Ethernet connection (top right Figure 18.) to the computer hosting the remote command line. The position of the Pi 3B+ does not impact the signal path being analysed in a CSI capture as it is extracting the CSI from the Pi 4. To execute a CSI extraction the Pi 3B+'s WiFi NIC needs to intercept the packets from the Pi 4 reliably, so must be in proximity. Within 4 or 5 meters is recommended, to ensure a reliable WiFi signal.



Figure 18. Raspberry Pi 3B+ with required connections for CSI capture

To capture and log CSI from a specific device the channel and bandwidth of the WiFi network must be known and the MAC address of the device from which CSI is going to be captured from. The WiFi NIC's MAC address of the Raspberry Pi 4 is: E4:5F:01:6A:CC:BC. The MAC address can be displayed by using the output of the *ifconfig* command and locating the details of the WiFi NIC in the output. The channel and bandwidth of the network can be set via the AP. In the project WiFi channel parameters are configured via the TP-Link webserver, accessed by connecting to the routers IPv4 address via a web browser.

Using the Raspberry Pi 3B+ terminal the command to configure and run CSI extraction can then be inputted. The *mcp* (makecsiparams) command configures the extraction. The arguments passed to *mcp* define the parameters of the CSI extraction. A list of arguments is included in Table 4. Not all possible arguments must be passed to *mcp* as any not included will revert to default values.

Table 4. Arguments for the *mcp* Command

Argument	Function	Description
-h	help	print this message - list of arguments
-e	on/off	enable/disable CSI collection (0 = disable, default is 1)
-c	chanspec	Channel specification <channel>/<bandwidth>
-C	coremask	bitmask with cores where to activate capture (e.g., 0x5 = 0b0101 set core 0 and 2)
-N	nssmask	bitmask with spatial streams to capture (e.g., 0x7 = 0b0111 capture first 3 ss)
-m	addr	filter on this source mac address (up to four, comma separated)
-b	byte	filter frames starting with byte
-d	delay	really needed for 3x4, 4x3 and 4x4 configurations, without it is enforced automatically
-r		generate raw output (no base64)

The output of the *mcp* command will be a base64 encoded parameter string formatted which can be parsed to the CSI extractor (Gringoli et al. 2019). The *nexutil* (Nexmon utility) command then initialises the extractor. The parameter string outputted by *mcp* is parsed to *nexutil* as a custom argument. Values of other arguments to *nexutil* remained static throughout the project and are intended for use with other firmware patches implemented using Schulz, Wegemer & Hollick's (2016) tools. A full list of all arguments and options is shown in Figure 19. The CSI samples can then be logged using the *tcpdump* command as they will be collected in the form of UDP packets originating from IP address 10.10.10.10 destined for 255.255.255.255 on port 5500 (Gringoli et al. 2019).

```

root@raspberrypi:/home/pi# nexutil --help
Usage: nexutil [OPTION...]
nexutil -- a program to control a nexmon firmware for broadcom chips.

-b, --custom-cmd-value-base64  Define that custom-cmd-value should be
                                interpreted as base64 string
-B, --broadcast-ip=CHAR        Broadcast IP to use for UDP tunneling (default:
                                192.168.222.255)
-c, --scansuppress[=INT]       Set/Get scan suppress setting to avoid scanning
-d, --disassociate             Disassociate from access point
-g, --get-custom-cmd=INT       Get custom command, e.g. 107 for WLC_GET_VAR
-i, --custom-cmd-value-int     Define that custom-cmd-value should be interpreted
                                as integer
-I, --interface-name=CHAR      Set interface name (default: wlan0)
-k, --chanspec[=CHAR/INT]      Set chanspec either as integer (e.g., 0x1001, set
                                -i) or as string (e.g., 64/80).
-l, --custom-cmd-buf-len=INT   Custom command buffer length (default: 4)
-m, --monitor[=INT]           Set/Get monitor mode
-o, --dump-objmem=INT          Dumps objmem at addr INT
-p, --promisc[=INT]           Set/Get promiscuous mode
-r, --raw-output               Write raw output to stdout instead of hex dumping
-R, --base64-output            Write base64 encoded strings to stdout instead of
                                hex dumping
-s, --set-custom-cmd=INT       Set custom command, e.g. 108 for WLC_SET_VAR
-v, --custom-cmd-value=CHAR/INT
                                Initialization value for the buffer used by custom
                                command
-V, --revinfo                  Dump revision information of the Wi-Fi chip
-w, --dump-wl_cnt              Dump WL counters
-x, --security-cookie[=INT]    Set/Get security cookie
-X, --use-udp-tunneling=INT     Use UDP tunneling with security cookie INT
-?, --help                     Give this help list
--usage                        Give a short usage message
--version                      Print program version

```

Figure 19. Help view of the Nexmon utility

3.2.3 CSI Capture Operating Instructions and Example

A capture of 1000 CSI samples from the Raspberry Pi 4 on WiFi Channel 157 – 5785MHz, with 20MHz bandwidth was undertaken to test the CSI extraction system and detail the operation. The TP-Link router is configured so that it will only operate a single 5GHz network in the IEEE 802.11ac mode via the TP-Link webserver. The routers IP Address was set to 192.168.1.1. The selection of IP Address is not significant to the RF parameters of the CSI extraction but is used to access the TP-Link webserver and to generate traffic between the router and the Pi 4. Since the router facilitates dual-band operation, the 2.4GHz IEEE 802.11b/g/n functions are disabled to ensure there can be no unintentional connection and minimise sources of interference. This is configured by: Wireless 2.4GHz > Wireless > select Disable Option – as shown in Figure 20.

tp-link AC750 Wireless Dual Band Router
Model No. Archer C20

Status
Quick Setup
Operation Mode
Network
Dual Band Selection
Wireless 2.4GHz
- Basic Settings
- WPS
- Wireless Security
- Wireless MAC Filtering
- Wireless Advanced
- Wireless Statistics
Wireless 5GHz
Guest Network

Wireless Settings(2.4GHz)

Wireless: ☐ Enable ☒ Disable

Wireless Network Name: (Also called SSID)

Mode:

Channel:

Channel Width:

☒ Enable SSID Broadcast

Figure 20. TP-Link webserver – disabling the 2.4GHz IEEE 802.11b/g/n network

The 5GHz 802.11 a/n/ac networking functions are then enabled and set to operate via a fixed channel 157 and channel width 20MHz via: Wireless 5GHz > Wireless > select Enable Option, Channel 157 and Channel Width 20Mhz – as shown in Figure 21.

The network is named “CSI Test” for easy identification when connecting to the Pi 4. Since the Pi 4’s Broadcom WiFi NIC is compliant with IEEE 802.11ac the TP-Link router and the Pi 4 will utilise IEEE 802.11ac during their communication session (Ward 2012; *Preliminary Data Sheet BCM43455* 2016).

tp-link AC750 Wireless Dual Band Router
Model No. Archer C20

Status
Quick Setup
Operation Mode
Network
Dual Band Selection
Wireless 2.4GHz
Wireless 5GHz
- Basic Settings
- WPS
- Wireless Security
- Wireless MAC Filtering
- Wireless Advanced
- Wireless Statistics
Guest Network

Wireless Settings(5GHz)

Wireless 5GHz: ☒ Enable ☐ Disable

Wireless Network Name: (Also called SSID)

Mode:

Channel:

Channel Width:

☒ Enable SSID Broadcast

Figure 21. TP-Link webserver, the 5GHz IEEE 802.11ac settings for test extraction

The Raspberry Pi 4 needs to be connected to the WiFi network hosted by the TP-Link router and then generate wireless traffic so that the Pi 3B+ can record and log the CSI. Using the Pi operating system, the Pi 4 can be

connected to the WiFi network by selecting the “CSI Test” network in the WiFi control palette in the right-hand corner of Pi 4 desktop. Traffic is then generated using the *ping* command to create a communication session between the Pi 4 and the router. The size and transmission frequency of packets can be controlled using the arguments of the ping command. For sensing applications, a high frequency of transmissions will provide more data enabling a higher chance of effective detection. It important to note that only the root user in the Pi 4 can initiate flood pings and pings set to intervals less than 200ms.

Table 5. Arguments for the *ping* Command (incomplete list relevant commands only)

Argument	Function	Description
-f	flood ping	for every ECHO_REQUEST sent a period "." is printed, while for ever ECHO_REPLY received a backspace is printed. This provides a rapid display of how many packets are being dropped. If interval is not given, it sets interval to zero and outputs packets as fast as they come back or one hundred times per second, whichever is more. Only the super-user may use this option with zero interval.
-i	interval	wait interval seconds between sending each packet. The default is to wait for one second between each packet normally, or not to wait in flood mode. Only super-user may set interval to values less 0.2 seconds.
-s	packetsize	specifies the number of data bytes to be sent. The default is 56, which translates into 64 ICMP data bytes when combined with the 8 bytes of ICMP header data.

(Anderson 2006)

```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ sudo su
root@raspberrypi:/home/pi# ping -i0.01 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=6.29 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=1.19 ms

```

Figure 22. Pinging the TP-Link router via the Raspberry 4 with 10ms ping interval

Once the Pi 4 and router are configured to create the wireless network that will generate the CSI, the Pi 3B+ must be configured to extract and log the CSI. The inputs to the terminal of the Raspberry Pi 3B+ are detailed below:

1. Input:

pi@raspberrypi:~ \$ sudo su

Comments:

Enter the root environment.

2. Input:

```
root@raspberrypi:/home/pi# mcp -C 1 -N 1 -c 157/20 -m E4:5F:01:6A:CC:BC -b0x88
```

Comments:

Configure CSI extraction to extract from a single core and spatial stream (all that is possible with a Pi) on channel 157 with 20MHz bandwidth. Filtering for MAC address E4:5F:01:6A:CC:BC, the address of the Pi 4 for frames starting with 0x88. Frames starting with 0x88 are QoS frames including traffic generated by pings. This creates a method of filtering frames that are specifically generated for testing from other unwanted traffic.

Output:

```
ndABEQGIAQDkXwFqzLwAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA==
```

3. Input:

```
root@raspberrypi:/home/pi# ifconfig wlan0 up
```

Comments:

Turn on the WiFi NIC.

4. Input:

```
root@raspberrypi:/home/pi# nexutil -lwlan0 -s500 -b -l34 -  
vndABEQGIAQDkXwFqzLwAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA==
```

Comments:

Configure the extractor to extract from the WiFi NIC (wlan0) by passing the parameter string generated. The other arguments are custom values which remain static regardless of configuration.

5. Input

```
root@raspberrypi:/home/pi# iw dev wlan0 interface add mon0 type monitor
```

Comments:

Enable monitor mode on the WiFi NIC.

6. Input:

```
root@raspberrypi:/home/pi# ip link set mon0 up
```

Comments:

Turn on monitor mode.

7. Input:

```
root@raspberrypi:/home/pi# tcpdump -i wlan0 dst port 5500 -vv -w prtest.pcap -c 1000
```

Comments:

Capture packets from the WiFi NIC for destination port 5500. Save to file “prtest”. Stop logging at 1000 samples. The contents of prtest.pcap will be 1000 CSI samples.

```

pi@raspberrypi:~$ sudo su
root@raspberrypi:/home/pi# mcp -C 1 -N 1 -c 157/20 -m E4:5F:01:6A:CC:BC -b0x88
ndABEQGIAQDkXwFqzLwAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA==
root@raspberrypi:/home/pi# ifconfig wlan0 up
root@raspberrypi:/home/pi# nexutil -Iwlan0 -s500 -b -134 -vndABEQGIAQDkXwFqzLwAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA==
root@raspberrypi:/home/pi# iw dev wlan0 interface add mon0 type monitor
root@raspberrypi:/home/pi# ip link set mon0 up
root@raspberrypi:/home/pi# tcpdump -i wlan0 dst port 5500 -vv -w prtest.pcap -c 1000
tcpdump: listening on wlan0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
1000 packets captured
1038 packets received by filter
0 packets dropped by kernel
root@raspberrypi:/home/pi#

```

Figure 23. Remote Terminal of Pi 3B+ During CSI Capture

During the test traffic was generated by the Pi 4 by pinging the TPLink router using the standard packet length of 32 bytes at a frequency of 10ms via the command *ping -i 0.01 192.168.1..* The captured packets were copied to the USB HD formatted as a packet capture .pcap file. The format of the payload of the .pcap file is provided in the user guides for the Nexmon CSI tools and is detailed in Table 6.

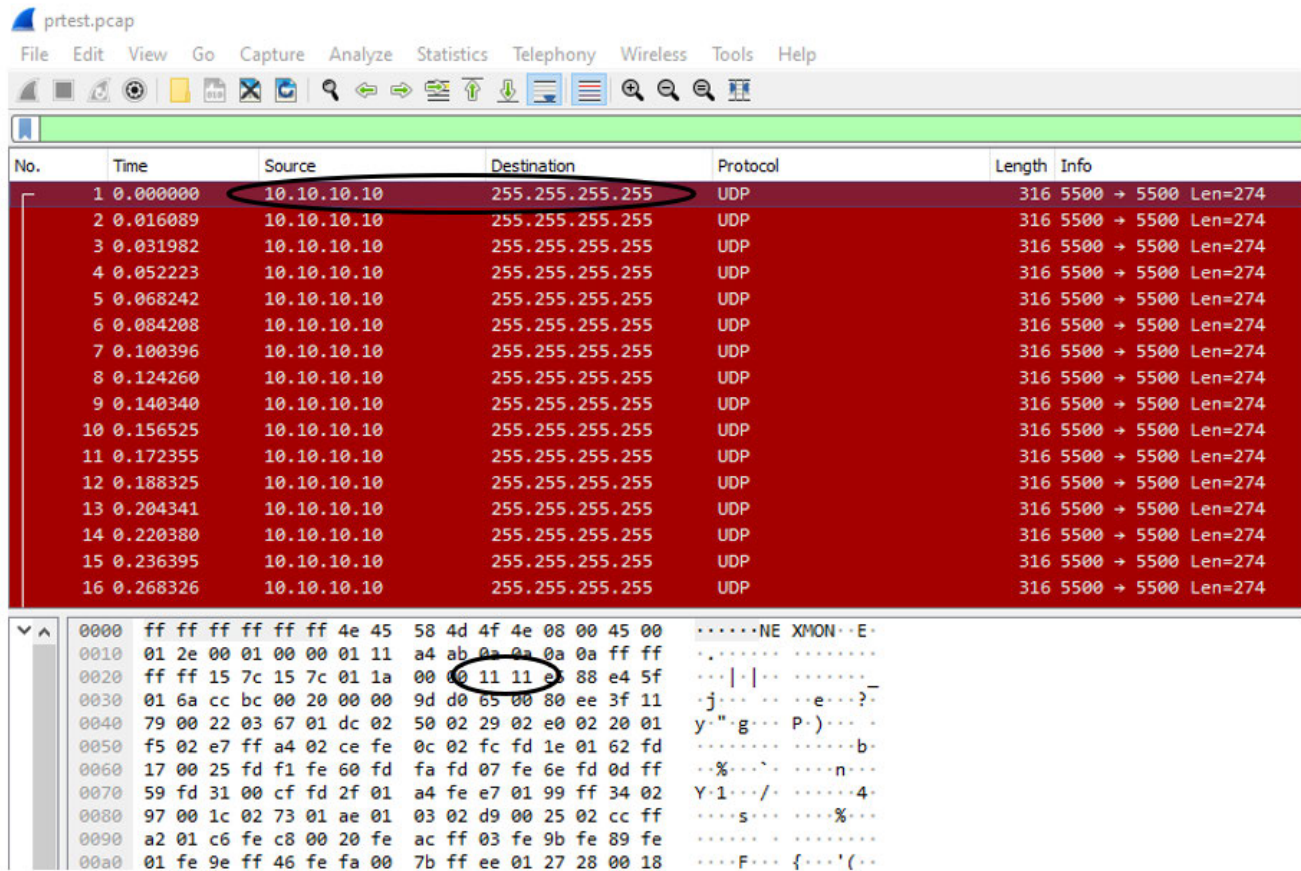
Table 6. Format of CSI Samples

Bytes	Type	Name	Description
2	uint16	Magic Bytes	0x1111
1	uint8	RSSI	RSSI in Two's Complement form
2	uint8	FrameControl Byte	Byte that shows the WiFi Frame Type
6	uint8[6]	Source Mac	Source Mac ID of the WiFi Frame
2	uint16	Sequence Number	Sequence number of the WiFi Frame
2	uint16	Core and Spatial Stream	Lowest 3 bytes indicate the Core, and the next three bits indicate the Spatial Stream number.
2	uint16	Chanspec	Chanspec used during extraction. See nexutil -k.
2	uint16	Chip Version	Chip Version
variable	int16[]	CSI Data	Each CSI sample is 4 bytes with interleaved Int16 Real and Int16 Imaginary. There are bandwidth * 3.2 OFDM subcarriers per channel, and a CSI sample for every subcarrier is present.

(Reddy 2022)

The .pcap files can be opened by software applications designed for network management and analysis such as Wireshark. Examining the raw contents of the file is useful to ensure the capture functioned correctly and the data is not corrupted. Figure 24. shows the test capture data opened in Wireshark (version 4.04). The first 16 of 1000 packets are shown. All UDP packets have source address 10.10.10.10 and destination address 255.255.255.255 - circled top of Figure 24. and all payloads contain the 2 magic bytes 0x111 at the 11th and 12th

bytes of row 0020 – circled bottom of Figure 24. 2 bytes forward of 0x111, is the MAC address of the Pi 4, “0x5f:01:6a:cc:bc”. These features align with Gringoli et al.’s (2019) description of the output of the extraction tools confirming capture is CSI data.



The image shows a Wireshark capture of CSI data. The top part is a packet list table with columns: No., Time, Source, Destination, Protocol, Length, and Info. The bottom part is a detailed view of packet 0020, showing the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.10.10	255.255.255.255	UDP	316	5500 → 5500 Len=274
2	0.016089	10.10.10.10	255.255.255.255	UDP	316	5500 → 5500 Len=274
3	0.031982	10.10.10.10	255.255.255.255	UDP	316	5500 → 5500 Len=274
4	0.052223	10.10.10.10	255.255.255.255	UDP	316	5500 → 5500 Len=274
5	0.068242	10.10.10.10	255.255.255.255	UDP	316	5500 → 5500 Len=274
6	0.084208	10.10.10.10	255.255.255.255	UDP	316	5500 → 5500 Len=274
7	0.100396	10.10.10.10	255.255.255.255	UDP	316	5500 → 5500 Len=274
8	0.124260	10.10.10.10	255.255.255.255	UDP	316	5500 → 5500 Len=274
9	0.140340	10.10.10.10	255.255.255.255	UDP	316	5500 → 5500 Len=274
10	0.156525	10.10.10.10	255.255.255.255	UDP	316	5500 → 5500 Len=274
11	0.172355	10.10.10.10	255.255.255.255	UDP	316	5500 → 5500 Len=274
12	0.188325	10.10.10.10	255.255.255.255	UDP	316	5500 → 5500 Len=274
13	0.204341	10.10.10.10	255.255.255.255	UDP	316	5500 → 5500 Len=274
14	0.220380	10.10.10.10	255.255.255.255	UDP	316	5500 → 5500 Len=274
15	0.236395	10.10.10.10	255.255.255.255	UDP	316	5500 → 5500 Len=274
16	0.268326	10.10.10.10	255.255.255.255	UDP	316	5500 → 5500 Len=274

Offset	Hex	ASCII
0000	ff ff ff ff ff ff 4e 45 58 4d 4f 4e 08 00 45 00NE XMON..E..
0010	01 2e 00 01 00 00 01 11 a4 ab 0a 0a 0a 0a ff ff
0020	ff ff 15 7c 15 7c 01 1a 00 00 11 11 e3 88 e4 5f
0030	01 6a cc bc 00 20 00 00 9d d0 65 00 80 ee 3f 11	..j...e...?
0040	79 00 22 03 67 01 dc 02 50 02 29 02 e0 02 20 01	y"-g...P.)...
0050	f5 02 e7 ff a4 02 ce fe 0c 02 fc fd 1e 01 62 fdb..
0060	17 00 25 fd f1 fe 60 fd fa fd 07 fe 6e fd 0d ff	..%...n..
0070	59 fd 31 00 cf fd 2f 01 a4 fe e7 01 99 ff 34 02	Y.1.../...4..
0080	97 00 1c 02 73 01 ae 01 03 02 d9 00 25 02 cc ffs....%...
0090	a2 01 c6 fe c8 00 20 fe ac ff 03 fe 9b fe 89 fe
00a0	01 fe 9e ff 46 fe fa 00 7b ff ee 01 27 28 00 18F...f...('..

Figure 24. Wireshark captured of CSI Data

3.2.4 Parsing CSI Data for Analysis

The CSI data must be parsed into MATLAB so that analysis can be undertaken, and detection techniques can be tested. Utilising the function created by Gringoli et al. (2019) to read .pcap files the CSI samples captured in UDP packets can be ingested into MATLAB as numeric values. Gringoli et al. (2019) created several MATLAB functions that are designed for use with the CSI extraction tools. All are generic and can be used with hardware other than the Broadcom bcm43455c0. Each different WiFi hardware option the CSI extraction tools are compatible with will format the raw CSI data differently, so the parsing process differs also. The functions also only enable basic plotting of each captured packet’s phase and amplitude. As the project requires a significant volume of data be ingested, visualised and manipulated in more complex ways, only Gringoli et al.’s (2019) functions to read .pcap files was utilised in the project. The functions were also modified to decode only UDP packets from the bcm43455c0 without requirement for any arguments to be parsed. A full code listing of the modified versions of Gringoli et al.’s (2019) functions is located in Appendix E.

Once ingested into MATLAB the raw CSI data is in the form of a matrix of complex numbers. The real part of each complex number represents the amplitude, and the imaginary part will represent the phase of the CSI sample. Rows contain a CSI measurement from each captured packet and columns contain the measurement of each subcarrier within the packet. The number of subcarriers in each sample is dependent on the bandwidth of the capture.

Table 7. Number of subcarriers for each Channel Width in IEEE 802.11ac

Bandwidth (MHz)	Number of Subcarriers
20	64
40	128
80	256
160	512

(Ward 2012)

To test and verify the operation of the CSI tools and extraction system the UDP packets which were captured following the process detailed in the previous section were ingested into MATLAB. The code used to ingest the CSI data is listed below: (using the default colour scheme and formatting for MATLAB code, see MATLAB documentation for more details: https://au.mathworks.com/help/matlab/matlab_prog/edit-and-format-code.html).

3.2.5 Matlab Code, Ingesting CSI:

```
%-----
%                               Ingesting CSI
%-----
%Ingests CSI data from tools developed by Gringoli et al. 2019
%Requires functions from Gringoli et al. 2019 MATLAB CSI Reader:
%  readpcap.m
%Avialbe at: https://github.com/seemoo-lab/nexmon_csi
%Only suitable for use with CSI captured from Broadcom 43455c0 WiFi NIC
%-----
%Packet Capture File read Parameters
File = '20MHzTest.pcap'; %file name of captured CSI data
BW = 20; %either 20MHz, 40MHz or 80MHz bandwidth
Max_UDP = 1000; %maximum number of UDP packets to read from capture file

%Ingest CSI from .pcap decoder
csi_raw = readCSI(File, BW, Max_UDP);
%csi_raw is matrix of CSI samples from captured packets, columns contain
%CSI data from each packet, rows contain CSI samples for each subcarrier

%Arrange CSI in subcarrier order with centre frequency = subcarrier
% index 0, as per 802.11ac
csi_raw = fftshift(csi_raw,2);
```


%%

%-----

Verifying the operation of the tools is important to ensuring data collected in testing is valid. The first verification step is to ensure the output variable *csi_raw* is a matrix containing 1000 rows, one for each captured UDP packet and 64 columns, one for each subcarrier of a 20MHz bandwidth capture, with each entry being a complex number.

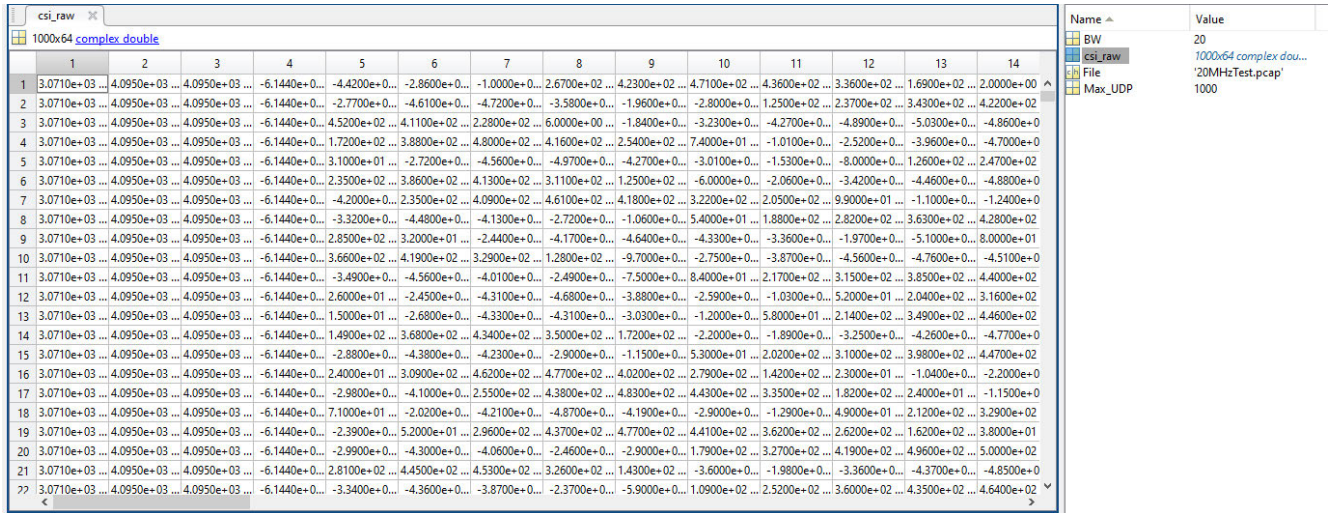


Figure 25. Test capture CSI data ingested, *csi_raw* Variable

The raw CSI data can then be visualised. The objective of visualising the raw data is simply to ensure that it contains the features that would be expected of a CSI data. The MATLAB code below was used to plot the CSI data from a single captured packet for inspection, the resulting plot is shown in Figure 26.

3.2.6 Matlab Code, Plotting Raw CSI:

```
%-----  
%  
%                               Plotting Raw CSI  
%Plots Raw CSI Data for inspection, magnitudes of all captured frames and  
%magnitude and phase of any selected frame  
%-----  
%Create subcarrier index  
subc_index = -(size(csi_raw,2)/2):1:(size(csi_raw,2)/2-1);  
  
%Plot Raw CSI Magnitudes:  
figure (10)  
plot(subc_index , abs(csi_raw.'))  
grid on  
xlim([subc_index(1) subc_index(length(subc_index))])  
xlabel('Subcarrier')  
ylabel('Magnitude')  
title('Raw Channel State Information')  
  
%Plot Raw CSI - Single Frame Magnitude and Phase:  
pack_no = 54; %select packet for plotting
```

```

figure (11)
subplot(2,1,1);
plot(subc_index , abs(csi_raw(pack_no, :)).')
grid on
xlim([subc_index(1) subc_index(length(subc_index))])
xlabel('Subcarrier')
ylabel('Magnitude')
title('Raw CSI From File: ' + convertCharsToStrings(File) + ...
      ', Frame no.: ' + pack_no)
subplot(2,1,2);
plot(subc_index , rad2deg(angle(csi_raw(pack_no, :)).'))
grid on
xlim([subc_index(1) subc_index(length(subc_index))])
ylim([-180 180])
xlabel('Subcarrier')
ylabel('Phase °')
%%
%-----

```

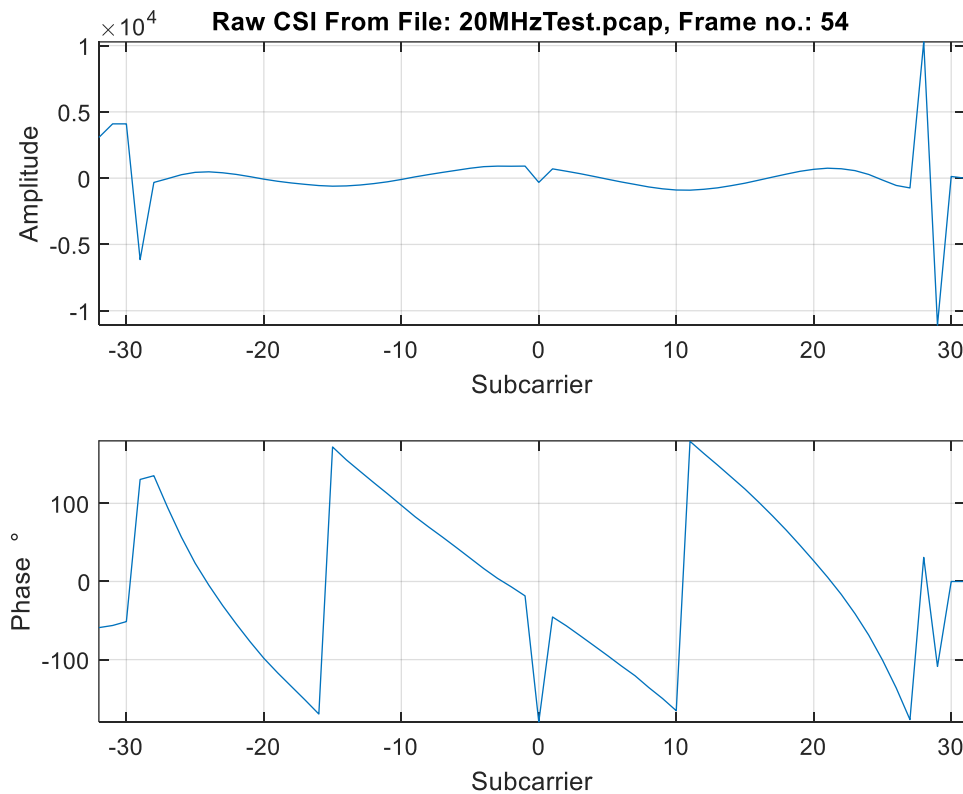


Figure 26. Plot of CSI data from test capture: Phase and Amplitude vs Subcarrier Index

The subcarriers in the x-axis are indexed as they are described in IEEE 802.11ac, for explanation refer to the plot in Figure 7. from Gast (2013). The phase changes in Figure 26. are uniform increments of approximately 15° per subcarrier. This can reasonably be considered expected behaviour of a CSI as there is uniform rotation between subcarriers (Ward 2012).

The amplitude in Figure 26. shows a sinusoidal pattern with significant outliers in the lowest, highest, and centre ordered subcarriers. These are also expected behaviour as they are the locations of the null subcarriers which contain a DC offset (Ward 2012). Table 8. lists the subcarriers used for data transmission in IEEE 802.11ac. Subcarriers outside of this range contain a DC offset and thus always contain outliers in amplitude and also will not contain any useful information about the properties of the signal path or the transmission (Ward 2012).

Table 8. List of data carrying subcarriers IEEE 802.11ac

Bandwidth	No. of Subcarriers	Data Tx. Subcarriers
20	64	-28 to -1 : 1 to 28
40	128	-58 to -2 : 2 to 58
80	256	-122 to -2 : 2 to 122
160	512	-250 to -130 : -126 to -6 6 to 126 : 130 to 250

(Ward 2012)

A similar plot to Figure 26. can be produced but with the null subcarriers removed to visualise the CSI behaviour without outliers.

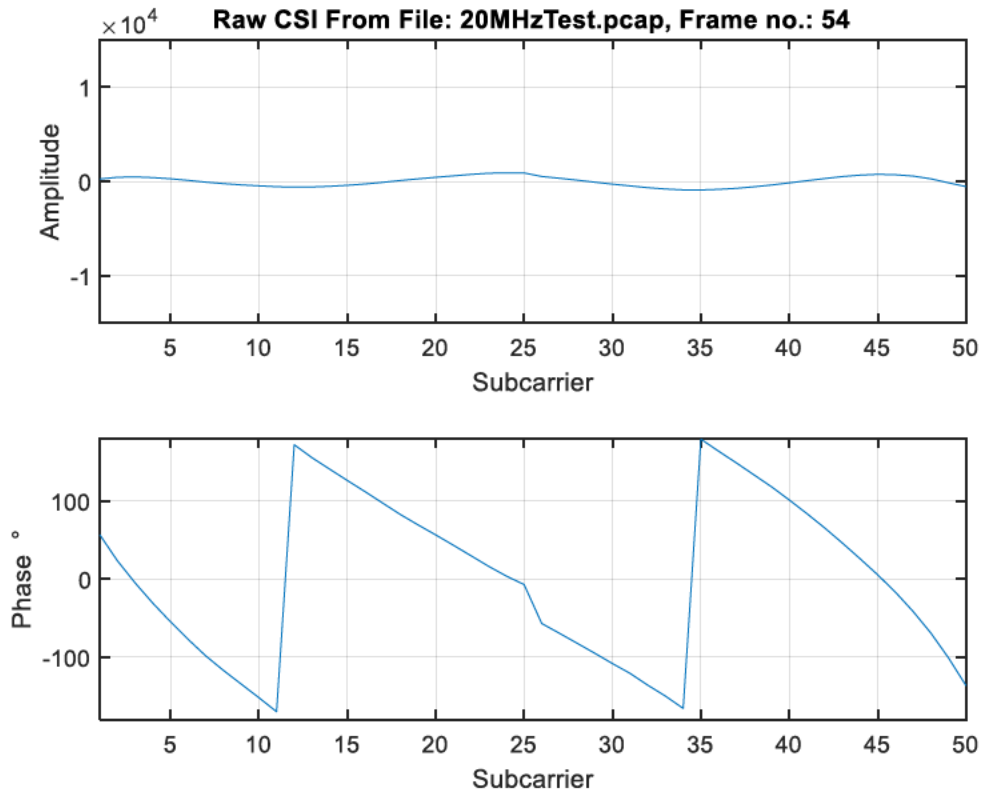


Figure 27. Plot of CSI data from test capture, null subcarriers removed

To keep the x-axis continuous, the subcarrier index is no longer similar to what is defined in IEEE 802.11ac and now represents the data subcarriers in order of frequency – lowest to highest in numerical order. While the CSI data appears to be extracted successfully, the validity of any testing undertaken in the project is highly dependent on the correct operation of the CSI tools. Ideally CSI measurements would have been taken from another source and compared to the CSI data extracted by the tools used in this project. This could be achieved via the use of instrumentation, such as a spectrum analyser taking measurements from the WiFi hardware or via assistance from a party with access to the development tools used to design and construct the WiFi hardware. While either of these methods would be complex and are outside the scope of the project, it is important to ensure that as much verification as possible is undertaken to ensure that the tools are functioning as intended.

While less robust, another method of verification is to compare the CSI data extracted in the test against CSI data extracted from a different toolset used by other researchers. Figure 28. plots CSI data extracted from the Atheros CSI Tool created by Xie, Li & Li (2015). Like the test extraction performed in this project, Xie, Li & Li (2015) note that the channel was stable during the extraction. While the Atheros CSI Tool is slightly older than Gringoli et al.'s (2019) tool, and operates with 802.11n as opposed to 802.11ac it should be similar.

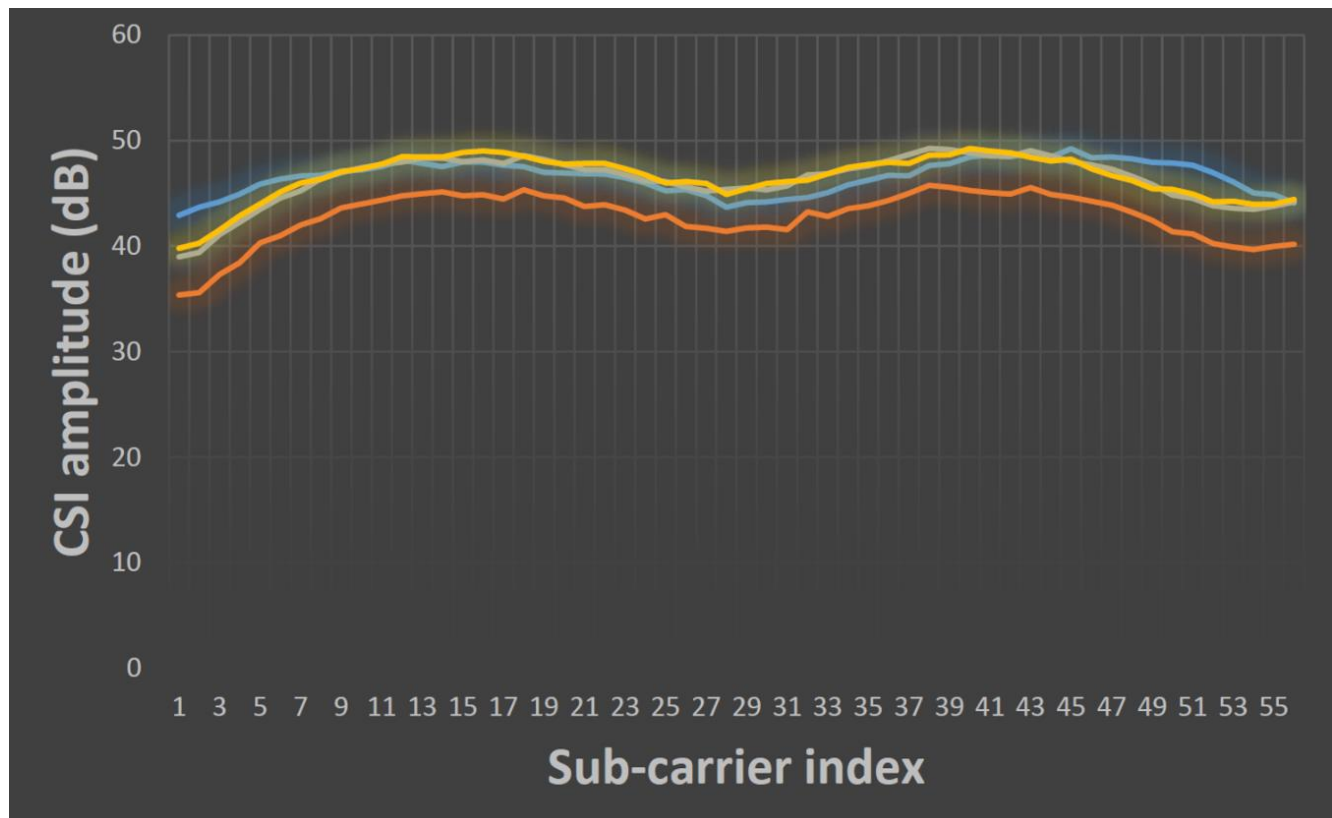


Figure 28. Test capture of four CSI samples from the Atheros CSI Tool

(Xie, Li & Li 2015)

Xie, Li & Li (2015) don't detail exactly how subcarriers are arranged in the x-axis or what was used as a reference for the conversion to decibels in the y-axis but it is clear that the CSI samples amplitudes follow a substantially similar pattern to the test capture undertaken in this project.

3.3 Verifying the Test Equipment

A challenge when using open-source tools and firmware created by researchers is that they are often developed with limited oversight, testing and compliance requirements in comparison to commercial products. Also, in the case of the tools used in this project without the direct assistance and verification of the manufacturers of the WiFi equipment they interact with (Gringoli et al. 2019). Prior to collecting any CSI data intended to assess the feasibility of detecting vermin the test network was verified as far as was practical using the equipment available to the project.

3.3.1 Filtering Functions

Without correct operation of the filtering functions the CSI samples logged could be from devices other than the Pi 4 and from packets that aren't the intentionally generated sensing pulse pings. To verify the operation of the filtering functions the Pi 4 was connected to a busy WiFi network being used within a household to provide internet connection to various devices. The testing produce used is detailed in Table 9.

Table 9. Testing the CSI filtering functions

Procedure	Results	Comments	Purpose
1. Identify the Channel being utilised by the WiFi network.	Channel 36	<p>Check router settings or use <i>iwlist wlan0 scan</i> on the Pi 4.</p> <p>The channel used is not significant for the test but must be known to configure the CSI extraction.</p> <p>The channel width is also not significant as packets will still be captured even if the extraction width doesn't match the captured packet.</p>	Configure Test.
2. Extract CSI with no filtering.	CSI measurements captured rapidly	Do not enter an address (<i>-m</i>) or byte filter(<i>-b</i>) argument to: <i>-mcp</i> , command when configuring the CSI extraction.	Confirm CSI extraction works with default parameters.
3. Extract CSI with filtering of the Pi 4's MAC address.	Captures packets at a slower rate than previous step	<p>Configure CSI extraction using MAC address filtering: <i>-mE4:5F:01:6A:CC:BC</i></p> <p>Ensure the Pi 4 will generate traffic on the WiFi network, web browse or ping the router.</p>	Confirm CSI measurements are captured when filtering for packets from the Pi 4 only.

4. Turn the Pi 4 off and repeat capture in Step 3.	No captured packets	No other device should have the same MAC address as the Pi 4. If packets are captured indicates error.	Confirm filter does not log measurements when the Pi 4 is not on network.
5. Turn on Pi 4 and ping another WiFi device on the network at the default interval, Extract CSI using MAC and address QoS frame filtering	Packets Captured approximately once a second. Occasional bursts of 4 packets captured in quick succession were observed.	Any device on the network can be pinged, it is not recommended to ping a device outside of the local network to ensure consistent trip time. Configure CSI extraction using frame filtering: <code>-b0x88</code> , and MAC address filtering: <code>-mE4:5F:01:6A:CC:BC</code> Default ping interval is 1 second. Ping will continue indefinitely if initiated from Pi 4's terminal.	Confirm CSI extraction occurs at expected packet rate when frame filtering is used.
6. Stop pinging from Pi 4 and generate WiFi traffic via other method e.g., web browsing. Continue CSI extraction from Step 5.	Packets only captured in irregular bursts of 4 usually 15 – 30 seconds apart.	Only specific traffic should contain the frame heading for QoS. Short bursts of 4 packets are likely responses to beacon frames or other network management traffic.	Confirm CSI extraction with QoS frame filtering will not measure CSI from other frames.
7. Turn off Pi 4 and initiate pinging between two other devices on the WiFi network e.g. between router and PC. Continue CSI extraction from Step 5.	No captured packets.	Filtering should only log measurements from QoS frames from the Pi 4.	Confirm CSI extraction with QoS frame filtering and MAC address filtering will not measure QoS frames from devices other than the Pi 4.

Testing confirmed there was no unexpected behaviour of the filtering functions except for occasional CSI measurements were when no ping was initiated and QoS frame filtering was being utilised. These measurements are almost certainly caused by beacon frames or other network management traffic. Beacon frames are transmitted to announce the presence of a WiFi device and contain information to allow devices to begin communication sessions, such as modulation type and compatible standards (Ge et al. 2022). They are transmitted frequently from APs, typically every 100ms but devices connected to the AP do not respond to every beacon. The exact frequency of a device connected to an AP transmitting QoS frames is hard to predict and depends on events occurring within the communication session (Ge et al. 2022).

No method of filtering out these unsolicited QoS frames could be implemented however the impact they have on testing is deemed minimal. In the case where QoS frames are transmitted between the Pi 4 and the TP-Link router in response to beacon frames, they should still contain information about the signal path under test but will not be generated at a known interval. Given they are generated extremely infrequently (15-30 seconds based

on testing) compared to the frequency of traffic required for sensing ($\approx 10\text{ms}$) this won't adversely impact testing. If the frames are transmitted between the Pi 4 and another device not part of the test network, they may not appear in the capture at all as they will not necessarily be on the same channel the CSI extraction is occurring on. However, if they are transmitted over the same channel, they will be captured and will be an outlier to the other CSI measurements as they will not be generated from the same transceivers or signal path. In the next section of this dissertation the pre-processing methodology of the CSI data collected during testing is detailed which aims to remove such CSI captures prior to sensing analysis.

3.3.2 Controlling Bandwidth and Channel

The channel and bandwidth of the test network (and WiFi networks in general) are controlled by the router or AP. Prior to the selection of the Archer C20 TP-Link router other models were trialled. Initially, a Netgear Nighthawk AC1900 was intended to be used as it was the most feature rich and powerful router available to the project. However, unlike the TP-Link router, the Netgear router could not be configured to operate with a fixed channel width. Both the TP-Link and Netgear routers could be set to automatically find an optimal channel (the recommended setting) or be configured to remain on a fixed channel. But the configuration options available for channel width differed. The Netgear router provides three configuration options to set maximum speed that correspond to 20MHz(289Mbps), 40MHz(600Mbps) and 80MHz(1300Mbps) channel width see Figure 29. The TP-Link router allowed a specific channel width to be selected and set see Figure 21.

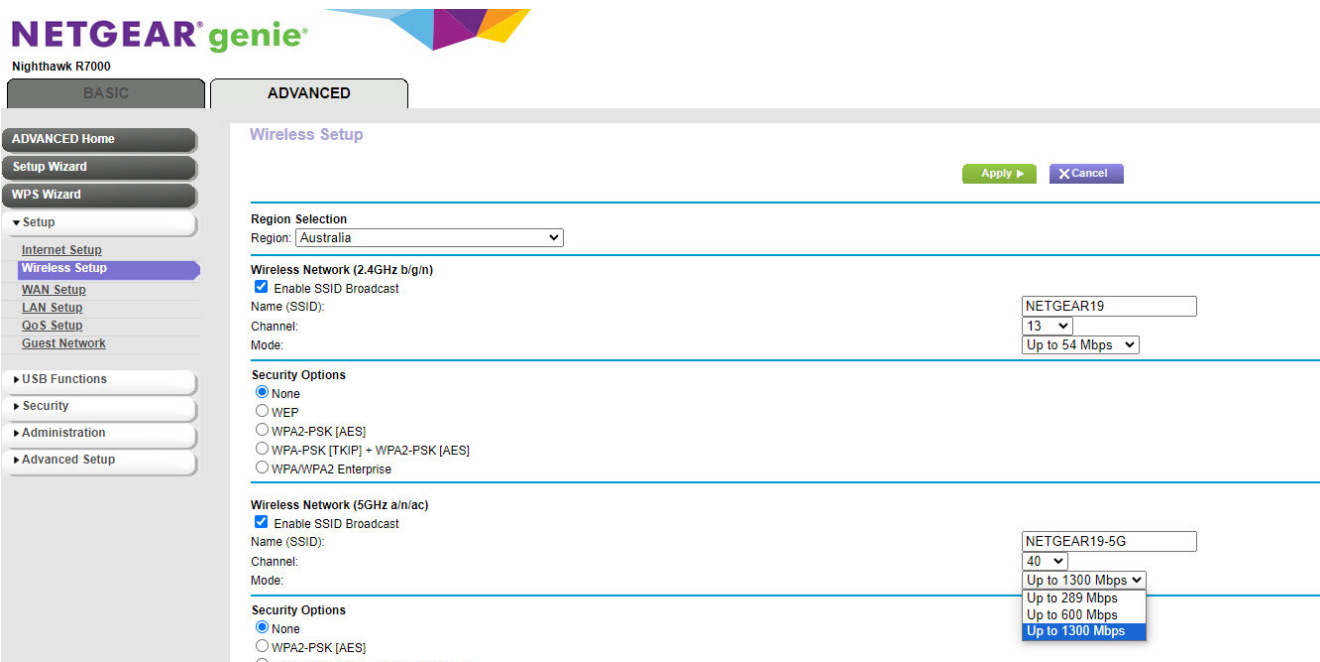


Figure 29. Netgear configuration webserver

The Pi 4 can scan all networks (via: `iwlist wlan0 scan`) and display the parameters of each network in range, including the channel. Scanning formed an important part of testing to ensure the channel used was vacant and

to ensure that the Pi 4 was connected correctly to the TP-Link router and with strong signal strength. The output of a WLAN scan performed on the Pi 4 is shown in Figure 30. It is notable that the standard IEEE 802.11i listed in the scan in Figure 30. refers to an access standard and not a wireless networking standard such as IEEE 802.11ac.

```

pi@raspberrypi: ~
File Edit Tabs Help

Cell 02 - Address: 3A:22:E2:9A:04:53
Channel:149
Frequency:5.745 GHz (Channel 149)
Quality=52/70 Signal level=-58 dBm
Encryption key:on
ESSID:"DIRECT-53-HP OfficeJet 8010"
Bit Rates:6 Mb/s; 9 Mb/s; 12 Mb/s; 18 Mb/s; 24 Mb/s
          36 Mb/s; 48 Mb/s; 54 Mb/s
Mode:Master
Extra:tsf=0000000000000000
Extra: Last beacon: 60ms ago
IE: Unknown: 001B4449524543542035332D4850204F66666963654A657
42038303130
IE: Unknown: 01088C1218243048006C
IE: Unknown: 2D1A2C0103FF00000001000000000000000000000000000000
0000000000
IE: IEEE 802.11i/WPA2 Version 1
Group Cipher : CCMP
Pairwise Ciphers (1) : CCMP
Authentication Suites (1) : PSK
IE: Unknown: 3D169508000000000000000000000000000000000000000000
0
IE: Unknown: 7F080500000021000000
IE: Unknown: 3B130301020304050C161718191A1B1C1D1E1F2021

```

Figure 30. Scan of WiFi network in range of the Pi 4

Determining channel width is more difficult than determining channel number. The most robust method is to use CSI to verify the channel width throughout a capture. To do this the CSI extraction must be configured to measure 80MHz CSI samples (the highest width available). The channel bandwidth can then be identified using the value of the CSI measurements. Throughout the data subcarriers the magnitude of the CSI will fluctuate but should maintain a reasonably consistent value, well above zero and the noise floor. If CSI is captured from a transmission within the same channel region but with lower bandwidth, the magnitude of each subcarrier will be insignificant until a subcarrier is sampled that aligns with data a carrying subcarrier. For example, referring to Figure 31. if channel 155 is an 80MHz channel with 256 subcarriers and channel 157 is a 20MHz channel using 64 subcarriers, then subcarriers 0-64 (of -128 – 128) of channel 155 will align with the 20MHz channel 157.

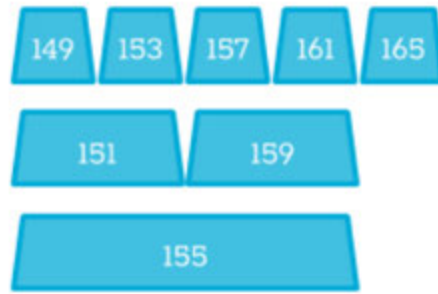


Figure 31. Overlapping WiFi channels of varying width

(IEEE Standards Association 2013)

It is straightforward to determine the channel bandwidth when inspecting a plot of the subcarrier index vs CSI magnitude. With the CSI capture configured to sweep 80MHz, six packets were captured, 3 from a 20MHz channel width and 3 from an 80MHz channel width. Figure 32. is the resulting plot identifying the channel width.

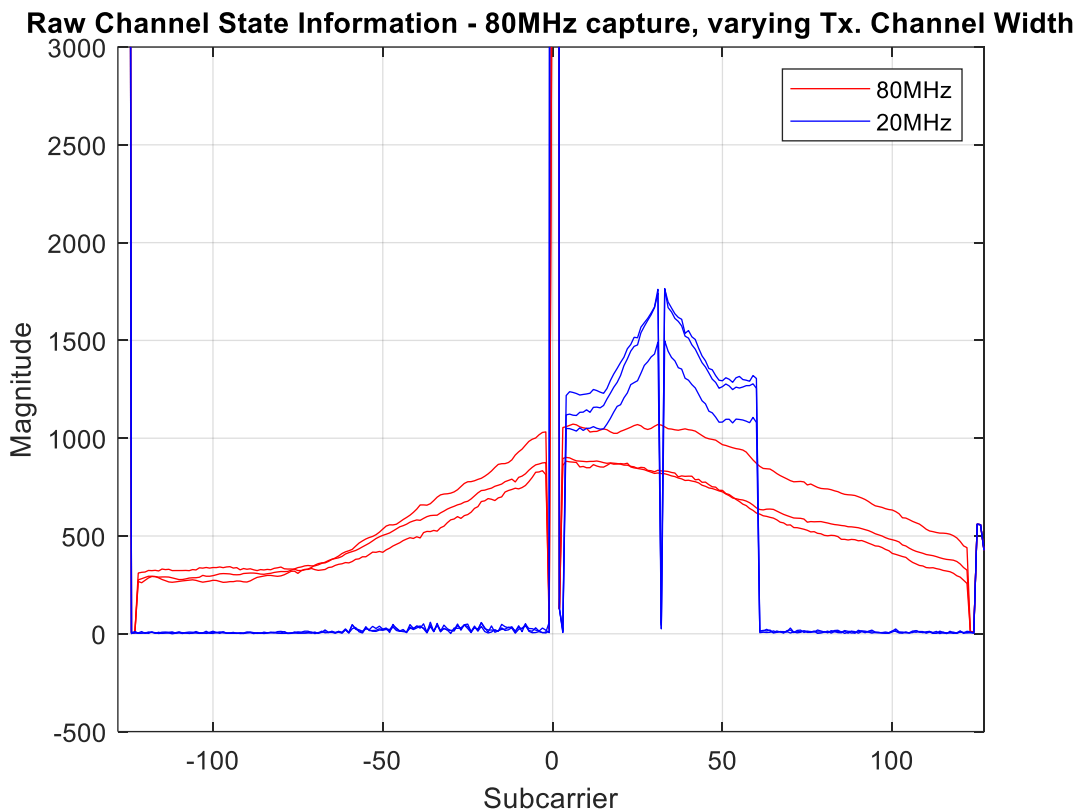


Figure 32. Plot to identify channel width

Utilising CSI plotting to determine channel width, a testing procedure was developed to verify the behaviour of the TP-Link and Netgear routers to determine the most suitable to use for testing.

Table 10. Router testing and verification procedure

Procedure	Results	Comments	Purpose
1. Configure the router under test to host a network on a fixed Channel with 80MHz bandwidth. Scan networks on the Pi 4 to verify which channel the network is operating on.	<p>Netgear Router – WiFi network is on configured channel - 157.</p> <p>TP-Link Router – WiFi network is on configured channel – 157.</p>	<p>Routers are configured via webserver.</p> <p>Only the 5GHz band was used in this project but both routers have dual band capability and can also host a 2.4GHz region network.</p> <p>To scan on Pi 4: <i>iwlist wlan0 scan</i></p>	Verify the webserver configuration tools and manual channel selection are functioning.
2. Configure another router to host a network on the same channel as the router under test. Generate traffic to this router and the router under test simultaneously. Scan networks on the Pi 4 to verify which channel the router under test is operating on.	<p>Netgear Router – WiFi network remained on configured channel.</p> <p>TP-Link Router – WiFi network remained on configured channel.</p>	<p>The Pi 4 can only join one network at a time so at least one other WiFi enabled (IEEE 802.11ac) device is needed.</p> <p>Flood pings are a simple way to generate significant traffic.</p> <p>To flood ping: <i>ping -f <IP Address/DNS name></i></p>	Confirm router will not move channel even when there is interference from another network.
3. Switch off the additional router from Step 2 leaving only the router under test hosting a network. Connect the Pi 4 to the network. Configure a CSI extraction targeting the Pi 4 on the channel used with 80MHz bandwidth. Generate a varying rate of traffic between the Pi 4 and the router over 10 seconds (maximum). Plot magnitude vs subcarriers of the captured CSI.	<p>Netgear Router – varying channel width throughout CSI capture.</p> <p>TP-Link Router – channel width remained fixed at 80MHz.</p>	<p>Alternating between flood pinging and default (1s) pinging of the router from the Pi 4 will generate a varying rate of traffic. Web browsing or any networking task that requires a varied data rate will also be suitable. eg. web browsing, then video streaming.</p> <p>It is important to vary the traffic quickly to avoid having an enormous number of CSI samples that will be difficult to plot coherently. The CSI capture can be paused while the usage of the network is changed and then restarted.</p>	Confirm channel width remains fixed at varying data rates.

Testing revealed that the Netgear router adjusted the channel width automatically based on the rate of traffic. An inconsistent channel width is undesirable for sensing so only the TP-Link router was used during testing.

3.4 Testing Arrangement and Configuration of Test Network

Ensuring that ethical standards are adhered to limits the location of testing to the animal's normal habitat, in this case that is the room the mouse used in testing is normally kept. The room is approximately 3m by 4m with 2.5m high ceilings. The room is a bedroom used mainly for keeping animals within a 5-bedroom brick veneer house, located in Townville, North Queensland. A rectangular area approximate 3m by 1.7m was free of obstacles and available to set up the test signal path. The mouse's usual enclosure is a steel cage which would be undesirable to introduce to the signal path during testing as it may impact the CSI measurements. The owner explained that the mouse is often taken out of its enclosure for cleaning and socialising and kept on a 300mm by 300mm mostly plastic tray with some husbandry items shown in Figure 33. (left). Usually, the tray is placed on top of the mouse's enclosure or on a self and never on the ground.



Figure 33. The mouse positioned for testing (left) and close up of the mouse used during testing

The Pi 4 and the TP-Link router were placed on plastic boxes 350mm above the ground and 2.5m apart creating a 2.5m long LOS signal path. A third plastic box also 350mm in height (shown in the left of Figure 33.) was placed in the centre of the 2.5m signal path to house the mouse within the signal path. Figure 34. was taken in the room where testing was conducted and shows the TP-Link router on the left, the box used to elevate the mouse in the centre and the Pi 4 on the right.



Figure 34. Singal path used for testing



Figure 35. Close up of the TP-Link router and Pi 4 in the test network

During testing the Pi 3B+ could be kept in the same room approximately 1.3m from the Pi 4. A LOS signal path 2.5m long can be considered suitable for sensing testing as there should be no significant attenuation caused by distance and the length of signal path is reasonably consistent with that used in most other documented sensing testing based on a survey conducted CSI based sensing research, shown in Table 11.

Table 11. Survey of signal path length used in CSI sensing research

Title	Author	Distance Between WiFi Devices
WiFi-Based Real-Time Calibration-Free Passive Human Motion Detection	(Gong et al. 2015)	5 meters
Non-Invasive Detection of Moving and Stationary Human With WiFi	(Wu et al. 2015)	2 – 6 meters
Channel State Information Based Human Presence Detection using Non-linear Techniques	(Palipana, Agrawal & Pesch 2016)	5 meters
Complex Motion Detection Based on Channel State Information and LSTM-RNN	(Zhang et al. 2020)	3 meters
Device free human activity and fall recognition using WiFi channel state information (CSI)	(Damodaran et al. 2020)	Non-fixed locations in 4m by 4m room
Revisiting Indoor Intrusion Detection With WiFi Signals: Do Not Panic Over a Pet!	(Lin et al. 2020)	Distance not stated but performed in a typical domestic room
WiFi-based Human Activity Recognition using Raspberry Pi	(Forbes, Massie & Craw 2020)	Distance not stated but performed in a typical domestic room – “<i>same height above the ground</i>”
Fire Detection Using Commodity WiFi Devices	(Li et al. 2021)	90cm
Human Activity Recognition Using CSI Information with Nexmon	(Schäfer et al. 2021)	Diagonally opposite in 3.5m by 4.5m room

3.4.1 Testing Procedure

Two types of tests were undertaken, vermin detection tests where the mouse is within the signal path while CSI measurements are extracted and logged and control tests. Control tests were conducted when the mouse was removed from the signal path but with all the same objects, including the mouse's husbandry items in the signal path and were always undertaken in the same room. Control tests were also always undertaken on the same day immediately after vermin detection tests to ensure weather conditions and sources of interference were similar. The room contained a ceiling fan and there was a WiFi network operating within the house where testing was conducted. To avoid interference during testing the fan and all WiFi devices in the house were switched off and no persons were present within the room during testing. The Pi 3B+ was interfaced with via an Ethernet patch lead connected to the computer hosting the remote terminal line and the Pi 4 was controlled via a keyboard. Tests were observed and the equipment was operated from a position several meters behind (opposite direction to the Pi 4) the TP-Link router outside the bedroom doorway.

An 80MHz channel width was used for all testing as it is the widest bandwidth compatible with the Pi's. A wider bandwidth produces more CSI data, and a greater spectrum increases the chance of a subcarrier within the spectrum interacting with an object in the signal path in a significant way, as Tan, Zhang & Yang (2018) demonstrated when attempting to sense fruit ripeness. The choice of channel was largely based on avoiding interference as opposed to sensing efficacy. The tests were conducted in the inner suburbs of Townville, and it is highly likely that numerous other WiFi networks were operating nearby in adjacent dwellings. The Pi 4 can scan for other WiFi networks using the `iwlist wlan0 scan` command. The output of this command details the signal strength and channel of any networks that are within range of the Pi 4 and so any channels present could be avoided to prevent interference. As a default, a high value channel, 149+, was used as these channels seemed to be utilised less often. Possibly because there are greater restrictions around their use or because WiFi devices search for an available channel in ascending order (ACMA 2021a, 2021b). By using a channel not identified by the Pi 4 scanning utility, the chance reduces significantly of interference being caused by a nearby WiFi network as it can be determined that no WiFi networks are operating on that channel within range of the Pi 4 at the time of the scan.

To generate traffic and create transmissions between the Pi 4 and the router the Pi 4 pings the router at 10ms intervals. A common problem encountered with CSI sensing is irregular data-distribution (Zhang et al. 2022). This mostly arises from being unable to determine the exact Tx and Rx partners of a CSI sample and the intervals between CSI samples being irregular as WiFi communication is not synchronized. This is somewhat overcome in the test network by filtering CSI samples so that only QoS packets are captured and 10ms was selected as experimenting revealed that ping times less than 10ms started to become far more irregular when the network was being used for normal communications.

For example, as an experiment, 2000, 10ms pings were generated between the Pi 4 and the TP-Link router while the Pi 4 was also constantly refreshing a web page. This caused an average time of 10.03ms between pings. The

same test was undertaken with 2000, 2ms, pings and the average time was 2.3ms. Notably the response time for each test was similar so it can be inferred that the Pi 4 is the device unable to maintain 2ms ping which is expected as it's WiFi hardware only has a single core and can only generate a single spatial stream were as the TP-Link router (which was used to host internet access) has 3 antennas, 4 cores and can generate 8 spatial streams.

10ms is considered as a potentially reasonable value that could be used to generate a “sensing pulse” in a WiFi system that is functioning as both a communication network and a sensing system without consequentially compromising communications but while still producing enough CSI data to enable sensing. It is important to note that the 10ms ping will still not generate a completely uniformly disturbed sensing pulse, nor will the filtering functions only log CSI samples from ping packets as other traffic is also categorised as QoS.

Conducting testing was the highest risk from a safety and wellbeing perspective phase of the project and involved technical tasks such as configuring the test network as well as housekeeping and data management aspects. Testing also involved the added complications of an animal and the requirement to turn off the WiFi network to a family household. To ensure testing was repeatable and conducted efficiently a procedure and checklist were developed which is detailed in Table 12.

Table 12. Testing Procedure

Procedure	Check List	Comments
1. Arrange and initialise test equipment: Pi 4, TP-Link router and stand for stimuli, 350mm above ground level. Pi 4 and TP-Link router arranged to be 2.5m apart from front edge of devices. TP-Link router to be oriented with antennas behind single path. Pi 3 positioned close to signal path but outside LOS. Energise equipment and verify connectivity: Pi 4 connects to Test CSI network hosted by TP-Link router. Pi 3 is within WiFi range of Pi 4 at least > -60dBm	<input type="checkbox"/> Equipment layout verified and cabling secure <input type="checkbox"/> Pi 4 connects to “CSI Test” network <input type="checkbox"/> Pi 3 is in range of Pi 4 <input type="checkbox"/> Terminal to Pi 3 active <input type="checkbox"/> KVM to Pi 4	Each device takes up to 90 seconds to boot. Pi 3 wireless networking will be off by default – if testing signal strength prior to CSI test use: <i>ifconfig wlan0 up</i> to activate. Refer risk assessment 2122.

<p>2. Isolate sources of potential interference: ceiling fans, persons moving within 6m of signal path, nearby microwaves, Bluetooth devices and WiFi devices</p> <p>Scan for nearby WiFi networks via the Pi 4.</p>	<p><input type="checkbox"/> All sources of interference isolated as far as reasonably practical</p> <p><input type="checkbox"/> Scan for nearby WiFi networks</p> <p>Record Signal Strength and Channel of nearby networks:</p> <p>Temperature and Humidity</p>	<p>Sources of interference will still likely be present but selecting a vacant channel reduces the chance of significantly affecting sensing.</p> <p>As well as WiFi networks document other sources of interference when known, as well as temperature and humidity to provide further background information on CSI data captured.</p> <p>Scanning for WiFi networks will only detect access points and not clients.</p>
<p>3. Configure parameters of “CSI Test” network on TPLink router.</p> <p>Select a Channel that is far as possible from any known channels detected in the scan in Step 3.</p> <p>80Mhz Bandwidth, IEEE 802.11ac.</p>	<p><input type="checkbox"/> TPLink, “CSI Test” network configured</p> <p><input type="checkbox"/> Pi 4 still connected to “CSI Test” network if reconfigured</p> <p>Channel used for testing:</p>	<p>See Figure 21. for TP-Link configuration via webserver.</p> <p>Adjusting WiFi network parameters should not disconnect Pi 4 from network, but verification of connection if changing channel is advised.</p>
<p>4. Configure CSI extraction.</p> <p>Input commands on Pi 4 to ping TPLink router but do not press enter.</p> <p>Configure CSI extraction on Pi 3 and input the <i>tcpdump</i> command with required arguments but do not press enter.</p>	<p><input type="checkbox"/> CSI extraction can be started by executing next commands on Pi 3 & 4</p> <p>Ping Frequency:</p> <p>Number of CSI capture frames:</p> <p>File name of CSI data:</p> <p>Estimated length (time) of Capture:</p>	<p>Time of capture is ping frequency 10ms ($-i0.01$) x No. of capture frames usually 1000 ($-c1000$)</p> <p>Refer “Setting up Raspberry Pi for CSI Extraction” section for instructions on CSI extraction setup.</p> <p>It is critical to ensure that once signal path stimuli are in place, particularly when the mouse is in the signal path, CSI extraction can begin quickly. Complete all preparation tasks for CSI extraction in this step.</p>
<p>5. Prepare signal path for testing ensure any stimuli are in position in centre (1.25m) of LOS between Pi 4 and TP-Link router.</p> <p>If animal is within signal path refer Risk Assessment 2122 and Ethics Approval ETH2023-0118.</p>	<p><input type="checkbox"/> Valid Test completed; or</p> <p><input type="checkbox"/> Test Invalid</p> <p>Any significant observations:</p>	<p>Invalid tests should be completed where possible. Validity refers to assessing whether the CSI data collected is suitable for sensing analysis. Invalid CSI data may still be useful for other analysis.</p>

<p>Move as far away from LOS signal path as possible while still able to access Pi 3 & 4 terminals.</p> <p>Execute Pi 4 pinging router and Pi 3 CSI extractions.</p> <p>Observe signal path and ensure:</p> <ul style="list-style-type: none"> • no sources of interference arise during test e.g. object falls into signal path • mouse stays within 300mm-by-300mm zone (if applicable) • Extraction completes within +-3s of estimated time in previous Step. <p>Test is not valid if any of the observation conditions above are not met.</p> <p>Test Complete once <i>tcpdump</i> completes on Pi 3.</p>		
<p>6. Copy CSI data from Pi 3 to USB storage drive and reboot Pi 3.</p> <p>Return to Step 4 if completing additional CSI extractions.</p>	<p><input type="checkbox"/> CSI data backed up to USB storage drive</p>	<p>Rebooting Pi 3 ensures correct operation of CSI extraction.</p> <p>If completing a significant number of tests, consider deleting captures from Pi 3 memory and from USB storage drive once backed up to prevent consuming all available disk space. A single 80MHz captured frame requires 1000 bytes of storage space.</p>

4. Results and Analysis

Over 200 captures of CSI data were undertaken during the project. The majority of CSI data captured in the project was from channel 157 via sessions of 1000 packets triggered by a 10ms interval ping. It was noted that tests involving captures beyond 10s in length were significantly more likely to encounter either technical issues or when testing with the mouse in the signal path, the mouse would leave the 300mm by 300mm zone in the

LOS signal path of the Pi 4 and TP-Link router. Only data captured from tests that met the conditions in Step 5. in Table 12. were used to attempt sensing. The CSI data from each test was saved in the outputted .pcap file and named using a numeric identifier and a one or two word description containing either “mouse” or “blank”. Noting that “blank” refers to control tests used for sensing analysis that contained no mouse in the signal path and is not meant to imply the signal path was void of all objects.

4.1 Removing Outliers and Unwanted Subcarriers

Before undertaking an analysis of CSI data with the intent of sensing, it is important to remove anomalies and outliers from the CSI data. Null and pilot subcarriers and ambiguities caused by hardware and firmware are known to cause anomalies and outliers that are not reflective of the environment within the signal path (Schäfer et al. 2021). Any anomalies and outliers caused by these effects are insignificant to sensing but will likely create features that could be miscategorised as the environment in the signal path changing. It is important they are isolated and removed wherever possible from the CSI data before attempting sensing analysis.

In the case of null and pilot subcarriers they can be implicitly removed since their locations are determined by the networking standard IEEE802.11ac. Subcarriers outside of the data subcarriers in Table 13. and the listed pilot subcarriers in the left most column of Table 13. can be removed from each CSI capture prior to any analysis aimed at sensing.

Table 13. Null and pilot subcarriers for channel bandwidths available on Pi 4

Bandwidth	No. of Subcarriers	Data Tx. Subcarriers	Pilot Subcarriers
20	64	-28 to -1 : 1 to 28	$\pm 7, \pm 27$
40	128	-58 to -2 : 2 to 58	$\pm 11, \pm 25, \pm 53$
80	256	-122 to -2 : 2 to 122	$\pm 11, \pm 39, \pm 75, \pm 103$

4.1.1 Hampel Filtering for General Outlier Removal

In the other case of ambiguities caused by hardware and firmware, identification and removal are less straightforward. Schäfer et al. (2021), Li et al. (2021) and Wang, Yang & Mao (2020) utilised the Hampel filtering function to remove outliers during CSI data pre-processing prior to inputting to sensing algorithms, see Figure 8. for Schäfer et al.’s (2021) implementation (Wang, Yang & Mao 2020; Li et al. 2021; Schäfer et al. 2021). The Hampel filter examines incremental segments of the input data termed “windows” (Pearson et al. 2016).

$$\mathbf{W}_k^K = \{x_{k-K}, \dots, x_k, \dots, x_{k+K}\}$$

(Pearson et al. 2016)

Where \mathbf{W} , is the data window, x is the input data samples and K is the window length, a positive integer identifying the number of samples each side of the centre of the window (Tukey 1974; Pearson et al. 2016).

The Hampel identifier is used to identify outliers. To determine the Hampel identifier, the median value of the window, denoted m_k , is determined.

$$m_k = \text{median}(\mathbf{W}_k^K) = \text{median}\{x_{k-K}, \dots, x_k, \dots, x_{k+K}\} \quad (\text{Pearson et al. 2016})$$

The median value is then scaled by a factor of the median absolute deviation (MAD) often denoted by κ , which for normally distributed data is:

$$\kappa = \frac{1}{\sqrt{2} \text{erf}^{-1} \frac{1}{2}} \cong 1.4826$$

Where erf is the error function.

The Hampel identifier tS_k can then be determined.

$$tS_k = t \cdot \kappa \cdot m_k \quad (\text{Davies \& Gather 1993})$$

Where t is factor of the MAD and a filter tuning parameter, κ is MAD (≈ 1.4826) and m_k is the median value of the window.

If the distance between the input sample to the Hampel filter and the median of the window (m_k) is greater than the Hampel identifier (tS_k), the input sample is identified as an outlier, and it is replaced by median of the window (m_k).

The filter response can be given by:

$$y_k = \begin{cases} x_k & |x_k - m_k| \leq tS_k, \\ m_k & |x_k - m_k| > tS_k. \end{cases} \quad (\text{Davies \& Gather 1993; Pearson et al. 2016})$$

Where y_k is the output, x_k is the input, m_k is the median value of the window and tS_k is the Hampel identifier.

The scaler t , used to determine the Hampel identifier was set to 3, which aligns with the empirical rule of statistics in that 99.7% of a normally distributed population will reside within three standard deviations of the median value (Kaye & Freedman 2011).

By setting the scaler t , to 3, the only variable tuning parameter to the Hampel filter then becomes window size. Window size is defined by number of samples on each side of the sample subject to the filter. In both Schäfer et al.'s (2021) and Wang, Yang & Mao's (2020) pre-processing, window lengths of 3 – 5 samples were used. Li et al. (2021) used a significantly longer window of 11 samples and was also anticipating much finer features in the CSI data collected as the aim was to detect fire with extremely close transceivers. While vermin are likely to induce finer changes in the CSI data as opposed to humans, which were the target of Schäfer et al.'s (2021) and Wang, Yang & Mao's (2020) testing, his project uses longer intervals between frames in the CSI data collected.

These longer intervals are a key feature of the sensing system that is implemented in this project but also likely increase the chance of variance between samples when compared to Schäfer et al.'s (2021) and Wang, Yang & Mao's (2020) testing. If something is moving in the signal path a longer time between samples will correlate to a larger physical change and it is also reasonable to assume that fluctuations caused by hardware and firmware will be more pronounced over time as well. Since the variance amongst samples is expected to be a higher, a courser filter with a shorter window length should be less likely to remove samples that may be significant to sensing but a length of at least 3 should be utilised as proved effective in Schäfer et al.'s (2021) and Wang, Yang & Mao's (2020) pre-processing.

4.1.2 Implicit Removal of Unwanted CSI Captures

When raw CSI data captured in testing was visualised in plots, two other phenomena were noticed that also create data unwanted for sensing. Some CSI captures contained data that was captured from a 20MHz transmission as opposed to the exclusively 80MHz transmissions intentionally generated for sensing. Others appeared to contain measurements not triggered by a transmission with most samples having zero magnitude and some with minor values just above the noise floor.

The source of the 20MHz captures is almost certainly another device not part of the test network attempting to start a communication session with the Pi 4. The captures containing no significant magnitude are most likely not transmissions sent on the test channel. It is suspected they are QoS frames transmitted by the Pi 4 to another WiFi network on a channel nearby to the test channel. This frame is misinterpreted by the Pi 3B+ and as there is some cross talk and induced noise on the test channel some non-zero subcarriers are captured and logged.

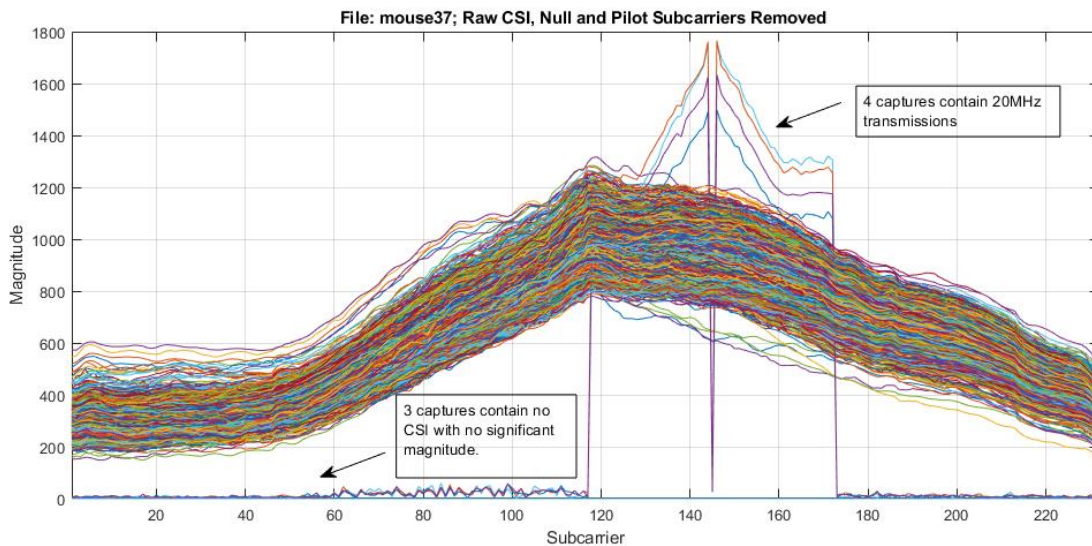


Figure 36. CSI data from test 37

Figure 36. plots the CSI captures from a test that contained four captures from 20MHz transmissions and three captures with no significant magnitude. The implementation of the Hampel filter is designed to remove outliers occurring in individual subcarrier measurements but, in these cases the entire capture should be discarded as it

will not contain any data useful for sensing. The most robust method trialled in this project to identify and remove these captures is to exclude all captures that contain over 50 subcarrier measurements (approximately one fifth of all subcarriers) with magnitude less than 3% of the mean of all measurements throughout the capture.

4.1.3 Pre-processing Methodology and MATLAB Code

The pre-processing methodology utilised in the project can be described by six steps:

1. Ingest Raw CSI from “.pcap” file.
2. Remove null and pilot subcarriers.
3. Remove any captures not containing measurements from 80MHz transmissions.
4. Separate CSI data into components; magnitude, amplitude, and phase.
5. Remove other outliers – apply Hampel filter to successive samples of each subcarrier.
6. Output CSI data as matrices of components; rows-captures, columns-subcarriers

The MATLAB code is used to implement these steps is listed below:

```
%-----  
%                               Ingesting CSI  
%-----  
%Ingests CSI data from tools developed by Gringoli et al. 2019  
%Requires functions from Gringoli et al. 2019 MATLAB CSI Reader:  
%  readpcap.m – called by “readCSI”  
%Avialbe at: https://github.com/seemoo-lab/nexmon\_csi  
%Only suitable for use with CSI captured from Broadcom 43455c0 WiFi NIC  
%-----  
%Packet Capture File read Parameters  
File = 'mouse15.pcap'; %file name of captured CSI data  
BW = 80; %either 20MHz, 40MHz or 80MHz bandwidth  
Max_UDP = 1000; %maximum number of UDP packets to read from capture file  
  
%Ingest CSI from .pcap decoder  
csi_raw = readCSI(File, BW, Max_UDP);  
%csi_raw is matrix of CSI samples from captured packets, columns contain  
%CSI data from each packet, rows contain CSI samples for each subcarrier  
  
%Arrange CSI in subcarrier order with centre frequency = subcarrier  
% index 0, as per 802.11ac  
csi_raw = fftshift(csi_raw,2);  
%%  
%-----  
%                               Pre-Processing CSI Data for Analysis  
%-----  
csi = csi_raw; %move to new array for processing  
%-----  
%Remove null and pilot subcarriers  
%For 80 MHz bandwidth, null outside of -122 - 2 and 2 to 122
```

```

%pilot subcarriers at +-(11, 39, 75, 103)

%Create list of Null and Pilot subcarriers
%List subcarriers to remove:
sub_remove=[1:6, 26, 54, 90, 118, 127:131, 140, 168, 204, 232, 252:256];
csi(:, sub_remove) = []; %delete subcarrier columns from raw CSI data
%-----
%Remove any outlier captures, captures not containing significant power to
%be considered valid 80MHz bandwidth captures should have all subcarriers
%removed, captures removed should be either 20MHz or 40MHz channel width
%or captures containing only noise

%Calculate 3% of mean power for all CSI data - 3% can be used as tuning
parameter
out_vu = mean(reshape(abs(csi.').^2,1,[])) * 0.03;
cap_remove = []; %list captures to be removed

    for index = 1:length(csi) %scan all CSI captures
        %number of subcarriers with power below 3% of mean
        outliers = sum(abs(csi(index, :)).^2<out_vu);
        if outliers >= 50 %remove when 50 or more subcarriers
            cap_remove = [cap_remove index]; %list captures to be removed
        end
    end

csi(cap_remove , :) = []; %delete capture rows from CSI data
%-----
%Filter CSI from outliers using the Hampel filter
%separate CSI data into separate vectors of magnitude, amplitude and phase
%filter is applied separately
%Filter is applied to successive samples of each subcarrier - column wise
%for CSI data matrix

%tunning parameters
w_l = 4; %window length
t = 3 %scalar of standard deviation used for the Hampel identifier

csi_mag = hampel(abs(csi), w_l, t); %processed magnitude CSI data
csi_amp = hampel(real(csi), w_l, t); %processed amplitude CSI data
csi_phase = hampel(imag(csi), w_l, t); %processed phase data
%%
%-----

```

Visualising the raw versus pre-processed CSI data is an effective way to verify the pre-processing method.

Figure 37. plots all CSI magnitudes in data collected during the 37th test conducted during the project. The data collected in the 37th test had a higher than average rate of outliers. Of the 264,000 CSI measurements logged 39,006 were either removed or modified by the Hampel filter during pre-processing. The magnitudes of the processed CSI data are plotted in Figure 38.

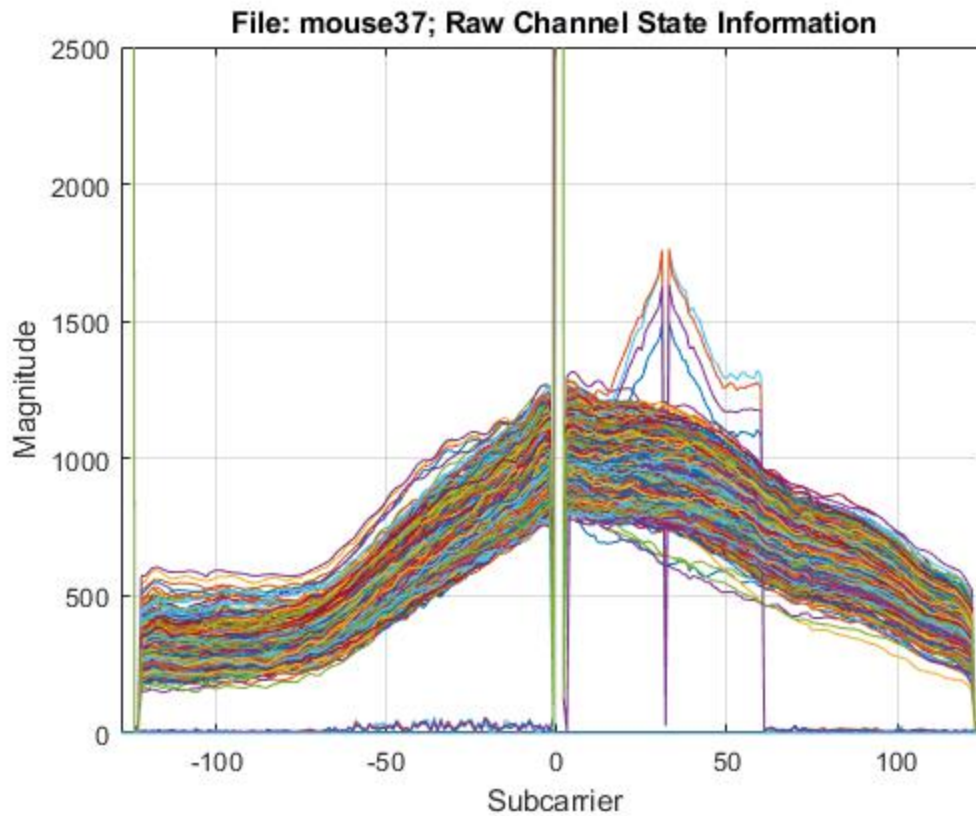


Figure 37. CSI data from test 37, all raw data before pre-processing

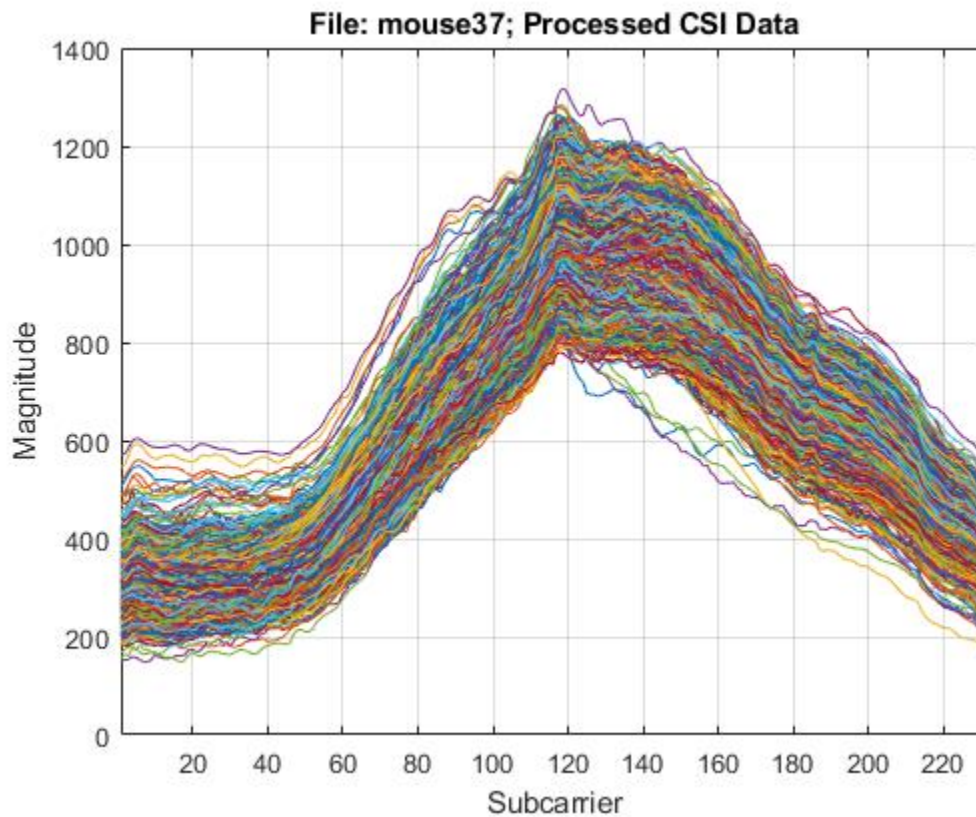


Figure 38. CSI data from test 37, processed data

4.2 Utilising CSI Data for Sensing

Once CSI data has been pre-processed it can be investigated for features that are dependent on the environment within the signal path. The phase and amplitude of the CSI data are impacted by slightly different characteristics within the signal path. The amplitude of CSI quantifies the attenuation of the signal path for the relevant subcarrier. Any material within the signal path will attenuate the signal to some extent but of particular significance to vermin sensing is moisture, as the bodies of mammals contain significant amounts of water. Water absorbs microwaves considerably more effectively than most other materials. Multipath fading will also cause changes in the amplitude of CSI and the occurrence of multipathing is influenced by movement within the signal path (Liu, Wang & Deng 2021).

The phase of CSI measurements are a representation of the phase response of each sampled subcarrier. The phase response of each subcarrier is more likely to contain information about the signal path than amplitude data (Liu, Wang & Deng 2021). Comparatively minor changes in the signal path that cause reflection and refraction will cause changes to the phase response. However as shown in Figure 27, the phase between subcarriers also changes periodically and this periodic change is not synchronised between the transceivers that generate each CSI capture causing a pseudo random shift between captured frames. Experiments have shown that the phase of CSI data is more sensitive to changes in the signal path than amplitude (Zeng et al. 2014). However, phase data is also impacted so much by variations caused by the hardware and firmware via carrier frequency offset, phase-locked-loop initialisation, unsynchronized frequency oscillators between transceivers etc. it is considered unsuitable for sensing when collected from commodity WiFi devices as it is not practical to separate this distortion from changes caused by the signal path (He et al. 2020).

The amplitude (real) component of CSI data will also exhibit the periodic phase shifts created by the WiFi hardware. But, by utilising the magnitude of CSI measurements, periodic phase shifts are normalised so that variations in magnitude are reflective of the gain of the signal path of each subcarrier and there is no need to account for the phase shift between captured frames. For this reason, a recent feasibility analysis of fall detection systems based on CSI extracted from commercial WiFi devices identified magnitude as the most effective input to sensing algorithms (Guo et al. 2023). Figure 39, examines the 500th frame captured from test 37.

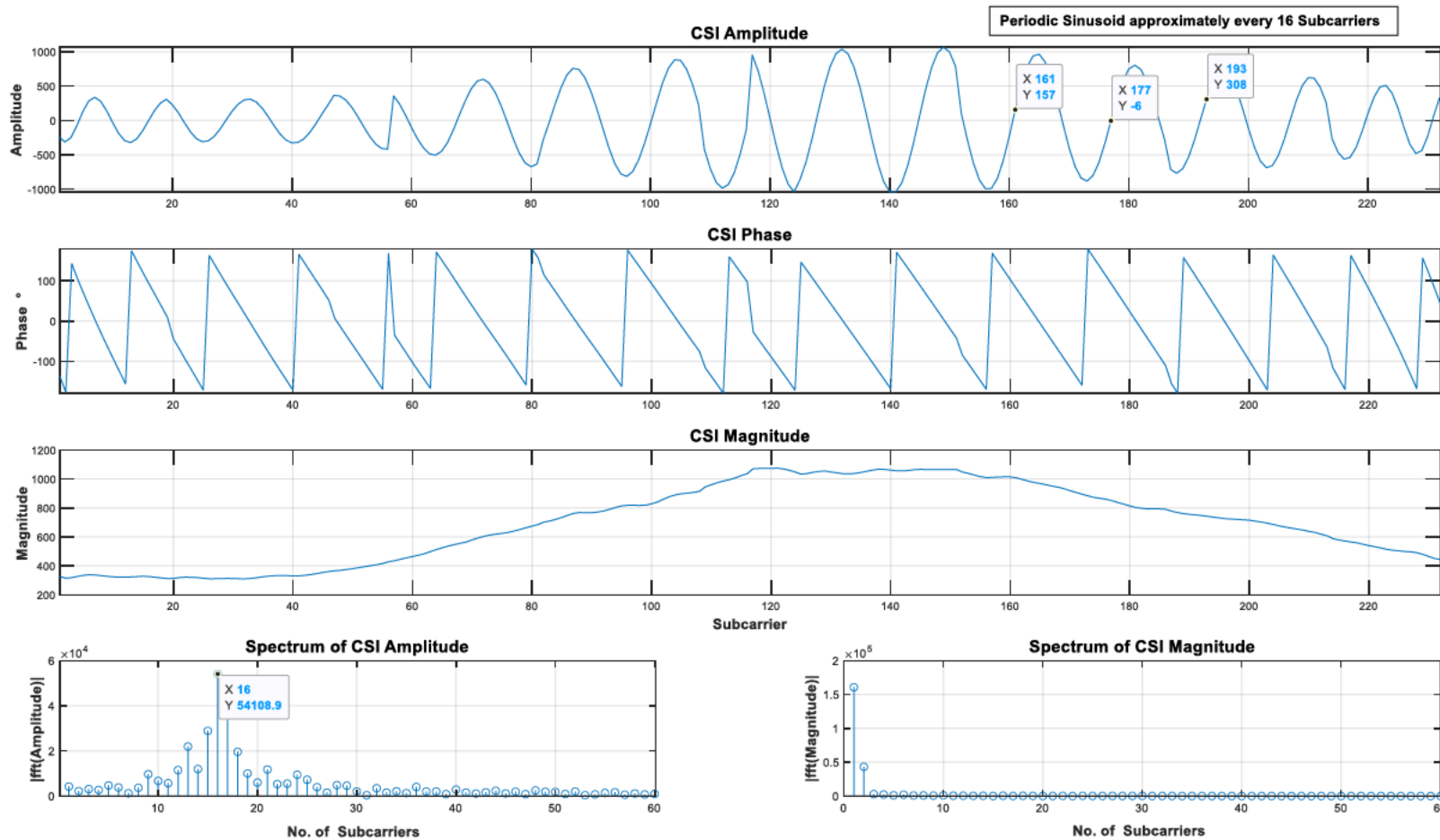


Figure 39. CSI frame components and Fourier analysis of amplitude and magnitude for frame 500 from test 37

Each CSI parameter is plotted against the processed subcarrier index (null and pilot removed). The phase shift is mostly uniform throughout the subcarrier index with two distinct phase changes around subcarriers 55 and 115 and some minor fluctuations. The periodic phase change creates a sinusoid in the amplitude measurements. Fast Fourier Transform analysis of the amplitude reveals a significant Fourier component after 16 subcarriers. However, there is no periodic waveform across subcarriers in the magnitude measurements. This enables comparison of magnitudes between CSI frames without need to reference the phase difference between frames. CSI magnitude should thus be considered the most effective input for sensing when it is not possible to isolate phase changes caused by the signal path from phase changes caused by hardware and firmware.

There is a noted inconsistency in terminology between amplitude and magnitude of CSI data in the published literature, for example what is referred to as CSI magnitude in this project, is referred to as “amplitude” by Wang, Yang & Mao (2020), Zhang et al. (2020) and Ibrahim & Brown (2021). However, Schäfer et al. (2021), Ma (2019) and Guo et al. (2023) use similar terminology to this project and refer to “magnitude”. This inconsistency is likely exacerbated by varying extraction formats from different tools. Since the tools used in

this project extract CSI in Cartesian form, CSI amplitude and magnitude reference Cartesian form nomenclature. It is important to note though that what is referred to as CSI magnitude in this project may be referred to as CSI amplitude in another sensing research.

4.3 Features Created in CSI from Vermin in Signal Path

For a mouse to be detected reliably by a sensing system, it must cause changes significant enough in the magnitude measurements that they are detectable in comparison to the constant fluctuations caused by all sources of noise. Then, if detectable the effect the mouse has on the magnitude measurements must create a unique signature that can be recognised against the signature created by other stimuli in the signal path. This signature can be termed a CSI fingerprint (Liu, Wang & Deng 2021). To determine if vermin would be a feasible sensing target for a WiFi sensing system the initial testing needs to establish that a mouse can be detected when the presence of a mouse is the only change to the signal path. If the mouse can be detected with high confidence and the changes in the CSI data are significant when the mouse is present, there is the potential that creating vermin sensing system is feasible.

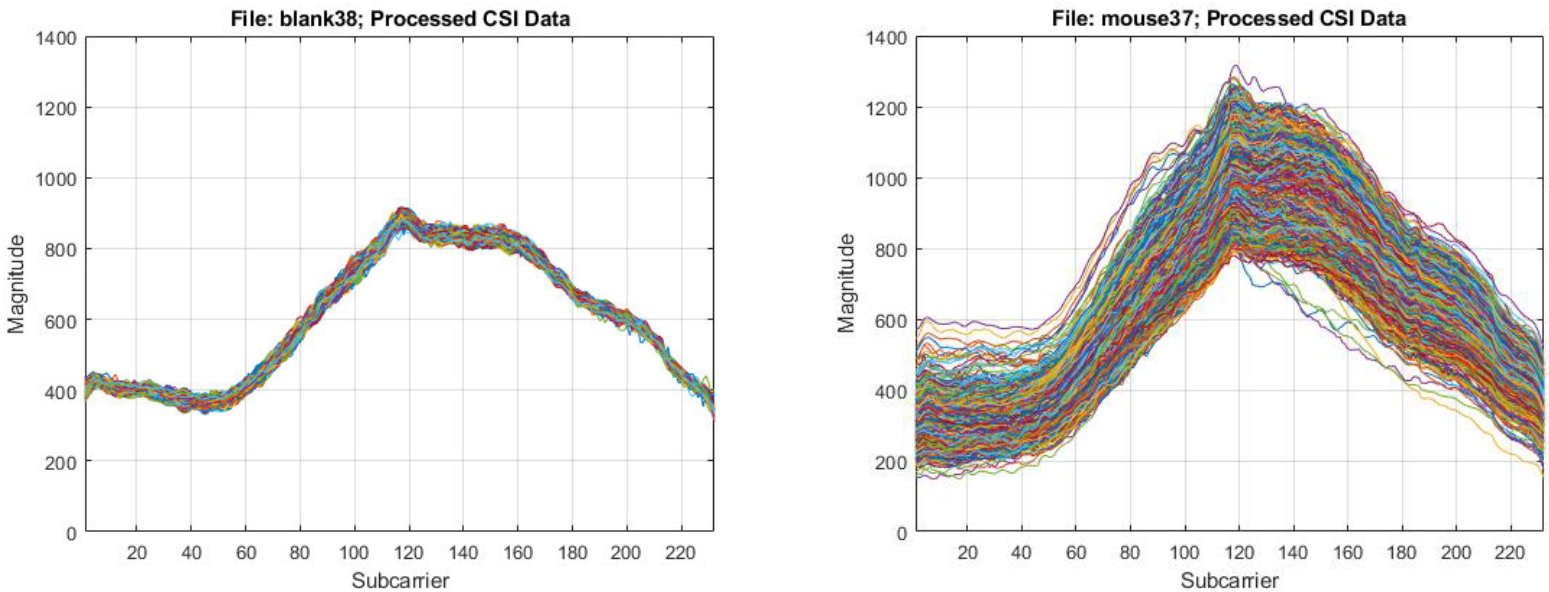


Figure 40. Comparison of CSI data, left control test, right test with mouse

The two tests that yielded the data plotted in Figure 40. were conducted on the same day within several minutes. The plots show CSI magnitudes after pre-processing. The left plot is a control test, and the right had the mouse in the signal path. There is a significantly higher variance in the magnitude of the captured CSI samples when the mouse is in the signal path. This difference can be visualised more conspicuously by examining the fluctuation in the power in each captured frame. A scalar quantity that is indicative of the power in each frame can be calculated by:

$$P_{pf} = \sum_{i=1}^I |CSI_i|^2$$

Where P_{pf} is a scalar quantity representative of the power in each captured CSI frame, i is the subcarrier index and $|CSI_i|$ is the magnitude of each CSI sample.

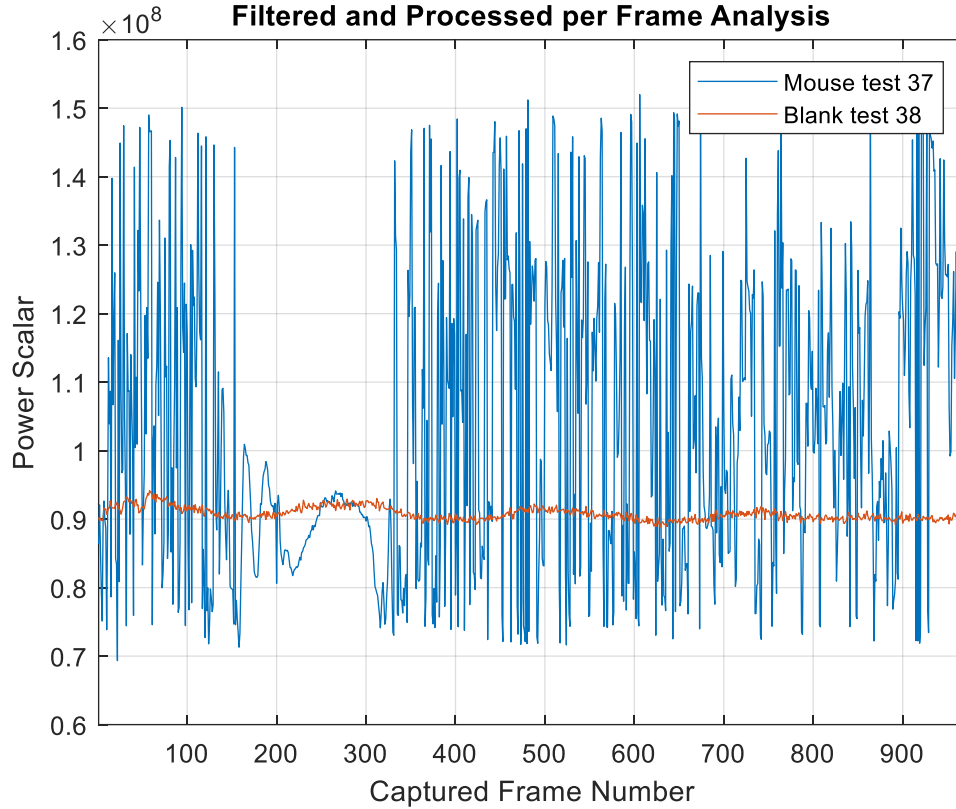


Figure 41. Power per frame comparison of Tests 37 and 38

The impact the mouse has on the CSI data when in the signal path is very apparent when comparing the power per frame to a blank control test. In the data captured with the mouse in test 37, between frames 200 and 300 there is a notable stabilising of the power per frame. Since both tests were conducted with a 10ms ping interval time, this period is approximately 1 second. Since fading is highly dependent on movement, this likely represents a period where the mouse kept still (Liu, Wang & Deng 2021). It is difficult to closely observe the mouse during testing as being close to the signal path would cause interference. Five tests were conducted with the mouse in the signal path and a smartphone camera in a fixed position filming the mouse. In all filmed tests the mouse was never completely still (always sniffing, twitching, or preening to some extent) but at times did briefly stay in the same location in it's enclosure.

4.3.1 Observations of the Mouse in Filmed Testing

In Test 73 the mouse was filmed within the signal path and the CSI data captured is visualised in Figure 42. During periods when the mouse was moving less, there is a noticeable reduction in the distribution of power in the captured CSI data. The CSI capture was 10 seconds in duration with 10ms interval pings, triggering measurements. The duration of the test is approximated in the x-axis of Figure 42. When the capture begins the mouse emerges slowly from the tube it houses in, then walks quickly to one corner of the enclosure which takes approximately 3 seconds. Then the mouse pauses for about 0.5 second before moving to it's food bowl. At its food bowl it pauses again briefly about 7.5 seconds into the test, then begins looking around while making small stuttering movements.

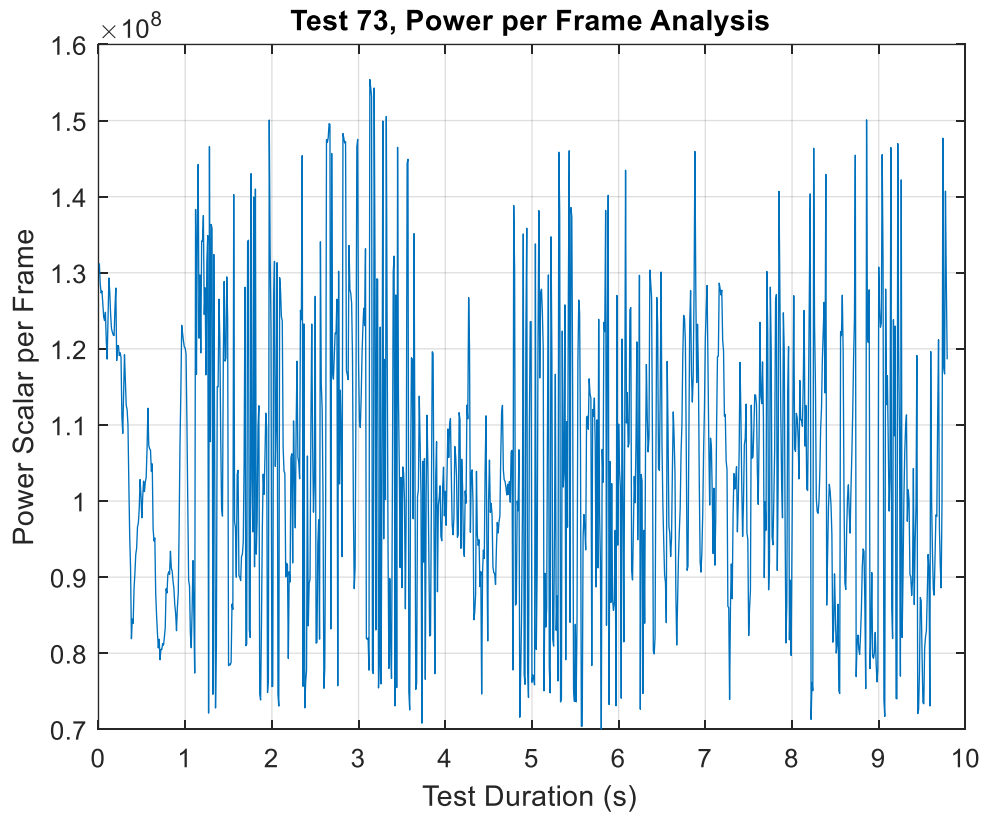


Figure 42. Results from filmed test 73, Power vs Approximate Test Duration

4.3.2 Variance in Power Distribution

In all tests conducted with the mouse in the signal path there was a similar amount variance in the distribution of power within the spectrum. Variance appears to fluctuate based on how much the mouse moved around during the test but was always significantly and conspicuously higher than the variance in the distribution of power during a blank control test. On the 14th of August 2023, a testing session was conducted that yielded 20 CSI captures suitable for analysis. 10 containing the mouse in the signal path and 10 blank control tests with a similar signal path but without the mouse present. Figure 43. plots the variance between the power in captured frames (after pre-processing) in each test. The variance is normalised between 0 and 1 to aid visualisation.

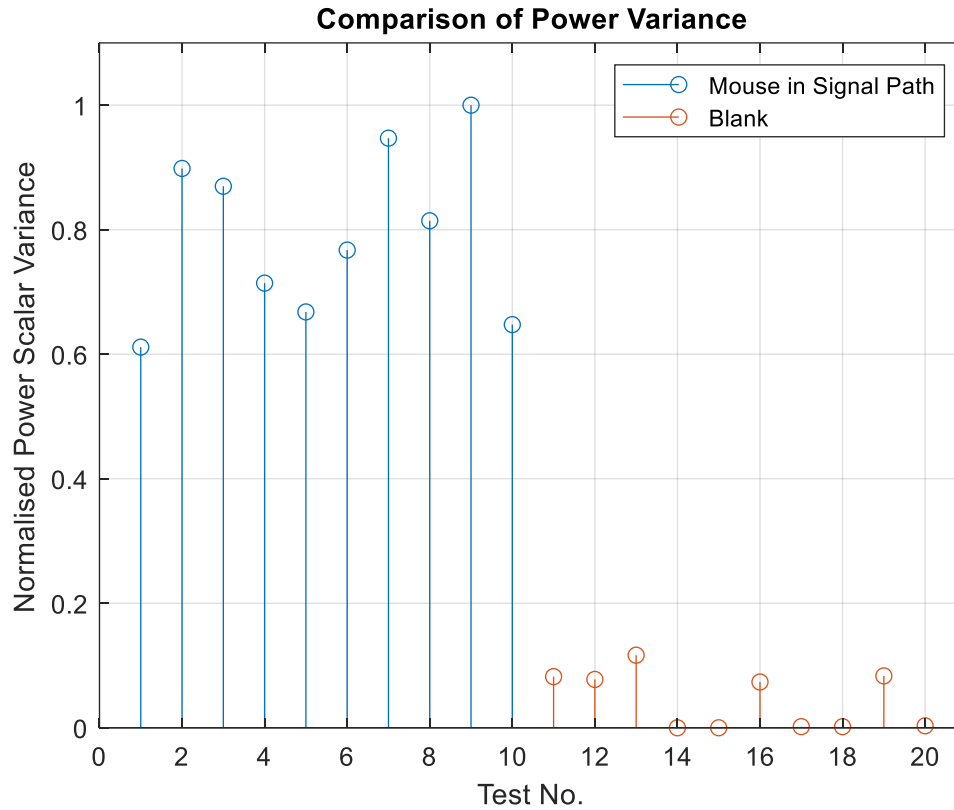


Figure 43. Variance of power per frame throughout 20 Tests

Some blank control tests that had only static objects in the signal path still show a significant increase in variance in comparison to other blank tests. This was always caused by the same phenomena and creates features in the CSI data distinct from those caused by the mouse. Unlike when the mouse is moving within the signal path, the CSI magnitudes always exhibit tight grouping. However, in some tests there were discrete groups of magnitudes centred around different values. The left side of Figure 44. plots the processed CSI data from test 16 (one of the tests yielding data for the plot in Figure 43.), and the right side plots the most extreme example of this phenomenon captured in all testing throughout the project.

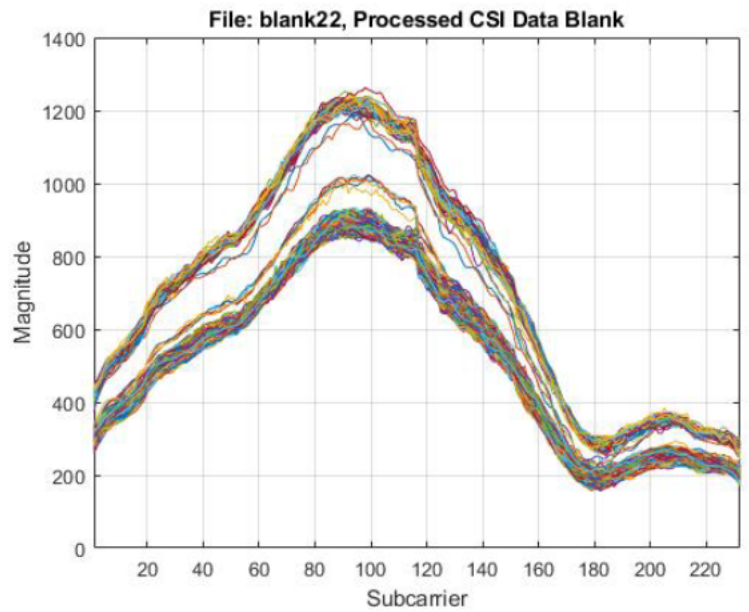
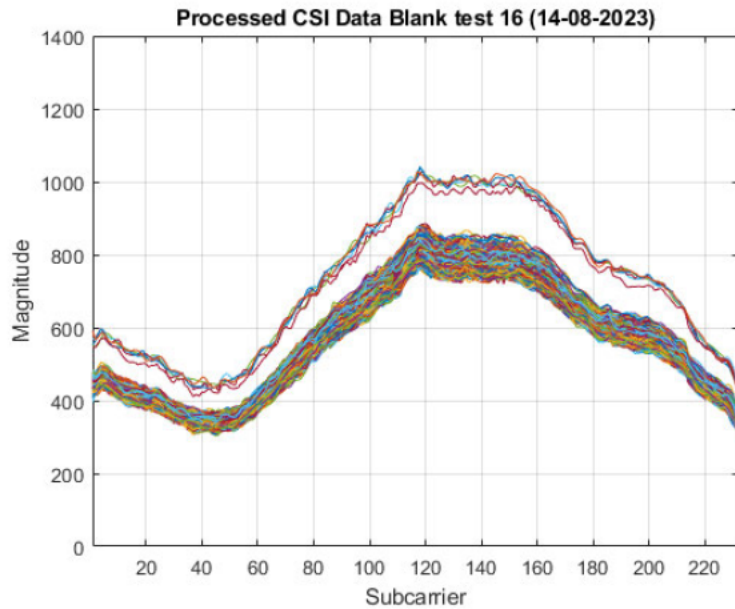


Figure 44. Blank test data containing split magnitude grouping

Split grouping of magnitudes only occurred in around 5% of blank tests and no correlation between the occurrence of split grouping and any variables in the test could be identified. Split grouping can be attributed to two potential causes, either beamforming, which can change the direction of propagation and gain through the signal path to improve the communication session or as the TP-Link router has multiple antennas it may be that the communication session is hopping between antennas on the TP-link router to load share (Gast 2013). It is an example of how features of WiFi networking designed to optimise performance can impact sensing. Split grouping was not identified in any tests where the mouse was in the signal path. However, it is possible that the fluctuations seen are in part a result of beamforming or antenna hopping as opposed to being fluctuations caused directly by the mouse itself via interactions with the RF signal.

4.3.3 Changes between Subcarrier Measurements

Examining the power distribution across entire frames is an effective way to visualise large volumes of CSI data and demonstrate that the mouse has a significant impact on the CSI data. More subtle features can be found by examining the measurements of individual subcarriers across clusters of successive frames. The amount different materials will affect the gain response of the signal path varies across the spectra (Tan, Zhang & Yang 2018). By isolating subcarriers, the effect the signal path has on a discrete frequency component is examined as opposed to examining the entire channel. It also normalises the fluctuation across subcarriers. All frames in each capture show a repetitive shape across the subcarrier index that appears to be independent of the signal path. In all captures the magnitude tends to rise around the middle subcarriers sometimes with minor peaks towards the lowest and highest subcarriers. Figure 44. demonstrates two typical shapes. This pattern is assumed to be determined by features of the hardware and firmware mostly likely caused by filtering of the channel as less attenuation is seen in the centre of the channel. Regardless of the exact cause of the shaping of CSI frames,

analysing individual subcarrier measurements across different frames avoids the significance of changes being amplified by the subcarriers position.

Figure 45. examines and compares data from two tests. One test conducted with the mouse in the signal path, with the CSI data labelled: “mouse39” and one blank control test with the CSI data labelled: “blank41”. Nine subcarriers spread evenly across the subcarrier index are analysed. In the right-side plots of Figure 45., the magnitude of each subcarrier over 100 measurements is plotted with the x-axis representing the CSI capture packet index and the y-axis subcarrier magnitude. The vertical lines in the two left side plots of Figure 45. show the location of the subcarriers analysed in the right-side plots.

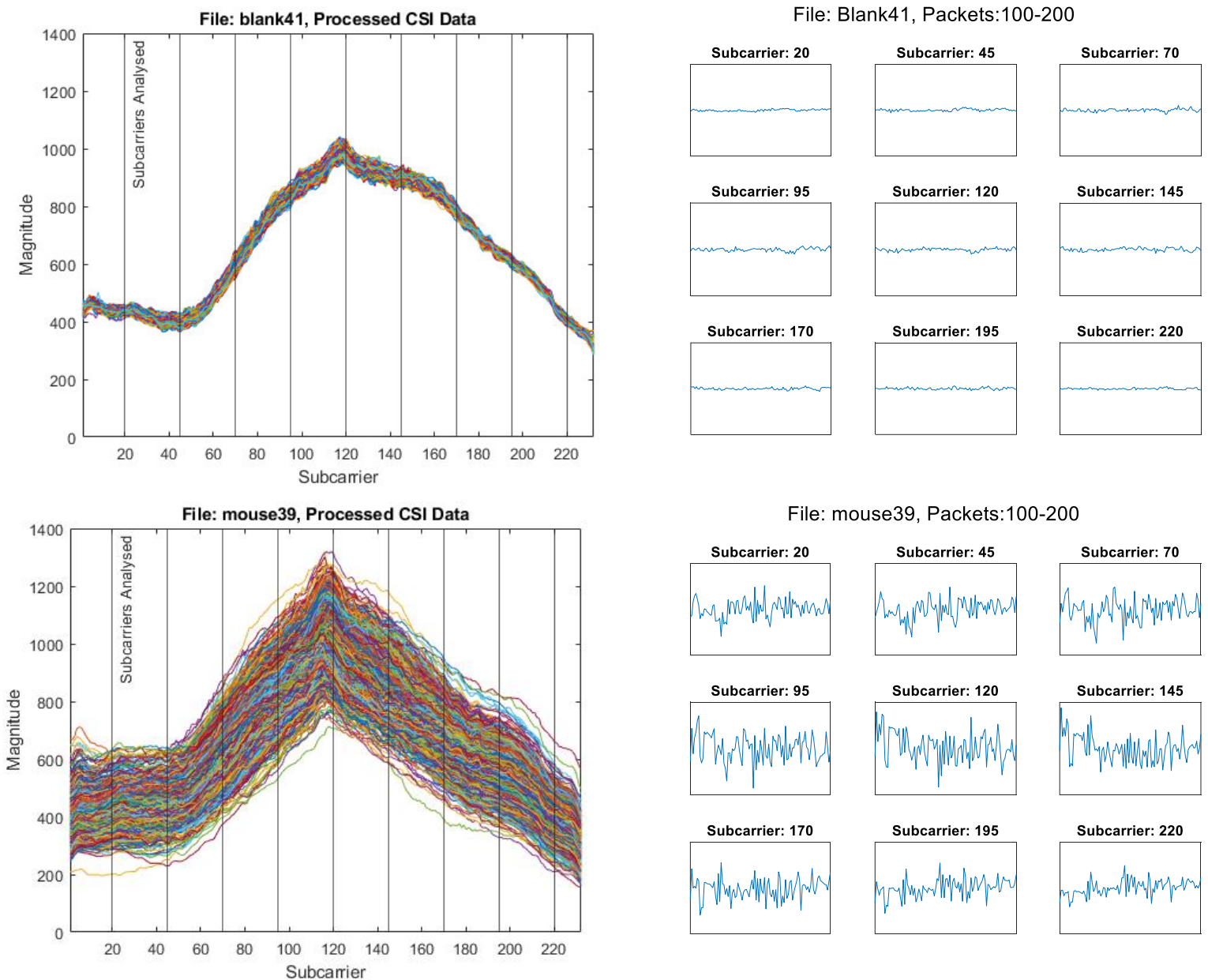


Figure 45. Subcarrier analysis, Mouse vs Blank Control

Not all subcarrier measurements increase and decrease consistently between captured packets, demonstrating that the spectra used within the channel is not uniformly affected by the signal path. It is also notable that subcarriers closer together exhibit more similarities. For example, subcarrier 20 and 45 from “mouse39” in Figure 45, contain mainly similar features but subcarriers 20 and 220 show no obvious similarity. This highlights how varied regions of the RF spectrum will interact with the signal path differently and is why CSI measured from a wider channel can potentially discern more information about the signal path as Tan, Zhang & Yang (2018) proposed when attempting to create 600MHz sweep in the sensing system they designed. By examining variations in measurements between individual subcarriers there is more potential to identify any unique features that would be created by vermin.

4.4 Constructing a Sensing System

The testing undertaken demonstrates that a mouse does have a significant influence on CSI extracted from commodity WiFi devices. Figure 43, suggests the mouse could be reliably detected in the CSI data collected in this project by simply identifying an increase in CSI magnitude. One of the challenges when attempting to create a useful system that will identify vermin in a wider range of scenarios and not generate false positives is that any movement, especially from materials that interact strongly with microwaves will cause changes to the gain response of the signal path. The tests undertaken in the project involved close antenna placement with the mouse directly in the LOS signal path. When vermin are at the extremities of the range of the WiFi network or not in the LOS signal path, the effect on the CSI measurements will almost certainly be less significant. Gathering CSI data from testing with varied antenna placement and other moving stimuli in the signal path are beyond the scope of this project. However, using the data gathered a method is proposed that could form the basis of a sensing system that could be trained to identify vermin in a wider range of scenarios.

The first stage requires dividing the CSI data into segments representative of a time window to enable block by block analysis of the CSI data. Then, features extracted from each individual block will be used as the input to a machine learning algorithm. The statistical features used by Zhang et al. (2020) (see Table 1.) will be used as well as certain features from the WiFi-Based Intrusion Detection System proposed by Tain et al. (2018).

4.4.1 Block by Block Analysis and Feature Extraction

Analysing the CSI data in discrete blocks enables the sensing system to classify the events occurring in the signal path within a time window. A block size of 50 CSI measurements, representing a 0.5 second time window given the 10ms ping time used in testing, will be utilised. The coefficient of variation defined by Tian et al. (2018) was shown to be an effective metric to classify movement within the signal path and can be adapted to suit the CSI data collected during the testing in this project. It is calculated by first finding the auto covariance matrix \mathbf{C} , of the CSI block:

$$\mathbf{C} = [\text{cov}(|\mathbf{H}_{\Delta T}|, |\mathbf{H}_{\Delta T}|)]_{m \times m}$$

Where $\mathbf{H}_{\Delta T}$ is the CSI matrix captured in the time window and m is number of subcarriers.

Then to find the coefficient of variation ω , the square root of the maximum eigen value of the covariance matrix is divided by the mean of CSI magnitudes $|\mathbf{H}_{\Delta T}|$:

$$\omega_{\Delta T} = \frac{\sqrt{\max(\text{eigen}(\mathbf{C}))}}{\text{mean}(|\mathbf{H}_{\Delta T}|)}$$

(Tian et al. 2018)

The coefficient of variation can be utilised as a feature of the CSI block for input to machine learning processes, but Tian et al. (2018) also demonstrated that the ratio of coefficients of variation between two adjacent CSI blocks is effective way of determining if the amount of movement within the signal path changed significantly between blocks.

$$R = \frac{\omega_{\Delta T}}{\omega_{\Delta T-1}}$$

(Tian et al. 2018)

4.4.2 MATLAB code for block by block Processing and Feature Extraction

```
%-----
%                               CSI Feature Extraction
%-----
%Input pre-processed CSI data for feature extraction. Extracts features
%from blocks of CSI data intended for input to sensing system.
%-----
%Load Processed CSI data, magnitudes only
load("blank45_csi.mat");
csi = csi_mag;
%File Name for Extracted Features
feat_name = "blank45_feat.mat";
%-----
%Block Processing of CSI Data
block_len= 50; % number of packets per block
no_blocks = floor(length(csi)/block_len); %number of blocks to process
block_index = 1; %initialise block index

%Select Subcarriers for Analysis
sub_ana = [20 40 60 80 100 120 140 160 180 220]; %subcarriers for analysis

feat = []; %array to store features extracted from CSI

for block_no = 1:no_blocks %process CSI data block by block
    feat_block = []; %array to store features extracted from the block

    %extract block from CSI data
    block = csi(block_index:block_index + block_len-1, :);

    C = cov(block); %calculate autocovariance matrix
    eig_max = max(eig(C)); %find the maximum eigenvalue

    uA_block = mean(block, 'all'); %mean of all CSI measurements
    wT(block_no) = sqrt(eig_max)/uA_block; %coefficient of variation
```

```

feat_block = [feat_block wT(block_no)]; %store features

%add additional block feature calculations here

%Analyse Subcarriers
for sub_i = 1:length(sub_ana) %iterate for the number of subcarriers
    %extract subcarrier measurements
    sub_block = block(:,sub_ana(sub_i));

    %Feature Extarction
    u_sub = mean(sub_block); %mean
    feat_block = [feat_block u_sub];%store features

    med_sub = median(sub_block); %median
    feat_block = [feat_block med_sub]; %store features

    std_sub = std(sub_block); %standard deviation
    feat_block = [feat_block std_sub]; %store features

    skew_sub = skewness(sub_block); %skewness
    feat_block = [feat_block skew_sub]; %store features

    kurt_sub = kurtosis(sub_block); %kurtosis
    feat_block = [feat_block kurt_sub]; %store features

    %add additional subcarrier feature calculations here

end
%store all features extracted from the block,
% rows-blocks, columns-features, 1st column-coefficient of variation
feat(block_no ,:) = feat_block;

block_index= block_index + block_len; %move index to next block
end
%-----
save(feat_name, "feat")
%-----

```

4.4.3 Sensing via Machine Learning

The earliest well documented instance of utilising machine learning algorithms for CSI sensing was in 2014 when Han et al. (2014) proposed a human activity classification system based on a one-class Support Vector Machine (SVM) (Han et al. 2014). More recently Neural Networks have been proposed as the most effective machine learning technique for CSI sensing system (Damodaran et al. 2020; Zhang et al. 2020; Schäfer et al. 2021). Neural Networks are a machine learning process that is loosely based on a biological brain and are generally considered to be optimal for pattern recognition problems but have many different types of architectures that are suitable for a variety of applications. The Long Term-Short Term Recurrent Neural Network (LTSM-RNN) architecture is commonly used in CSI sensing research (Damodaran et al. 2020; Zhang et al. 2020; Schäfer et al. 2021). LTSM-RNN is considered the best architecture for handling time series data where temporal ordering is significant (Ma 2019). As a proof of concept, a Neural Network optimised for

pattern recognition will be used as a machine learning algorithm to classify blocks of CSI data collected in this project as either “blank” or “mouse”. Features extracted after block by block processing are used to train the Neural Network. After training the features from CSI data can inputted to the Neural Network and the network will output predicted classification in the form of weighted probabilities for each class.

4.4.2 The MALAB code to Implement Neural Network

```
%-----
%                               Neural Network CSI Sensing System
%-----
%Ingests features extracted from CSI data, trains Neural Network to
%classify signal path. Proof of concept implementation specifies two
% classes - Mouse and Blank.
%
%Input data is matrix, with format:
%   Row 1: integer class identifier e.g. 1 = Mouse, 2 = Blank
%   Rows 2-end: Variables - features of CSI data
%   Columns: Observations - blocks of CSI data
%-----
%                               Training Data Preparation
%-----
%Load Features Extracted from CSI data
csi_data = load("csi_features.csv");

%Extract Class Identifier
class = csi_data(1, :); %class identifiers
csi_data(1, :) = []; %remove class identifier

[no_blocks, no_feat] = size(csi_data); %dimensions of CSI data

%Number of Classes - Additional classes must have labels added
no_class = max(class); %number of classes
class_labels = [ "Mouse", "Blank"]; %class labels

%Create Matrix of Target Classes
% Rows: Classes
% Columns: 1=belonging to class, else 0
sp_class = zeros(no_class, no_blocks);

%Assign Classes to Target Matrix
for index = 1:no_class
    sp_class(index, (class==index)) = 1;
end
%-----
%                               Neural Network
%-----
%Initialise Neural Network
%10 hidden nodes, training method: Variable Learning Rate Gradient Descent,
%Cross Entropy performance evaluation
net = patternnet(10, 'traingdx', 'crossentropy')

%Portion Input data for Training, Validation and Testing
net.divideParam.trainRatio = 70/100 % 70% Training
```

```

net.divideParam.valRatio = 15/100 % 15% Validation
net.divideParam.testRatio = 15/100 % 15% Testing

%Use the Softmax Transfer Function
net.layers(Gringoli et al.).transferFcn = 'softmax';

```

```

%Train Network
net = train(net, csi_data, sp_class)

```

```

%-----
%Predications can be made using the Neural Network to perform sensing
%classification by calling the "net" function and parsing CSI features in
%same format as the training data.
%-----

```

The Neural Network was trained with features extracted from CSI data captured from 30 tests analysed in blocks 50 frames long, with features extracted from 10 subcarriers disturbed evenly across the subcarrier index. This translates to information discerned from approximately 7 million captured CSI frames. To reduce the chance of biasing the network the ratio of mouse to blank test data used to train the network was 50:50. To examine the performance of the network data from mouse and blank control tests not used for training were inputted to the network after pre-processing and feature extraction. The resulting predictions are plotted in the confusion matrix in Figure 46.

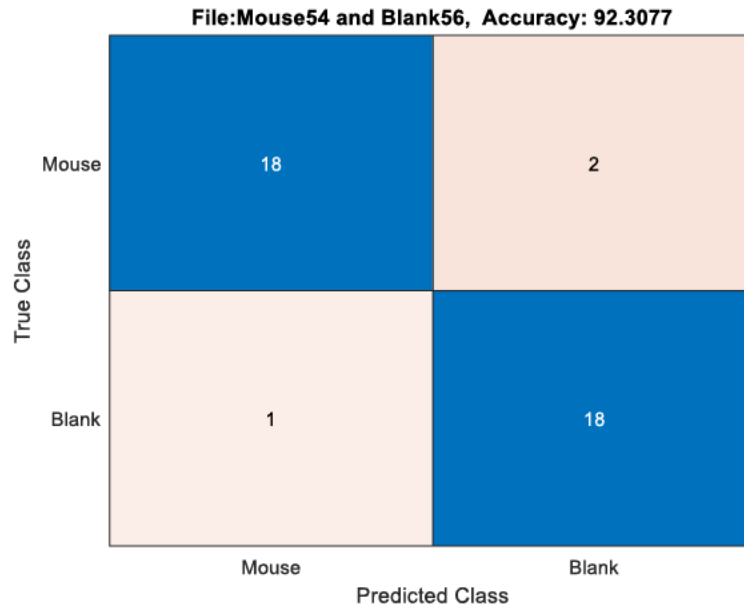


Figure 46. Confusion matrix from Neural Network test

The Neural Network was able to correctly predict the contents of the signal path with high accuracy. A block of CSI data misidentified as being blank when the mouse was in the signal path was captured between frames 400 and 450 of test 54. Figure 47. plots the power per frame of the section of CSI data around frames 400 to 450. There is a clear reduction in the distribution of power between frames 400 and 450 which most likely represents a time when the mouse was not moving around within the signal path.

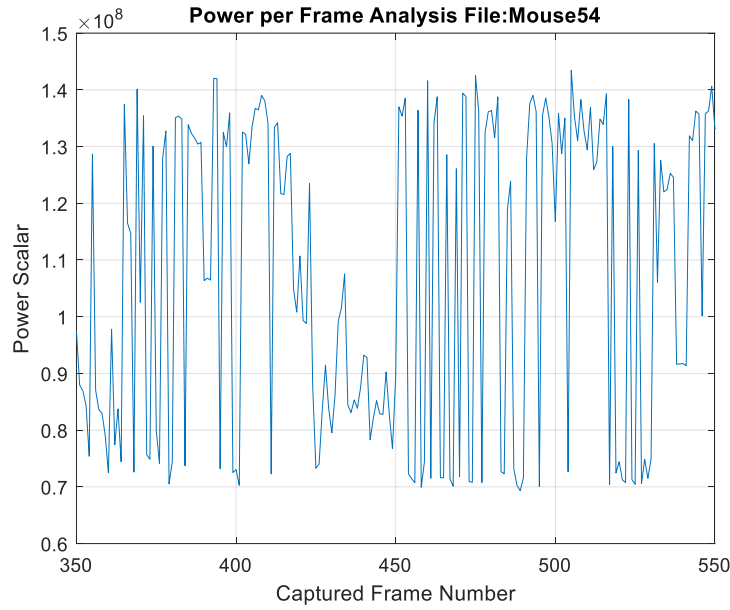


Figure 47. Analysis of misidentified CSI data from mouse

All blocks misidentified as containing the mouse when the CSI data was extracted from a blank control signal path exhibit split grouping of magnitudes. The right side plot of Figure 48. shows the magnitudes from a correctly identified block and the left side of Figure 48. is a block misidentified as containing the mouse by the Neural Network.

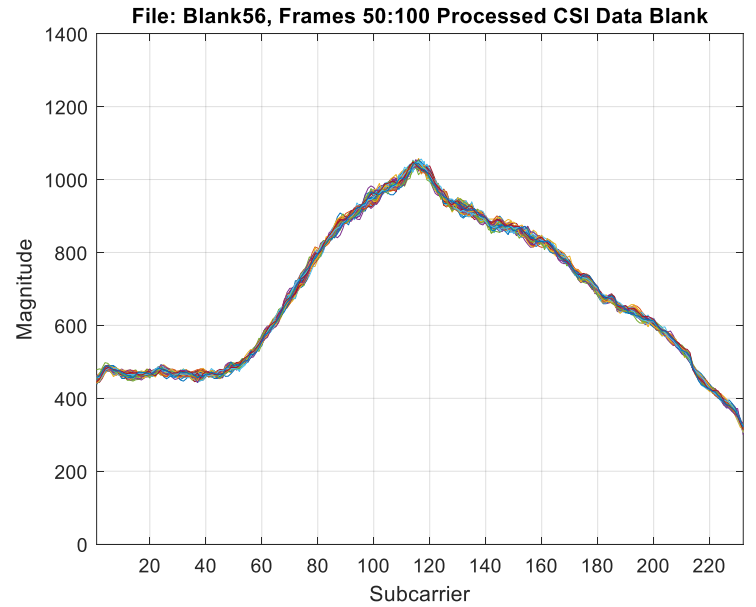
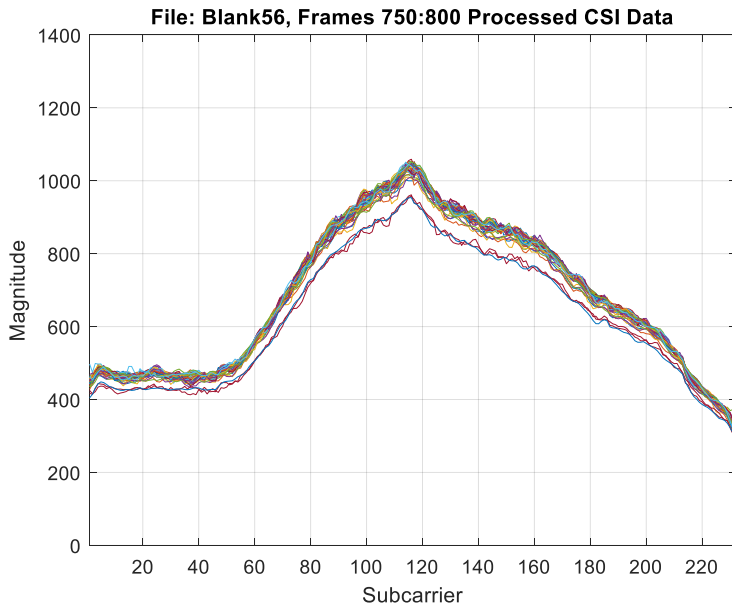


Figure 48. Correctly identified blank control CSI block (right) and misidentified block (left)

The implementation of the Neural Network is intended to demonstrate how a sensing system can be implemented using machine learning and was not heavily refined and optimised. Refining should always aim to remove unwanted captures and remove or rationalise outliers and anomalies prior to inputting data to a Neural Network or other machine learning algorithm. For example, assuming split grouping of magnitudes is caused by antenna hopping on the TP-Link router, a more feature rich CSI extraction system than the one used in this

project could identify which pair of antennas generated the CSI measurement. This information could then be used to categorise the different signal paths between different pairs of antennae generating CSI measurements and analyse them as individual streams.

4.5 Components of the WiFi Sensing System

Refining the Neural Network or utilising other machine learning techniques that may yield higher classification accuracy is only one aspect that contributes to improving the efficacy of the entire sensing system. The implementation of the Neural Network as a classification algorithm forms the final component of the sensing system developed in this project. The complete sensing system implemented in this project can be conceptualised as four main elements as shown in Figure 49. The input to the system is the data from an operating WiFi network and the output is whether vermin are detected.

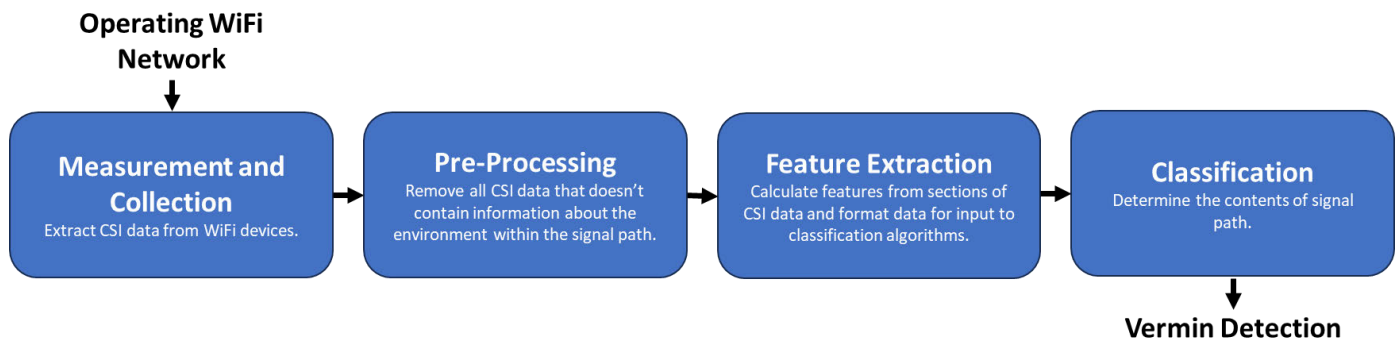


Figure 49. Components of sensing system

While all elements of the system are interdependent, they are individual subsystems with separate inputs and outputs. Improvements, refinements, and changes can be made to individual elements without redesigning the entire system. With reference to Figure 49, a high-level overview of how each element of the sensing system was implemented in this project is listed:

- **Measurement and Collection:** A WiFi network operates between a TP-Link router and Raspberry Pi 4. A Raspberry Pi 3 extracts CSI measurements from the Pi 4 and outputs the CSI data in .pcap files.
- **Pre-Processing:** CSI data in .pcap files is parsed to MATLAB where pre-processing occurs. Unwanted subcarriers and captures are implicitly removed then Hampel filtering removes outliers.
- **Feature Extraction:** Processed CSI data is portioned into temporal blocks and statistical features from each block are calculated.
- **Classification:** A trained Neural Network classifies the features and outputs the probability of the mouse being within the signal path.

5. Conclusion

The concept of embedding a sensing utility function in a WiFi network to detect vermin is feasible. The mouse created conspicuous features in the CSI data and significantly increased the distribution of power throughout the spectrum of the WiFi channel. After pre-processing and extracting features from temporal blocks of CSI data a Neural Network was able to reliably identify when the mouse was within the signal path. The design, configuration and validation of the equipment used for testing is thoroughly documented within this dissertation and provides context to the CSI data collected and ensures that the assembly of the test network is repeatable.

This was the first project proposing vermin as a potential target of a passive WiFi sensing system and the first project that collected CSI data from testing with mouse in the signal path of a WiFi network. It was demonstrated that by extracting CSI data from typical commodity WiFi devices, it is possible to identify when a mouse is present within the signal path. The network used for testing was also capable of performing sensing functions and normal communication functions in parallel, suggesting that it may be feasible to embed a utility vermin sensing function into a household WiFi network. The ping interval used to create a sensing pulse from which CSI measurements were taken was comparable to the rate at which WiFi APs generate beacon frames and the devices which generated the transmissions used for CSI data capture were not modified in a way that would prevent their normal use.

5.1 Reflection and Achievement of Objectives

The objectives of the project are listed below for reference and are also found in Section 1 and in the initial project specification accepted by the University of Southern Queensland in Appendix A. A reflection of the achievements of the project is listed below each of the original objectivities.

5.1.1 Project Objectives

1. Survey, review and analyse previous WiFi sensing research and experimentation. Conduct initial background research into using radio frequency signals, specifically microwaves for sensing and the operational aspects of WiFi networks which will affect sensing.

The literature review documented in Section 2 was used to identify the existing gaps in WiFi sensing research. Surveying previously implemented CSI extraction systems used for testing and experimentation was essential to determining the testing methodology used in this project.

2. Procure and configure WiFi hardware that will facilitate the extraction of CSI and parse the CSI data into a suitable software application e.g. MATLAB that can perform statistical analysis and implement detection algorithms.

The choice of hardware was justified, and the test network used to extract CSI measurements for this project was implemented successfully and the data captured was ingested into MATLAB.

3. Design a test apparatus that simulates a WiFi network, where stimuli can be placed in the signal path including vermin (mice) to capturing and log CSI.

Objectives 2 and 3 were the most challenging and consumed a significant portion of the time resources available to the project. There were significant lead times to procuring Raspberry Pi's due to global shortages and high demand. Designing, constructing, and validating the test apparatus used for CSI extraction was challenging. The difficulty in accessing CSI data without support from manufacturers and vendors of WiFi equipment is a significant obstacle to undertaking testing and experimentation to develop WiFi sensing systems. The configuration and validation processes detailed in Section 3 of this dissertation will contribute to making CSI data more accessible.

4. Gather data from testing that can be used to examine the feasibility of using WiFi sensing as a vermin detection system.

The ethical obligations required of testing with live animals were met. The mouse used in testing was not interfered with and remained within its usual habitat. The data gathered from testing was sufficient to determine that a mouse can be detected using CSI extracted from typical commodity WiFi devices.

5. Determine if it is possible to detect vermin via CSI what limitations and constraints may impede the development of a system intended to be used as an additional utility function in a typical WiFi communication network.

The key challenges and constraints to using CSI data and commodity WiFi devices for sensing that were exposed during the project are detailed and further work is suggested in the next section of this dissertation to determine to what extent they can be overcome. The project was completed successfully, and all objectives were met. The continuation of research into passive WiFi sensing could eventually enable the deployment of sensing utility functions into WiFi networks and this project demonstrates that detecting vermin with WiFi is feasible.

5.2 Further Work

The testing methodology in this project can be expanded to include a wider range of scenarios. This could include test scenarios with varied antenna placement and signal path arrangement as well as with other stimuli in the signal path, for example, a human and mouse simultaneously. Distinguishing the mouse from other stimuli

reliably will almost certainly require further feature extraction and more complex classification techniques. Aside from the continuation of testing, further work is needed to enhance the capability of the sensing system.

Lin et al. (2020) demonstrated that interference height estimation could be used to classify if a pet or human is within the signal path with the aim of improving WiFi sensing based intrusion detection systems. Being able to determine the height of stimuli within the signal path would be a powerful feature to a sensing system aimed at detecting vermin. The geometry used for interference height estimation assumes that the transceivers are at the same height (see Figure 9.), which may not be the case in a typical WiFi network where a vermin sensing system would be deployed (Lin et al. 2020). But it may be possible to adapt Lin et al.'s (2020) method to estimate interference height when a user inputs the antennae height of static WiFi devices. This would allow interference height estimation with dissimilar transceiver height providing the WiFi devices position is static but needs to be investigated.

Classifying very fine movements accurately will need to be a key feature of a useful vermin sensing system. This is extremely difficult given the inherent distortion in CSI data collected from commodity WiFi devices. Xie, Li & Li (2015) investigated using power delay profiling to identify when multipathing is occurring and presented the "Splicer" software tool that can derive high resolution power delay profiles from CSI measurements. Splicer was tested with CSI data gathered from 20MHz and 40MHz channels from IEEE 802.11n devices and while functional it was noted that for detecting very fine movement 200MHz of bandwidth was required (Xie, Li & Li 2015). Splicer and the techniques used by Xie, Li & Li (2015) could be applied to the CSI data from this project which was captured from 80MHz channels. Also other IEEE802.11ac devices compatible with the CSI extraction tools used in this project are capable of measuring CSI from 160MHz channels (Gringoli et al. 2019). Utilising Xie, Li & Li's (2015) Splicer with wide bandwidth CSI measurements could provide a method for detecting and classifying very fine movements.

The limitations of a CSI sensing system are unlikely be defined by the strength of the interactions between target stimuli within the signal path and the RF signals used in WiFi or the capability of machine learning. Isolating and removing the distortion in CSI data caused by the behaviour of WiFi hardware and firmware will present the greatest challenge. Inputting phase measurements from CSI data will enhance the efficacy of a sensing system but the challenge of effectively removing or attenuating distortion created by noise and ambiguities is significant. In a survey of WiFi based sensing, recognition and detection systems, He et al. (2020) propose and review some methods of calibrating and processing phase data to identify and remove distortion. One method is to connect the transmitter and receiver via coaxial cabling to obtain a reference signal for comparison to phase data obtained when the same transmitter and receiver communicate wirelessly (He et al. 2020). The CSI extraction tools, and equipment used in this project could be used to undertake a comparison of phase measurements captured from wirelessly transmitted packets against phase data captured from packets transported via coaxial cabling. The comparison would aim to determine which features of phase data are

independent of the wireless signal path and propose methods to remove them using the techniques discussed by He et al. (2020) as a starting point.

6. References

Abbas, WaA, Nasim and Majeed, Uzma and Khan, Sadaf 2016, 'EFFICIENT STBC FOR THE DATA RATE OF MIMO-OFDMA', *Science International Journal* 1013-5316, vol. 28, pp. 247-55.

ACMA 2021a, *Communications and media in Australia: How we use the internet*, viewed 17 August 2022, <www.acma.gov.au/publications/2021-12/report/communications-and-media-australia-how-we-use-internet>.

ACMA 2021b, 'Exploring RLAN use in the 5 GHz and 6 GHz bands Discussion and options paper', *Proceedings of the Australian Communications and Media Authority*.

ACMA 2022, *Our role to manage spectrum*, Australian Communications and Media Authority viewed May 01, <<https://www.acma.gov.au/our-role-manage-spectrum>>.

Ades, E 2018, 'Species specific information: rat', *Johns Hopkins University*.

Anderson, G 2006, *ping(8) - Linux man page*, die.net, United States, viewed May 5, <<https://linux.die.net/man/8/ping>>.

Association, IS 2013, *IEEE Computer Society: "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications; Amendment 4: Enhancements for Very High Throughput for Operation in Bands Below 6 GHz"*, *IEEE Std 802.11 ac™*, The Institute of Electrical and Electronics Engineers, Inc.

AustralianMuseum 2022, *Black Rat*, NSW Government viewed 30 September, <<https://australian.museum/learn/animals/mammals/black-rat/>>.

CSIRO 2021, *Tackling Australia's persistent mouse problem*, CSIRO, viewed 15 September <<https://www.csiro.au/en/research/animals/pests/mouse-control>>.

Damodaran, N, Haruni, E, Kokhkhharova, M & Schäfer, J 2020, 'Device free human activity and fall recognition using WiFi channel state information (CSI)', *CCF Transactions on Pervasive Computing and Interaction*, vol. 2, no. 1, pp. 1-17.

Davies, L & Gather, U 1993, 'The identification of multiple outliers', *Journal of the American Statistical Association*, vol. 88, no. 423, pp. 782-92.

E. Cianca, MDSaSDD 2017, 'Radios as Sensors', *IEEE Internet of Things*, vol. 4, no. 2, pp. 363-73.

Forbes, G, Massie, S & Craw, S 2020, 'Wifi-based human activity recognition using Raspberry Pi', *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, IEEE, pp. 722-30.

Fromaget, P 2020, *What's the Best Micro SD Card for Raspberry Pi?*, RaspberryTips, viewed April 15, <<https://raspberrytips.com/best-sd-card-raspberry-pi/>>.

Gast, MS 2013, *802.11 ac: a survival guide: Wi-Fi at gigabit and beyond*, " O'Reilly Media, Inc."

Ge, Y, Taha, A, Shah, SA, Dashtipour, K, Zhu, S, Cooper, JM, Abbasi, Q & Imran, M 2022, 'Contactless WiFi Sensing and Monitoring for Future Healthcare-Emerging Trends, Challenges and Opportunities', *IEEE Reviews in Biomedical Engineering*.

Gong, L, Yang, W, Man, D, Dong, G, Yu, M & Lv, J 2015, 'WiFi-based real-time calibration-free passive human motion detection', *Sensors*, vol. 15, no. 12, pp. 32213-29.

Gringoli, F, Schulz, M, Link, J & Hollick, M 2019, 'Free your CSI: A channel state information extraction platform for modern Wi-Fi chipsets', *Proceedings of the 13th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*, pp. 21-8.

Guo, R, Li, H, Han, D & Liu, R 2023, 'Feasibility analysis of using Channel State Information (CSI) acquired from Wi-Fi routers for construction worker fall detection', *International journal of environmental research and public health*, vol. 20, no. 6, p. 4998.

Halperin, D, Hu, W, Sheth, A & Wetherall, D 2011, 'Tool release: Gathering 802.11 n traces with channel state information', *ACM SIGCOMM computer communication review*, vol. 41, no. 1, pp. 53-.

Han, C, Kaishun Wu, Yuxi Wang & Ni, LM 2014, 'WiFall: Device-free Fall Detection by Wireless Networks', *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pp. 271-9.

He, Y, Chen, Y, Hu, Y & Zeng, B 2020, 'WiFi vision: Sensing, recognition, and detection with commodity MIMO-OFDM WiFi', *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8296-317.

Ibrahim, M & Brown, KN 2021, 'Vehicle in-cabin contactless WiFi human sensing', *2021 18th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, Institute of Electrical and Electronics Engineers (IEEE), pp. 1-2.

IEEE 2013, *Telecommunications and information exchange between systems - Local and metropolitan area network Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE, New York, NY 100016-5997 USA.

Islam, MT, Dias, P & Huda, N 2020, 'Waste mobile phones: A survey and analysis of the awareness, consumption and disposal behavior of consumers in Australia', *Journal of Environmental Management*, vol. 275, p. 111111.

Jansons, J & Dorins, T 2012, 'Analyzing IEEE 802.11 n standard: outdoor performance', *2012 Second International Conference on Digital Information Processing and Communications (ICDIPC)*, IEEE, pp. 26-30.

Jiang, Z, Luan, TH, Ren, X, Lv, D, Hao, H, Wang, J, Zhao, K, Xi, W, Xu, Y & Li, R 2021, 'Eliminating the barriers: demystifying Wi-Fi baseband design and introducing the PicoScenes Wi-Fi sensing platform', *IEEE Internet of Things Journal*, vol. 9, no. 6, pp. 4476-96.

- Kanda, T, Sato, T, Awano, H, Kondo, S & Yamamoto, K 2022, 'Respiratory rate estimation based on WiFi frame capture', *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*, IEEE, pp. 881-4.
- Kaye, DH & Freedman, DA 2011, 'Reference guide on statistics', *Reference manual on scientific evidence*, pp. 211-302.
- Leis, JW 2018, *Communication Systems Principles Using MATLAB*, John Wiley & Sons, Incorporated, Newark, UNITED STATES.
- Li, J, Sharma, A, Mishra, D & Seneviratne, A 2021, 'Fire detection using commodity wifi devices', *2021 IEEE Global Communications Conference (GLOBECOM)*, IEEE, pp. 1-6.
- Li, W, Piechocki, RJ, Woodbridge, K, Tang, C & Chetty, K 2020, 'Passive WiFi radar for human sensing using a stand-alone access point', *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 3, pp. 1986-98.
- Lin, Y, Gao, Y, Li, B & Dong, W 2020, 'Revisiting indoor intrusion detection with WiFi signals: do not panic over a pet!', *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10437-49.
- Link, J, Zakharov, M, Schulz, M & Singh, G 2019, *Nexmon Channel State Information Extractor Issues*, Secure Mobile Networking Lab, viewed March 7, <https://github.com/seemoo-lab/nexmon_csi/issues>.
- Liu, W, Wang, X & Deng, Z 2021, 'Csi amplitude fingerprinting for indoor localization with dictionary learning', *Entropy*, vol. 23, no. 9, p. 1164.
- Ma, Y 2019, 'Improving WiFi sensing and networking with channel state information'.
- Palipana, S, Agrawal, P & Pesch, D 2016, 'Channel state information based human presence detection using non-linear techniques', *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments*, pp. 177-86.
- Paul, T & Ogunfunmi, T 2008, 'Wireless LAN comes of age: Understanding the IEEE 802.11 n amendment', *IEEE Circuits and systems magazine*, vol. 8, no. 1, pp. 28-54.
- Pearson, RK, Neuvo, Y, Astola, J & Gabbouj, M 2016, 'Generalized hampel filters', *EURASIP Journal on Advances in Signal Processing*, vol. 2016, pp. 1-18.
- Preliminary Data Sheet BCM43455*, 2016, Cypress Semiconductor Corporation.
- Raspberry Pi Documentation*, 2023, Raspberry Pi Ltd, viewed May 10, <<https://www.raspberrypi.com/documentation/computers/raspberry-pi.html>>.
- Reddy, A 2022, *Nexmon Channel State Information for Raspberry Pi*, Secure Mobile Networking Lab, viewed 17 February <https://github.com/nexmonster/nexmon_csi/tree/pi-5.10.92>.

- SAHealth 2022, *Rats and mice - prevention and control*, Government of South Australia, viewed 28 September, <<https://www.sahealth.sa.gov.au/wps/wcm/connect/public+content/sa+health+internet/conditions/bites+stings+and+pests/rats+and+mice+prevention+and+control>>.
- Schäfer, J, Barriwal, BR, Kokhkhharova, M, Adil, H & Liebehenschel, J 2021, 'Human activity recognition using CSI information with nexmon', *Applied Sciences*, vol. 11, no. 19, p. 8860.
- Schulz, M, Wegemer, D & Hollick, M 2016, 'Using NexMon, the C-based WiFi firmware modification framework', *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, pp. 213-5.
- Sydney Struggles to Get Rat Problem Under Control*, 2021, Australian Institute of Food Safety, Brisbane QLD AU 4001, viewed 30 September, <<https://www.foodsafety.com.au/news/sydney-struggles-rat-problem>>.
- Tan, S, Zhang, L & Yang, J 2018, 'Sensing fruit ripeness using wireless signals', *2018 27th international conference on computer communication and networks (ICCCN)*, IEEE, pp. 1-9.
- Tian, Z, Li, Y, Zhou, M & Li, Z 2018, 'WiFi-based adaptive indoor passive intrusion detection', *2018 IEEE 23rd International Conference on Digital Signal Processing (DSP)*, IEEE, pp. 1-5.
- Tukey, J 1974, 'Nonlinear (nonsuperposable) methods for smoothing data', *Cong. Rec. EASCON'74*, vol. 673.
- UniSQ 2022, *Applications and reports, How to Submit an Exemption*, University of Southern Queensland, CRICOS: QLD 00244B, viewed 30 September, <<https://www.unisq.edu.au/current-students/academic/higher-degree-by-research-students/conducting-research/animal-ethics/applications-reports>>.
- Vermin-Managing Rats in Your Home*, 2019, Townsville City Council, Townsville, <https://www.townsville.qld.gov.au/_data/assets/pdf_file/0021/4548/Vermin-managing-rats-in-your-home-EHRS-September-2019.pdf>.
- Viswanathan, M 2014, *Model and characterize MIMO channels*, GeneratePress, viewed March 12, 2023, <<https://www.gaussianwaves.com/2014/08/characterizing-a-mimo-channel/>>.
- Wang, C, Tang, L, Zhou, M, Ding, Y, Zhuang, X & Wu, J 2022, 'Indoor Human Fall Detection Algorithm Based on Wireless Sensing', *Tsinghua Science and Technology*, vol. 27, no. 6, pp. 1002-15.
- Wang, R, Zhou, X, Wang, B, Zheng, Z & Guo, Y 2022, 'A Subcarrier Selection Method for Wi-Fi-based Respiration Monitoring using IEEE 802.11 ac/ax Protocols', *2022 IEEE MTT-S International Microwave Biomedical Conference (IMBioC)*, IEEE, pp. 189-91.
- Wang, X, Yang, C & Mao, S 2017, 'PhaseBeat: Exploiting CSI phase data for vital sign monitoring with commodity WiFi devices', *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, IEEE, pp. 1230-9.

- Wang, X, Yang, C & Mao, S 2020, 'On CSI-based vital sign monitoring using commodity WiFi', *ACM Transactions on Computing for Healthcare*, vol. 1, no. 3, pp. 1-27.
- Wang, Z, Huang, Z, Zhang, C, Dou, W, Guo, Y & Chen, D 2021, 'CSI-based human sensing using model-based approaches: a survey', *Journal of Computational Design and Engineering*, vol. 8, no. 2, pp. 510-23.
- Ward, L 2012, '802.11ac Technology Introduction', vol. 1MA192, no. 7e.
- WorkSafe 2020, *Setting up your workstation*, The State of Queensland, viewed April 10, <[.](https://www.worksafe.qld.gov.au/safety-and-prevention/hazards/hazardous-manual-tasks/working-with-computers/setting-up-your-workstation#:~:text=A%20fixed%20sitting%20desk%20should,and%20avoid%20sharp%20desk%20corners.>)
- Wu, C, Yang, Z, Zhou, Z, Liu, X, Liu, Y & Cao, J 2015, 'Non-invasive detection of moving and stationary human with WiFi', *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 11, pp. 2329-42.
- Xiao, J, Wu, K, Yi, Y, Wang, L & Ni, LM 2012, 'Fimd: Fine-grained device-free motion detection', *2012 IEEE 18th International conference on parallel and distributed systems*, IEEE, pp. 229-35.
- Xie, Y, Li, Z & Li, M 2015, 'Precise power delay profiling with commodity WiFi', *Proceedings of the 21st Annual international conference on Mobile Computing and Networking*, pp. 53-64.
- Yadav, SK, Sai, S, Gundewar, A, Rathore, H, Tiwari, K, Pandey, HM & Mathur, M 2022, 'CSITime: Privacy-preserving human activity recognition using WiFi channel state information', *Neural Networks*, vol. 146, pp. 11-21.
- Yang, Z, Zhou, Z & Liu, Y 2013, 'From RSSI to CSI: Indoor localization via channel response', *ACM Computing Surveys (CSUR)*, vol. 46, no. 2, pp. 1-32.
- Zeng, Y, Pathak, PH, Xu, C & Mohapatra, P 2014, 'Your ap knows how you move: fine-grained device motion recognition through wifi', *Proceedings of the 1st ACM workshop on Hot topics in wireless*, pp. 49-54.
- Zhang, D, Wu, D, Niu, K, Wang, X, Zhang, F, Yao, J, Jiang, D & Qin, F 2022, 'Practical Issues and Challenges in CSI-based Integrated Sensing and Communication', *arXiv preprint arXiv:2204.03535*.
- Zhang, P, Su, Z, Dong, Z & Pahlavan, K 2020, 'Complex motion detection based on channel state information and LSTM-RNN', *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*, IEEE, pp. 0756-60.
- Zhou, Z, Wu, C, Yang, Z & Liu, Y 2015, 'Sensorless sensing with WiFi', *Tsinghua Science and Technology*, vol. 20, no. 1, pp. 1-6.

Appendix A Project Specification

ENG4111/4112 Research Project

Project Specification

For: Ben Balanzategui

Title: WiFi Based Passive Vermin Sensing, Proof of Concept and Feasibility Analysis

Major: Electrical and Electronic

Supervisors: TBA Expected: John Leis

Enrollment: ENG4111 – EXT S1, 2023

ENG4112 – EXT S2, 2023

Project Aim: Investigate novel and potentially useful applications of WiFi sensing that could be deployed as an additional utility function in a typical WiFi communication network. Significant research and testing have demonstrated the sensing capability of WiFi networks via the analysis of Channel State Information. The project will investigate the concept of using passive WiFi sensing to detect vermin.


Programme: Version 1, 15th March 2023

1. Conduct initial background research into using radio frequency signals specifically microwave signals for sensing and how wave propagation is affected by stimuli in the signal path.
2. Review and analyze previous WiFi sensing research and experimentation.
3. Procure and configure WiFi hardware that will facilitate the extraction of Channel State Information (CSI) and parse the CSI data into a suitable software application that can perform statistical analysis and implement detection algorithms.
4. Design a test apparatus consisting of WiFi access points, a signal path where stimuli including vermin (rats) are introduced and a device capturing and logging CSI.
5. Gather data from testing that can be used to examine the feasibility of using WiFi sensing as a vermin detection system.
6. Determine if it is possible to detect vermin and distinguish vermin from other stimuli and determine what limitations and constraints may impede the development of a system intended to be used as an additional utility function in a typical WiFi communication network.

If time and resource permit:

7. Investigate if fire detection is a feasible application – also a similar alternative if animal testing is unable to occur during the project.
8. Investigate using the Beamforming Matrix as the input to detection algorithms as opposed to using CSI values as CSI is not readily presented in WiFi hardware.

Appendix B Risk Management Plan

NUMBER	RISK DESCRIPTION	TREND	CURRENT	RESIDUAL
2122	WiFi Sensor Testing WiFi Based Passive Vermin Sensing, Proof of Concept and Feasibility Analysis		Low	Not Assessed
RISK OWNER	RISK IDENTIFIED ON	LAST REVIEWED ON	NEXT SCHEDULED REVIEW	
Ben Balanzategui	13/03/2023			
RISK FACTOR(S)	EXISTING CONTROL(S)	PROPOSED CONTROL(S)	OWNER	DUE DATE
Working with electrical appliances - laptops and WiFi access points	Control: All equipment is inspected prior to use and appliances are compliant with Australian standards, powered from outlets protected by RCD(safety switch) Student performing testing is a QLD licenced electrician, will inspect electrical equipment prior to energisation Where possible extra low voltage appliances will be substituted. i.e. the battery charger of a laptop will be left unplugged during testing			
Injury from lifting or bending, cuts and abrasions	Control: Team lift if required - nothing in the test apparatus should be >1kg Appropriate non slip footwear Suitable gloves (such as Ansell Hyflex) to be worn when required			
Slips trips and falls and when positioning WiFi access points	Control: Before each test housekeeping inspection of			

	<p>trafficable areas Prevent unexcepted access to test area with barriers or spotter Non slip footwear</p>	
<p>Live Pet Rat will be used a stimuli for testing</p>	<p>Control: Rats to remain within usual enclosure at all times Wifi Access points are positioned to place Rat within signal path, Rats must never be repositioned for the purpose of testing and are to remain within their usual habitats at all times No handling of Rat permitted, Rat must remain within enclosure If rat requires husbandry activities to be undertaken by owner or usual caregivers only</p>	
<p>Exposure to Microwave Frequency signals generated by the WiFi network used in the test apparatus</p>	<p>Control: All equipment is compliant with IEEE 802.11 and intended for regular use in residential and commercial settings. The Australian Communications and Media Authority has deemed such devices harmless to humans and pets via the relevant parts of the Radiocommunications Act 1992. As a precaution only, WiFi equipment will be deenergized when not required to avoid unnecessary exposure</p>	

Appendix C Ethics Approval

Ethics ETH2023-0118 (AEC): Mr Ben Balanzategui (Student) (Negligible Risk/Exempt)

Academic/Researcher	Mr Ben Balanzategui (Student)
Project	Prof John Leis WiFi Based Passive Vermin Sensing, Proof of Concept and Feasibility Analysis
Division	Academic Division
Faculty/Department	Deputy Vice Chancellor (Academic Affairs)

Ethics application

Overview

Application initiated by: Mr Ben Balanzategui
(Student)

Ethical Considerations

Are you working with animals or humans?

Animals

Do you have a current approval from another Ethics Committee to conduct this project? No

Project title

WiFi Based Passive Vermin Sensing, Proof of Concept and Feasibility Analysis

Project summary

Investigate novel and potentially useful applications of WiFi sensing that could be deployed as an additional utility function in a typical WiFi communication network. Significant research and testing have demonstrated the sensing capability of WiFi networks via the analysis of Channel State Information. The project will investigate the concept of using passive WiFi sensing to detect vermin and assess the potential feasibility of using a typical WiFi communications network to alert to the presence of vermin in a dwelling.

Host department

[School of Engineering](#)

Project duration

1 year

Is your research being conducted within Australia?

Yes

Select all that apply: Queensland

Does this project relate to, and/or extend on a previously approved project.

No

Is this project funded? Yes

Funding

How is the project being funded?

Please provide further details.

Self-funded Honours research project

Investigators

Principal Investigator

☐

Prof John Leis

UniSQ ID

[REDACTED]

Person type

Staff

Organisational area

School of Engineering

Other affiliations

Field of Research (FoR)

400607. Signal processing; 400907. Industrial electronics; 510202. Lasers and quantum electronics;

Co-investigator (UniSQ Staff)

Co-investigator (UniSQ Student)

☐

Mr Ben Balanzategui (Student)

UniSQ Student ID

[REDACTED]

Type of student

UGRD Student

Program

BENH - Bachelor of Engineering (Honours)

Organisational area

School of Engineering

Field of Research (FoR)

Does the project involve co-investigators from another university or organisation?

No

Conflict of interest

Does the Principal Investigator have an actual, perceived, or potential personal or financial Conflict of Interest (Col) in relation to the project?

No

Do any of the Co-Investigators or External Investigators have an actual, perceived, or potential personal or financial Conflict of Interest (Col) in relation to the project?

No

Outline the Conflict of Interest (Col) and advise on how it will be managed.

Qualifications and Experience

Principal Investigator - qualifications and experience

Principal Investigator

Prof John Leis

Qualifications relevant to project

Have taught telecommunications and related courses at tertiary level. Understand radio frequency engineering.

Experience relevant to project

Have supervised a similar project in 2022.

Co-Investigator - qualifications and experience

Co-Investigator

Mr Ben Balanzategui (Student)

Qualifications relevant to project UGRAD Engineering Student

Experience relevant to project

UGRAD Engineering Student

Operational Items

Does this project include: not applicable

The following options were available for selection:

- *Genetically Modified Organism (GMO)*
- *biological material (non-GMO), e.g. work with toxins, mutagens, teratogens, carcinogens etc.*
- *biological material native to Australia that was (or will be) collected in Queensland for commercial purposes*
- *radioactive substances and/or ionising radiation? (e.g. DXA, X-ray)*

Does this project include: not applicable

The following options were available for selection:

- the export, supply, publishing, or brokering of controlled goods, software, or technology
- an arrangement with a foreign government or foreign university that does not have institutional autonomy not applicable
- not applicable

If you have not previously submitted an Research Data Management Plan (RDMP) please provide details around 1. Storage, 2. Access, 3. ownership and 4. sharing research data.

1.personal desktop computer of student studying ONL

2.personal OneDrive account password protected and sent to Principal Investigator

3.Usual process for student honours projects, Data will be retained for a minimum of 7 years. After the minimum 7 year period, data may be stored indefinitely or securely deleted, if it is no longer of use, dissertation may published to UniSQ ePrints

Additional Information

Do you have a UniSQ Risk Management Plan relating to the activities being undertaken in this project? Yes

RMP Reference number

2122

UniSQ RMP Project Title

WiFi Sensor Testing WiFi Based Passive Vermin Sensing, Proof of Concept and Feasibility Analysis Low Not Assessed

Status of approval

Current

Date of Approval

13 Mar 2023

Upload a copy of the RMP

Ethical considerations - Animal

In what way does your project incorporate animals?

no interference with animals no abnormal disruption of habitat

Outline of project

Using plain language provide a description of what will be undertaken

There is no intervention with the rat required, the test apparatus need only be placed close to the rat (in its usual housing) to confirm it can detect the rat. The test methodology requires no interference, (handling, moving, interrupting normal routine etc.) of the pet rat.

Testing will be conducted to determine if a WiFi network can be used to detect the presence of vermin and distinguish between vermin and other stimuli i.e. static and moving objects. The test apparatus will include WiFi access points

connected to software that will collect and log performance data from the WiFi network, mainly Channel State Information. Pet rats owned by a supporter of the student undertaking the project will be used as stimuli in the test apparatus. The antennas of the WiFi access points will be placed so that a rat is within the signal path while the WiFi network is operating and performance statistics are being logged.

The radio equipment in WiFi hardware used in the test apparatus will not be modified and all equipment used is commercially available and intended for continuous use in typical domestic WiFi communication networks and will comply with the relevant parts of the Radiocommunications Act 1992 and adhere to IEEE 802.11. Such devices are deemed harmless to humans and pets.

The testing only requires data be captured that describes the propagation of the radio signals used by the WiFi network. It does not involve modifying the transmission power or any other parameters that would impact the electromagnetic compatibility of the WiFi network used in the test apparatus. The pet rats reside inside a dwelling that operates a WiFi network so the testing should not be considered to be introducing any source of harmless electromagnetic noise that is not already present in their normal habitat.

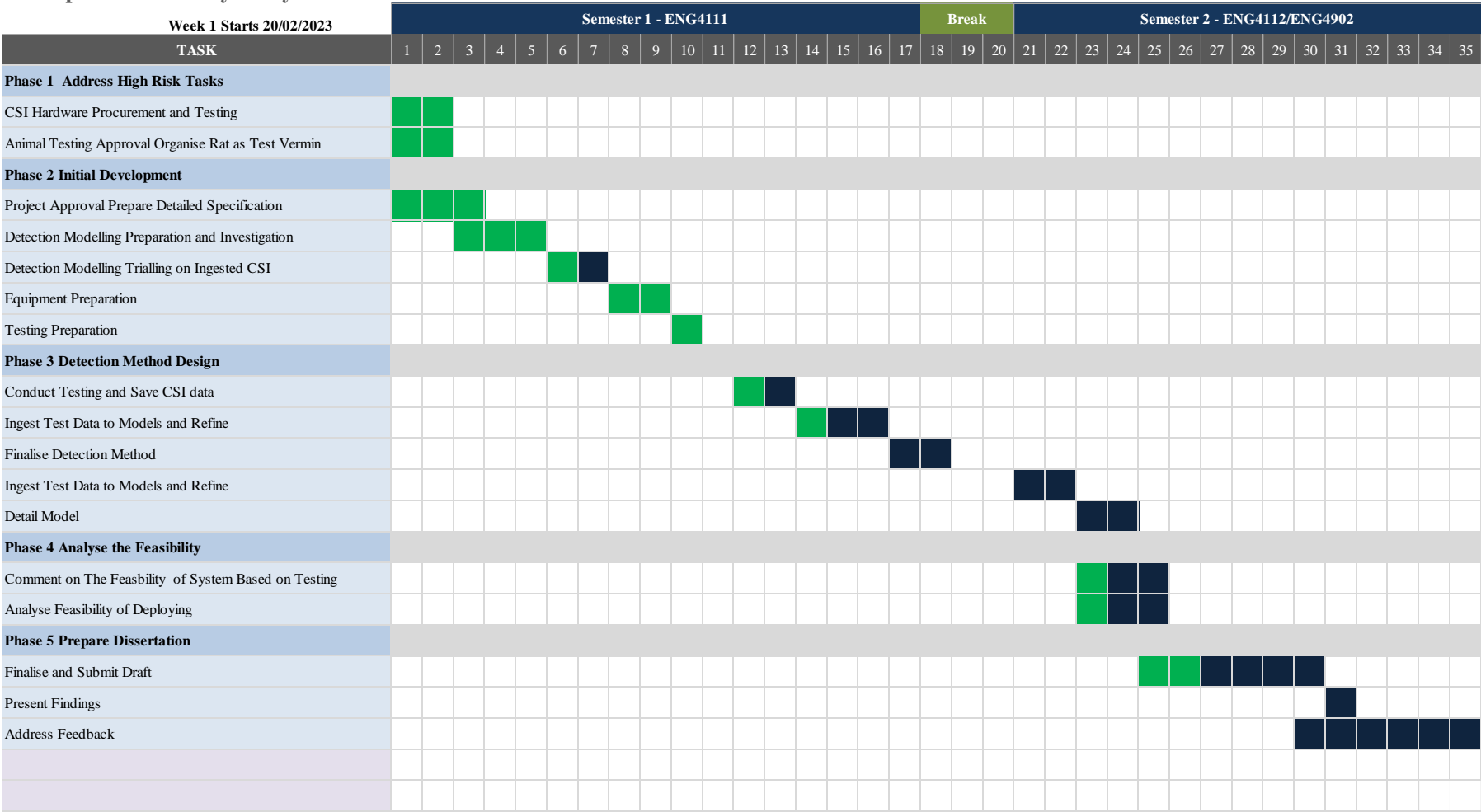
The project only aims to prove the concept of a WiFi based vermin detection system is potentially feasible. Testing will be limited in scope and only require the presence of a rat within the signal path of the WiFi antennas. No testing will involve interfering with the rat or removing it from its usual housing.

Attached files

RiskExport.docx

Appendix D Gantt Chart

WiFi Sensor Testing WiFi Based
Passive Vermin Sensing, Proof of
Concept and Feasibility Analysis



Appendix E MATLAB Function Ingesting CSI

```
function [csi_buff] = readCSI(File, BW, Max_UDP)
HOFFSET = 16;           % header offset
NFFT = BW*3.2;         % fft size
p = readpcap();
p.open(File);
n = min(length(p.all()),Max_UDP);
p.from_start();
csi_buff = complex(zeros(n,NFFT),0);
k = 1;
while (k <= n)
    f = p.next();
    if isempty(f)
        disp('no more frames');
        break;
    end
    if f.header.orig_len-(HOFFSET-1)*4 ~= NFFT*4
        disp('skipped frame with incorrect size');
        continue;
    end
    payload = f.payload;
    H = payload(HOFFSET:HOFFSET+NFFT-1);
    Hout = typecast(H, 'int16');
    Hout = reshape(Hout,2,[]).';
    cmplx = double(Hout(1:NFFT,1))+1j*double(Hout(1:NFFT,2));
    csi_buff(k,:) = cmplx.';
    k = k + 1;
end
end
```

(Gringoli et al. 2019)