Signal classification at discrete frequencies using machine learning

C. J. Swinney

A thesis submitted for the degree of

Doctor of Philosophy

Royal Air Force

Department of Computer Science and Electronics Engineering

University of Essex

Date of submission for examination August 2022

Acknowledgements

First, I would like to express my thanks to my supervisor Dr. John C. Woods. Without his extensive experience, academic support and unwavering dedication for the amount of experiments we have achieved, this work would not have been possible. This was even more paramount during the COVID-19 pandemic whereby progress could have been hindered due to lockdown restrictions. With Dr. Woods continuous support throughout that time, both pastorally and academically, the pandemic did not affect the output. I would also like to extend my thanks to Professor Reinhold Scherer, University of Essex, for the continued support and expertise, in particular with regards to machine learning and cross validation techniques. A special thank you is also extended to UAS Pilot Jim Pullen who has given up his time to fly all the UAS experiments details in this thesis and for the production of the DroneDetect dataset.

I would like to extend my thanks to the Royal Air Force Engineering and Cyberspace Profession, and Chief of Staff Support, for the sponsorship of this PhD. A special thank you to the Royal Air Force 591 Signals Unit for the use of their facilities for a number of the experiments in this thesis. In terms of Royal Air Force support, this PhD would not have been possible without the unwavering support of the Air and Space Warfare Centre (ASWC) in everything that has been achieved with this work. The ASWC have gone out of their way to make sure I have had all the support necessary to make this work count both academically and for the benefit of wider Defence. They have also taken special care to ensure I have been supported pastorally through every stage of the PhD and throughout the pandemic, which is something I am most grateful for. Without their support to me and this work it would not have happened. My deepest gratitude is extended to my hierarchy in the ASWC. Lastly I will extend my thanks to my family who always support all of my endeavors in every way.

Abstract

Incidents such as the 2018 shut down of Gatwick Airport due to a small Unmanned Aerial System (UAS) airfield incursion, have shown that we don’t have routine and consistent detection and classification methods in place to recognise unwanted signals in an airspace. Today, incidents of this nature are taking place around the world regularly. The first stage in mitigating a threat is to know whether a threat is present. This thesis focuses on the detection and classification of Global Navigation Satellite Systems (GNSS) jamming radio frequency (RF) signal types and small commercially available UAS RF signals using machine learning for early warning systems. RF signals can be computationally heavy and sometimes sensitive to collect. With neural networks requiring a lot of information to train from scratch, the thesis explores the use of transfer learning from the object detection field to lessen this burden by using graphical representations of the signal in the frequency and time domain. The thesis shows that utilising the benefits of transfer learning with both supervised and unsupervised learning and graphical signal representations, can provide high accuracy detection and classification, down to the fidelity of whether a small UAS is flying or stationary. By treating the classification of RF signals as an image classification problem, this thesis has shown that transfer learning through CNN feature extraction reduces the need for large datasets while still providing high accuracy results. CNN feature extraction and transfer learning was also shown to improve accuracy as a precursor to unsupervised learning but at a cost of time, while raw images provided a good overall solution for timely clustering. Lastly the thesis has shown that the implementation of machine learning models using a raspberry pi and software defined radio (SDR) provides a viable option for low cost early warning systems.

Table of Contents

Tables

Figures

Glossary

Term	Definition
Unmanned Aerial System (UAS)	Otherwise known as a ‘drone’. An uncrewed aircraft or ship (and associated equipment) guided by remote control or onboard computers. [1]
Radio Frequency (RF)	“Electromagnetic wave frequencies that lie in the range extending from below 3 kilohertz to about 300 gigahertz and that include the frequencies used for communications signals (as for radio and television broadcasting and cell-phone and satellite transmissions) or radar signals”. [2]
Software Defined Radio (SDR)	“A radio in which some or all the physical layer functions are software defined”. [3]
Global Navigation Satellite System (GNSS)	“Global Navigation Satellite System (GNSS) refers to a constellation of satellites providing signals from space that transmit positioning and timing data to GNSS receivers. The receivers then use this data to determine location”. [4]
Global Positioning System (GPS)	“The Global Positioning System (GPS) is a U.S.-owned utility that provides users with positioning, navigation, and timing (PNT) services”. [5]
Machine Learning (ML)	“Machine learning is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and

	algorithms to imitate the way that humans learn, gradually improving its accuracy”. [6]
Deep Learning (DL)	“Deep learning is a subset of machine learning, which is essentially a neural network with three or more layers. These neural networks attempt to simulate the behavior of the human brain—albeit far from matching its ability—allowing it to “learn” from large amounts of data.” [7]
Convolutional Neural Network (CNN)	“Convolutional neural network (CNN) is a class of deep learning methods which has become dominant in various computer vision tasks and is attracting interest across a variety of domains, including radiology. A CNN is composed of multiple building blocks, such as convolution layers, pooling layers, and fully connected layers, and is designed to automatically and adaptively learn spatial hierarchies of features through a backpropagation algorithm”. [8]

Chapter 1 - Introduction

The thesis is broken down into the following chapters. The introduction includes an overview of the research problem and scope, the contribution of the research to the academic community and a background to the work. Chapter 2 ‘Chapter 2 - Literature Review’ contains the literature review which is broken down into UAS detection and classification; GNSS jamming detection and classification; and transfer learning. Chapter 3 ‘

Chapter 3 - RF Profiling’ describes the datasets, dataset generation and the representation of signals as graphical representations which can be saved as images. Chapter 4 ‘Chapter 4 - Supervised Learning’ describes the CNN feature extraction, machine learning classifiers and experimental results for classification of UAS and GPS jamming signals. Chapter 5 ‘Chapter 5 - Unsupervised Learning’ discusses the unsupervised learning algorithm K-Means and its application and experimental results for small UAS clustering and for GNSS jamming signal clustering. Chapter 6 ‘Chapter 6 - Early Warning including Unknown Signal Detection’ considers the implementation of a low cost early warning system made up of a Raspberry Pi and BladeRF SDR. It also considers the classification of unknown signal types using the supervised models created in Chapter 4. Lastly Chapter 7 ‘Chapter 7 - Conclusions’ will draw out the conclusions from the thesis and recommended future work.

1.1 Research Problem and Scope

The use of Radio Frequency (RF) signals as an access vector for malicious cyber-attacks is a growing concern in a world made up of connected systems. Security professionals often regard Cyber threats as mitigated against by disconnecting a system from the internet. However, RF connectivity in many systems is pervasive, underpinning fundamental services and often overlooked in the consideration of cyber security. Wireless standards are inherently open and often published on the internet, leaving RF links as an opportunity for adversaries to exploit and conduct cyber-attacks. However, open security has been shown to be an overall better solution than security by obscurity. The first stage of mitigating cyber-attacks which use RF as an access vector is to detect the presence of an unwanted signal. Classifying the signal type can further help with attribution of an attack to a perpetrator and to inform time critical risk calculations. The application extends itself to other signals of interest such as the detection and classification of small commercially available Unmanned Aerial Systems (UAS) and RF signals which deliberately interfere with critical services such as Global Navigation Satellite Systems (GNSS).

Arguably the largest incident of UAS disruption happened in 2018 at Gatwick Airport where over 1000 flights were grounded for 36hrs at a cost of over 50 million pounds to the UK economy. An in depth report conducted by the Guardian newspaper concluded there was minimal evidence to suggest that a small UAS was even present at all [9]. Incidents of this nature have not diminished since 2018 but rather are still causing disruption all over the world. In 2022 the United States Department of Homeland Security officials revealed they received over 2,000 sightings near airports in the last year [10]. In the UK a 2018 study by Dedrone [11] reported almost two detections of small UAS per day at four different UK airports, with a 2019 study highlighting that 62% of near miss incidents between small UAS and aircraft posed a significant risk [12].

Another example of RF signals of interest relates to the worldwide reliance on GNSS such as Global Positioning System (GPS) to provide vital position, navigation and timing services. In

2017 the potential economic impact from a loss of services for 5 days was assessed at £5.2bn [13]. In 2022 we are highly dependent on GNSS for precise timing signals to enable applications from power distribution to emergency services and 5G performance [14]. Future technologies such as driverless cars are critically reliant on these signals for safety. Even with the growing economic reliance on GPS, intentional interference incidents are increasing with a European report showing recorded events to have increased 20 times over a period of 2 years [15]. This thesis concentrates on UAS RF signals and GNSS jamming RF signals due to the availability of open source datasets, which allow for peer comparison of results, and due to the prevalence of incidents which have the potential for highly significant impact to the economy. However, future application of this work could include any RF signal of interest.

Whether it is UAS or GNSS interference signals under consideration, it is apparent that routine and consistent detection and classification methods are not widely available or in place to recognise unwanted signals. Human analysts are commonly used to manually scan the RF spectrum to identify signals of interest [16]. This process can be timely and creates a burden on specialist operators which subsequently instills a high dependency on human experience. Deep learning and machine learning present an opportunity to reduce the manual scan time significantly and point a human operator to a signal of interest. RF signal data is inherently large and can be computationally resource intensive. RF signals can also be sensitive to collect due to certain legal constraints in many countries such as the UK. Transfer learning offers an opportunity to work with smaller datasets while maintaining high accuracy results and has been successful in other fields including medical condition diagnosis and audio signal classification.

Signals from small UASs of the same manufacturer tend to be evolutionary, with a datalink being upgraded between UAS platforms from the same manufacturer. One of the world leading manufacturers of small commercial UAS is Da-Jiang Innovations (DJI). They have recently evolved the LightBridge datalink which is hardware based using a field programmable gate array (FPGA), to the OcuSync datalink which utilises a Software Defined Radio (SDR) based approach. The evolution of datalinks has potential to lend itself to supervised learning to provide a probability indication that an unwanted UAS signal is present. However, unsupervised learning provides an opportunity to detect signals which have not been seen before in a much quicker time frame. Unsupervised and supervised learning operating together also presents the scope to detect and classify signal types in a timely and accurate manner as part of a larger early warning system.

Several industry programs claim to be able to perform these functions [17] [18] but incidents of malicious UAS activity and GNSS interference are still occurring across the world on a daily basis. One potential reason why detection and classification systems are not common place is due to the cost of implementing such a system. This thesis will therefore also consider whether these techniques can be achieved on a low cost platform to enable future widespread use of such a system.

1.2 Contribution of the Research

The contributions of the work as shown in the thesis are as follows:

(1) Classifying the flight mode of a UAS signal with high accuracy is an important step forward in the field and could provide vital information on the scene of a major incident for

risk assessment. We improve accuracy by over 45% to 91% from previous research of flight mode classification confirmed using the same open source dataset. Further confirmation is proved with a larger dataset containing 11 more classes and tested in the presence of interference.

(2) Pre-trained CNNs for image classification can be employed for feature extraction using transfer learning on image based graphical representations of RF UAS and GPS jamming signals producing high accuracy results over 98% and reducing the need for large datasets and computational resources.

(3) Supervised machine learning algorithms utilising transfer learning are capable of detecting UAS signals not captured in the original dataset, including evolutionary and non-evolutionary datalinks. This has significant implications for detecting unknown threats.

(4) Frequency domain graphical signal representations and deeper CNN architectures provide features which are robust to interference from other signals operating in the same frequency band such as Wi-Fi and Bluetooth.

(5) CNN feature extraction and transfer learning produces high performance clustering (over 0.8 v-measure) for unsupervised learning compared to clustering raw data from the SDR and raw images but at a cost of time (6s). Raw images are a good overall solution for timely clustering (under 0.3s) which could form part of an early warning system to confirm and/or cue other sensors.

(6) A low cost raspberry Pi and SDR based machine learning classification system can predict signals it was trained against, and signals it was not trained against, in a live environment.

Overall the work has produced 8 published and peer reviewed papers, 2 papers which are currently under peer review and 2 papers which have been peer reviewed and are ready for print imminently, one of which is to be printed in the IEEE Transactions Journal series. I have presented at 6 conferences and other contributions include 2 published open source datasets, an invitation to be a reviewer for IEEE Transactions on Intelligent Systems Journal and an invited seminar at Manchester University as part of their Digital Trust Series on the exposure of cyber vulnerabilities using software defined radios. Sections within the thesis which include material from published papers in the following list are referenced within the section headings.

1.2.1 Journal papers

Title	**Journal**	**Status**	**Reference**
K-means Clustering Approach to UAS Classification via Graphical Signal Representation of Radio Frequency Signals for Air Traffic Early Warning	IEEE Transactions on Intelligent Transportation Systems	Published 2022	[19]
(Invited) Low Cost Raspberry Pi based UAS Detection & Classification System using Machine Learning	MDPI Aerospace Special Issue Journal - Unmanned Aerial Vehicles en-Route Modelling and Control	Published 2022	[20]
A Review of Anomaly Detection in the RF Spectrum for the Early	International Journal of Critical Infrastructure Protection	Under Peer Review	

Warning of RF enabled Cyber-Attacks using Software Defined Radio			
Challenges of Artificial Intelligence: Britain’s Bright Future.	Air and Space Power Journal	Scheduled to be published in Autumn/Winter 2022 edition	
A Review of Security Incidents and Defence Techniques relating to the Malicious Use of Small Unmanned Aerial Systems	IEEE Aerospace and Electronic Systems Magazine	Published 2022	[21]
(Invited) Faithful Wingmen or Killer Robots? Artificial Intelligence Applications for Air Power	Air and Space Power Review	Published 2022	[22]
(Invited) GPS Jamming Signal Classification with CNN Feature Extraction in low Signal-to-Noise Environments	International Journal on Cyber Situational Awareness IJCSA Volume 6, 2021	Published 2022	[23]
(Invited) The Effect of Real-World Interference on CNN Feature Extraction and Machine Learning Classification of Unmanned Aerial Systems	MDPI Aerospace Special Issue Journal - AI/Machine Learning in Aerospace Autonomy	Published 2021	[24]
Unmanned Aerial Vehicle Operating Mode Classification Using Deep Residual Learning Feature Extraction	MDPI Aerospace	Published 2021	[25]

1.2.2 Conferences

Title	Conference	Status	Reference
(Invited) Deep Residual Learning Feature Extraction for GNSS Jamming Signal Classification and the effect of low Signal-to-Noise Ratio	Artificial Intelligence, Machine Learning and Data Science World Forum 2022	Presentation 2022	
(Invited) K-means Clustering Approach to GNSS Jamming Detection via Graphical Representations of Radio Frequency Signals	Conference on Applied Science, Engineering and Technology (GECAET-2022)	Presentation 2022	
Unmanned Aerial System Detection and Classification	U.S. Department of Defence Artificial Intelligence Symposium	Poster Presentation 2021	
RF Detection and Classification of Unmanned Aerial Vehicles in Environments with Wireless Interference	2021 International Conference on Unmanned Aircraft Systems (ICUAS)	Presentation & Conference Paper Published 2021 IEEE Explore	[26]
GNSS Jamming Classification via CNN, Transfer Learning &	2021 International Conference on Cyber	Presentation & Conference	[27]

the Novel Concatenation of Signal Representations	Situational Awareness, Data Analytics and Assessment (CyberSA)	Paper Published 2021 IEEE Explore	
Unmanned Aerial Vehicle Flight Mode Classification using Convolutional Neural Network and Transfer Learning	2020 16th International Computer Engineering Conference (ICENCO)	Presentation & Conference Paper Published 2021 IEEE Explore	[28]

1.2.3 Other Contributions

Title	**Description**	**Status**	**Reference**
(Invited) - Reviewer	Reviewer for Transactions on Intelligent Transportation Systems	1 x paper review in 2022	
(Invited) - Security Challenges presented by Malicious Small UASs	Book Chapter with Nova Science Publishers, Inc.	Due to be published Aug 22 (hardcover and electronically)	

(Invited) Software Defined Radios: RF enabled cyber vulnerabilities and early warning	Seminar delivery to Manchester University as part of the Digital Trust & Security Seminar Series	Delivered May 2022	[29]
DroneDetect Dataset: A Radio Frequency Dataset of Unmanned Aerial System (UAS) Signals for Machine Learning Detection & Classification	IEEE Open Source Dataset Publication	Published 2021	[30]
Raw IQ dataset for GNSS GPS jamming signal classification	Zenodo Open Source Dataset Publication	Published 2021	[31]
(Invited) - Reviewer	Reviewer for book chapter ‘Cyberspace Operations’	Published 2021	[32]

1.3 Background

To introduce the background to these signals of interest this thesis will first consider the significance of the small commercially available UAS used for malicious intent. Next the section will discuss RF signals which interfere with GNSS before introducing SDRs and their use for both the delivery of RF access based cyber-attacks and for the early warning of unwanted signals in prohibited areas.

1.3.1 Malicious Small UASs [21]

In recent years the market for small UAS has expanded from hobbyists to supporting the economy commercially. In the "Commercial Drone Market, 2021-2028" report, the market is predicted to grow exponentially until 2028 [33]. Amazon and Domino's Pizza in the US are trialling deliveries using UAS and industry specific usages such as farming and geological surveying are benefitting from the technology. In the US the Department of Transport are proposing to ease regulations to enable greater use of small UAS [34]. However increasing the ease of use comes with security risks. A study carried out in 2018 by Dedrone across 4 different UK Airports observed nearly two small UAS detections per day. Dedrone also concluded that while Da-Jiang Innovations (DJI) is a world leader in the small UAS market, they only made up 44% of the incursions monitored [11]. In 2017 in the UK a study was commissioned to understand whether a small UAS could cause critical damage to an aircraft from a mid-air collision. The report concluded that factors including the size of the UAS, whether the aircraft was bird strike certified and the aircraft type all contributed to whether critical damage occurred [35]. In 2019 62% of the near miss incidents between small UAS and aircraft in the UK were deemed to have held significant risk [12]. Following a collision between a small UAS and an aircraft in Canada in 2017 [36] the University of Dayton Research Institute conducted mid-air collision investigations between a DJI Phantom 2 quadcopter and an aircraft. They showed that even a small UAS such as the Phantom 2 could cause significant damage to an aircraft [37].

Malicious activity from small UAS does not always involve physical damage, we can break down malicious activity into 3 categories:

- Physical attacks
- Cyber attacks
- Surveillance

Physical Attacks

Physical attacks could be caused by mid-air collisions with a moving target as previously discussed or carrying a payload against a fixed target, for example chemical, radiological, biological, nuclear or explosive materials. A prevalent example of an attempted physical attack was seen in 2018 at a speaking event in Caracas. Venezuelan President Maduro was subject to an assassination attempt when two UAS carrying explosives detonated near his stand [38]. However, as far back as 2015 we have seen small UASs cause disruption, for example a small UAS was flown into power lines causing a blackout of electricity for 650 people in California in 2015 [39]. In 2017 during the Golden State Race Series a Phantom 4 small UAS hit a tree and then went on to hit a cyclist [40]. In 2016 a small UAS hit Seattle’s Space Needle while fireworks were being set up for the New Year’s Eve display [41]. In

2017 we observed a collision occurring in Canada between a light aircraft which was landing and a small UAS [42]. Then as far back as 2017 small UASs were being weaponised to carry explosives and drop munitions by Islamic State (IS) in Iraq [43]. In April 2021 the Institution of Engineering and Technology (IET) conducted an open data analysis study looking at consumer UASs or drones being armed for terror attacks by conflict groups. They discuss the use of weaponised UASs for propaganda, small UAS use by the Taliban for dropping mortars and Houthi rebels using UASs for naval mine attacks [44]. It is without a doubt that the use of small UASs is becoming more prominent in warfare. 5 years ago Hartmann and Giles [45] stated that with plans in place for UASs fitted with physical weapons to target other UASs, this would see the first documented case of warfare using UAS on UAS. They liken it to the First World War where pilots were armed with rifles in the cockpit of reconnaissance aircraft.

Another concern is that small UASs could be equipped with chemical, radiological, biological, nuclear or explosive materials. David Cameron (former UK prime minister) warned in 2016 small UASs carrying an aerosol with nuclear materials could be used in attacks against Western cities [46]. DeFranco [47] suggests that a new challenge is presented to countering biological weapons from UASs. Along with producing a thorough taxonomy of threats, Majeed et al. [48] assert that equipping UASs with armed objects by terrorist groups is a matter of concern. In 2019 the South China Morning Post reported that criminal gangs were using UASs to spread swine fever over pig farms in an attempt to profit from the market [49]. The Missile Technology Control Regime (MTCR) is an agreement that includes governing UASs and includes 35 countries. The MTCR met in October 2019 and looked at the prevention of Weapons of Mass Destruction released using unmanned delivery systems [50]. In 2020 China released footage of a swarm of military UASs known as the ‘suicide drones’ which operate as one unit to destroy ground targets [51]. In 2021 the first documented use of a swarm of UASs was used in combat against Hamas militants by the Israel Defense Forces (IDF). The swarm which was reported to be guided by Artificial Intelligence (AI) and used to locate, identify and attack targets in Gaza in May 2021 [52].

Cyber Attacks

Small UAS also could provide a platform to carry out cyber-attacks through the interception and breaching of wireless networks [53]. At the CanSecWest Conference 2021 security researchers Weinmann and Schmotzle used a DJI Mavic 2 carrying a Wi-Fi dongle to hack open a Tesla Model X car door [54]. In 2019 at the DroneDeploy Conference, Rhea Naidoo, co-founder and Director of Automated Solutions at Cambrian Cyber Group, broke cyber-attacks using UASs into 3 categories. Attacks which compromised the confidentiality of data, for example exfiltrating confidential information; attacks which compromised the integrity of data such as manipulating controls or disabling alarms and lastly compromising the availability of services by taking processes or networks offline [55]. Devices such as the Raspberry Pi can be fitted onto a UAS and used to access insecure networks and retrieve personal data or perform techniques such as spoofing [56]. Known as a ‘man in the middle’ attack, a UAS can be used to broadcast a Wi-Fi network which has been connected to in the past, all the data from that persons phone will then pass through that network allowing an attacker access to valuable information [57]. Barker shows that when a directional antenna is used that computer network attacks and data exfiltration can be effective at distances larger than 800m [58].

UASs can be used as a platform for the interception of wireless networks [53]. Le Roy et al. [59] consider attacks using an SDR embedded on a UAS and conclude this an opportunity for attacks. A UAS was shown to hack a Bluetooth mouse in an office building while hovering outside the window. The attack successfully installed malware through the connection onto the computer and subsequently received malicious communication using light pulses and the camera fitted on the UAS [60]. Generally speaking any cyber-attack which uses RF as the access vector and can be performed on an SDR is potentially viable. While cyber-attacks using RF access points are often described with a person parked up in a car with a laptop and a device such as a Wi-Fi Pineapple Nano with targets such as airports, coffee shops and hotels, the use of the UAS provides access to geographically disparate areas where the person can be a significant a distance away [61]. Further, without a robust registration process for small UASs it may be hard to identify attackers for prosecution. The UAS allows a cyber-attack to be performed both remotely and anonymously [62], an attractive prospect for an attacker. However, it is not just the UAS platforms themselves which can cause a threat in the cyber domain. In December 2020 the U.S. government imposed serious restrictions on DJI amidst claims from cyber security firms of data collection from affiliated applications [63].

Another consideration is that a small UAS being used for legitimate purposes could be hacked or hijacked using cyber capabilities. Walters [64] shows a list of vulnerabilities in small commercially available UASs which is kept up to date and includes cyber-attacks from reverse engineering to denial of service and hijacking. Krishna and Murphy examine the range of cyber-attacks on small UASs which have been simulated or studied and suggest that GPS spoofing attacks are the most researched. They also highlight GPS jamming, de-authentication attacks, zero-day vulnerabilities, video replay attacks and the interception of data feeds. Krishna and Murphy also indicate that attacks using viruses are not common [65]. Some attack methodologies and source code are openly available online which means that anyone could download and execute the code [45]. However, if we consider cyber-attacks exploiting small UAS vulnerabilities, there is one incident whereby a small UAS in South Korea was affected by GPS jamming and crashed killing an engineer [65].

Surveillance

Lastly, many small UAS have cameras as a standard fitting which could be used for surveillance and mapping of sites. This capability has led to enormous benefits for security in agriculture, search and rescue missions and many more applications. However, along with these benefits come the risk of malicious use. In 2020 a Welsh man was jailed for using a small UAS to spy on and monitor the activities of an ex-partner [66]. In Australia a small UAS was used to spy on a woman skinny dipping in her own back garden [67]. A couple who lived in a fifth floor apartment reported seeing a small UAS at the balcony door spying on them getting ready for a shower [68]. It's not only individuals that are at risk from spying, sports such as football have seen accusations of small drones being used to spy on practises [69]. In 2019 an incident occurred whereby UAS swarmed a U.S. warship and hovered in its vicinity for days [70]. It is without a doubt that UAS pose many security threats and robust detection, classification and counter measure systems are required to mitigate the risk posed. Figure 1 shows the different technologies being considered for UAS detection, classification and mitigation.

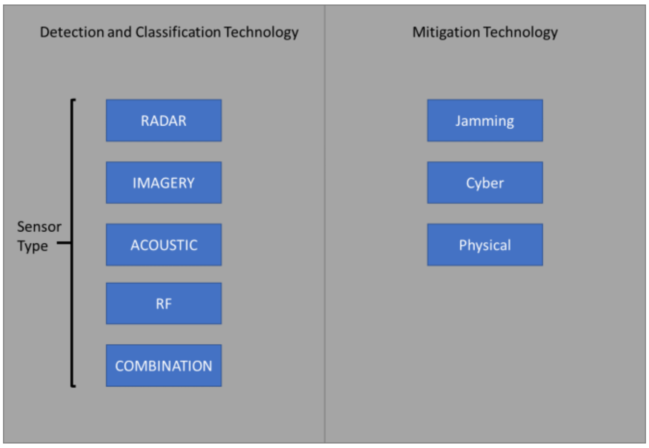


Figure 1 - UAS Detection and Classification Sensor Types, and Mitigation Technologies

The various sensors which have been researched for detection and classification all come with different benefits and their own challenges. Radar for example boasts long ranges and has advantages over imagery based methods as it is resilient to weather considers but it has been shown to have difficulties distinguishing between small UAS and birds. Acoustic sensors have proved very accurate in short range and like radar performance they do not degrade with visibility but acoustics systems do suffer when noise is present, including wind. Imagery based systems using electro-optical cameras are generally low in cost and human verification is possible but detection and classification accuracy is affected by environmental conditions and time of day. Laser and thermal based systems can be costly and thermal has shown to be less effective with power efficient UAS such as some quadcopters.

RF systems can produce ranges similar to radar, they don’t suffer from environmental conditions and classification can specify what the UAS is doing – i.e. stationary, hovering or flying. It is worth noting that up until 2019 when classification was referred to with UAS it considered issues such as distinguishing between UAS and birds and classifying the UAS type. Al-Sa’d et al. [71]. As we come to review recent security incidents in this paper we can keep in mind that understanding what a UAS is doing, for example capturing video or hovering over a particular site, could impart valuable information contributing to an assessment of risk. Classifying the flight mode is an important step forward and a significant benefit of RF detection and classification systems. However, these systems cannot detect UAS flying without active RF datalinks such as UAS flying in a GPS autonomous mode. RF detection when combined with machine learning can increase both reliability and system performance [72].

When a malicious UAS has been detected and classified we need to know what to do with it. It may be that there is an imminent threat that needs to be dealt with straight away, as with the attack on the Venezuelan President in 2018, or we may want to prevent access to an area such an active airspace around an airport. Arguably the most researched counter-measure is jamming, emitting another signal in the same frequency band which is higher in power and subsequently blocks the transmission between the UAS and its controller. Physical measures include firing nets and using physical weapons against UAS. Lastly cyber methods have been investigated to disrupt malicious UAS activity. All of the counter measure methods which are covered in more details within Chapter 2 ‘Chapter 2 - Literature Review’ present legal, policy and practical challenges which can vary depending on the environment the system needs to operate in. Before moving on to consider the literature in the field, recent security incidents related to Critical National Infrastructure including Airports and Nuclear Facilities caused by malicious UAS will be discussed.

Security Incidents

The largest incident to cause airport disruption is without a doubt the 2018 UAS sightings at Gatwick Airport in the UK. After a number of sightings flights were grounded for nearly 2 days and no suspect has ever been found [9]. The incident did highlight the cost to the economy as the disruption affected over 82,000 passengers and at a cost of over 50 million pounds [73]. Since 2018 there have been a number of incidents in restricted airspaces across the world. In 2019 in New Jersey, U.S. flights were disrupted following a UAS flying within 9m of an aircraft [74]. In 2019 in Japan a UAS was observed hovering above the runway at Kansai Airport [75]. In 2019 Heathrow Airport in the UK grounded flights for 30 minutes after a number of UAS sightings [76]. 2019 also saw disruption at Frankfurt Airport in Germany twice that year [77], flight diversions at Dublin Airport [78], suspension of flights in Dubai [79] and 37 flights delayed at Changi Airport in Singapore due to the UAS sightings [80]. In 2020 a confirmed UAS sighting by a military Apache helicopter at Stansted Airport in the UK was reported and one arrest was made [81]. Again Frankfurt Airport in Germany in 202 was subject to 2 hours of grounded flights and diversions due to a pilot sighting of a UAS [82]. In 2020 two pilots reported sighting a UAS which resulted in grounded flights for 1 hour in Spain [83]. Most recently in 2021 we have seen disruption at North Carolina Airport in the U.S. [84] and at Auckland Airport in New Zealand where the UAS was sighted only 30m away and 5m above a helicopter [85]. Although we haven’t seen another incident with the scale of disruption that was caused at Gatwick, events are still occurring regularly. The Federal Aviation Authority in the U.S. report UAS sightings at airports at over 100 a month [86].

It’s not just airports who have had recorded incidents of disruption from small UAS, Nuclear Facilities have also seen increased media attention. The first widely reported issue concerning small UAS and nuclear power plants occurred in France in 2014. The incidents involved multiple UAS, which were never identified, flying into restricted airspace over 13 different nuclear power plants [87]. In 2014 the malicious use of small UAS also came into the attention of the British media, with the first conviction in the UK made after a recreational UAS flew within 50 metres of a nuclear facility [88]. Aside from an incident involving Greenpeace who crashed a superman shaped UAS into a French nuclear facility to expose vulnerabilities [89], there was a gap in media reporting of incidents at nuclear facilities between 2014 and 2019. However a recent freedom of information request revealed there had been 57 incidents at nuclear power plants in the U.S. between December 2014 and October 2019, with 85% of the incidents being unresolved in terms of attributing the incident to a perpetrator [90], [91]. In September 2019 a swarm of UAS entered the restricted airspace of the Palo Verde Nuclear Plant, the largest power plant in the U.S. The report stated that 5 or 6 small UAS flew around the protected area for 80 minutes, indicating that this was not conducted by a popular consumer UAS such as the DJI Phantom which has a much lower flight time [92]. In June 2021 one or more small quadcopters were reported to have caused substantial damage to a nuclear facility in Iran [93]. This highlights the importance of being able to classify the UAS type to help with attribution of incidents to perpetrators. Next GPS jamming will be discussed.

1.3.2 **GPS Jamming** [27] [23]

The term GNSS covers various different satellite constellations that provide position, navigation and timing (PNT) information including GPS (United States), Glonass (Russia), Beidou (China) and Galileo (Europe). This thesis focusses on GPS satellite constellation and receivers.

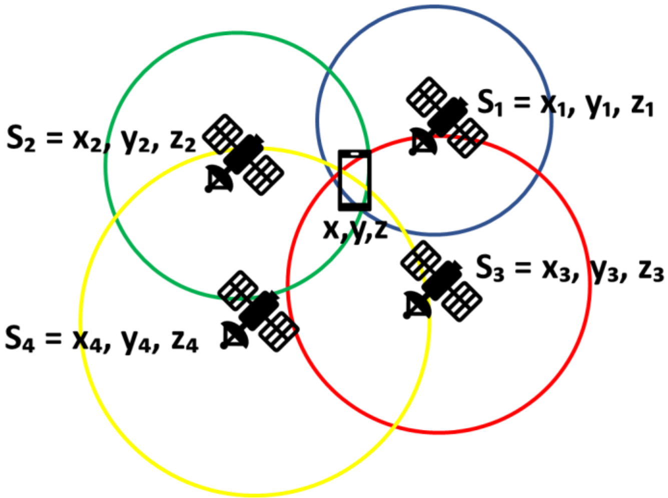


Figure 2 – GNSS Pictoral Represenations of Triangulation for Calculting Receiver Position

GNSS receiver position (x,y,z) is worked out using the calculation of triangulation. Satellites are constantly broadcasting position and timing information. The first 3 satellites are required to perform the triangulation and the fourth satellite is used to calculate a timing error. This is because the satellites each have atomic clocks, so their timing signals are highly accurate to each other. Ground receivers generally do not have access to atomic clocks as it would be impractical to carry these around in our phones for example. The fourth satellite is required to calculate a timing offset for the receiver so that the distances can be accurately measured. Figure 2 shows a pictorial representation of this.

GPS is comprised of satellites, control segment, monitoring stations and the user segment which is made up of many GNSS receivers. The GPS signal is modulated using a carrier frequency before it transmits to the receiver from the satellite. The main two frequencies in use for GPS are 1575.42 MHz or the L1 band and 1227.60 MHz also known as the L2 band. By the time these signals reach the receiver they are very weak. GPS is susceptible to interference from jamming as the signal from the satellite is approximately -130dBm when it reaches the receiver [94]. The weak signal at the receiver has been described as the Achilles heel of GPS with satellites technically producing less power than a typical car headlight from 20,000km into space [95].

For many sectors, GNSS provides vital position, navigation and timing services but are extremely vulnerable to interference due to the signals being weak at the receiver. Space communications in the UK are deemed critical to national security and therefore classed as Critical National Infrastructure [96]. In 2017 the potential economic impact from a loss of services for 5 days was assessed at £5.2bn [13]. Although this study has not been repeated, it can be assumed that the figure is much higher today. In 2021 society is highly dependent on GNSS for applications from power distribution, emergency services, travel and even 5G performance for a precise timing signals [14]. A further concern is the cascading effect on secondary and tertiary sectors from GNSS disruption [97]. Intentional interference known as jamming occurs when a jamming signal is transmitted at high power in a GNSS band, disrupting the GNSS service. Jamming equipment is illegal in many countries. In the UK the use of a jamming device is an offence under the Wireless Telegraphy Act but it is not illegal to buy or own the equipment [98]. Their availability on the open market can be seen from as little as £10.99 [99]. This availability combined with an increase in the use of UASs causing public annoyance, has created fears of a stark increase in usage, making them a serious and credible threat to satellite navigation. An increased use of UASs by law enforcement, hobbyists and commercial, has uncovered a potential motivation for civilians to purchase jamming devices to illegally combat their use. Morong *et al.* [100] carry out a study of GNSS jamming in a real world environment. They assess that GNSS jammers are very dangerous to aircraft and UASs, especially those that are flying at low height. The IET recently did a study to look at how easy it would be to purchase a Drone Jammer Gun from Asia revealing a straightforward practise [101].

However, a high powered jamming gun is not required to create a significant and critical effect. In 2013 a truck driver in New Jersey, USA, drove a route past Newark airport over several months whilst utilising a GNSS jammer in his vehicle to hide his movements from his employer. He was fined $32,000 for interfering with the Air Traffic Control System in Newark Airport [102]. With an increase in employers tracking employee movements there has been an increase in use and they have been found to be used in Mexico in 85% of vehicle thefts [103]. The European Global Navigation Satellite Systems Agency monitored GNSS interference in 23 countries over 2 years. The study showed 450,000 events whereby 73,000 had an impact on GNSS which was deemed significant. 66,000 of those were shown to have originated from jammers [104]. GNSS jamming equipment can be purchased easily online. Jammers which plug into car cigarette lighters are easily available online from under £10 [99]. The first step towards the mitigation of GNSS jamming is the detection and classification of the signal.

Today, with low cost SDRs on the market it is possible to re-produce many different jamming signals. Lineswala and Shah show this in [105] for jamming the Indian Regional Navigation Satellite System. Ferreira et al. [106] show the use of GNURadio software to produce jamming signals which are then transmitted on a BladeRF SDR to jam GPS signals for the disruption of UAS operation. Glomsvoll and Bonenberg [107] showed maritime GPS receivers to be affected by even low power jamming signals from 1600 metres away. Positional accuracy was affected from under 1000m and showing up to 10m discrepancy in position. The potential harm which could result from the use of a low cost jamming device becomes more critical when future reliance of Connected and Autonomous Vehicles (CAV) such as driverless cars on GNSS is considered. Pham and Xiong [108] present a survey of 184 papers considering state of the art attacks on CAVs. With regards to GNSS jamming they determine that the timely detection of a jamming incident in enough to ensure CAV safety and can be a pre-cursor to filtering out the attack signal so that the CAV can continue its operation in certain circumstances.

Jamming types have been classified into various categories in previous studies. A graphical representation of this can be seen in Figure 3 below.

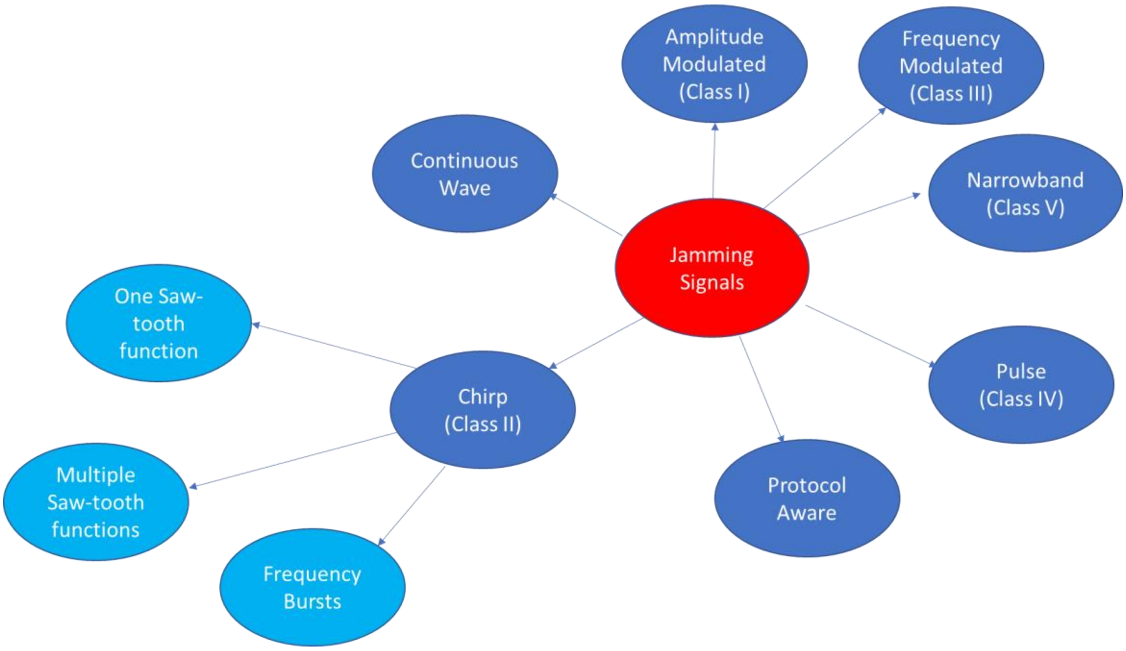


Figure 3 - Graphical Representation of Jamming Signal Types as described in Literature

Kraus *et al*. [109] in 2011 suggested four classifications for civilian GNSS jammers. In 2019 Ferre *et al.* [110] expand this to 6 classes; Class I: Amplitude Modulated jammers; Class II: Chirp jammers; Class III: Frequency modulated jammers; Class IV: Pulse or Distance Measurement Equipment (DME) jammers; Class V: Narrowband (NB) jammers; and Class

VI: No jamming signal present. These classes act as a basis for the datasets used and generated in section Chapter 3.

1.3.3 Software Defined Radio (SDR)

Traditional radio systems work by providing various capabilities entirely by hardware elements [111], designed and built for one specific purpose. Reconfiguration commonly requires changing elements of the hardware design. The term SDR was first used by Joseph Mitola in the early 90s as a class of radio which could be reconfigured through only using software [112]. His vision was a radio with just an antenna and an Analogue to Digital Converter (ADC) as the physical analogue components. All other functionality would be programmable through software [113]. The IEEE definition of an SDR is a "radio in which some or all the physical layer functions are software defined" [3]. In 2004 Eric Blossom, the developer of GNU Radio, said "software radio is the technique of getting code as close to the antenna as possible. It turns radio hardware problems into software problems" [114].

Figure 4 shows the functional blocks contained in a typical SDR receiver. The RF Front End takes the RF signal and converts it to analogue Intermediate Frequencies (IF) using a mixer with a local oscillator and amplification. The ADC then samples the IF signal at discrete intervals. Next a process called Channelisation occurs within the Digital Down Convertor (DDC). The Channelisation process modulates the IF to centre the channel of interest in line with the baseband frequency [115]. The Digital Signal Processor (DSP) implements a wide range of functions for example error correction and cryptography. General Purpose Processer (GPP), programmable hardware such as Field Programmable Gate Arrays (FPGAs) and

Application Specific Integrated Circuit (ASIC) are also potential devices for implementing baseband processing.

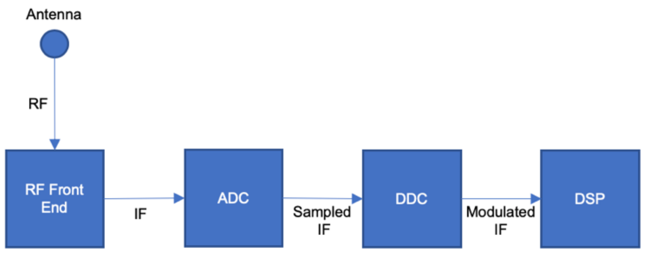


Figure 4 - Functional Blocks which make the basis of a SDR which includes the RF front end; Analogue to Digital Convertor (ADC), Digital Down Convertor (DDC) and the Digital Signal Processor (DSP)

Antenna Selection and RF Front End

Traditionally developers designed and built radios with the ability to transmit and receive in specific narrow bands. For example, an FM radio could not receive mobile phone signals. There are advantages to directional narrow band antennas as they reduce interference and tend to have increased performance [116]. As discussed earlier, the advantage of an SDR is its flexibility and re-configurable nature, so its usage suits a wide band antenna to take full advantages of those features. Antenna design for a truly wideband antenna (e.g. 20 MHz to 6 Ghz) is incredibly difficult, especially compared with traditional narrowband systems. Antenna size is proportional to the wavelength so a wideband antenna will always suffer from performance issues compared with narrowband antennas [117].

In some circumstances smart antennas could start to bridge this gap. For example, beam steering antennas use phased arrays to steer beams dynamically to users. Another example is beam switching which switches between directional beams according to user locations [118]. Smart or Intelligent antennas select and deselect capacitors and inductors, allowing for a changeable frequency response [119]. Once the chosen antenna receives the signal, the RF front end changes that RF signal into an Intermediate Frequency (IF) signal. A Low Noise Amplifier can be placed as close to the antenna as possible to reduce noise and amplify the signal and then a mixer is used with a Local Oscillator to convert the signal to IF [120].

ADC and DDC

In the past developers found the Analogue to Digital Converter (ADC) a limiting factor in SDR development. The ADC turns the IF signal into digital samples. To ensure the signal sampled replicates the original signal, Nyquist states that the sampling frequency must be at least twice the bandwidth of the signal for correct representation. At baseband this equates to twice the highest frequency. Power consumption and efficiency other parameters in the design phase which require attention, especially for battery powered SDRs. Battery powered SDRs have motivated the development of ADCs which are more energy efficient [120].

Another consideration is the effect of jitter caused by the high-speed sampling clock. When the clock varies, it effects the sampling of the input and this effect increases in severity as the frequency increases [121]. It should also be noted that higher sampling rates imply faster electronics and hence increased power consumption.

The signal then passes from the ADC to the Digital Down Converter (DDC). A DDC typically consists of a mixer and an oscillator. The mixer multiplies digital IF signals with complex sinusoidal waves and down converts the data to a baseband frequency. Once the baseband signal is acquired, a low-pass filter is used to filter out the harmonic frequencies [122].

Baseband processing

There are different options for signal processing which each have advantages and disadvantages. The first is the use of a General-Purpose Processor (GPP) such as the x86 microprocessor. A GPP is defined as a “digital circuit that is clock-driven and register-based, and is capable of processing different functions, operating on data streams represented in a binary system” [123]. GPPs do not have the processing power for real time requirements but their flexible nature makes them attractive for some SDR systems [124]. Digital Signal Processors (DSPs) are a type of GPP that can conduct high speed instruction and arithmetic processing. DSPs have been optimised for digital signal processing [125] and promote flexibility, for example they are able to be reprogrammed when standards change [126].

Application-specific integrated circuit (ASIC) advantages include power consumption and efficiency. Example applications include the implementation of wireless standards. However, ASICs are hard wired and not re-programmable after the design phase. Therefore ASICs do not lend themselves well to the concept of an SDR [124]. Further, ASICs can have a long development time and at a large cost. An alternative is Field Programmable Gate Array (FPGAs), arrays of logic blocks where the routing is programmable in seconds [127] and reconfigured in milliseconds. However, FPGAs are slower and less efficient than ASICs. In order for FPGAs to compete with ASICS in low power applications, FPGA designers must address the issue of power efficiency. Grover and Soni [128] found that FPGA optimisation at the system level resulted in significant power savings. Previously FPGA development required experienced VHSIC Hardware Description Language (VDHL) software programmers. Today High-Level Synthesis (HLS) allows the design of applications in C++ for example, without needing to understand or program in VHDL. The HLS compiles the code in the correct language for the FPGA making it more accessible for software programmers [120]. Designers in general consider FPGAs less flexible than GPPs. However, FPGAs perform reconfigurable computation with much higher speed and efficiency and lend themselves to real time processing applications [123].

History

The concept of a programmable radio was first considered almost 50 years ago and the first design was developed through United States research departments within government organisations in 1970s–1980s [129]. US and UK military ground troops used an ADC connected to an 8085 microprocessor as a VLF radio [130]. In 1984 E-Systems (now Raytheon) published a newsletter which referred to a ‘software radio’. It used arrays of processors to perform functions such as adaptive filtering [131]. In 1991 the US Defence Advanced Research Projects Agency (DARPA) released details of a military program called SPEAKeasy. SPEAKeasy was a radio which operated between 2MHz and 2 GHz, supporting up to ten different protocols in an open architecture, with software implementing all physical layer components [131].

In 1992 at the National Telesystems Conference Joseph Mitola presented a radio which could be reconfigured using only software and included the ability to both transmit and receive [112]. The community credits him with the term Software Radio and he was the first person to publish on the technology. In 1996 a partnership between industry, government and academia, at the initiation of the US Department of Defence (DoD) was set up called MMITS. The DoD wanted MMITS to define an open architecture for SDRs [132]. In 1997 the DoD set up the Joint Tactical Radio System (JTRS) to try and collate various stove piped service led programs into one main effort. The intention was for JTRS radios to be compatible with legacy radios and provide further capability for the warfighter to access information [133]. The US government shut down the JTRS program in 2011 but it undoubtably made a significant contribution to the worldwide community of SDR enthusiasts.

In the late 1990s, the use of the SDR spread from military application to the commercial sector, in particular cellular networks and base stations were considered a potential application [134]. In 2000 a company called Lyrtech worked in conjunction with Mathworks to create a development environment for fast prototyping. The hardware platform was created using Digital Signal Processors and Field Programmable Gate Arrays (FPGA) and the Mathworks Simulink formed a bridge for simulation and execution [135]. This advancement provided a commercially available development suite for SDRs. In 2001 Eric Blossom, funded by Sun Microsystems, founded an open source framework called GNU Radio. The framework allowed for the development of SDR using a free toolkit within a PC environment [136]. Developers use GNU Radio with a variety of platforms and hardware to perform signal processing and to create SDRs. Applications can be programmed in both Python and C++ languages [137].

In 2005 a company called Vanu released the first commercial SDR base station system certified by the US Federal Communications Commission [138]. Aimed at the 3G market, Picochip introduced the PC102, dramatically reducing the power, cost and size of the equipment[136]. Up until the introduction of the PC102 commercial SDRs had been based on x86 processors. In 2009 Lime Microsystems released a FPGA which could handle the RF Front End on one chip. Research into the use of FPGAs continued and in 2016 Osma proved an FPGA-based SDR using Xilinx System Generator programming and Matlab [139].

SDR Comparison

Today SDRs are used for a multitude of tasks from wireless base stations to military communications [140]. The real advantage of an SDR is flexibility, the ability for functionalities such as changing frequency bands and updating protocols to be implemented through a software download rather than full hardware replacement [141].

Table 1 - SDR Comparison

	RTL-SDR V3 [142]	HackRF One [143]	BladeRF x40 [144]	Ettus USRP B210 [145]
Frequency Range	500 kHz – 1766 MHz	1 MHz - 6 GHz	300MHz to 3.8GHz	70 MHz – 6 GHz
Bandwidth	2.4 MHz	20 MHz	28MHz	56MHz
Sample Rate	3.2 Msps	20 Msps	40 Msps	61.44Msps
ADC	8 bit	8 bit	12 bit	12 bit

TX	0	1	1	2
RX	1	1	1	2
Duplex	N/A	Half	Full	Full
Interface	USB 2.0	USB 2.0	USB 3.0	USB 3.0
Chipset	RTL2832U	MAX5864, MAX2837, RFFC5072[146]	LMS6002D	AD9361
Price	£16.73 [147]	£217.95 [148]	£408.75 [149]	£1,100

Table 1 shows a comparison between the main SDR competitors available for purchase today. The Ettus USRP series represents the higher price end of the market. The mid-price range includes the full duplex BladeRF which uses a high speed USB3 connection with an FPGA. A single chip, the LMS6002D, performs most of the functionality including the mixer and Analogue to Digital Converter. It can cover 300 MHz to 3.8 GHz on the spectrum, but frequency range is limited due to the single chip approach. The Ettus B210 uses the AD9361 which has similar functionality but uses 2 chips. The HackRF uses several different components to achieve its frequency coverage. The Ettus B210 and the BladeRF both utilise the Cypress FX3 microcontroller (ARM9 core), while the HackRF does not comprise of an FPGA but uses complex programmable logic and an NXP microprocessor [150]. At the bottom end of the market the RTL-SDR digital TV receiver is low cost and easily converted for SDR use. An observer should not discount the RTL-SDR due to its lower specifications detailed in Table 1. This very low cost SDR has proved it usage in multiple capacities, for example its ability to receive live weather satellite imagery through reception of an Automatic Picture Transmission signal [151]. Due to the advances in computational power and their low cost, SDRs provide researchers with the ability to design and evaluate cyber-attacks on different wireless systems [152]. As technology rapidly continues to advance and RF communications become even more embedded into our everyday lives; from intelligent transport systems to IoT devices in the home, to the control and navigation of aircraft in the sky, RF vulnerabilities increase. SDRs allow RF spectrum analysis which can allow an attacker to reverse engineer a signal in order to devise new attack methodologies [153]. SDRs can be used to both expose cyber vulnerabilities and to provide early warning of unwanted signals in a prohibited airspace. Next their use for exposing cyber vulnerabilities will be considered.

1.3.4 SDR RF enabled Cyber

RF signals provide access points into many connected systems, in some cases bypassing security controls. Many devices integral to our daily lives contain analogue sensors, from mobile phones to Supervisory Control and Data Acquisition (SCADA) systems. The information and data provided by these sensor inputs can form part of critical decision paths which can be vulnerable. The application of Cyber Security does not generally include any consideration of the RF spectrum. However, spectrum access underpins many types of technology of which cyber security is a concern. In a 2018 white paper [154], QinetiQ proposes the term cyber-spectrum security and predicts that attacks utilising the spectrum will increase due to factors such as the interconnectivity of radio systems.

Examples of RF access based attacks have been seen in different forms. In the U.S. an SDR was used to compromise an RF link which formed part of an Emergency Services System.

The attack set off 150 weather sirens across the city for over 90 minutes [155]. Significant media coverage regarding Remote Keyless Systems on new vehicles highlights a practical example of cyber security vulnerabilities exploited via the RF spectrum. Ibrahim *et al.* [156] demonstrate a hijack attack on a car lock by jamming, listening and then replaying a signal. Even with the prominence of these attacks, recent government literature on connected vehicles is still focussed on the security of information over spectrum vulnerabilities [157]. Another example is seen in [158] where Tripple *et al.* proved the insertion of digital values into microprocessors and embedded systems using intentional acoustic interference through the addition of steps onto a Fitbit[1].

In March 2019 the UK National Air Traffic Services announced the start of operational trials of a space based Automatic Dependent Surveillance Broadcast (ADS-B)[2] following a $69 million investment in 2018 [159]. Moser *et al.* [160] show ADS-B is vulnerable to spoofing even with the addition of multi-lateration techniques to try and verify the integrity of a signal. Vulnerabilities in ADS-B is another example of the use of RF links as a viable access vector for a cyber-attack. Spoofing has also been seen with other RF signals such as with the operation of Satellites and the vulnerabilities with technologies that provide PNT information. Wang *et al.* [161] proves the use of a low cost SDR to change the date and time of a target device. With SDR technology decreasing in cost and increasing in complexity, security professionals must look to secure any dependence on wireless connectivity. The first stage of mitigating against RF enabled cyber-attacks is the detection and early warning of unwanted signals in an airspace. SDRs can also be used in an early warning capacity.

[1] Wireless enabled activity tracker that measures data such as steps walked, heart rate and other fitness metrics.

[2] An aircraft automatically uses satellite navigation such as GPS to determine its location and broadcasts it to other sensors and aircraft

Chapter 2 - Literature Review

2.1 UAS Detection and Classification [21] [28] [25] [24]

This section is split into the main research areas for small UAS detection and classification including imagery based systems (electro-optical and thermal), acoustic, radar, RF, LiDAR and LADAR based systems. For each sensor type the review is conducted chronologically ending with the most recent literature. Each of the techniques have strengths and weaknesses but RF is seen as the best tool for detection at distance. The ability to detect and classify a UAS with enough time for decision making regarding the threat is key to dealing with the threat [162].

2.1.1 Imagery

Thai et al.[163] use a CNN for feature extraction of motion patterns and k-nearest neighbour (kNN) to classify results with an accuracy of 93%. Acceptable accuracy values would be defined in the system requirements for an early warning system and would be highly dependent on what the use case for the system in question. Schumann et al. [164] use a convolutional neural network (CNN) to detect UASs for the birds vs. drones competition. They find that when they increase the training data to include images from the web from many varied scenarios, their results increase in accuracy. They use the metric average precision and achieve their highest overall result as 80%. The birds vs. drones challenge [165] is set up because drones can be confused with birds. Saqib et al. [166] compare various CNNs for UAS detection including pre-trained VGG-16 using transfer learning. They conclude that faster R-CNNs produce the highest performing results (average precision over 60%) when comparing all the architectures. Schumann et al. [167] present a CNN which is trained on video which have been pre-processed with median background subtraction. The classifier is able to distinguish the UAS from birds and other background noise. Rozantsev et al. [168] use 3d Histograms of Gradients (HoG3D) and a CNN to perform detection using one camera. They utilise a regression approach for motion stabilisation and produce an open source dataset for the community and achieve 86% average precision. Aker and Kalkan [169] propose a CNN for small UAS object detection using background subtracted images. They show that birds can be distinguished from small UAS using this method.

Yoshihashi et al. [170] use deep learning and a multi-frame approach for identifying small objects. They present a Recurrent Correlational Network which can perform detection and tracking simultaneously using multiple frames. Peng et al. [171] address the need for large datasets to train neural networks by creating an artificial dataset of photorealistic UAS images using a process called Physically Based Rendering Toolkit. The new images rendered vary with orientation, camera specs, background details and UAS positions/size. The network is trained using Faster R-DNN and achieves higher accuracy with the larger dataset than using a smaller one. Lee et al. [172] uses camera imagery and a CNN to classify the UAS type. Their detection system is based on a second UAS and the camera imagery collected from it. The dataset consists of google images and the OpenCV library to identify the location of the UAS on the image and the make/model of the UAS. The system has an accuracy of 89%. Unlu et al. [173] use General Fourier Descriptors as features for UAS detection and classification from birds. In their work they show that using these features with a CNN produces the highest accuracy for classification. Nalamati et al. [174] consider various CNN architectures such as ResNet and Inception. Transfer learning is employed due to limited data and they show that ResNet using the Faster-RCNN is preferable for higher accuracy results (improving accuracy by 15%). Coluccia et al. [175] evaluate the highest performing algorithms produced for the 2020 Drones v. Birds competition, achieving 80% average precision. Challenges for video detection and classification were observed when classifying between birds and UAS at greater distances. The other problem perceived was with moving cameras. Coluccia et. al recommend training data to be updated to include greater distances and for categories such as real time detection and computing complexity be considered in the future.

We will now consider the use of thermal imagery for small UAS detection and classification. Andraši et al. [176] investigate the use of thermal signatures for small UAS detection. They find that quadcopter style UAS which are electrically powered do not produce a significant amount of heat compared with UAS which use fuel consumption for power. This is due to the efficient nature of electrically powered motors and the air circulation within the quadcopter. Diamantidou et al. [177] propose a fusion process using a neural network. The fused information includes thermal imaging but does not expand on specifics details such as resolution [178]. Wang et al. [179] produce a monitoring system using visible and thermal imagery. They suggest that the largest issue for this work using deep learning techniques is a lack of data. They address this using augmentation techniques and show that systems trained on synthetic data performs well on real world UAS images, even those incorporating complex backgrounds. Svanstrom et al. [178] propose a multi-sensor detection system for UAS. They observe that the thermal camera which has a lower resolution has equal performance to an electro-optical camera.

2.1.2 LiDAR and LADAR

LiDAR stands for Light Detection And Ranging and is commonly used for object detection for example it is utilised for collision avoidance with autonomous vehicles using simultaneous localization and mapping (SLAM). However the range detection on the sensor is limited to 100m [180]. LADAR stands for LAser Detection And Ranging. The basis of the technology is the same but LADAR systems have a greater distance and are therefore more suited and commonly used for detection and classification systems.

Kim et al. [180] highlight the problems of LiDAR for detecting UAS at distance and propose a 3D LADAR sensor which can perform detection up to 2km. Depending on the size of an airfield, which can vary in the UK between anywhere from 0.2 miles to 2.7 miles, with Gatwick Airport being 2 miles in length [181], multiple sensors may be required to cover that space. They propose a data augmentation method and a clustering algorithm variable-radially bounded nearest neighbour (V-RBNN) which overcomes issues with prior research using an RBNN. However, they do not consider the issue which LADAR suffers in distinguishes between UAS and birds but they do propose this for future work. Khan et al. [182] observe the issue of LADAR datasets lacking long distance targets. They propose a fusion data generation methodology for small targets and present an approach for real time small UAS detection. Kim et al. [183] extend their previous work to consider a double pan-tile scan laser radar to detect a 0.5m small UAS in real time from a distance of 2km. They show that this is achievable with complex background scenario with a calculation time of 20 millisecond per frame. Salhi and Boudriga [184] consider arrays of laser sources and concentrators in a spherical form with a photodiode in the middle. This method is evaluated by Salhi and Boudriga for UAS detection.

2.1.3 Acoustic

Audio signals have been studied using microphones pointed in different directions that can capture sound up to 30ft away. Mezei et al. [185] use mathematical correlation as a way of fingerprinting audio signals from different UASs. This method concentrates on capturing the motor sound which resonates around 40KHz but this technique struggles in urban areas where the base noise level is high. Busset et al. [186] combine the first approach with a video system [164] achieving a detection range of 160-250m (UAS type dependant). Nijim and Mantrawadi [187] use data mining techniques for UAS in flight identification. They use a Hidden Markov Model to perform phenome analysis with good results. However, the work does not consider the issue of noise or any real time considerations. Jeon et al. [188] do consider real-life environments when analysing the audio data from UAS by augmenting the audio with environmental sound produced from diverse environmental settings. They consider a CNN, a Gaussian Mixture Model and a Recurrent Neural Network (RNN) for detection and the RNN provides the highest accuracy using 240ms of audio. Bernardini et al. [189] use features from the time and frequency domain such as spectral centroid, spectral roll-off and Mel Frequency Cepstral Coefficients with machine learning classifier support vector machine to detect an audio fingerprint from a UAS. Their work uses audio with sampling rates higher than 48kHz taken from the internet. Flying UAS are classified against nature sounds, traffic noises, crowded areas and trains. They produce high accuracy of 96.4% using this method. Kim et al. [190] tackle the aspect of real time monitoring and detection using acoustic signals. They implement a FFT on the data in real time and perform detection with machine learning classifiers kNN and plotted image machine learning (PIL). Kim et al. use a mixture of indoor and outdoor audio signals with environmental noise created using

YouTube. They further produce a framework to allow other researchers to test other classifiers in the future.

Yue et al. [191] use acoustic signals collected using an SDR and machine learning algorithm support vector machine for small UAS detection. They focus on the frequency domain representation Power Spectral Density (PSD) which they showed reduced noise and produced successful detection rates. They also added in Additive Gaussian White Noise (AGWN) so they could control the SNR and observe the results in different noise scenarios. Seo et al. [192] study the 2D features produced by the Short Time Fourier Transform (STFT) for UAS detection. They use 20ms samples with AWGN noise added and the produced STFT image is then classified using a CNN. They found that they could successfully distinguish between scooters and motorcycles using this method but suggest further research to consider wider acoustic signals which have similar harmonics. Matson et al. [193] use features derived from STFT and mel-frequency cepstral coefficients for inputs as 2D images to support vector machine and CNN for detection of small UAS. The work was limited to the testing of one type of UAS the Parrot AR and background activities. STFT features and support vector machine produced the highted accuracy for detection. Shi et al. [194] focus on real time detection and localisation of small UAS using audio signals through an array of microphones and [195] follow the same methodology. Localisation is performed with Time Distance on Arrival (TDOA) and using a Bayesian filter.

2.1.4 Radio Frequency (RF)

Shin et al. [196] investigate the security of UAS controllers with a particular emphasis on the employment of Frequency Hopping Spread Spectrum (FHSS) technology. Shin et al. show it is possible to extract the hopping sequence of a controller using a Universal Software Radio Peripheral (USRP) SDR. Nguyen et al. [197] suggest investigate two methods for UAS detection systems. Firstly they look at an active approach which sends an RF signal and waits for a returned reflection. The second is a passive method whereby the signal is observed and analysed. Both approaches use the USRP SDR and they find that as the distance is increased the detection accuracy suffers. Shi et al. [198] use support vector data description with has fingerprints to detect small UAS operating in the 2.4GHz frequency band. They focus on producing the envelope of the RF signal before creating a fingerprint. The system is only tested in an indoor environment and accuracy decreases with the addition of AWGN. Nguyen et al. [199] examine whether features of UASs such as body vibration can be used to distinguish UAS from moving Wi-Fi or mobile phone signals. They use a SDR to test the detection accuracy at various distances using this technique. Abeywickrama et al. [200] present an autoencoder DNN for UAS direction finding using a single channel RF receiver. They achieve this using a circular antenna array which is directional and the accuracy of the direction finding was shown to be over 90%. Zhao et al. [201] classify UAS RF signals collected from a SDR using Auxiliary Classifier Wasserstein Generative Adversarial Networks (AC-WGANs). In an indoor setting the system shows an accuracy of 95%.

Ezuma et al. [202] use a Markov models-based naïve Bayes decision mechanism with UAS RF signals for UAS detection of 15 types of UAS controllers. Once the UAS is detected, 5 different machine learning classifier were evaluated for classification. Ezuma et al. evaluate the systems performance at various SNR and includes two of the same controller in the set. Huang *et al*. [203] expand detection to include a low cost method of localisation using multiple HackRF SDRs. Al-Sa'd et al. [71] are the first authors to consider classifying the flight mode of the UAS - whether it is switched on and connected to the controller, hovering or flying with or without the video feed. The use a DNN to detect the UAS and classify its type and flight mode. Accuracy declines as the number of classes increase. They produce 99.7% accuracy for UAS detection, 84.5% for classifying the UAS type and 46.8% for classifying the flight mode. They also contribute by producing the DroneRF dataset [204] for other researchers to use. Liang et al. [205] use the Hilbert Spectrum for features and an ANN for micro UAS detection. SNR is noticeably improved by using a higher order cumulant algorithm. Al-Emadi and Al-Senaid [206] use the DroneRF dataset and a CNN to improve the results of Al-Sa'd et al. by 0.1% for UAS detection, 1.3% for UAS type classification and 12% for UAS flight mode classification. Their work suggested that the CNN produces higher accuracy for classifying UAS signals over a DNN.

Nemer et al. [207] propose a hierarchical approach using ensemble learning and pre-processed RF data. The first classifier detects the UAS, the second classifies it by type and the following two specify the flight mode. The method is shown to produce classification accuracy of 99% but the flight mode classification is limited to the Bebop and AR UAS. Future work looks to work with larger and more diverse datasets.

Lv et al. [208] compare the Power Spectral Density (PSD) of a signal with an average PSD to achieve the detection of small UAS. The method boasts advantages when computational power and efficiency are constrained.

2.1.5 RADAR

The majority of the research we will present with respect to radar is regarding frequency modulated continuous wave (FMCW) and higher resolution radars which allow for the analysis of the micro-Doppler signature. FMCW has suffered issues with detecting and classifying small UAS due to their size, especially when classifying between other flying objects such as birds at distance, and micro-Doppler signatured based methods are limited in terms of range (70–120m [209]). De Wit et al. [210] show that micro-Doppler signatures can be used as a characteristic to classify small UAS. Molchanov et al. [211] extract features from micro-doppler signatures from a 9.5GHz radar. Pre-processing includes filtering and alignment to overcome the Doppler shift. Classification accuracy using Support Vector Machine produces an accuracy of 95% for birds, planes, small UAS and helicopters. Harmanny et al. [212] again tackle the issue of successfully discriminating between birds and small UAS using micro-Doppler signatures. Spectrogram and Cepstrograms are used to not only distinguish between bird and UAS but to visually determine the UAS type.

De Wit et al. [213] characterise between fixed and rotary based UAS by using Micro-doppler features after first distinguishing between a UAS and a bird. Singular Value Decomposition is used to extract features from spectrogram images. Mohajerin et al. [214] use a mixture of simulated UAS, bird and plane radar tracks and a range of statistical features for classification. The work however doesn’t fully take into account atmospheric and environmental effects. Fioranelli et al. [215] use multi-static RADAR and micro-Doppler signature analysis to classify between UASs carrying different payloads. They achieve accuracy over 90% using centroid features from the micro-Doppler signature. Zulkifli and Balleri [216] design and develop a radar system to detect nano-drones using a micro-Doppler technique. Semkin et al. [217] consider radar cross-sections for detection and classification in urban environments and suggest further work to include the use of machine learning. Practical issues with RADAR detection systems include the cost of implementation and the requirement for licenses for transmission [53].

Ritchie et al. [218] observe that micro UAS electromagnetic scattering caused by the rotor blades produces strong variations with azimuth and frequency. Jahangir and Baker [219] use Holographic Radar (HR) that uses a 2D antenna array to create a multi-beam 3d surveillance sensor. Decision tree which is a machine learning classifier is used to distinguish between false tracks successfully and overall detection of 88%. Lundén and Koivunen [220] use a deep CNN to extract features from high range resolution profiles (HRRPs) for multi-static radars. They show that large target aircrafts are distinguishable using this method but the work does not extend to smaller targets such as UAS. Torvik et al. [221] use polarimetric parameters to address the problem with radars distinguishing between birds and UAS which are comparable in size. They focus on reducing critical detection time by using polarimetric features with a nearest neighbour classifier. Oh et al. [222] perform automatic multicategory classification of mini UAS using empirical-mode decomposition. Statistical and geometrical features are fed to machine learning classifier support vector machine for prediction.

Kim et al. [190] were one of the first to consider a CNN with merged Doppler images to perform classification of UASs. They found frequency domain features to be more robust than micro-Doppler signatures, which prior work had concentrated on. Mendis et al. [223] use a deep belief network (DBN) and spectral correlation functions (SCF) from a Doppler radar. SCF are used due to their resilience to noise. They show that environments where SNR is less than 0 that high levels of accuracy (above 90%) can be maintained. However, their work does not consider UAS in motion but only in static positions. Ren and Jiang [224] highlight that existing micro-Doppler spectrogram, Cepstrogram and cadence representations do not include any phase information but only magnitude. They address this using a 2 dimensional complex Fourier transform and in doing so improve error rates for cadence velocity diagrams. Zhang et al. [225] enhance micro-Doppler signature robustness by using short-time Fourier Transform spectrograms in the K-band and X-band with principal component analysis for feature extraction. Support Vector Machine is again used as the machine learning classifier and they show that the fusion of multiple radar sensors produces a higher accuracy than a single radar feed. Regev et al. [226] highlight the issue that UASs present radars due to their low radar cross section and when detection has occurred, distinguishing the UAS from birds or by UAS type. They use a Multi-Layer Perceptron (MLP) neural network to classify UAS type using the baseband signal from the radar return.

Fuhrmann et al. [227] classify UAS types - quadcopters, octocopters, helicopters and fixed wing platforms of various sizes using features extracted from time frequency transforms including Short-Time Fourier Transform, Cadence Velocity and Cepstrograms. Using Support Vector Machine classification accuracy is produced at over 96%. Oh et al. [222] use empirical-mode decomposition for UAS classification. They use eight statistical and geometrical features and SVM as the machine learning classifier. The utilise the unique patterns in micro-Doppler produced by the motion of the UAS blades. Ma et al. [228] investigate entropies Shannon, spectral, log energy, approximate, fuzzy and permutation to enhance mini UAS classification accuracy. Support Vector Machine again is used as the machine learning classifier. They show higher accuracy than compared work but with increased computational power requirements. Sun et al. [229] classify and localise drones using micro-Doppler signatures and dimensionality reduction. Sun et al. show robust feature selection which works at lower frequencies. Habermann et al. [230] introduce a new type of feature and use point cloud features generated from the radar return to classify helicopters and UAS. The classification uses an MLP artificial neural networks and can classify between different types of helicopters and UAS even in very low SNR environments.

Wang et al. [231] compare CNN detection methods with traditional CFAR methods. The CNN had the highest performance especially in low SNR environments. They also found that coincidence detection could improve CNN results. However, their work is limited to simulated data and is yet to be tested on real radar emissions. Samaras et al. [232] highlight issues with existing research requiring long illumination times and therefore a tracking radar architecture rather than a surveillance radar. This is due to existing methods being based on the Fourier spectra. Samaras et al. propose a deep learning solution for surveillance radar data to distinguish between UAS, birds and noise, producing accuracy of 95%. Choi and Oh [233] use a deep CNN to classify micro-Doppler signatures from different UAS. Wan et al. [179] prove that HRRP feature extraction for automatic target recognition using a CNN and spectrograms. However, again this work does not extend to smaller targets such as UAS.

Guo et al. [234] use a one dimensional CNN to overcome priori issues with HRRP sensitivity. Again the work does not consider smaller UAS targets. Chen et al. [235] develop motion models for UAS and flying birds and then calculate the variance in the time domain of the model occurrence probability to estimate the target before identifying and classifying it. The model results are validated using ground truth radar data from airport and coastal environments. Messina et al. [236] automatically classify UAS using machine learning and surveillance radar signals. The show classification between bird, planes and cars at higher than 98% and classifying between fixed wing and quadcopters at over 93%. Coluccia et al. [237] review the method of RADAR for the detection and classification of UAS systems. In particular they consider frequency modulated continuous wave (FMCW) radar as per the latest technology advances and also sensors which are spatially distributed. Amongst this they also discuss the main challenges for detection, verification and classification of small UAS. Passafiume et al. [238] investigate the reliability of FMCW radar images for classifying a UAS by the amount of motors it has and they rule out the rotation speed from effecting classification.

2.1.6 Critical Evaluation of Detection and Classification Methods

Table 2 shows a summary of critical analysis points observed from reviewing the current research in each field. It can be seen that distances of up to 3280ft can be achieved with RADAR for example. However, each method as already discussed has weaknesses. There is no one golden solution for small UAS detection, rather a fusion of sensors should be considered so the advantages of various method can be maximised.

Table 2 - Critical Evaluation of Detection Methods

Method	Example Detection Range	Example Detection Results	Critical Evaluation
Radar	3280ft [219]	88% Accuracy [219]	Dependant on good radar cross section. Struggles to distinguish between flying objects with similar radar cross section.
RF	1400ft [239]	99% Accuracy [240]	Unable to detect UAS operating in autonomous mode.

			Can be susceptible to inference in the same frequency band.
Acoustic	30ft [185]	96% Accuracy [189]	Most research focussed on detection. Short detection range.
LiDAR and LADAR systems	328ft [180] (LiDAR); 6561ft [180] (LADAR)	No comparable accuracy scored available as methodology based on clustering [180]	Sensitive to environmental factors. Difficulties distinguishing between UAS and birds.
Imagery (electro-optical)	350ft [239]	78% F1-score [178]	Lack of large open source datasets. Sensitive to environmental factors. Hard to perform 24x7 real time detection over a large area
Thermal	350ft [239]	76% F1-score [178]	Difficult to detect heat efficient UAS Lack of current research

Radar is one of the most researched sensor types as it is already the choice for large aircraft, it works in all-weather environments and is capable for truly long range detection. However, traditional radar systems were not designed to detect objects with small radar cross sections or objects that have similar cross sections to birds. So radar does suffer with these challenges. The next best detection ranges undoubtedly come from RF detection. However, RF methods will never be able to detect a UAS which isn't emitting any RF signals, i.e. a UAS operating in a GPS autonomous mode. Some RF methods have been shown to suffer when there are other signals present in the same frequency band. Acoustic signals have proved incredibly effective for UAS detection but the method is highly susceptible to noise and only works at a very short range. As with imagery systems, the success of LiDAR and LADAR systems is dependent on good environmental conditions as they will degrade in low visibility for example. They also struggle to distinguish between small UAS and birds, especially at distance.

While imagery based techniques have many advantages, including the ability to use camera systems already in place and the significant improvements which have been made using deep learning to distinguish birds from UAS, we note a number of considerations from our review. There is a lack of large open source datasets to support the deep learning research. This has been overcome with some success in various pieces of work through the use of transfer learning, using a pre-trained neural network or generating synthetic images to enlarge a dataset. In the papers we have reviewed we have not seen any work relating to detection range and this may not be achievable with this method. Lastly we note that line of sight is essential for this technique which leaves it vulnerable to changes environment conditions such as low visibility due to weather or to a purposeful denial of service by covering the camera with spray paint for example.

Although the results being produced with imagery are high in accuracy we feel it would need to be utilised alongside another method to ensure detection. Thermal imagery is also considered but has limited research which may be due to a lack of open datasets. Research from 2016 showed that quadcopters were harder to detect with this method due to the efficiency of the motors and inadequate heat emissions. Nijim and Mantrawadi [187] go as far as to say that it is near impossible to detect a battery powered UAS with thermal signatures. Subsequent research which has referred to thermal imagery has been in the context of multi sensor detection systems [179]. We can see that each individual method has advantages and challenges, there is no one golden ticket solution. This is why there has been an increasing number of researchers and commercial programs looking at either fusing sensor data to reap the benefits of more than one of these methods or considering a number of individual sensors together to make a detection prediction with more certainty.

2.1.7 Multi-Sensor Systems

In industry we have seen two European programs which are looking to use a combination of different types of sensor data. Safeshore [18] is a project which integrates together various detection methods with intelligent data fusion in an attempt to address maritime border security issues. The Aladdin project [241] looks to develop a system to detect, localise, classify and neutralise suspicious UAS. Detection, classification and localisation uses radar, optronic, acoustic and other sensors, while the neutralisation aspect is said to include jamming, hacking and physical methods. The project also includes the development of a deep learning algorithm to fuse the different sensor data together. The Drone Detection Grid [242] is a system made by DD Countermeasures which flags unknown transmitters within a certain frequency range. Systems like these are susceptible to false positives as no classification of the signal is performed, so it could be something other than a UAS flagging detection. [243] describes a solution based on a network of sensors which use energy detection to identify the

UAS and then correlation to classify them. Since the system uses a network of sensors, time difference of arrival can then be used to locate the device. However, due to the number of distributed sensers required these systems can be expensive.

In academia, Shi et al. [244] use multiple sensors to perform UAS detection, localisation and jamming as a counter measure. Acoustic, imagery and RF signals are collected and processed using Support Vector Machine in parallel. The data results are fused using a logical OR operation to produce a final detection decision. Diamantidou et al. [177] propose fusing features from multiple sensors for UAS classification and detection. They do this using a framework of a neural network to merge extracted features and increase accuracy from utilising the benefits of more than one sensor. Zhang et al. [245] perform small UAS detection, tracking and localisation with a multi-sensor approach. It uses visual imagery and deep learning object detection to define a bounding box. LiDAR is used to calculate distance measurements from the pixel data and thermal data to detect the bounding box and then track it. Although Zhang et al. are not fusing the data in order to improve detection and classification, they show how a combination of different sensor data can provide a richer overall system capable of multiple functions.

2.1.8 Counter Measure Systems

We split Counter-measures into three main categories, physical measures, jamming and cyber-attacks. Brust et al. [246] use a swarm of UAS, capable of self-organising when a malicious UAS is detected, to chase the malicious UAS. They achieve this using clustering and develop a system resilient to the loss of communications with the swarm. Rothe et al. [247] present a counter UAS method through catching a UAS in mid-air by using a formation of UAS and a net. Prior work has concentrated on single UAS with nets but this limits the net size. This research was limited to indoor testing but outdoor experiments are planned for the future. ‘Robotic Falcon’ developed by Michigan Technological University [248] and the SkyWall 100 which has just secured a contract with the European Union police forces use net capture technology [249]. Dutch Police trailed the use of birds of prey to tackle malicious UASs but there were issues with consistency and concerns over animal rights [250]. China developed an electric fence to stop malicious UASs entering prohibited areas [251]. France have tested a laser defence against small UASs ahead of the 2024 Paris Olympics Games [252]. BBC News report that the U.S. Air Force are employing a microwave based counter measure system [253] and India are exploring rubber bullets as a physical counter measure [254]. All of these physical countermeasures contain risk when employed. For example, shooting a small UAS with a rubber bullet over a crowded area could cause harm to civilians if it crashed to the ground.

Multerer et al. [255] combine a FMCW radar for UAS detection and a directional 2.4GHz jammer to produce an anti-drone system. The jammer again aims to disrupt the control signal between the UAS and the controller and the tracking function within the system allows the signal to continue to be jammed. Shi et al. [244] use RF jamming as a countermeasure for malicious UAS in wider multi sensor system which we presented earlier. The system receives the location of the UAS and calculates an azimuth angle for the jammer antenna to be steered towards. The goal of the jammer is to break the communication signal between the UAS and the controller. They observe the issues of controlling the jamming power and the unintentional consequences of interfering with other wireless communications in the same frequency band. Parlin et al. [256] propose protocol aware jamming for UAS controller signals which normal employ some form of frequency hopping. They use a SDR and experiments proved that a protocol aware jammer was more effective than sweep jamming and required less transmit power. Li et al. [257] target UAS eavesdroppers by getting physically close to the UAS and transmitting a jamming signal to disrupt the malicious UAS.

Sliti et al. [258] propose three types of attack scenario including jamming the control link between the UAS and its controller which usually instigates the UAS returning to a 'home' location. The 'home' location is something pre-set. It also propose a black hole attack whereby network traffic is dis-regarded and a replay attack which repeats real communication that is happening between the controller and the UAS. Chen et al. [259] consider false data injection onto UAS navigation algorithms on open source flight control systems. In particular they show they can compromise magnetometer measurements to directly affect the UASs state estimation and therefore seriously compromising navigation, stability and power consumption. Kwon et al. [260] specifically target the micro-air-vehicle communication (MAVLink) protocol which is an open source communication protocol for UAS. It is widely used with ground control based systems. They prove an attack which disables a UAS and its subsequent mission by exploiting a vulnerability with the protocol. As always when attack methodologies progress we see security measures progressing in parallel. One example is the work of Lei et al. [261] who propose a new lightweight authentication protocol for UAS. Westerlund and Asif [262] investigate Wi-Fi based vulnerabilities with two commercially available UAS which operate on Wi-Fi. Cyber based attacks including Denial of Service, De-authentication, man in the middle, root access and packet spoofing are specifically considered and proved, highlighting security vulnerabilities for Wi-Fi based UAS.

Aside from the methods considered above there are various reports of laser weapons for counter UAS [263], [264], projectiles [265],[266] and even the use of animals [267],[268].

Table 3 - Critical evaluation of counter-measure methods.

Counter Measure Method	Critical Evaluation
Physical	• Must be lightweight • Net mechanism must be large enough
Jamming	• Can interfere with other frequency band users • Illegal in some countries
Cyber	• Can be difficult to perform • Dependant on specific UAS type / communication protocol used which can be encrypted • Illegal in some countries

Table 3 shows, as with detection and classification methods, that there is not one gold plated solution. Jamming is incredibly effective but it comes at a cost of having other consequences. Cyber again is very effective but it is specific and therefore will incur cost and will likely not deal with unknown UAS. All methods must at the very least employ successful detection systems as a prior requisite and physical measures require tracking alongside this.

2.2 GNSS Jamming Detection and Classification [27] [23]

2.2.1 GNSS Specific

Related works to the detection and classification of GNSS jamming signals includes Lineswala and Shah [269] who consider Indian constellation signals and the use of the power spectral density (PSD) which is a measurement of power in the frequency domain to detect jamming. Although the work is successful at detection it does not look at the classification of the jamming signal type. Ferre et al. [110] take the work further and use spectrograms which are a representation of the signal power over time to both detect and classify the signal. They use machine learning classifier Support Vector Machine (SVM) to produce 95% accuracy and a CNN which produces 91% accuracy for the 6 jamming signal classes described above.

Other studies are less discriminative, Glomsvoll and Bonenberg [107] consider narrow and wideband jamming signals on high end receivers for GPS L1 and GLONASS L1 frequency bands. Their work concluded that the impact to GPS from jamming was more critical than jamming GLONASS. Lee et al. [270] consider a cloud based solution for the detection of GNSS jamming but rely on multiple receivers in the area of the jammer. They prove the detection of jammer type and estimate localisation based on the 2 dimensional time-frequency correlation between receivers. Kim et al. [271] again show the validity of considering the jamming signal as a 2 dimensional correlation in the time-frequency domain for detection of jamming type and estimated localisation. As in [270] this solution is based on generating a network of receivers. Xu et al. [272] propose the use of a DNN with time, frequency and transform domain features from the signal for jamming recognition. The DNN was able to detect 12 jammer types with over 99% accuracy and outperformed traditional machine learning classifiers. Sreeraj et al. [273] use PSD data and an adversarial autoencoder (AAE) to detect synthetically generated anomalies in the wireless spectrum. The model is trained in an unsupervised manner for mean squared error reduction and then via semi-supervised learning to learn the features where they achieve 80% accuracy.

2.2.2 Wireless Communications

Related work to the detection and classification of jamming signals in GNSS bands, is that of wireless communication links and spectrum monitoring. Tandiay *et al.* [274] use spectrograms stored as 2D images using a video prediction system called Prednet [275] to detect jamming through a comparison of the spectrum with no jamming signal present. O'Shea *et al.* [276] use spectrograms with a CNN in order to detect and classify signals for wireless spectrum monitoring, such as GSM, Bluetooth and LTE. Arjoune *et al.* [277] evaluate SVM, Random Forest and Neural Network as machine learning models to detect whether a jamming signal is present or not in wireless communications. This work only considers detection and not classification, which is proposed for future work. Wu *et al.* [278] assess CNN as a feature extractor for the classification of jamming signals in satellite communication networks and concentrate on the combination of multiple jamming signals. The work showed near 100% accuracy but capability to generalise on unseen data was not assessed. Yang and Zhu [279] consider the recognition of satellite interference signals using an incremental learning SVM model. They extend this model to allow it to function with multiple classes. Their overall conclusions were that although the model did not improve accuracy compared with traditional SVM models, it did reduce computing time and the memory capacity for training the models. It effectively improved efficiency and maintained accuracy compared to traditional methods but reduced the resources needed to produce though results.

For jamming signal detection and classification research also exists which utilises an SDR and a Raspberry Pi for real time classification. Price et al. [280] are concerned with the detection and classification of signals which jam the 2.4GHz control signal between a UAS and ground control station. They consider barrage, protocol aware, single tone and pulse jamming signal types using GNU-Radio and classify the signals using a random forest machine learning model. They then fit a small UAS with a HackRF SDR and use a raspberry Pi to make predictions in real time. They achieve accuracy of 93% and suggest further work needed for experimenting while the UAS is running and then implementing the classification as part of a wider counter system which could employ mitigations such as path re-scheduling.

2.2.3 RADAR

The detection of RADAR jammers is another area which is related to the detection and classification of GNSS jamming signals. Hao et al. [281] tackle the issue of detecting a specific type of jammer called a dense false target jammer. The jamming signal is designed specifically to interfere with pulse compression radar. Hao et al. use a feature extractor based on Gabor time-frequency atomic decomposition which are then fed to a SVM classifier for recognition. They found that for the problem of dense false target jammers that the Gabor atomic time-frequency parameter extraction and SVM where successful with a high detection rate (above 92%). Fu in [282] consider the issue of classifying different types of radar jamming signals. The extract features using the time domain, frequency domain and with the use of fractal dimensions. Fu uses neural networks to classify the signal types and specifically the paper looks a when there are more than two jamming signals operating at the same time.

Zhang and Cao [283] use SVM to consider the classification of radar interference signals. They argue that with the introduction of advanced driving systems that more and more interference issues concerning radar on radar are going to be experienced. They consider 6 different types of radar to radar interference. The driver assistance radars considered lie in the 77GHz range and Zhang and Cao consider frequency data with a linear SVM to produce real time classification for future driverless vehicles. They achieve 90.6% accuracy and use cross validation to highlight any overfitting in the model. Kong et al. [284] consider the recognition of interference signals for radio ground-to-air using SVM. In particular they consider an optimisation method called gravitational search algorithm which they conclude has high efficiency for recognition between interference and non-interference signals. Junfei et al. [285] consider barrage jamming of synthetic aperture radar (SAR) systems. They consider statistical characteristics of SAR echo signals and use a VGG-16 CNN to classify simulated interference types. Shao et al. [286] consider the classification of pulse compression RADAR interference to be an important step to be able to counter the jamming. They use a one dimensional CNN to consider classification with limited and resourceful datasets.

2.3 Transfer Learning [25]

Deep CNNs have proven very successful at object detection and classification over recent years. Many pre-trained CNNs are available which have been trained of large sets of images such as VGG-16 [287] and Residual Networks (ResNet) [288]. ResNet architectures allow for deep neural networks to be trained using a technique called skip connection which takes the output from an earlier layer and combines it with the a later layer. This technique overcame prior difficulties with training very deep neural networks whereby gradients would vanish due to repeated multiplication. ResNet50 has been commonly used for transfer learning research with a large scale image recognition database of over 14 million images called ImageNet [287]. Training the weights in a neural network from scratch can take a very long time and needs a large amount of training data, for example the 14 million images that trained the weights for ImageNet.

Transfer Learning allows other domains to benefit from the use of pre-trained weights for a new purpose. The main principle behind transfer learning is that the lower layers of the CNN are general and therefore we can find commonality and transferability between different types of domains. When transfer learning based pre-trained CNNs are used for feature extraction, traditional linear based classifier models which are quick to train can be used because they will have chosen any non-linear features in a robust manner using the CNN feature extraction [289]. The experiments within this thesis use two different CNNs for feature extraction using transfer learning. The first is the 16 layer VGG-16 [287] and the 50 layer ResNet-50 [290]. Both CNNs were pre-trained on ImageNet, an object detection database containing 1000 classes and 14 million images. Transfer learning is where a CNN is trained for one purpose but used for a different one. Although ImageNet does not contain images of this nature, spectrogram images in particular have been successful in other fields which will be detailed in the sections 2.3.1 to 2.3.3 below. The assumption with machine learning is that data needs to follow the same distribution as the model being trained. With transfer learning this is not necessary and existing data and models can be used for new environments [291].

2.3.1 Audio Classification

Tsalera et al. [292] describe a “promising paradigm” in their research surrounding transfer learning and spectrogram images. They compare three image based datasets of spectrograms and scalograms produced from audio signals on pre-trained CNNs using ImageNet. They compare the use of transfer learning to also training the CNNs from scratch. Transfer Learning out performs the CNNs trained from scratch significantly, producing an average of 27%, 57% and 25% higher classification accuracy results for GoogleNet, SqueezeNet, and ShuffleNet CNNs.

Their applicability for sound classification has been realised in various recent research whereby Palanisamy et al. [293] show that a CNN with pretrained weights on ImageNet can provide “a strong baseline for audio classification, even with a significant difference between spectrograms and ImageNet samples, assumptions gained from transfer learning hold firmly.” They classify various audio datasets using CNNs pre-trained on ImageNet with spectrogram images. Their work shows that using pretrained weights from ImageNet produces higher accuracy results than starting with weights which have been randomly initialised.

Koike et al. [294] use a heart sound classification dataset which was developed to consider new ways of detecting cardiovascular disease from the sound our hearts make. The development of sound classification for heart conditions has partly been due to the increase in wearable technology and also the increase in the use of machine learning techniques. Transfer learning is investigated using audio signals and also using pre-trained CNNs on ImageNet for use with spectrograms. It was found that the deep models pre-trained by audio signals outperformed the image based ones. Zhao et al. [295] use pre-trained CNN architectures on ImageNet to detect and classify abnormal heart beats using scalogram images of Phonocardiogram (PCG). They found that the features extracted from the VGG-16 CNN were more robust than an audio feature set which has been more widely used. They suggest future work to look at data augmentation and for datasets to include a greater number of heart sound classifications.

Zhou et al. [296] use CNNs pre-trained on ImageNet for acoustic scene classification. They highlight the fact that a lot of data is needed to be able to distinguish between different acoustic scenes as significant overlaps and similarities are contained within each one. Transfer learning was considered by Zhou et al. because of the success it has had in the imagery domain and they discuss whether transfer learning can be successful in a totally different classification task. They consider spectrograms of the audio signals as images, treating it as an image classification problem and produce classification accuracy scores of between 59.7% and 77.8%. Nagarajan and Oruganti in [297] classify emotions using transfer learning. They use spectrogram representation of the audio files which include the emotional responses and extract features using a pre-trained CNN AlexNet on ImageNet. A linear classifier was then trained using the extracted features from the CNN AlexNet. They were able to classify eight different emotions and built on the success of prior research in the field by 16% in terms of f1-score.

Muller et al. [298] determine that image based features are general purpose in their paper. They use pre-trained CNNs AlexNet, ResNet and Squeeze Net to extract features from mel spectrogram images and then train anomaly detection models using the features. Dufourq et al. [299] in their paper highlight the fact that designing architectures for CNNs requires extensive knowledge and that hyper-parameter optimisation and tuning can be time consuming and computationally intensive. Their work considers bioacoustics classification using CNNs which have had their weights pre-trained using ImageNet for passive acoustic monitoring. Twelve different CNNs are considered with four different datasets for passive acoustic monitoring. They find that the pre-trained CNNs utilising transfer learning can provide high f1-scores (82%) and robust models for passive acoustic monitoring.

2.3.2 Medical Diagnosis

CNNs which have been pre-trained on ImageNet have been used in the medical field for identifying conditions such as Diabetic Retinopathy (DR). Thota and Reddy [300] use a VGG-16 pre-trained on ImageNet with transfer learning and fine tune the CNN to classify how severe the conditions DR presented were. They implement data augmentation and were able to achieve an overall accuracy of 74%, improving on other research in the field. Jayakumari et al. [301] also consider transfer learning for the early detection and classification of DR. They find a higher accuracy can be achieved with the application of transfer learning through a pre-trained CNN on ImageNet. Their work achieves 98.6% accuracy. Weimann and Conrad [302] consider the fact that remote monitoring devices are producing a lot of data for health care patients to monitor heart activity. The increase in the amount of data also presents problems with needing more physicians to analyse the data. Salem et al. [303] highlight the fact that DNNs can produce higher accuracy results (when trained correctly) at detecting cardiac arrhythmias from ECG patterns than cardiologists.

They classify four types of ECG signals using a transfer learning based approach from the image classification domain. They show that the model can classify arrhythmias with an accuracy of 97.23% using cross validation.

Pal et al. [304] introduce a system called ‘Cardio Net’ which uses transfer learning with a neural network called Dense Net pre-trained with ImageNet to classify arrhythmia heartbeats. The neural network is fine-tuned and achieves accuracy levels of 98.92% for classification of 29 different types of heartbeats. Venton et al. [305] compare different types of CNN pre-trained on ImageNet whereby the neural network was fine-tuned on a database of ECG images. They conclude that out of the data given in their experiments. spectrograms and scalograms produced the best results. They also concluded that out of the different CNN models tried, there was no pattern with any one producing significantly higher accuracy results than the others. Their experiments considered AlexNet, GoogLeNet and SqueezeNet.

The diagnosis of COVID-19 has also benefited from the use of transfer learning with CNN feature extraction following by machine learning classification using logistic regression [306], random forest [307] and support vector machine [308]. Imran et al. [309] use transfer learning with cough samples to provide a COVID-19 preliminary diagnosis. Lahsaini et al. [310] consider confirmed COVID-19 patient’s chest CT images and patients without COVID-19. They evaluate various deep learning models including DenseNet121, DenseNet201, VGG16, VGG19, Inception Resnet-V2, and Xception and produce a model using DenseNet to perform COVID-19 detection in chest CT images. Garg et al. [311] consider whether CNNs utilising transfer learning can effectively detect COVID-19. They pretrain the following CNNs on ImageNet; VGG16, ResNetV2, InceptionResNetV2, DenseNet121, and MobileNetV2 CNN. They found that ResNetV2 and DenseNet121 were the best feature extractors to detect COVID-19 using X-Ray images. They also concluded that principle component analysis (PCA) can be used to increase efficiency. Overall for three classes they achieve an accuracy of over 94%. Alotaibi in [312] classifies between normal, COVID-19 and viral pneumonia using chest X-Ray images and pre-trained CNNs which are fine tuned. The CNNs tested were ResNet50, VGG19, DenseNet121, and InceptionV3. He found that the accuracy for all four models were similar and that 98.3% could be achieved for the classification of COVID-19 patients, normal patients and those with viral pneumonia.

Mormont et al. [313] consider transfer learning for various CNNs to perform classification for digital pathology and microscopy images. They produced the highest results with fine tuning the pre-trained CNN, however this came with the resource burden associated with the re-training of the network. Thota et al. [300] consider the use of transfer learning to improve the classification accuracy for the early detection and severity of DR. They improve upon other research in the field in terms of classification accuracy using the fine tuning of a 16 layer VGG-16 CNN. Kaur et al. [314] examine a pre-trained CNN using transfer learning to classify pathological brain images. Kaur et al. use cross-validation to ensure their results are not overfitting and find that the pretrained model with transfer learning out performed other current research. They also highlight as an advantage the fact that no hand crafted features need to be implemented as the CNN extracts its own features from the data.

Oktavian et al. [315] highlight the use of Magnetic Resonance Imaging (MRI) scans to detect Alzheimer’s disease and how deep learning is effective but very large datasets and high computational power are needed to produce high accuracy models. They use a ResNet-18 CNN architecture pretrained with ImageNet weights and therefore utilising transfer learning to detect for Alzheimer’s disease in MRI scans. Maqsood et al. [316] also consider MRI scan images for the early detection and classification of Alzheimer’s disease and dementia. The classification is further broken into four distinct phases for dementia that is shown in patients. AlexNet is used as the CNN pre-trained on ImageNet and fine-tuned for use with MRI scan images. They achieve an overall accuracy of over 92% for the classification of MRI scan images for dementia stage and Alzheimer’s disease classification. Khan et al. [317] use breast histopathology images to detect cancer and utilise transfer learning so that the benefits of deep learning can be utilised without the need for extensive datasets. Their results show that fine tuning a CNN model with prostate cancer images produced higher accuracy results that a model pre-trained on ImageNet (or from scratch on just breast cancer images). They show that cross domain transfer learning approaches are worth investigating for the detection of cancer in histopathology images.

2.3.3 Other

Chen et al. [318] use a pretrained CNN on ImageNet to extract features from short Fourier transform images for accurately detecting cracks in the pavement. Using pre-trained CNNs with ImageNet and the process of transfer learning with graphical representations of the RF signal has also been used for classifying small UASs. Saqib et al. [174] consider transfer learning for UAS detection in long-range surveillance videos and found residual CNNs to be strongest for classification. A second piece of work by Saqib et al. [166] considers the detection of UAS using imagery detection and transfer learning.

In their work entitled Deep Neural Networks to Enable Real-time Multi messenger Astrophysics’ Huerta et al. [319] conclude that transfer learning reduces the time for training classifiers and also improves accuracy in low SNR environments. George et al. [320] consider detecting anomalies such as glitches from gravitational wave data. They utilise transfer learning through pre-trained deep learning models to show that training time can be significantly reduced and that accuracy can remain very high at over 98.8%. Ackermann et al. [321] consider a transfer learning based approach for detecting galaxy mergers. They show that a pre-trained CNN and transfer learning can significantly improve performance for small datasets and compared to the current methods for detecting galaxy mergers. They showed their system robust to noise interference and also to distortions. The introduction of transfer learning improved accuracy overall and lowered error rates.

Dayang et al. [322] consider the identification of planetary nebulae from other object types. The architectures that were pre-trained on ImageNet were not optimised to the dataset and one issue was highlighted that astrology images are bigger than the maximum size expected by the pre-trained neural networks. However, even with these limitations they proved what they deemed to be “impressive results”. The neural network which provided the highest results was the DenseNet201. Wei et al. [323] consider the classification of compact start clusters using the Hubble Space Telescope ultraviolet optical imagery of spiral galaxies which are nearby. The amount of labelled star cluster samples is small so transfer learning is employed for this reason. Star clusters are classified into four classes and the ResNet18 and VGG-19-BN are used as the CNNs in the experiments. Their work shows that star cluster classification can be automated at scale and the next stage for this work would be to produce an agreed upon and standardised star cluster dataset to improve the study.

2.4 Contribution of Research

The following table shows were we have continued from previous research through identifying a gap from conducting the literature review and relating it to chapter headings and publications of our work.

Gap identified through Literature Review	**How experiments fill identified gaps**	**Own Work Publications**
UAS Flight Mode Classification. Al-Sa'd et al. [71] are the first authors to consider classifying the flight mode of the UAS but only achieve 46.8% accuracy using a DNN as a classifier. The work does also not consider interference in the same frequency band from other signals. There are also only 3 UAS types in the dataset and they are older platforms.	Our experiments in Chapter 4 - Supervised Learning carry on the work of Al-Sa'd et al. directly by using transfer learning and CNN feature extraction followed by machine learning classifiers to improve the results. From our own results we carry this work on further by introducing a larger dataset, deeper CNN architectures and looking at the presence of interference from signals operating in the same frequency band.	[22]–[27]
Transfer learning via CNN feature extraction has shown successful for audio signal	Chapter 3 - RF Profiling – considers the different ways that an RF signal can be	[22]–[27]

classification [292] [293] [294] and the diagnosis of medical conditions through scan images [300] [301] [306]. This process has not been applied to the fields of UAS detection & classification and GPS jamming classification via graphical representation of the RF signal as an image.	graphically represented and saved as an image for use in transfer learning CNN feature extraction. Chapter 4 - Supervised Learning looks at using transfer learning with CNN feature extraction as a pre-cursor for training supervised learning models.	
Transfer Learning CNN Feature Extraction as a Pre-curser to an Unsupervised algorithm. In literature CNN feature extractors have been shown to enhance clustering of complex natural images [324]. This has not been done with UAS detection & classification using transfer learning.	In Chapter 5 - Unsupervised Learning we look at using transfer learning with CNN feature extraction as a pre-cursor for training unsupervised learning models.	[19]
Low Cost Raspberry Pi/SDR Early Warning System. Price et al. [280] mount an SDR and raspberry Pi onto a UAS to detect and classify jamming signals	Chapter 6 - Early Warning including Unknown Signal Detection considered running the CNN feature extractor and machine learning classifier	[20]

targeted against the 2.4GHz control signal in real time. This work was in the context of jamming detection. No prior work was found to consider UAS detection and classification on a Raspberry Pi using machine learning.	model on a raspberry Pi with a low cost SDR in real time.	
Classifying UAS signals which the model had not been trained against. The UAS work considered in the literature review [199] [198] [199] [71] only look at signals the model was trained against.	Chapter 6 - Early Warning including Unknown Signal Detection considered evolutionary and non evolutionary datalink predictions using a model which was not trained against those UAS.	

Chapter 3 - RF Profiling

The RF Profiling Chapter considers the dataset generation, graphical representation of the signals as images and the introduction of wireless interference. The first section Chapter 3 describes the datasets used in the thesis which have either been produced as part of this work or are open source datasets, which allow the results of the work to be easily compared with other researchers in the field. The main dataset produced in the thesis is the DroneDetect dataset which has been made publicly available for the benefit of other academic researchers in the field. The second section 3.2 details how the signals are represented in graphical form as spectrograms, PSD, raw IQ constellation, histograms and a concatenation of all four. Lastly section 0 looks at the introduction of wireless interference in the band and how this effects the graphical signal representations.

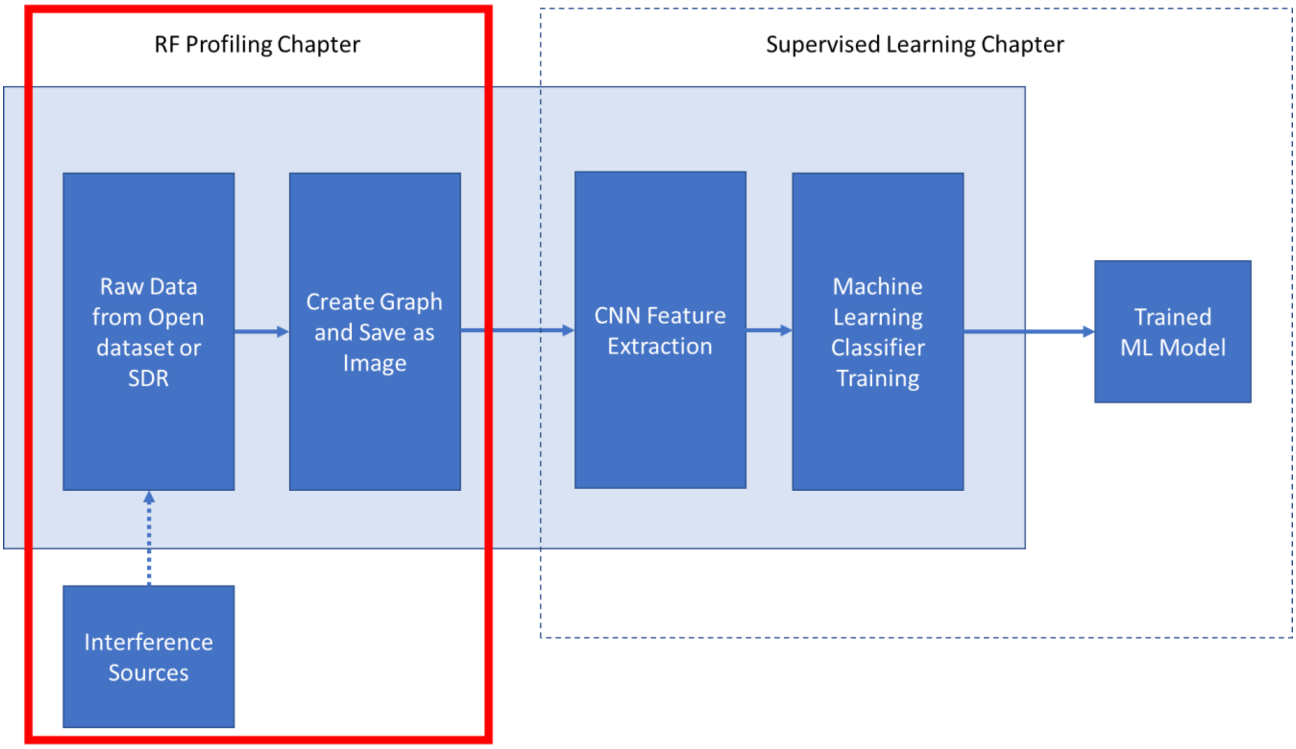

RF Profiling Chapter
Supervised Learning Chapter
Raw Data from Open dataset or SDR
Create Graph and Save as Image
CNN Feature Extraction
Machine Learning Classifier Training
Trained ML Model
Interference Sources

Figure 5 – How the RF Profiling Chapter fits into the Thesis

3.1 Dataset Generation

3.1.1 GPS Jamming Dataset [27] [31]

The first dataset used was an opensource dataset called Image datasets for jammer classification in GNSS [31]. The mathematical models used to generate the jamming signals are credited to Ferre *et al.* [110] [325], details can be found in the associated reference. Equation (1) represents the signal at the GNSS receiver, with g(t) representing the GNSS signal, j(t) the jamming signal and w(t) is background noise, represented in the signal generation by AWGN.

$$r(t) = g(t) + j(t) + w(t) \tag{1}$$

To incorporate varied scenarios, parameters are randomly generated but modelled with uniform distributions. The signal $r(t)$ was generated with a uniform distribution of AGWN whereby Carrier-to-Noise Ratio ($C/N0$) between 25-50dBHz and Jammer-to-Signal Ratio (JSR) 40-80dB. Signals were generated using open source Matlab files [325] which were only altered to save them as raw data. For the interference experiments $w(t)$ is omitted and a set level of noise n is calculated for set levels of SNR.

$$q(t) = g(t) + j(t) + n \tag{2}$$

To allow the evaluation of the signal at set levels of SNR we take our signal $g(t) + j(t)$ and add our noise n to give us signal $q(t)$ at a set level of SNR. This can be seen in (2). First the power in our signal $g(t) + j(t)$ is calculated, a noise spectral density using the desired level of SNR is worked out and from this the noise power n to be added to the original signal can be calculated. Once we have our signals in set levels of SNR we produce graphical signal representations.

3.1.2 JamDetect Dataset [23]

The JamDetect dataset was produced to extend the GPS Jamming dataset to include a protocol aware jammer and a barrage jammer. It also signals allowed those signals to be tested with specific levels of SNR. First, $s(t)$ is represented as the signal at the GPS receiver shown in Equation (3).

$$s(t) = g(t) + j(t) + w(t) \tag{3}$$

Whereby $g(t)$ is the GPS signal coming from the satellite, $j(t)$ is our jamming signal and $w(t)$ is the noise generated by AWGN across the wireless channel.

GPS Signal

First of all $g(t)$ is considered, the GPS signal coming from the satellite constellation. GPS is specifically be considered as the GNSS system. GPS is a US system consisting of 24 satellites across 6 orbits which operate over 12 hr periods. GPS has two carrier frequencies L1 and L2 which broadcast Binary Phase-Shift Keying (BPSK). The L1 band is the focus which operates at 1575.42MHz across a bandwidth of 24MHz. However, it has been proven due to the design of the L1 signal that a bandwidth of 9.66MHz is enough for tracking and navigation [326]. A bandwidth of 10MHz is used within our experiments.

To try and create a realistic signal a real GPS signal is generated using GPS-SDR-SIM [327]. GPS-SDR-SIM is python based code which generates GPS baseband data for the intention of being broadcast by an SDR. It allows you to define a static location, for our experiments the New York New York Piano Bar in Las Vegas, GPS co-ordinates 36° 10' 11.7876" N 115° 8' 23.3952" W which corresponds to Latitude Longitude (36.169941, -115.139832) are used. The GPS Satellite constellation is then specified using a daily GPS broadcast ephemeris file from the NASA Earth Data site [328]. For a date the 20 Dec 2014 is used for testing but ephemeris data is available up to 24hrs into the future, so applications can predict orbits. GPS-SDR-SIM uses the data to generate the pseudo range and Doppler for simulated GPS satellites in view at that time which is then converted to I/Q baseband samples.

Jamming Signals

Next the jamming signals $j(t)$ are considered and the details of the production of the JamDetect dataset. GNURadio was used to produce the jamming signals. GNURadio is an open source framework which includes a free toolkit for the development of SDRs in a PC environment [136]. The signals defined in [106] are used with GNURadio as a starting point and narrowband jamming is added to the jamming classes already defined.

Chirp Jammer

Chirp signals were generated by increasing frequency over time, also known as sweep jamming. Signals are constructed across a bandwidth of 10MHz with a fast sweep rate of 10KHz, the stop and start frequencies are calculated using Equation (4).

$$\begin{aligned} f_{min} &= f_c - \frac{BW}{2} = 1575.42\, x\, 10^6 - \frac{10\, x\, 10^6}{2} = 1570.42MHz \\ f_{max} &= f_c + \frac{BW}{2} = 1575.42\, x\, 10^6 + \frac{10\, x\, 10^6}{2} = 1580.42MHz \end{aligned} \tag{4}$$

The implementation used to produce the chirp signal in GNURadio is sourced from [329].

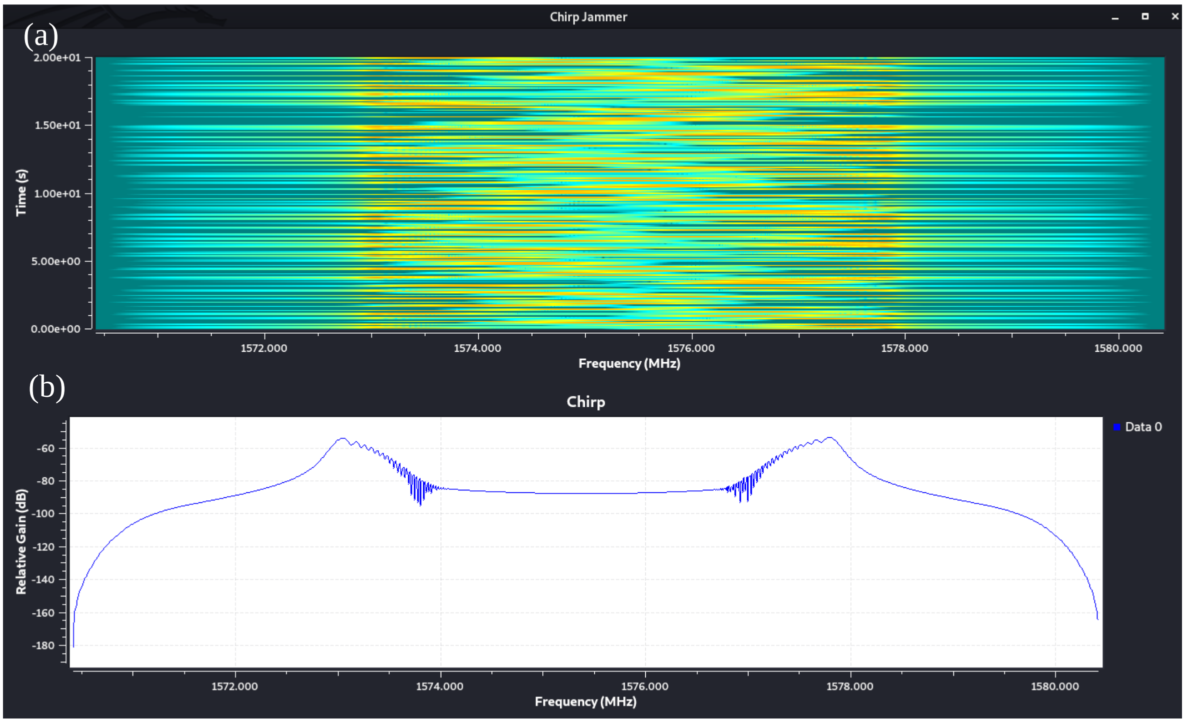
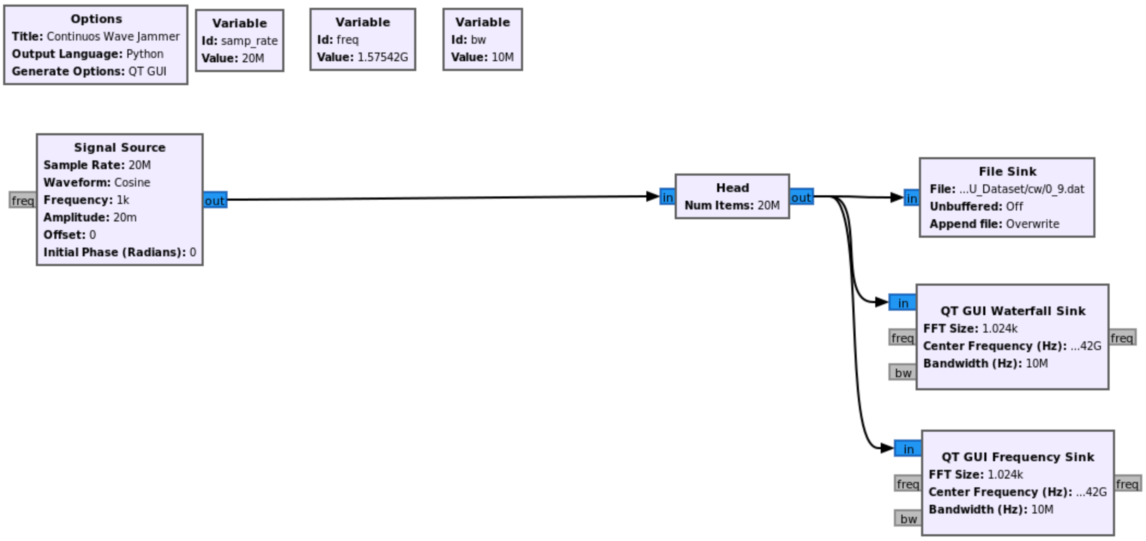


Figure 6 – Visual Representation of a Chirp Jamming Signal as a (a) Spectrogram and (b) PSD

Figure 6 shows the Spectrogram and PSD of the signal. The PSD shows the signal at its highest point is approximately -55dB/Hz.

Continuous Wave (CW) Jammer

The CW jamming signal is produced in GNURadio using a 1KHz cosine signal source. Figure 7 shows the implementation of the jamming signal generation in GNURadio.



Options
Title: Continuos Wave Jammer
Output Language: Python
Generate Options: QT GUI

Variable
Id: samp_rate
Value: 20M

Variable
Id: freq
Value: 1.57542G

Variable
Id: bw
Value: 10M

Signal Source
Sample Rate: 20M
Waveform: Cosine
Frequency: 1k
Amplitude: 20m
Offset: 0
Initial Phase (Radians): 0

Head
Num Items: 20M

File Sink
File: ...U_Dataset/cw/0_9.dat
Unbuffered: Off
Append file: Overwrite

QT GUI Waterfall Sink
FFT Size: 1.024k
Center Frequency (Hz): ...42G
Bandwidth (Hz): 10M

QT GUI Frequency Sink
FFT Size: 1.024k
Center Frequency (Hz): ...42G
Bandwidth (Hz): 10M

Figure 7 – The Generation of a CW Jamming Signal using GNURadio

In Figure 8 (a) it is observed that the CW jamming signal only occupies a small part of the GPS bandwidth.

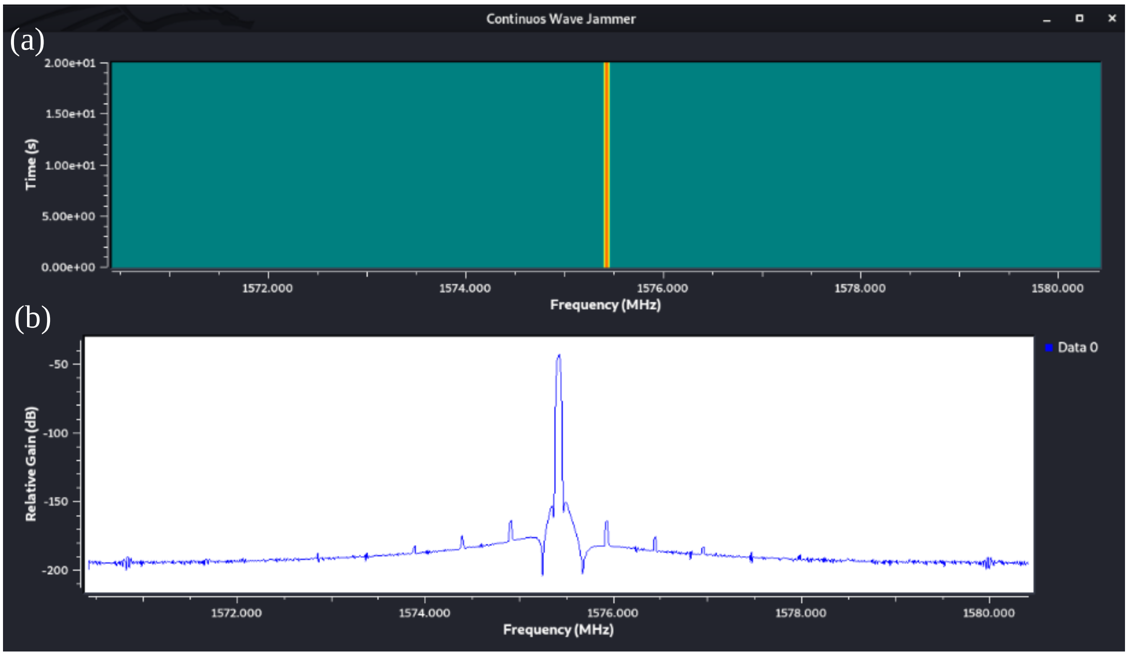
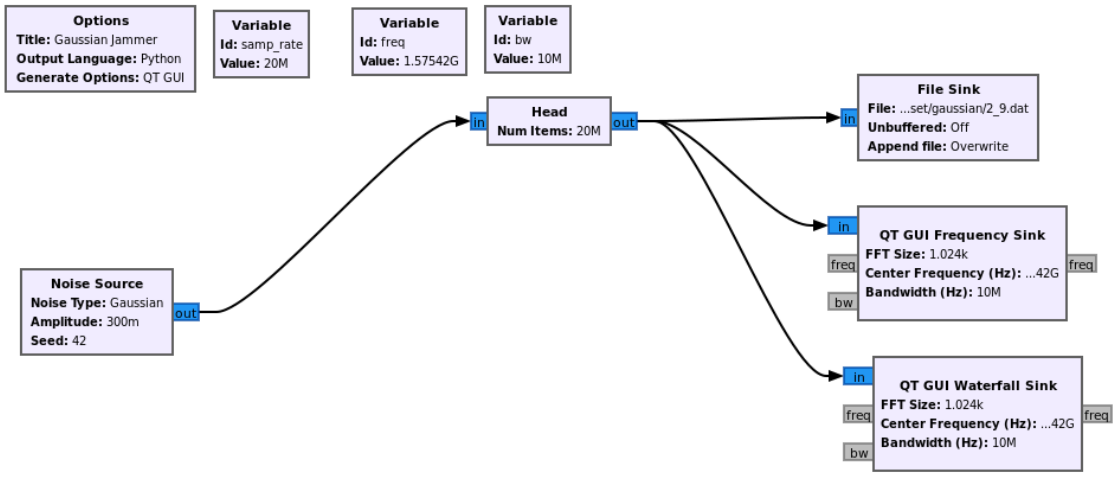


Figure 8 - Visual Representation of a CW Jamming Signal as a (a) Spectrogram and (b) PSD

This can be seen clearly in Figure 8 (b) which also shows the highest peak of the signal at approximately -45db/Hz on the PSD.

Barrage Jammer

The barrage jamming signal is produced using a gaussian noise source in GNURadio. Figure 9 shows the implementation in GNURadio.



Options
Title: Gaussian Jammer
Output Language: Python
Generate Options: QT GUI

Variable
Id: samp_rate
Value: 20M

Variable
Id: freq
Value: 1.57542G

Variable
Id: bw
Value: 10M

Head
Num Items: 20M

File Sink
File: ...set/gaussian/2_9.dat
Unbuffered: Off
Append file: Overwrite

Noise Source
Noise Type: Gaussian
Amplitude: 300m
Seed: 42

QT GUI Frequency Sink
FFT Size: 1.024k
Center Frequency (Hz): ...42G
Bandwidth (Hz): 10M

QT GUI Waterfall Sink
FFT Size: 1.024k
Center Frequency (Hz): ...42G
Bandwidth (Hz): 10M

Figure 9 - The Generation of a Barrage Jamming Signal using GNURadio

As opposed to the CW jamming signal, the barrage jammer occupies all the bandwidth. It is not a discrete choice of jammer but potentially more effective against GPS due to the spread spectrum nature of a GPS signal.

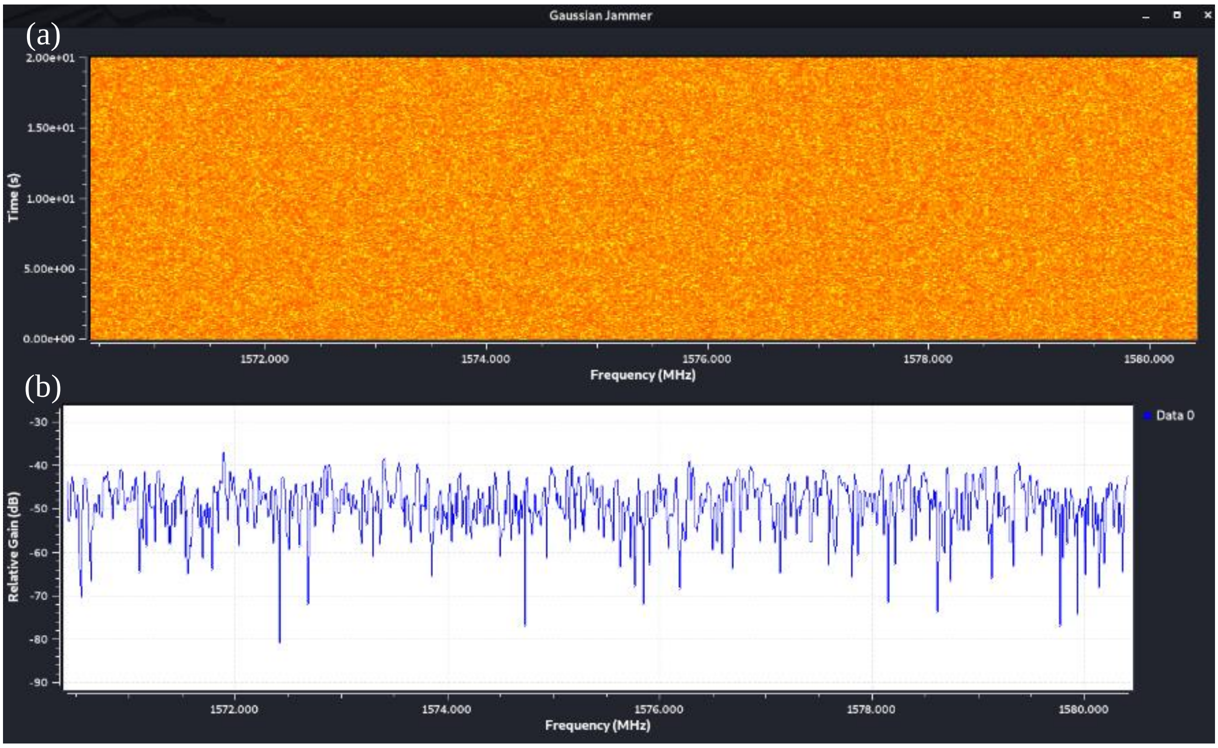


Figure 10 - Visual Representation of a Barrage Jamming Signal as a (a) Spectrogram and (b) PSD

The difference can be seen clearly when comparing Figure 8 for the CW jammer to Figure 10 for the barrage jamming.

Narrowband Jammer

The narrowband (NB) jammer is constructed in GNURadio by generating a QPSK signal which covers a bandwidth of 1.6MHz. Generation of the QPSK constellation was produced using the information in [330].

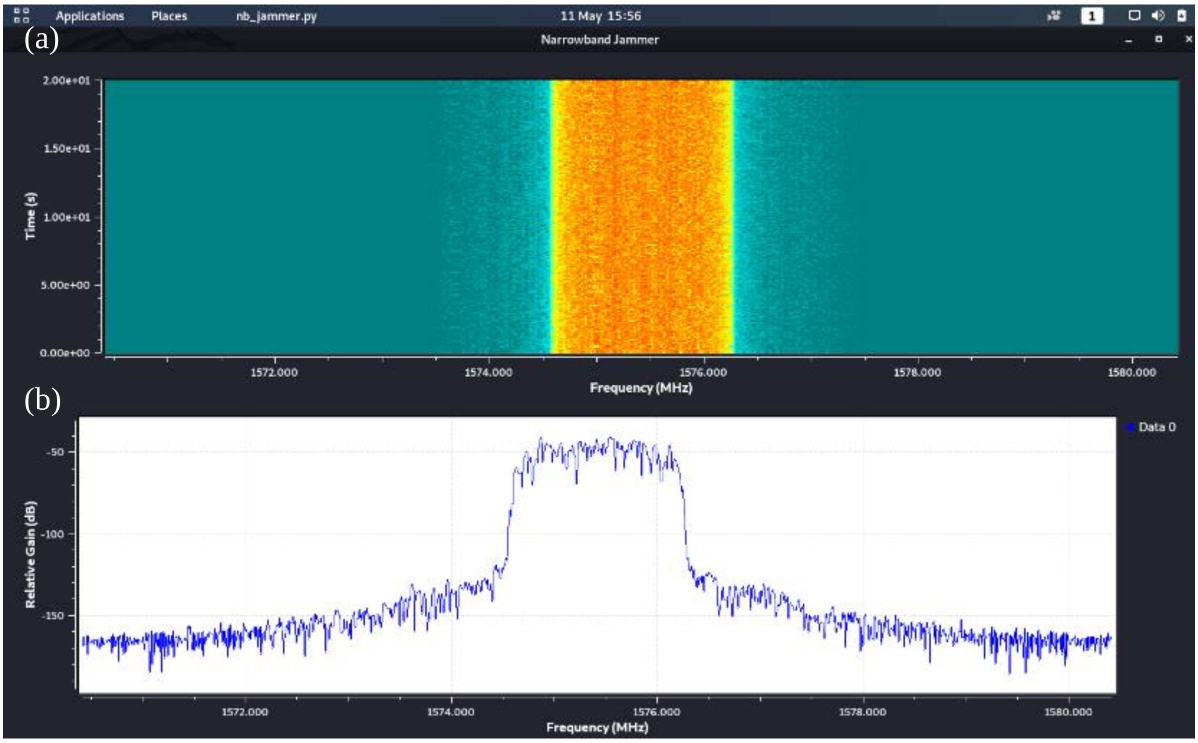
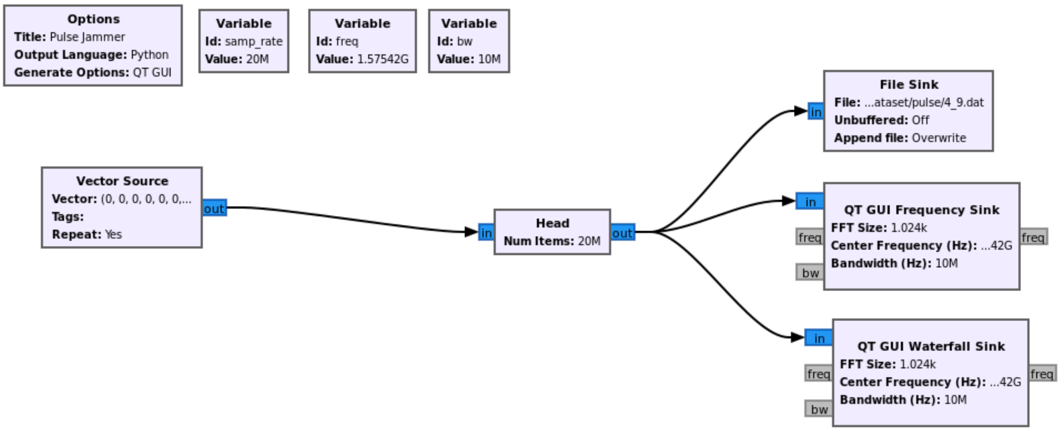


Figure 11 - Visual Representation of a NB Jamming Signal as a (a) Spectrogram and (b) PSD

Figure 11 shows the signal represented in both Spectrogram and PSD form. It is observed that the signal peaking at approximately -45dB/Hz.

Pulse Jammer

The pulse jamming signal was created using a vector source with a low duty cycle of 2% in GNURadio [106]. The implementation of the jamming signal generation is seen in Figure 12 below.



Options
Title: Pulse Jammer
Output Language: Python
Generate Options: QT GUI

Variable
Id: samp_rate
Value: 20M

Variable
Id: freq
Value: 1.57542G

Variable
Id: bw
Value: 10M

Vector Source
Vector: (0, 0, 0, 0, 0, 0,...
Tags:
Repeat: Yes

Head
Num Items: 20M

File Sink
File: ...ataset/pulse/4_9.dat
Unbuffered: Off
Append file: Overwrite

QT GUI Frequency Sink
FFT Size: 1.024k
Center Frequency (Hz): ...42G
Bandwidth (Hz): 10M

QT GUI Waterfall Sink
FFT Size: 1.024k
Center Frequency (Hz): ...42G
Bandwidth (Hz): 10M

Figure 12 - The Generation of a Pulse Jamming Signal using GNURadio

The sequences of pulses occupy the full bandwidth. Figure 13 shows the Spectrogram and PSD for the pulse jamming signal.

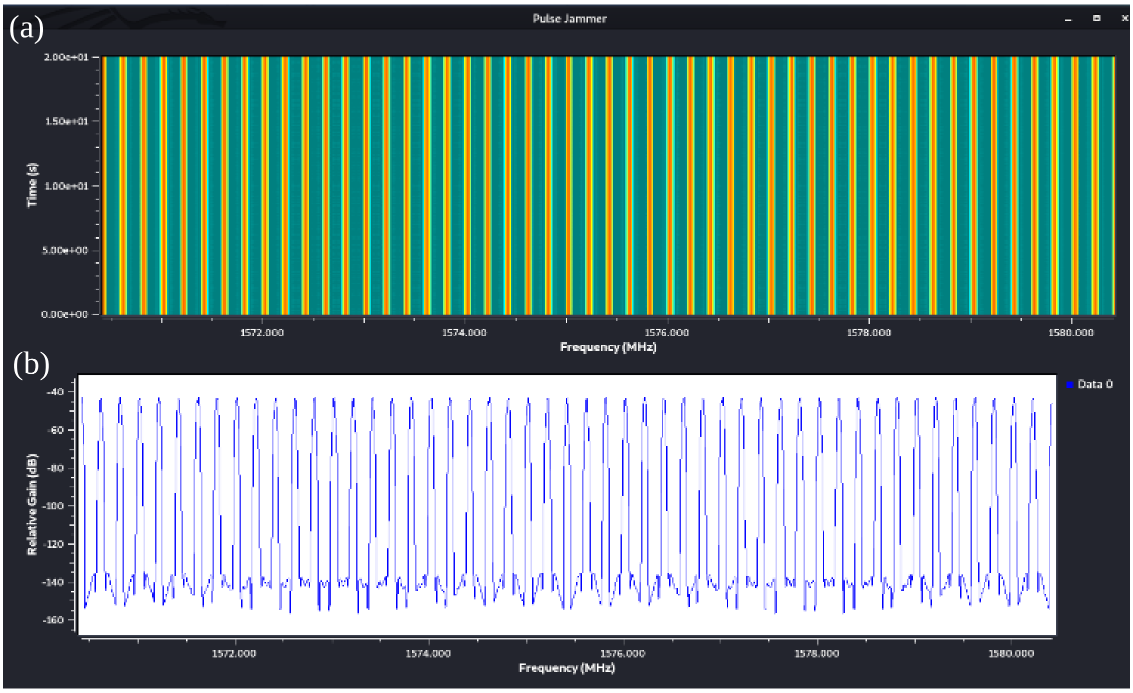


Figure 13 - Visual Representation of a Pulse Jamming Signal as a (a) Spectrogram and (b) PSD

Protocol Aware Jammer

The protocol aware jammer is constructed in GNURadio by generating a BPSK signal [331] to represent the structure of a GPS signal. Figure 14 shows the spectrogram and PSD representations of the protocol aware jamming signal.

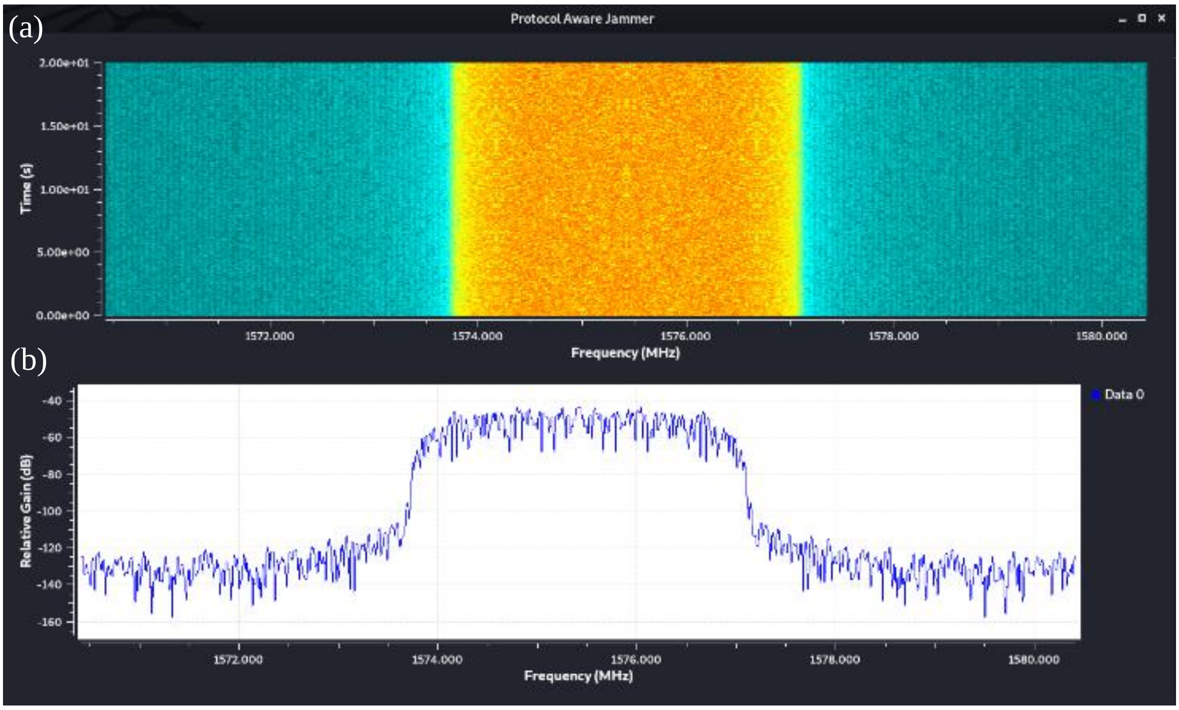


Figure 14 - Visual Representation of a Protocol Aware Jamming Signal as a (a) Spectrogram and (b) PSD

The GPS signal and the jamming signals were combined to achieve a JSR of 40dB.

Additive Gaussian White Noise (AWGN)

Lastly $w(t)$ modelled by AWGN to corrupt the signal and test classification accuracy for various levels of SNR is considered. AWGN was generated in Python 3 and added to the signal as shown in Figure 15.

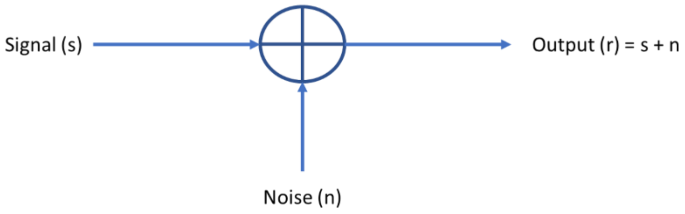


Figure 15 - AWGN Channel Process of the addition of noise to a signal

SNR is defined in (5) where P_r is the power in the signal, B the bandwidth and PSD of the noise $\frac{N_0}{2}$ [332].

$$SNR = \frac{P_r}{N_0 B} \tag{5}$$

To add the noise the power in our signal is first calculated. This is seen in (6) where N is the length of signal s.

$$P = \frac{1}{N} \sum_{i=0}^{N-1} |s_i|^2 \tag{6}$$

P is then used with our desired SNR to calculate the noise spectral density N_0 as seen in Equation (7).

$$N_0 = \frac{P}{SNR} \tag{7}$$

Lastly the noise power is calculated which is required to generate Gaussian random noise in Equation (8).

$$\sigma^2 = \frac{N_0}{2} \tag{8}$$

The calculations for generating AWGN and its implementation in python are referenced [333].

3.1.3 DroneRF Dataset

The open DroneRF dataset was produced by Al-S'ad *et al.* [204]. Table 4 shows the UAS classes assessed in the experiments from the DroneRF dataset used within this thesis. Three UAS types are included in the dataset – Parrot Bebop, Parrot AR (Elite 2.0) and the DJI

Phantom 3. Al-Sa’d *et al*. in [240] pick these UASs because they are commonly purchased for civilian applications.

Table 4 - UAS Classes

Class	**UAS Type**	**Mode**
1	No UAS	N/A
2	Parrot Bebop	Switched on and connected to controller
3	Parrot Bebop	Hovering automatically with no controller commands
4	Parrot Bebop	Flying without video
5	Parrot Bebop	Flying with video
6	Parrot AR	Switched on and connected to controller
7	Parrot AR	Hovering automatically with no controller commands
8	Parrot AR	Flying without video
9	Parrot AR	Flying with video
10	DJI Phantom 3	Switched on and connected to controller

The Bebop, AR and Phantom 3 are varied when it comes to price, size and overall capability. They increase with weight and size respectively and in terms of range the Phantom 3 can operate out to 1000m, while the Bebop and AR are restricted to 250 and 50m respectively. The Phantom 3 and the Bebop utilise the 5GHz and 2.4GHz Wi-Fi bands but during these experiments are limited to observing their activity in the 2.4Ghz band. The Bebop can be set to automatically select a Wi-Fi channel based on the countries legal requirements and channel congestion or you can manually set the channel yourself [334]. Capturing the whole Wi-Fi spectrum ensures that the UAS will be captured even if the device switches channel during operation due to interference.

In Table 4 we can see that for the Bebop and the AR various modes are captured, including switched on and connected to controller; hovering automatically with no input from the controller; flying with video transmission; and flying without video. The DJI Phantom 3 is recorded only in the first mode - switched on and connected to controller. Al-Sa'd *et al.* [240] is the first work to the author's knowledge which considers different modes of operation when classifying UASs. This RF forensic analysis would be extremely useful in helping to determine intent. Organisations such as the police could use this information to make an assessment on risk. For example flying with the video on could indicate an intelligence collection operation due to the real time feedback of imagery. Intelligence capabilities on UASs have been directly linked to targeted killing [335].

The DroneRF dataset was recorded using two USRP-2943 SDR. The USRP-2943 is a higher end SDR costing around £6350 each [336]. They operate between 1.2GHz and 6GHz frequency range with the ability to capture 40MHz of instantaneous bandwidth. Al-Sa'd *et al.* utilise two USRP-2943 simultaneously in order to cover 80MHz of the Wi-Fi spectrum (excluding channel 1 and 14). Another SDR would be required to cover the entirety of the band. The dataset is recorded in segments to include the class no UAS present which is essentially background noise with no UAS and then segments recorded for different UAS in modes switched on and connected to the controller, hovering, flying with and without the video feed present. The captures are 10.25 seconds in length for the no UAS class and 5.25

seconds for the other classes which produces a dataset over 40GB in size. This is broken down into 227 segments where each part contains 1 million samples that equate to the time domain amplitude of the raw RF signal. More information on the dataset can be found in reference [337]. For the experiments in this thesis which use the DroneRF dataset 1000 samples were taken from each class and split 80 % for training with cross validation and 20% kept entirely separate as a hold-out evaluation dataset. It should be noted that while the DroneRF dataset was a great starting point for understand if the methodology of transfer learning and CNN feature extraction was worth greater exploration, the dataset does have limitations. There are only 3 UAS evaluated, they are captured in a lab environment and the UAS types themselves were outdated at the time of the experimental work. Producing our own dataset allowed the expansion of the work by capturing up to date UAS, more UAS types and including real life interference in the frequency bands.

3.1.4 **DroneDetect Dataset** [25]

The DroneDetect dataset [338] was developed and made open source. This was our contribution for the benefit of the academic community.

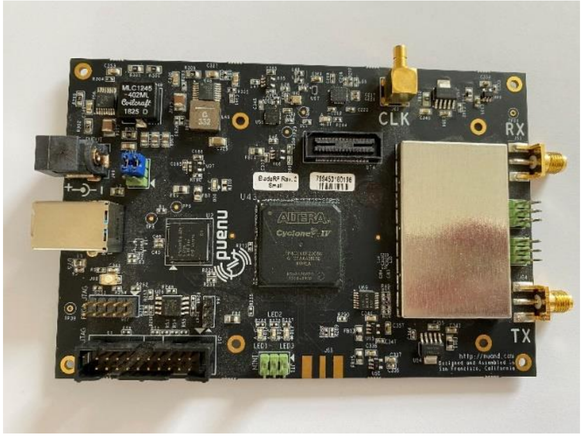


Figure 16 – Picture of a Nuand BladeRF SDR

The dataset was recorded using a SDR BladeRF shown in Fig.2 and a Palm Tree Vivaldi Antenna shown in Figure 16. The BladeRF is made by Nuand and is a popular choice as a low cost UAS due to its wide frequency range of 47MHz to 6GHz, its ability to transmit and receive at the same time and its low cost [339].

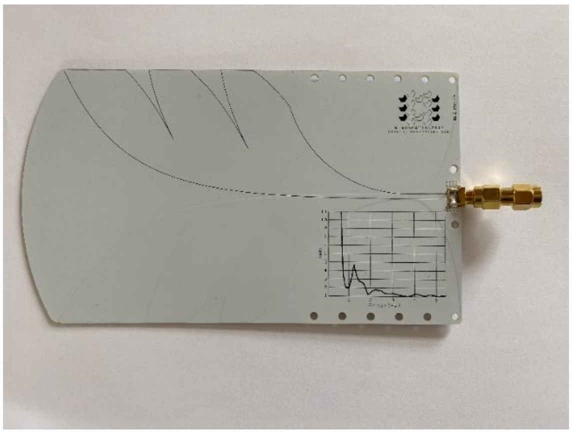


Figure 17 – Picture of a Palm Tree Vivaldi Wideband Antenna

In Figure 17 we see the low cost Palm Tree Vivaldi Antenna [340] [341]. The Vivaldi has a frequency range of 800MHz to 6GHz and is small, easily portable and costs only $18.99 [342]. Considering future possibilities for UAS detection systems, the Vivaldi would be a perfect addition to a HackRF connected to an Android phone for real time portable UAS detection and classification.

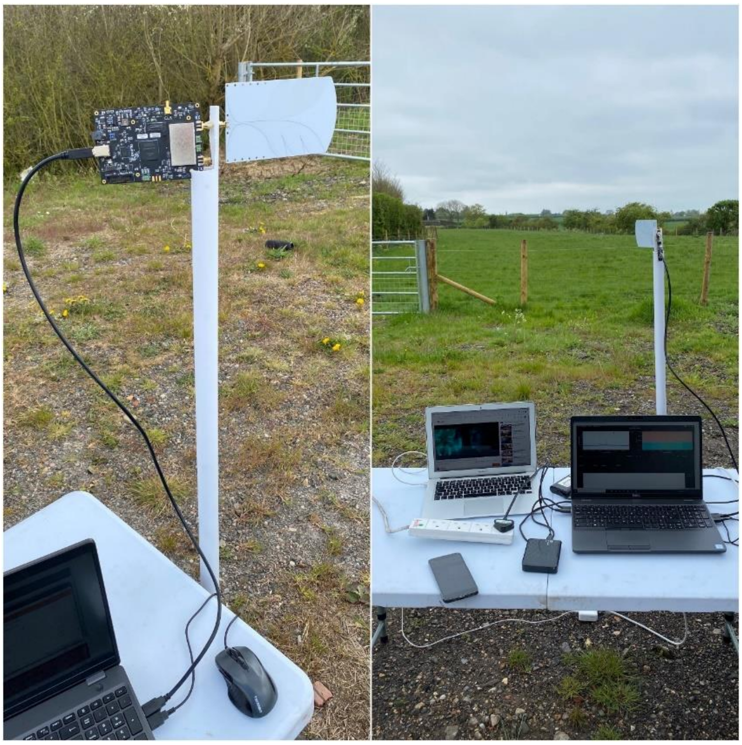
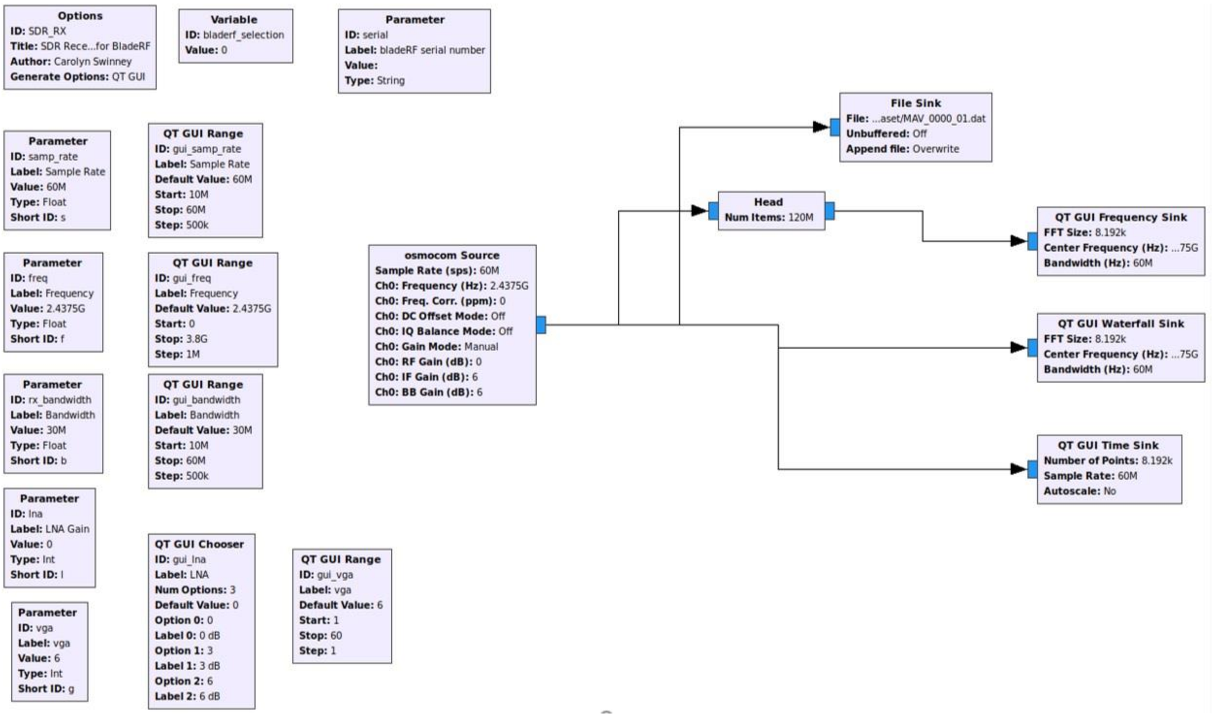


Figure 18 – Picture of the Experimental Setup for dataset collection and testing

As observed in Figure 18, the BladeRF and Vivaldi antenna were mounted on a piece of PVC piping approximately one meter high and connected to a laptop running GNURadio. A screenshot showing the GNURadio setup can be seen in Figure 19 below.



```
Options
ID: SDR_RX
Title: SDR Rece...for BladeRF
Author: Carolyn Swinney
Generate Options: QT GUI

Variable
ID: bladerf_selection
Value: 0

Parameter
ID: serial
Label: bladeRF serial number
Value:
Type: String

Parameter
ID: samp_rate
Label: Sample Rate
Value: 60M
Type: Float
Short ID: s

QT GUI Range
ID: gui_samp_rate
Label: Sample Rate
Default Value: 60M
Start: 10M
Stop: 60M
Step: 500k

Parameter
ID: freq
Label: Frequency
Value: 2.4375G
Type: Float
Short ID: f

QT GUI Range
ID: gui_freq
Label: Frequency
Default Value: 2.4375G
Start: 0
Stop: 3.8G
Step: 1M

Parameter
ID: rx_bandwidth
Label: Bandwidth
Value: 30M
Type: Float
Short ID: b

QT GUI Range
ID: gui_bandwidth
Label: Bandwidth
Default Value: 30M
Start: 10M
Stop: 60M
Step: 500k

Parameter
ID: lna
Label: LNA Gain
Value: 0
Type: Int
Short ID: l

QT GUI Chooser
ID: gui_lna
Label: LNA
Num Options: 3
Default Value: 0
Option 0: 0
Label 0: 0 dB
Option 1: 3
Label 1: 3 dB
Option 2: 6
Label 2: 6 dB

Parameter
ID: vga
Label: vga
Value: 6
Type: Int
Short ID: g

QT GUI Range
ID: gui_vga
Label: vga
Default Value: 6
Start: 1
Stop: 60
Step: 1

osmocom Source
Sample Rate (sps): 60M
Ch0: Frequency (Hz): 2.4375G
Ch0: Freq. Corr. (ppm): 0
Ch0: DC Offset Mode: Off
Ch0: IQ Balance Mode: Off
Ch0: Gain Mode: Manual
Ch0: RF Gain (dB): 0
Ch0: IF Gain (dB): 6
Ch0: BB Gain (dB): 6

File Sink
File: ...aset/MAV_0000_01.dat
Unbuffered: Off
Append file: Overwrite

Head
Num Items: 120M

QT GUI Frequency Sink
FFT Size: 8.192k
Center Frequency (Hz): ...75G
Bandwidth (Hz): 60M

QT GUI Waterfall Sink
FFT Size: 8.192k
Center Frequency (Hz): ...75G
Bandwidth (Hz): 60M

QT GUI Time Sink
Number of Points: 8.192k
Sample Rate: 60M
Autoscale: No
```

Figure 19 - GNURadio Configuration for the collection of the DroneDetect Dataset

An Osmocom source block was used to receive the data from the BladeRF in raw I/Q complex form. We can observe in Figure 19 that signals were recorded at a sample rate of 60Mbits/s over 28MHz bandwidth and with a centre frequency of 2.4375GHz. A head block was used to limit each recording to 1.2 x 10^8 complex samples, the equivalent of two seconds of recording time. Five recording were taken for each UAS flight mode and with no UAS present. From an ethical perspective recordings were completed in a rural setting away from any other frequency band users and this was confirmed by observing the activity in the spectrum prior to the recordings. Complex samples were recorded to file using a file sink block with the extension '.dat'. It is important to note that GNURadio saves complex data as interleaved floats and an example file is included with the dataset to allow the loading of the files into Python. For our machine learning experiments in our other papers we further split the files so that each sample is only 20ms in length and this provides a total of 500 samples per class (100 per individual file). This process is included with the dataset for ease of use for other researchers. We also considered sample lengths of 40ms and 80ms.

Flying was conducted within a radius of 40m of the antenna and at an altitude of 20m in height. For the ‘hovering’ samples the UAS was overhead at an altitude of 20m. Lastly for the ‘switched on’ samples the UAS was 4m from the antenna on the ground. For all of the samples the controller was approximately 4m from the antenna. This was done so that training samples were of good quality and it should be noted that reception at considerably greater distances is trivial.

Interference Sources

The end devices for Bluetooth and Wi-Fi interference were placed approximately 2m from the antenna and the connecting device directly next to the antenna. Table 5 shows the interference devices.

Table 5 - Interference Type & End Device & Connecting Device & Source

Interference Type	**End Device**	**Connecting Device**	**Source**
Bluetooth	JBL Charge Speaker	Android Phone	Music
Wi-Fi	MacBook Air	iPhone Personal Hotspot	YouTube Video

Data Structure

UASs were recorded with the following structure seen in Figure 20 and Table 6 shows the identification codes for the UAS and flight modes.

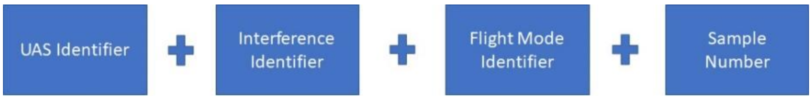


Figure 20 - File Configuration for the Drone Detect Dataset

Interference identifiers were 00 for a clean signal, 01 for Bluetooth only, 10 for Wi-Fi only and 11 for both Bluetooth and Wi-Fi interference presence. Table 3 expands on each description.

Table 6 - UAS Recording Description

UAS	**UAS ID**	**Flight Mode**	**Flight Mode ID**
No UAS	NO	No UAS	00
Air 2 S	AIR	Switched On	00
Air 2 S	AIR	Hovering	01
Air 2 S	AIR	Flying	00
Parrot Disco	DIS	Switched On	00
Parrot Disco	DIS	Flying	00
Inspire 2	INS	Switched On	00

Inspire 2	INS	Hovering	01
Inspire 2	INS	Flying	00
Mavic Pro	MP1	Switched On	00
Mavic Pro	MP1	Hovering	01
Mavic Pro	MP1	Flying	00
Mavic Pro 2	MP2	Switched On	00
Mavic Pro 2	MP2	Hovering	01
Mavic Pro 2	MP2	Flying	00
Mavic Mini	MIN	Switched On	00
Mavic Mini	MIN	Hovering	01
Mavic Mini	MIN	Flying	00
Phantom 4	PHA	Switched On	00
Phantom 4	PHA	Hovering	01
Phantom 4	PHA	Flying	00

An example code for the first sample of the Inspire hovering in the presence of Bluetooth interference would be:

$$INS + 01 + 01 + 01 = \text{INS\_0101\_01.dat}$$

The SDR is collecting everything that is happening in the frequency range of collection, whatever datalinks are present it will pick up. Table 7 lists the UASs we are looking at against their respective transmission systems.

Table 7 - UAS Transmission Systems

UAS Type	Transmission System
Air 2 S	OcuSync 3.0
Parrot Disco	Wi-Fi
Inspire 2	Lightbridge 2.0
Mavic Pro	OcuSync 1.0
Mavic Pro 2	OcuSync 2.0
Mavic Mini	Wi-Fi
Phantom 4	Lightbridge 2.0

The DJI range of UASs use three types of transmission system: Wi-Fi, Lightbridge and OcuSync. The Mavic Mini uses Wi-Fi and operates in the 5.8 GHz range (5.725-5.850 GHz}) and in the 2.4GHz range (2.4-2.4835GHz), with an effective isotropic radiated power (EIRP) or transmission power of 19dBm at 2.4~GHz [343]. The Lightbridge 2 has a maximum (interference free) transmission distance of 5km, the EIRP of the antenna is 100mW at 2.4GHz, and it can operate in the 5.8GHz range (5.725-5.825GHz) and the 2.4GHz range (2.4-2.483GHz) [344]. The OcuSync scans the band for any interference and decides which transmission channel is best. It then automatically switches between channels during the flight. OcuSync 1.0 and 2.0 have a range of 7km; they differ due to Ocusync 2.0 being able to utilise 2.4 and 5.8GHz frequency diversity. OcuSync 1.0 has an EIRP of 26dBm [345] and

OcuSync 2.0 an EIRP of 25.5 dBm. The OcuSync 3.0 range is increased again to 12km [346] and EIRP 26dBm [347]. The Parrot Disco uses the SkyController 2, which is a Wi-Fi-based protocol with a range of 2 km using MIMO antennas in the 2.4 and 5.0Ghz frequency bands [348].

OcuSync and Lightbridge are similar in the sense that they use Orthogonal frequency-division multiplexing (OFDM) to transmit the video (downlink). Unless manually changed, this channel is chosen when the platform switches on and it will stay on that channel for the flight unless interference is detected. The control link (uplink) on both Ocusync and Lightbridge uses Frequency-hopping spread spectrum (FHSS) which works by changing the carrier frequency very quickly over a large portion of the spectrum. The difference in the Ocusync and the Lightbridge really lies in how they were developed, the Lightbridge is more hardware based using an FPGA. The Ocusync moved towards a software defined radio approach, making changes to the datalink much easier as it can be implemented on the existing hardware via a software upgrade. For the Lightbridge to be updated it would need a hardware change [349].

3.2 Graphical Signal Representations [25]

3.2.1 Spectrogram

While the PSD looks at the distribution of signal strength in the frequency domain, a spectrogram looks at how the frequencies are changing with time. Spectrograms are used extensively in fields such as speech processing due to their ability to visualise bursts of activities at different frequencies over time.

The spectrogram shows the intensity of the STFT magnitude over time. It is a sequence of FFTs of windowed data segments which lets us visualise how the frequency content of the signal is changing over time. In Equation (9) we define the STFT [350].

$$\begin{aligned} X(w,m) &= STFT(x(n)) \\ &\coloneqq DTFT\ (x(n-m)\omega(n)) \\ &\coloneqq \sum_{n=-\infty}^{\infty} x(n-m)\omega(n)e^{-(i\omega n)} \\ &\coloneqq \sum_{n=0}^{R-1} x(n-m)\omega(n)e^{-(i\omega n)} \end{aligned} \tag{9}$$

Equation (9) describes a visual representation of the STFT magnitude $|X(\omega,m)|$, the spectrogram. In (4) $x(n)$ represents the signal, $\omega(n)$ the windowing function with a length of R and the windowing function determines the block length. As with the PSD, Matplotlib is used to plot the spectrogram with a Hanning windowing function and FFT length 1024. There is a correlation between the chosen FFT size and the frequency resolution of each spectral line, hence larger FFT sizes require a longer time to compute. During the beginning stages of the PhD FFT size was experimented with for varying signal types such as Wi-Fi and Bluetooth. A FFT size of 1024 was found to produce a frequency resolution which was sufficient for classification with a low computational time. This informed the choice for these experiments but FFT size is something which could be experimented with in future work to understand the subsequent effect on classification results.

UAS Signals

Figure 21 shows sample spectrograms produced using Matplotlib in python 3. On the left the class no UAS can be seen. This represents the baseline, the background activities captured in the spectrum when the collection took place. This should be minimal as the dataset was collected in an extremely rural and isolated environment with no nearby residential areas or activity in the spectrum.

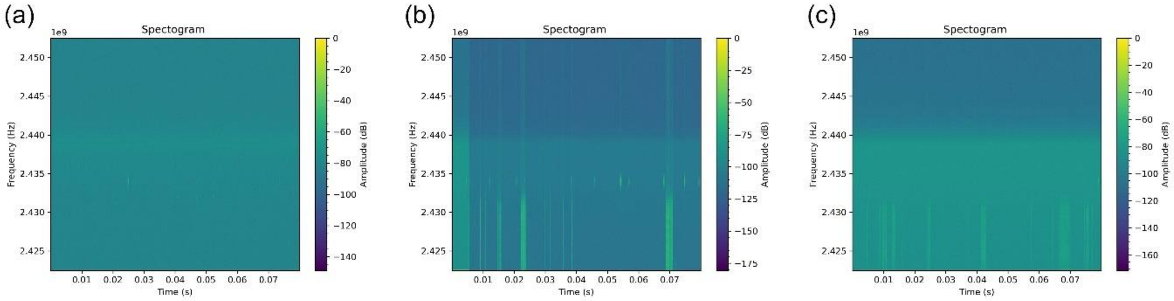


Figure 21 – Spectrogram Graphical Signal Representation for the classes (a) No UAS, (b) Parrot Disco Switched on and connected to the controller and (c) Parrot Disco flying

The middle of Figure 21 shows the Parrot Disco switched on (b) and on the right hand side shows the spectrogram of the Parrot Disco flying (c). On both graphs we can see a signal in the bottom half of the frequency range which in the time domain shows itself as yellow bursts of activity. The Disco operates using the Wi-Fi protocol which is effectively what is being displayed here. On the right hand side of Figure 21 there is slightly more noise apparent from the centre frequency downwards but the burst communication can still be seen across the time frame. The Parrot Disco is a fixed wing platform which is why the captures only include the classes switched on and flying. Fixed Wing platforms cannot physically hover over one location like a quadcopter.

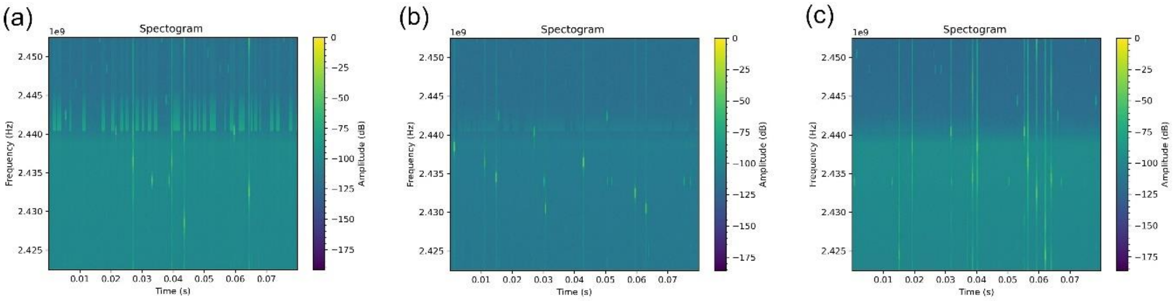


Figure 22 - Spectrogram Graphical Signal Representation for the classes (a) Air 2 S Switched on and connected to the controller, (b) Air 2 S Hovering and (c) Air 2 S flying

Figure 22 shows the spectrogram for the Air 2 S which uses the newest OcuSync 3.0 datalink for the control transmission link. The concentration of activity when the platform is switched on but not yet taken off (a) occurs in the top half of the frequency spectrum. When hovering (b) and flying (c) the activity is seen to spread further across the full range of spectrum shown with small concentrated bursts in the centre and lesser power lines extending up and down the frequency range. If we compare Figure 22 for the Air 2 S (a) (c) to Figure 21 with the Disco signal (b) (c), we are really comparing the Wi-Fi protocol to the Ocusync 3.0. Visually inspecting the signals, the Ocusync 3.0 presents like a Bluetooth burst protocol.

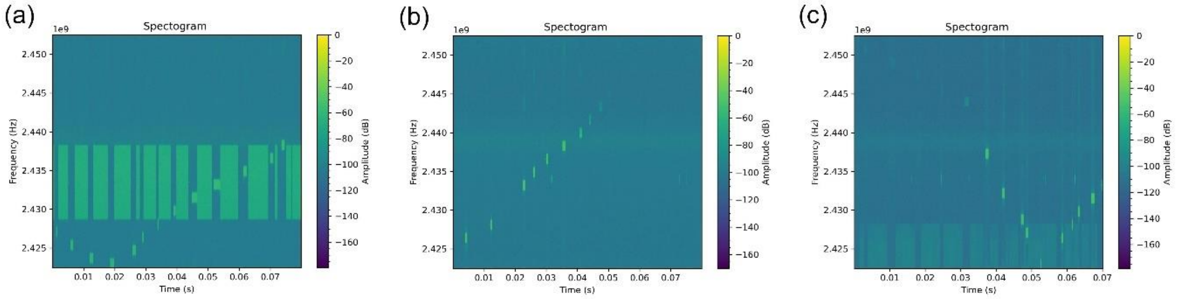


Figure 23 - Spectrogram Graphical Signal Representation for the classes (a) Inspire 2 Switched on and connected to the controller, (b) Inspire 2 Hovering and (c) Inspire 2 flying

Figure 23 show the Inspire 2 from left to right, switched on (a), hovering (b) and flying (c). In the left hand side spectrogram (a) a larger band of activity is visible and all the activity is happening within the lower end of the spectrum. Hovering (b) and flying (c) display more of a Bluetooth burst protocol type signal. The Inspire 2 uses the Lightbridge protocol for transmission.

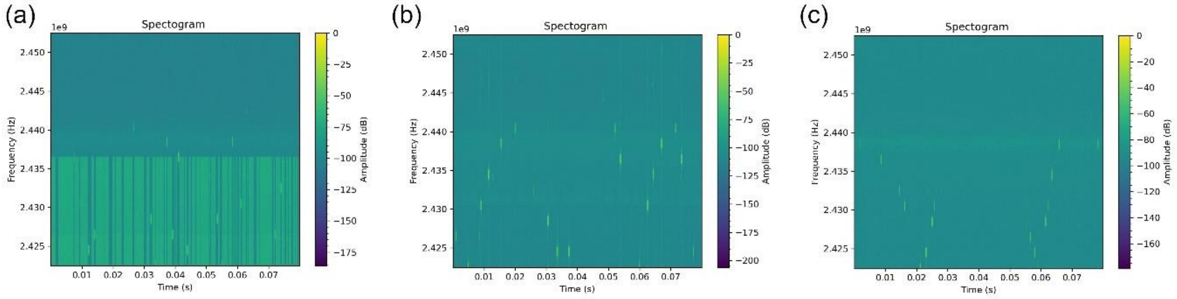


Figure 24 - Spectrogram Graphical Signal Representation for the classes (a) Mavic Pro Switched on and connected to the controller, (b) Mavic Pro Hovering and (c) Mavic Pro flying

The Mavic Pro uses the first edition of the Ocusync 1.0 which is more software based than the original Lightbridge protocol suite. If we compare Figure 24 Mavic Pro (a) (b) (c) to Figure 23 for the Inspire (a) (b) (c), which correspond to the Ocusync 1.0 and Lightbridge 2.0

respectively. The Ocusync 1.0 uses smaller and more frequent bursts of communication around the frequency range.

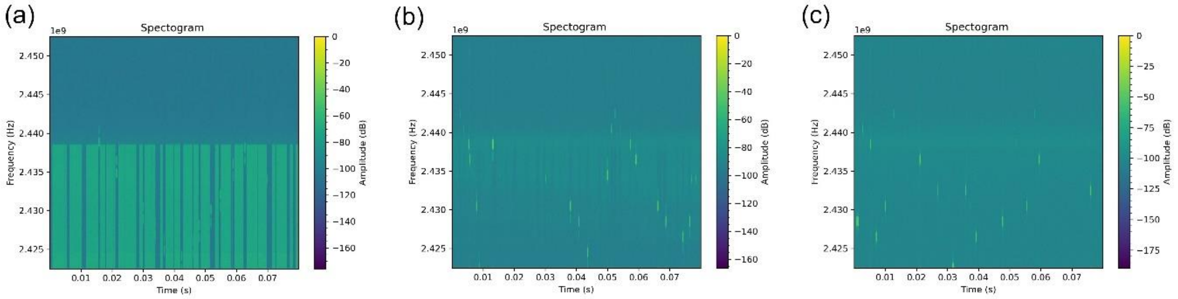


Figure 25 - Spectrogram Graphical Signal Representation for the classes (a) Mavic Pro 2 Switched on and connected to the controller, (b) Mavic Pro 2 Hovering and (c) Mavic Pro 2 flying

The Mavic Pro 2 uses the Ocusync 2.0 transmission protocol and the spectrogram graphical signal representation can be seen in Figure 25. Again the protocol displays as small bursts of activity in yellow which are more concentrated in the bottom half of the spectrum for hovering (b) and flying (c). A larger band is present when the platform is simply switched on but not started flying (a).

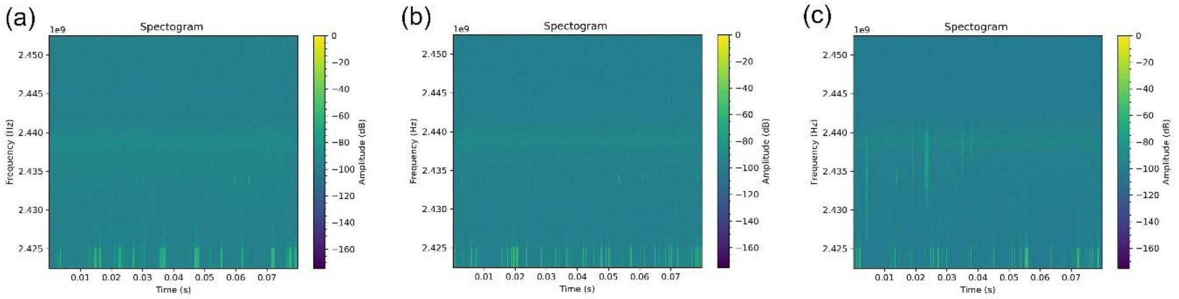


Figure 26 - Spectrogram Graphical Signal Representation for the classes (a) Mavic Mini Switched on and connected to the controller, (b) Mavic Pro Hovering and (c) Mavic Mini flying

Like the Parrot Disco, the Mavic Mini also uses a form of the Wi-Fi protocol to communicate. Figure 26 shows that there isn't much difference in terms of the signal between switched on (a), hovering (b) and flying (c). This is interesting to note for when the experiments are run to see if the features extracted from the CNN can distinguish between the flight mode of the UAS when visually we cannot.

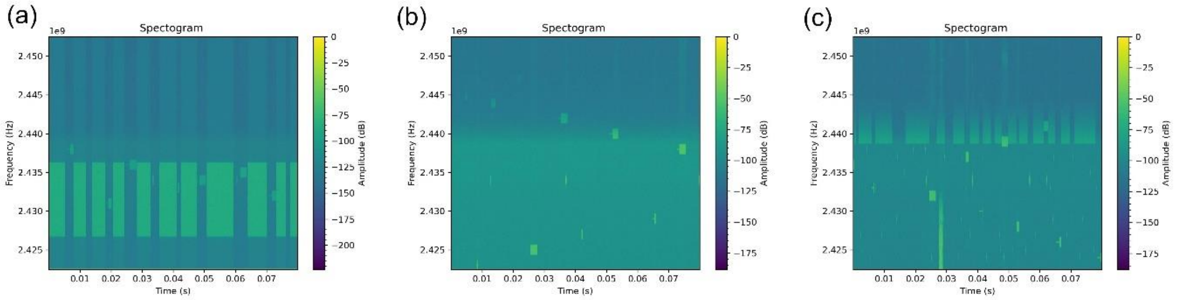


Figure 27 - Spectrogram Graphical Signal Representation for the classes (a) Phantom 4 Switched on and connected to the controller, (b) Phantom 4 Hovering and (c) Phantom 4 flying

Lastly the Phantom 4 spectrograms for switched on but not yet taken off (a), hovering (b) and flying (c) can be seen in Figure 27. As with the Ocusync 2.0, 1.0 and Lightbridge 2.0, a larger yellow band of activity can be seen when the platform is switched on but not yet taken off (a). The Phantom 4 also uses the Lightbridge 2.0 transmission system. For hovering (b) and flying (c) the signal then presents itself in smaller bursts of yellow activity spreading further into the range of the frequencies on show.

Overall the Ocusync and Lightbridge protocols look more like Bluetooth signals on the spectrograms than Wi-Fi signals. When the spectrograms are fed to the CNN they are striped of the writing, range values and labels.

GPS Jamming Signals

Spectrograms were produced in python3 with the PyPlot library from Matplotlib. Figure 28 shows a spectrogram plot of the class ‘no jamming signal present’. Effectively the spectrogram is showing a baseline plot of background AWGN described in more detail in section 3.1.2.

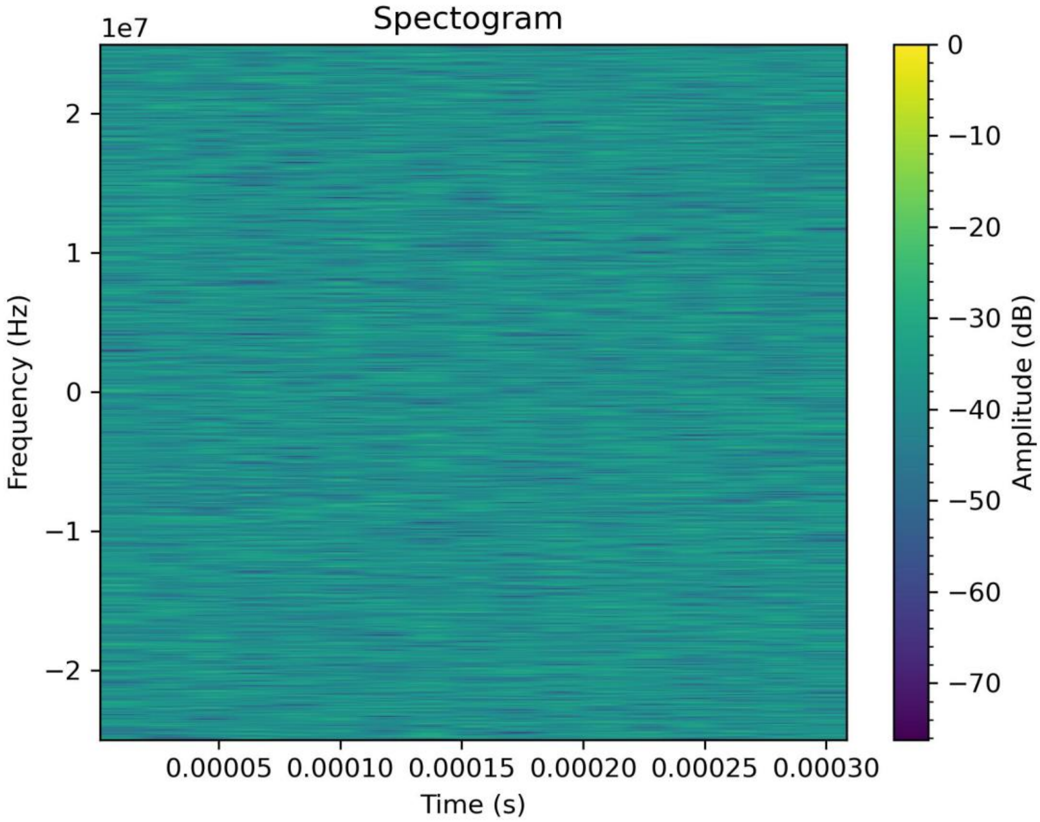


Figure 28 - Spectrogram for the class of No Jamming Present

Figure 29 shows the spectrogram plot for a narrowband jamming signal. From visually inspecting the graph a large yellow band can be seen around the centre frequency. This yellow band represents the increased power of the jamming signal compared to the green of the background. There is a clear separation between the signal and the background AWGN when Figure 28 and Figure 29 are compared.

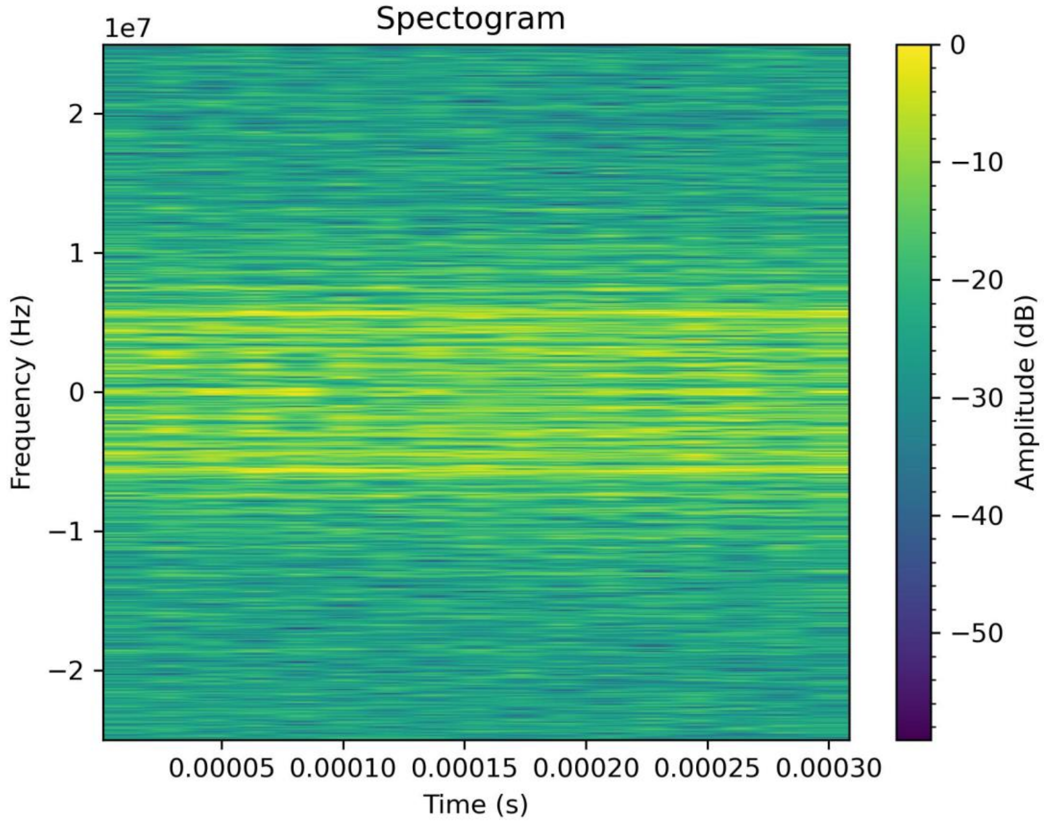


Figure 29 - Spectrogram for the class of Narrowband Jamming Signal

Figure 30 shows the spectrogram graphical signal representation of the chirp signal. It can be seen compared to Figure 29 that the signal has a higher concentration in the centre of the frequency band. If Figure 30 is visually inspected closely it can be seen that the yellow band is actually made up of smaller distinct bands.

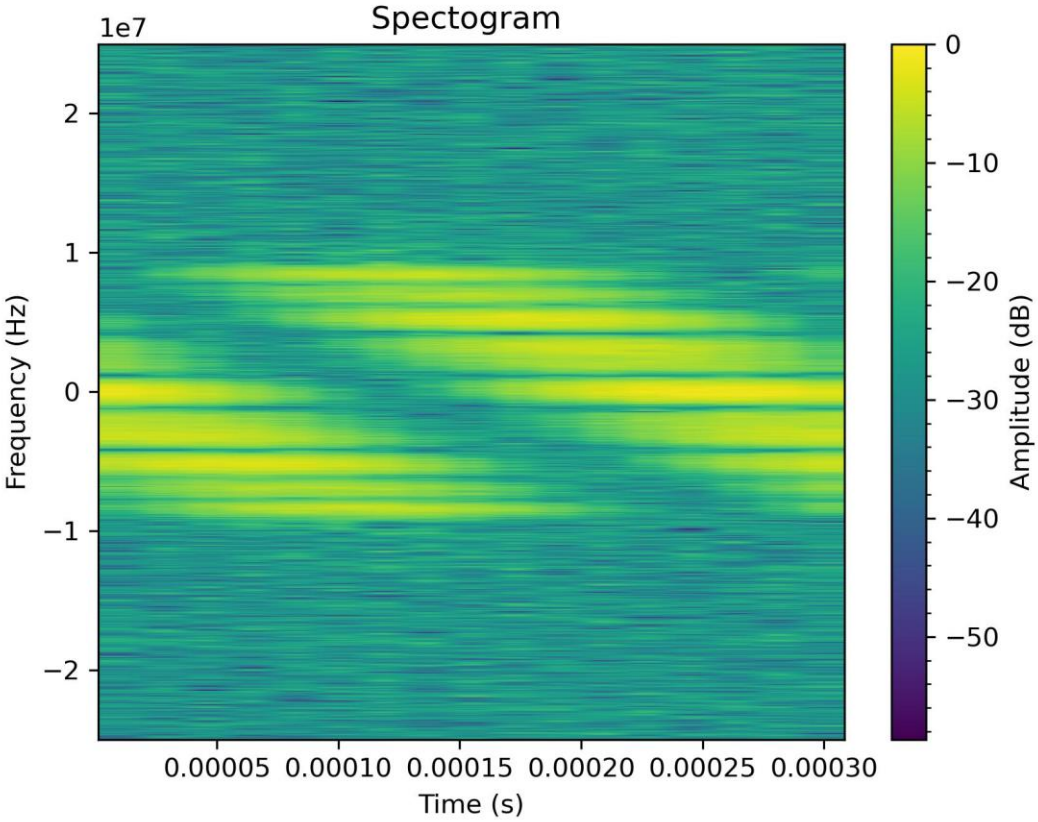


Figure 30 - Spectrogram for the class of Chirp Jamming Signal

Figure 31 shows the spectrogram plot for the FM jamming signal. It can be seen from observing the graph that separated jamming signals appear at discrete frequencies. The signals are periodic in nature occurring primarily upwards of the centre frequency. Comparing the FM signal pattern in Figure 31 to the chirp jammer in Figure 30 we observe a discrete jamming signal compared to the noise generated across the band from the chirp jammer.

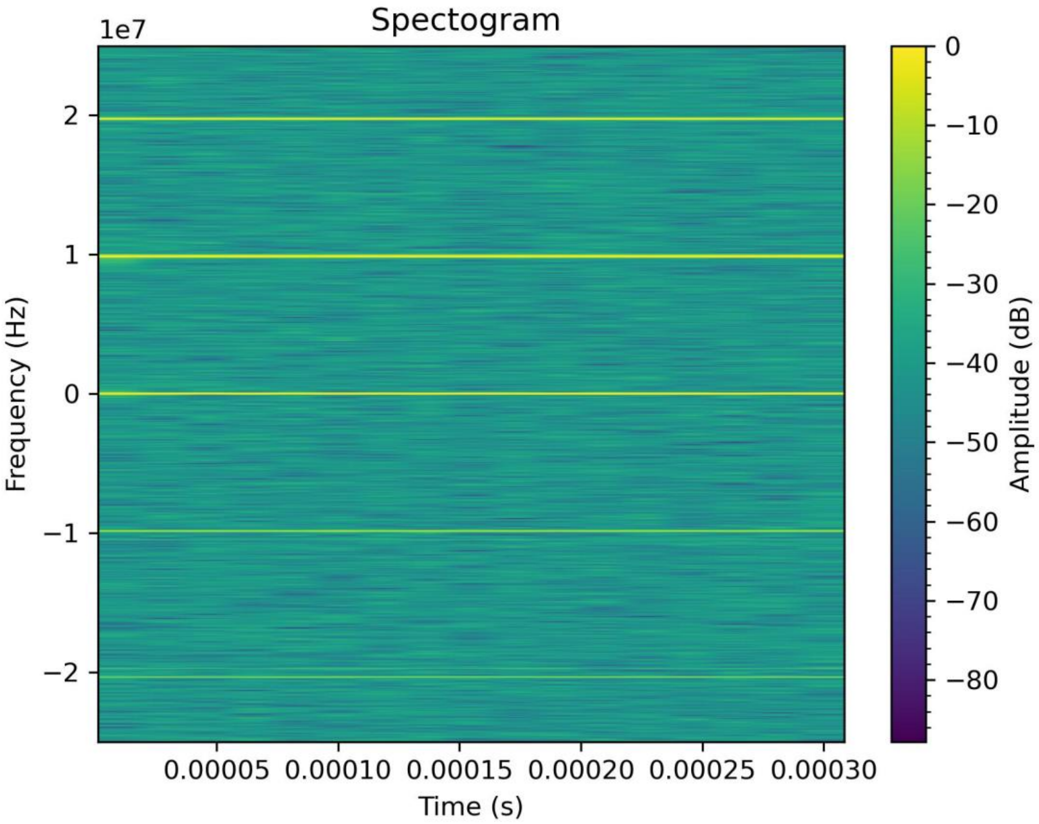


Figure 31 - Spectrogram for the class of FM Jamming Signal

Figure 32 displays the spectrogram graphical signal representation of the AM jammer. The AM jammer consists of a narrow line of noise at one frequency whereby the amplitude of that signal is modulated. The modulation of amplitude is apparent at the start time capture on the spectrogram whereby a glow can be observed around the signal.

$$sin\theta sin\phi = \frac{\cos(\theta - \phi) - \cos\,(\theta + \phi)}{2} \tag{10}$$

To explain Equation (10) can be consulted which shows the product of two sinusoids which represents the carrier signal and the envelope yielding signal at their sum and difference frequencies. If Figure 32 is then compared to the FM signal in Figure 31, it can be observed that the signals do not include any glowing due to the amplitude staying constant in the FM signals.

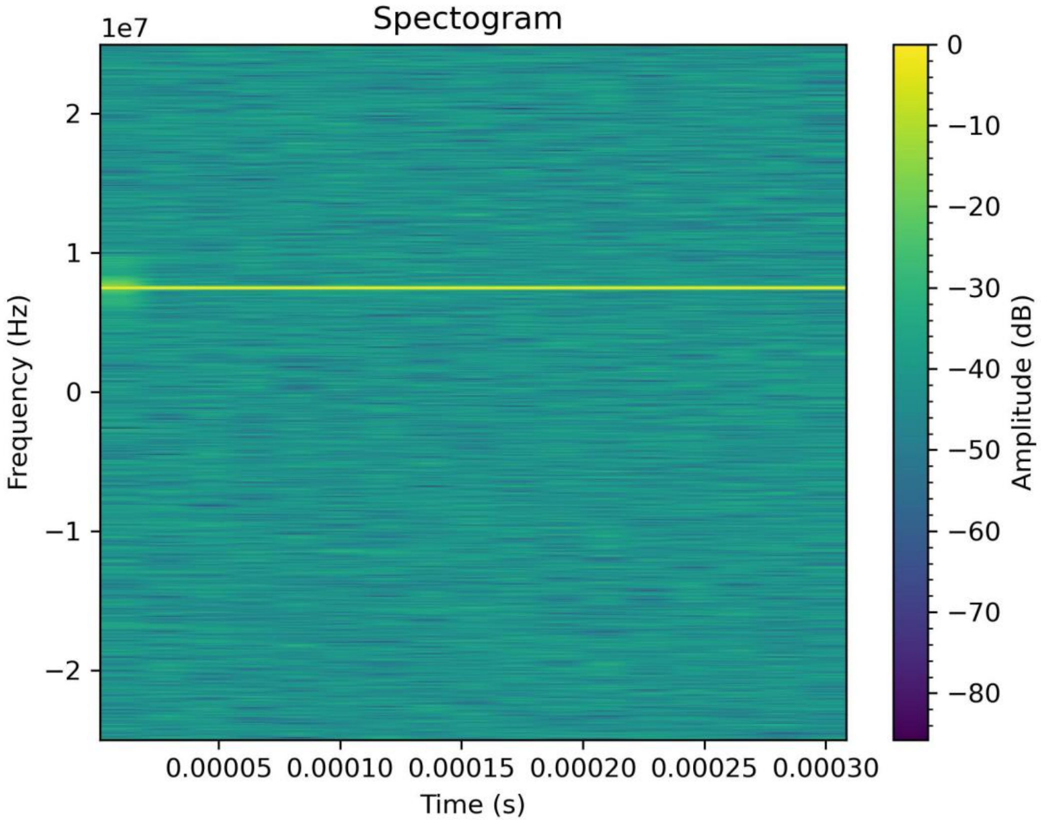


Figure 32 - Spectrogram for the class of AM Jamming Signal

Figure 33 shows the spectrogram for the DME pulse jammer. The pulse is clearly observed at the start of the time period in this capture by the yellow line which fades into green. This is in contrast to the constant jamming signals seen in the FM jammer in Figure 31 and the glow which appears on the constant signal of the AM jammer in Figure 32.

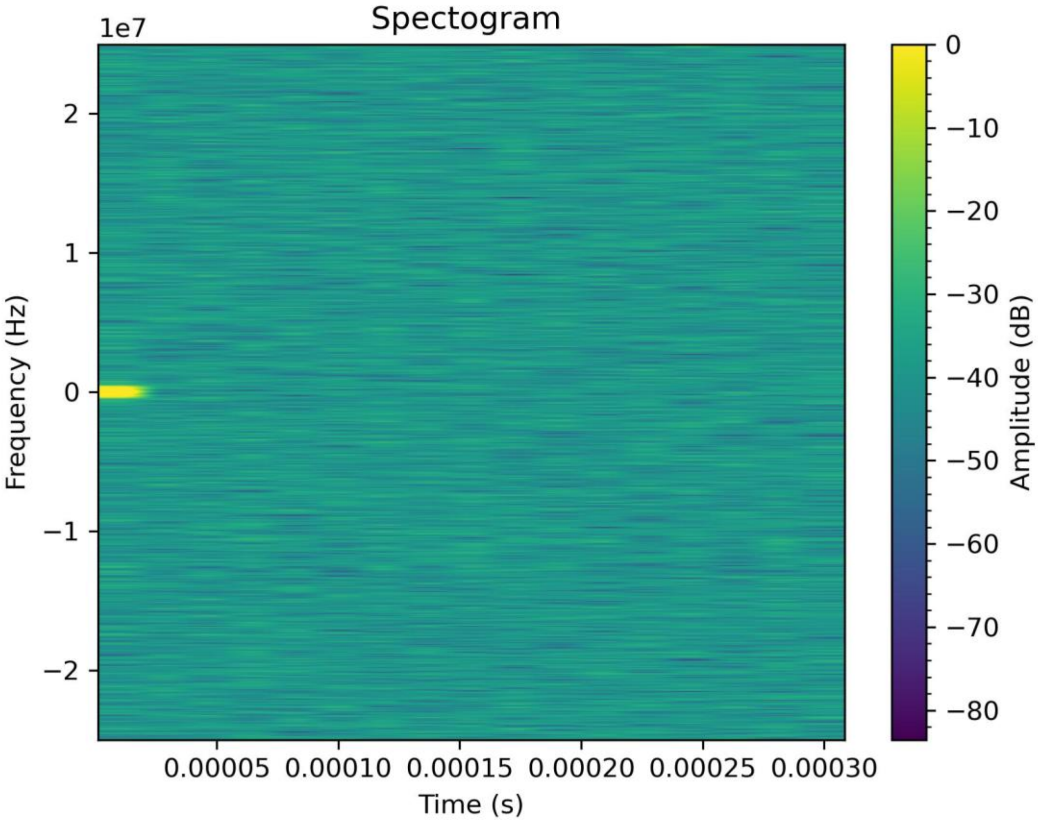


Figure 33 - Spectrogram for the class of DME Jamming Signal

3.2.2 Power Spectral Density (PSD)

PSD calculates the strength of a signal and the distribution of that strength in the frequency domain. This is done using Welch's method, an approach developed by Peter Welch that uses periodogram spectrum estimates and converts the signal from the time to the frequency domain. The Welch method is known for its ability to provide improved estimates when SNR is low but there is a trade-off between the reduction in variables to achieve this and the resolution of the PSD [351]. First the signal in the time domain is partitioned into blocks as shown in Equation (11) [352]. The signal x is broken into m windowed frames and Equation (11) shows the $m\text{-}th$ windowed frame from signal x where R is the hop size (the number of samples between each successive FFT window) and K is the number of frames [352].

$$\begin{gathered} x_m(n) \ \triangleq w(n)x(n+mR) \\ = 0,1,\dots,M-1, m = 0,1,\dots.,K-1 \end{gathered} \tag{11}$$

Next the periodogram of the m th block is calculated as show in Equation (12) [352].

$$Px_m, M(w_k) \triangleq \frac{1}{M} | \sum_{n=0}^{N-1} x_m(n) e^{\frac{-2j\pi nk}{N}} |^2 \tag{12}$$

Lastly in Equation (13) we calculate the Welch estimate which is an average of all the periodograms [352].

$$S_x^W(w_k) \triangleq \frac{1}{K} \sum_{m=0}^{K-1} Px_{m1}, M(w_k) \tag{13}$$

The implementation of PSD in this thesis uses Python 3 Matplotlib which utilises the Welch method. 1024 data points are used in each segment for the FFT and a windowing overlap of 120 points between segments using a Hanning windowing function. Figure 34 shows the PSD for the class No UAS on the left hand side, with the Parrot Disco in the middle switched on and the Disco flying on the right hand side. For our implementation when the PSD images are created they are bounded to -115 to -55dB/Hz on the y axis using the ylim() function. We perform this restriction because the default setting on matplotlib automatically chooses the range of the y-axis depending on the data it has to plot. Therefore if we didn’t not limit the range it could potentially vary between training classes. However, defining a range with the ylim() function allows a consistent range between classes for when the images are fed to the CNN for feature extraction.

UAS Signals

First the thesis will consider the PSD graphical signal representations produced for the UAS signals in the DroneDetect dataset.

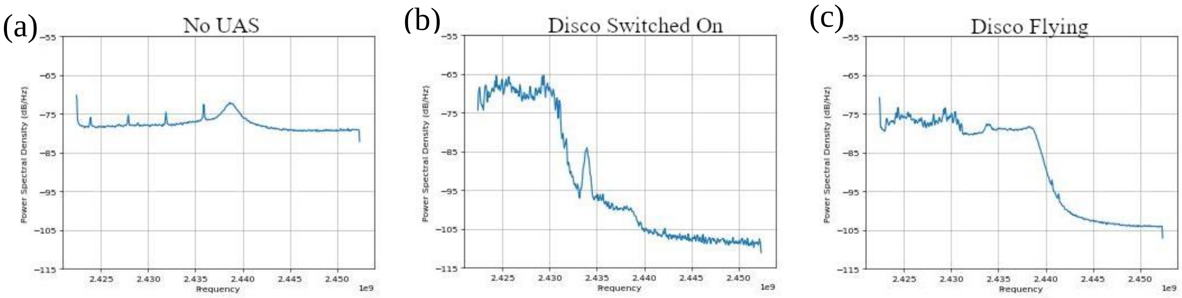


Figure 34 – PSD Graphical Signal Representation No UAS (a), Parrot Disco switched on and connected to the controller (b) and Parrot Disco Flying (c)

It can be seen in Figure 34 on the left hand graph (a) that the noise floor sits around -75dB with very minor background activities being displayed as small periodic peaks. When the disco is switched on (b) in the middle a spike can be seen in the lower third of the spectrum with a significant drop in the higher frequencies. When the disco is in flight (c) the power levels out for the first half of the frequency range and then drops off in the higher third of the spectrum covered. The disco uses the Wi-Fi protocol and is fixed wing so only the switched on and flying flight modes can be considered.

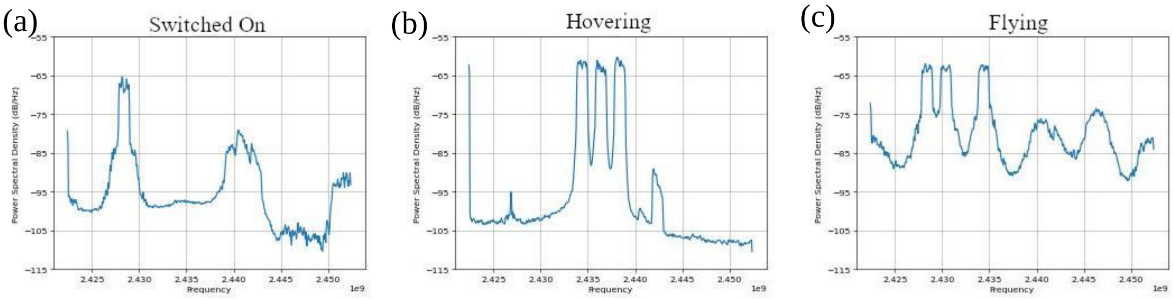


Figure 35 - PSD Graphical Signal Representation Air 2 S switched on and connected to the controller (a), Air 2 S hovering (b) and Air 2 S flying (c)

Figure 35 (a) (c) shows the Air 2 S which uses the Ocusync 3.0 datalink. Compared with Figure 34 (b) (c) there is a difference in both hovering (b) and flying (c) flight mode PSD representations which are easily distinguishable with the human eye. We can also see that the power peaks are around 10dB higher for the Ocusync 3.0 compared with Wi-Fi. Within Figure 35 for the platform switched on (a) there are peaks at 2.428GHz and 2.441GHz. For hovering (b) three peaks occur around the centre frequency and at a high power level. For the flying (c) class there are three main peaks again but they are spread out slightly further and occur in the lower half of the frequency range, rather than around the centre frequency.

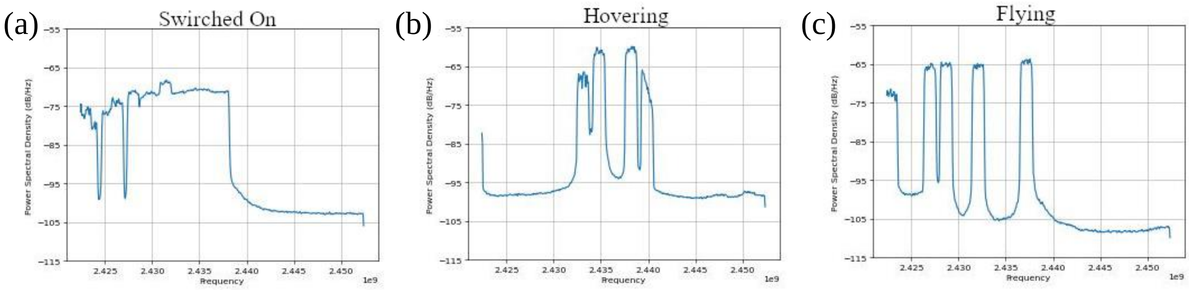


Figure 36 - PSD Graphical Signal Representation Inspire 2 switched on and connected to the controller (a), Inspire 2 hovering (b) and Inspire 2 flying (c)

Figure 36 shows the PSD for the Inspire 2 which uses the Lightbridge 2.0 transmission protocol. The PSD for the class switched on (a) is shown on the left and visually looks similar to the Wi-Fi protocol, with the signal dropping off significantly in the higher end of the spectrum. The PSD for hovering (b) and flying (c) are similar in terms of where the peaks occur to the Ocusync 3 in Figure 35. For hovering (b) the peaks occur again around the centre frequency (the middle peaks around -5db/Hz higher than the flying peaks). For the flying (c) class the peaks occur from the centre frequency down to the lower end of the spectrum.

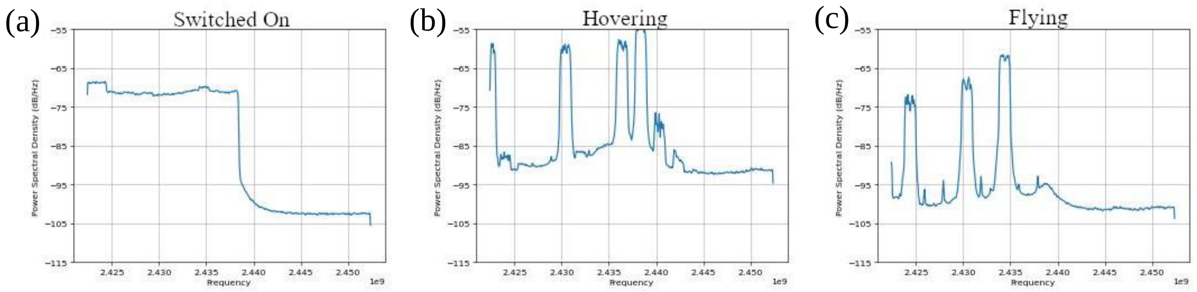


Figure 37 - PSD Graphical Signal Representation Mavic Pro switched on and connected to the controller (a), Mavic Pro hovering (b) and Mavic Pro flying (c)

Figure 37 shows the Mavic Pro which utilises the OcuSync 1.0 transmission link. Comparing Figure 37 with Figure 36 it can be seen that for the class switched on (a) the pattern is similar but the power is stable at around -70db/Hz for the Mavic Pro. Again for hovering (b) we see the majority of the peaks to occur around that centre frequency and with a higher level of power. Then for the flying (c) those peaks occur again the lower half of the frequency range at a lower power level.

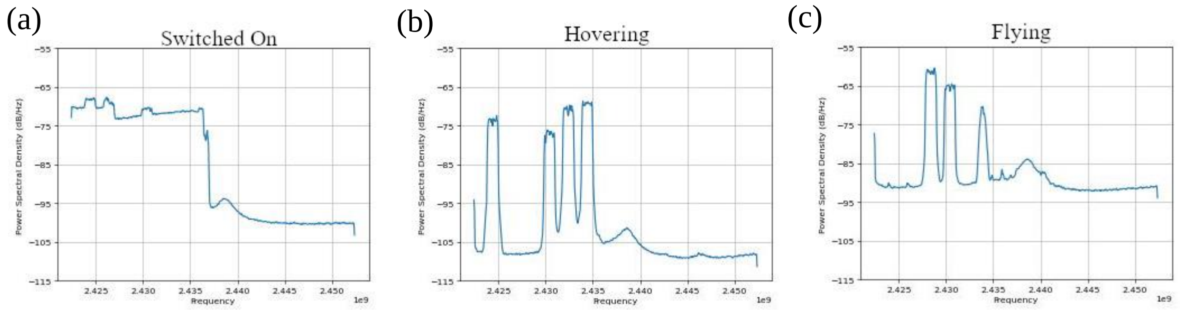


Figure 38 - PSD Graphical Signal Representation Mavic Pro 2 switched on and connected to the controller (a), Mavic Pro 2 hovering (b) and Mavic Pro 2 flying (c)

Figure 38 shows the Mavic Pro 2 which uses the Ocusync 2.0. The switched on (a) class visually has a similar pattern to both the Ocusync 1.0 in Figure 37 and the Lightbridge 2.0 in Figure 36. However for the hovering (b) and flying (c) we see a change in the pattern, the power is lower for the hovering peaks compared to the flying peaks. Also the peaks for hovering (b) occur in the lower end of the frequency spectrum along with the flying (c) peaks.

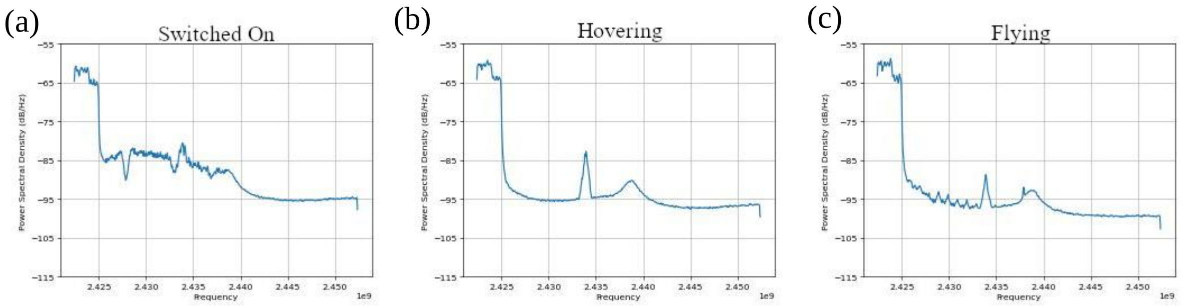


Figure 39 - PSD Graphical Signal Representation Mavic Mini switched on and connected to the controller (a), Mavic Mini hovering (b) and Mavic Mini flying (c)

Figure 39 is the PSD for the Mavic Mini which, like the Parrot Disco in Figure 34 (b) (c), uses the Wi-Fi protocol for communication. A peak can be seen at the same point in each PSD for switched on (a), hovering (b) and flying (c). This peak occurs at 2.434GHz. This could be the video transmission downlink or another constant within the overall signal. Overall the pattern of the signal for hovering (b) and flying (c) are very similar visually.

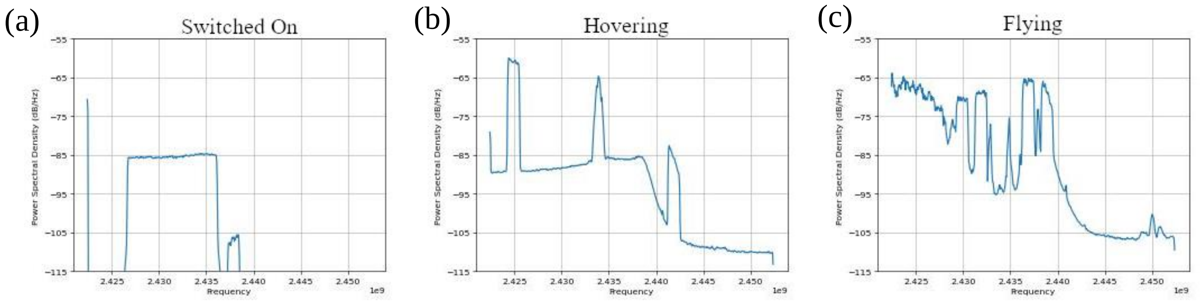


Figure 40 - PSD Graphical Signal Representation Phantom 4 switched on and connected to the controller (a), Phantom 4 hovering (b) and Phantom 4 flying (c)

Lastly in Figure 40 the PSD for the Phantom 4 can be seen. The Phantom 4 uses the Lightbridge 2.0 for transmission, the same transmission system used by the Inspire in Figure 36. The PSD in Figure 40 for switched on (a) is of a similar pattern to Figure 36 but with a large drop in power around 2.424GHz and with an overall drop in power compared to the Inspire. The class hovering (b) in Figure 40 has three peaks which are fairly spread out with one occurring above the centre frequency and the other two below it. The flying class (c) PSD has more peaks and there is more power in the lower end of the spectrum. Although the Phantom and Inspire are operating using the same datalink, there are visual differences between the corresponding PSD images in Figure 36 and Figure 40.

GPS Jamming Signals

Figure 41 shows the PSD graphical signal representation of the no jamming class in the GPS Jamming Dataset described in section 3.1.1. Effectively the PSD graph is showing a baseline plot of background AWGN described in more detail in section 3.1.2.

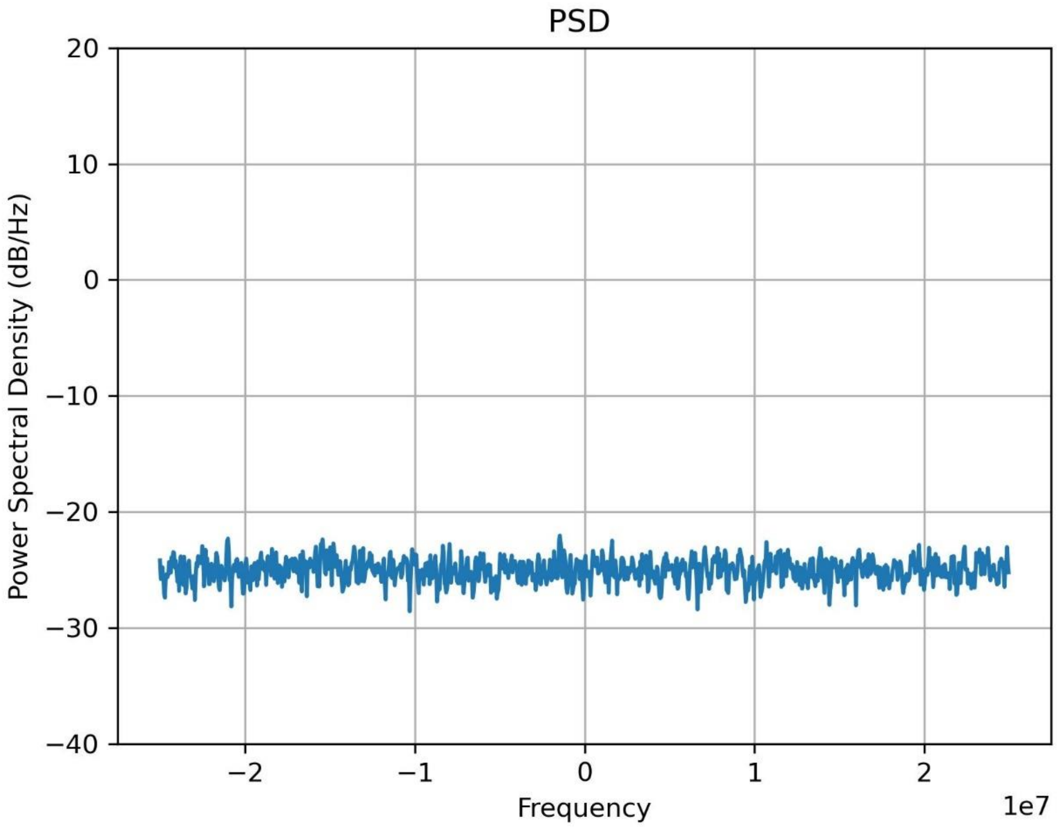


Figure 41 - PSD Graphical Signal Represenation for the class No Jammer Present

Figure 42 shows the PSD of the narrowband jamming signal. Compared with Figure 41 it can be seen that the power across the whole spectrum is increased by a minimum amount of 5-10dB/Hz and then there is a concentrated increase around the centre frequency of around 20-30dB/Hz. The jamming signal is clearly visible when comparing the narrowband signal in Figure 42 with the no jamming signal present class in Figure 41.

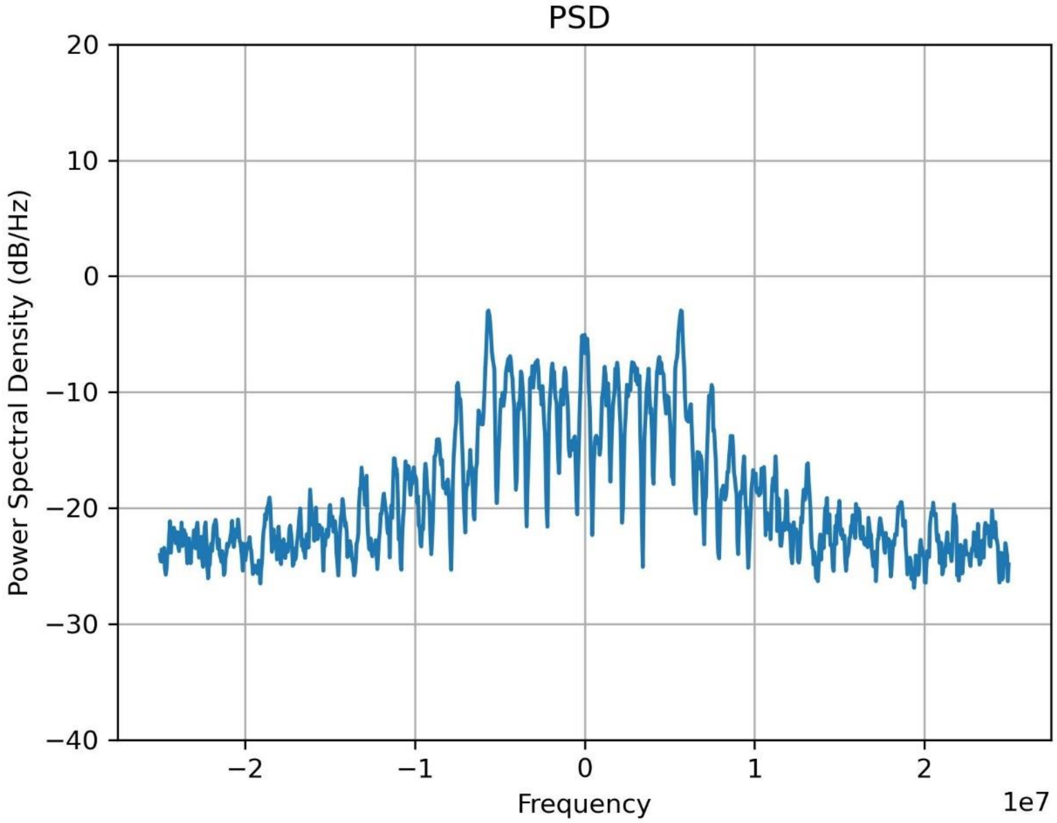


Figure 42 - PSD Graphical Signal Represenation for the class Narrowband Jammer

While with the narrowband jamming signal in Figure 42 an increase in power can be seen across the whole frequency band, the chirp jammer only emits the signal 20MHz around the centre frequency, as seen in Figure 43. The rest of the spectrum is comparable with the no jamming signal in Figure 41. This makes the chirp signal a different pattern visibly to that of the narrowband jammer in Figure 42.

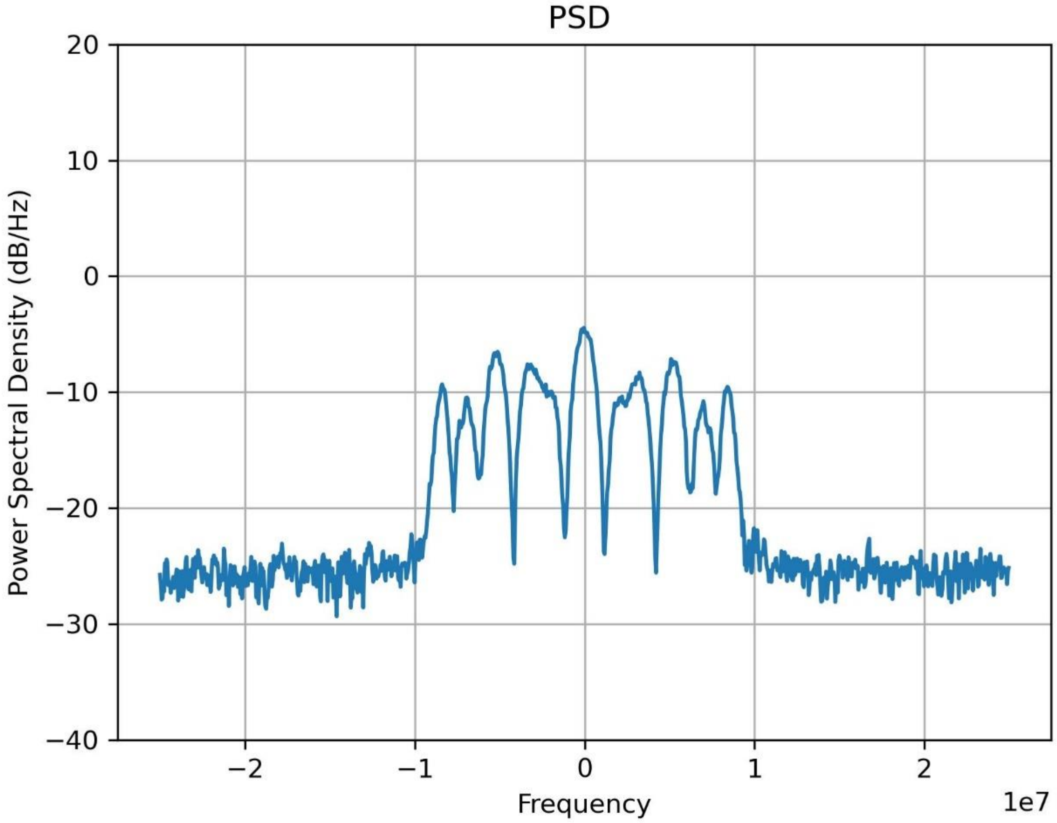


Figure 43 - PSD Graphical Signal Represenation for the class Chirp Jammer

Figure 44 shows the PSD for the AM jamming signals and the spike in power can be seen around 7MHz above the centre frequency. The dB/Hz value will likely chance for this signal depending on when the capture is taken but the frequency will remain the same.

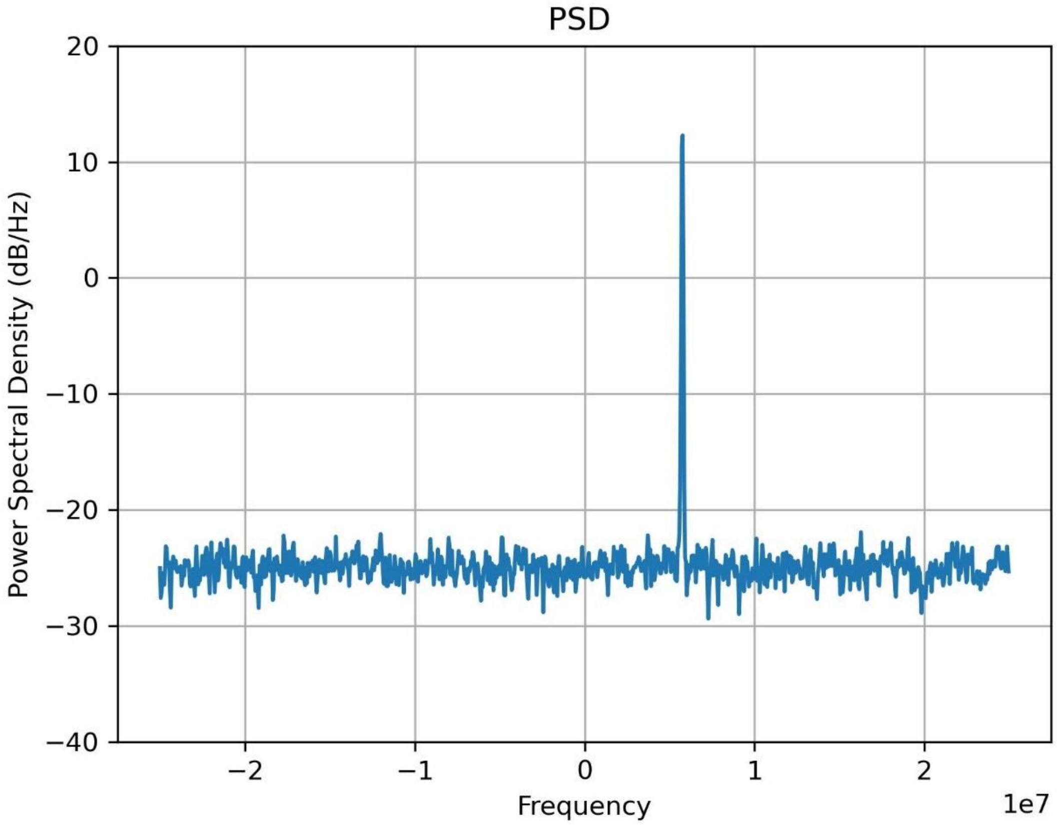


Figure 44 - PSD Graphical Signal Represenation for the class AM Jammer

Figure 45 shows the PSD for the FM jamming signal. As with the AM signal in Figure 44 where one spike of activity was observed, the FM signal displays multiple spikes at various different frequencies. The peaks are highest around the centre frequency and then decrease in power as they move away from the centre frequency. This forms a distinct patter again compared to the no jamming signal class in Figure 41.

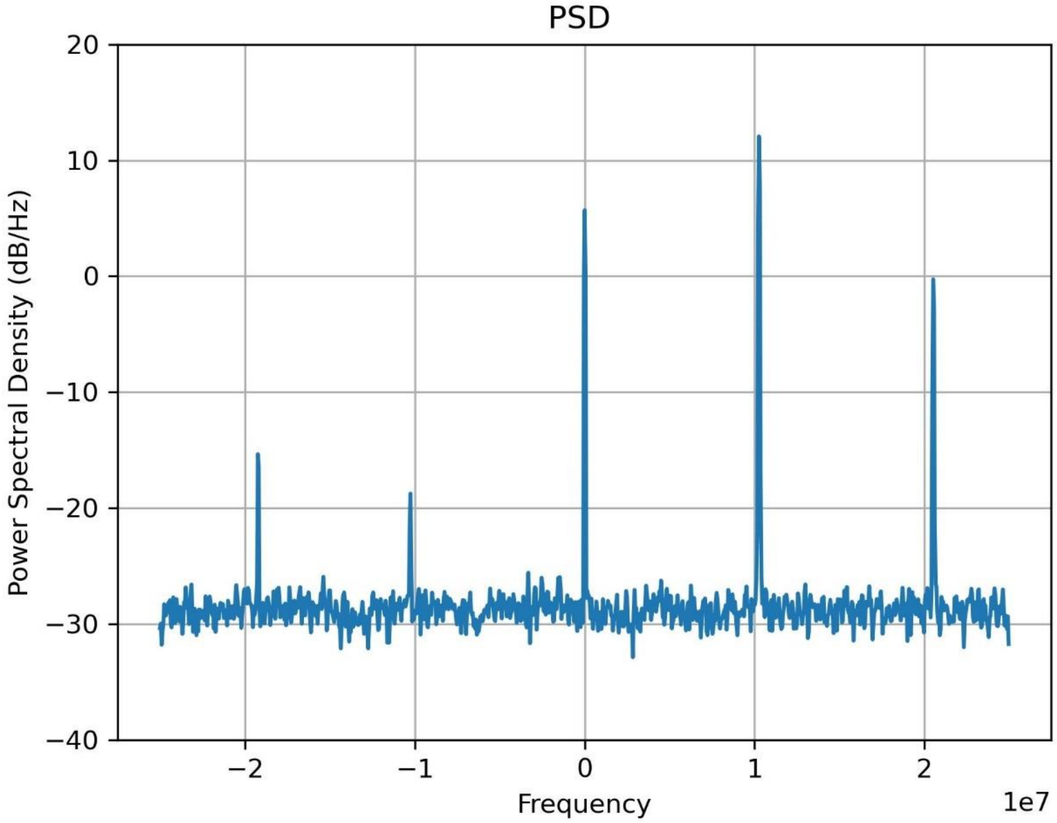


Figure 45 - PSD Graphical Signal Represenation for the class FM Jammer

Lastly the PSD for the DME pulse jamming signal is observed in Figure 46. On the PSD the DME pulse looks similar to the AM signal in Figure 44. However, if the signals are observed closely it can be seen that the AM signal is a straight line which fans out at the bottom while the DME is more triangular in shape. This is a pattern makes the two signals distinguishable on the PSD representation.

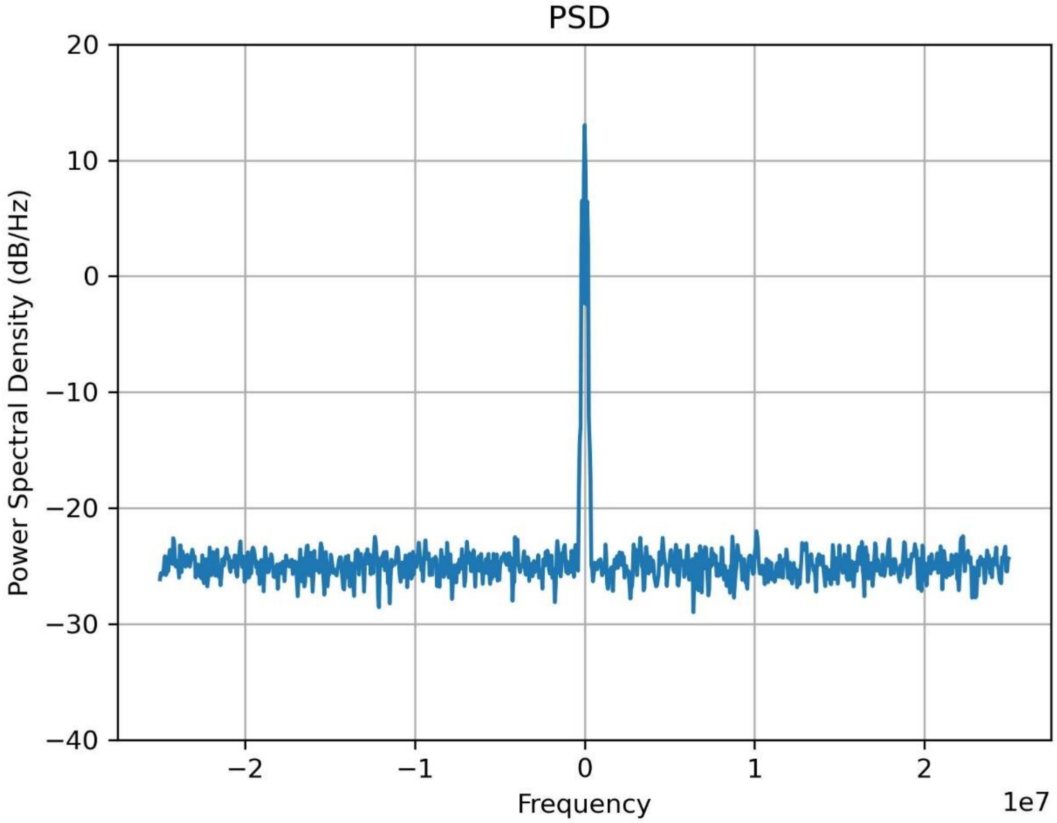


Figure 46 - PSD Graphical Signal Represenation for the class DME Pulse Jammer

Overall the both the Spectrogram and PSD graphical signal representations show distinct visual differences that an experienced human operator could likely easily identify. However, that operator may have to manually scroll through a significant amount of spectrum plots which contain no jamming activity. This is time consuming and errors can be made, events can be missed, especially when operators are tired. Machine learning presents an opportunity to direct a human operator to a place of interest, increasing efficiency and providing an indication of the class of the signal in a much quicker time frame.

3.2.3 Histogram and Raw IQ Plot

Next the histogram and the raw IQ plots were also considered. The raw data captured from an SDR is IQ data. I stands for In phase and Q for Quadrature, the I representing the real component of the signal and Q the imaginary, a complex number. A very basic SDR receiver connected to an antenna is shown in Figure 47 [353].

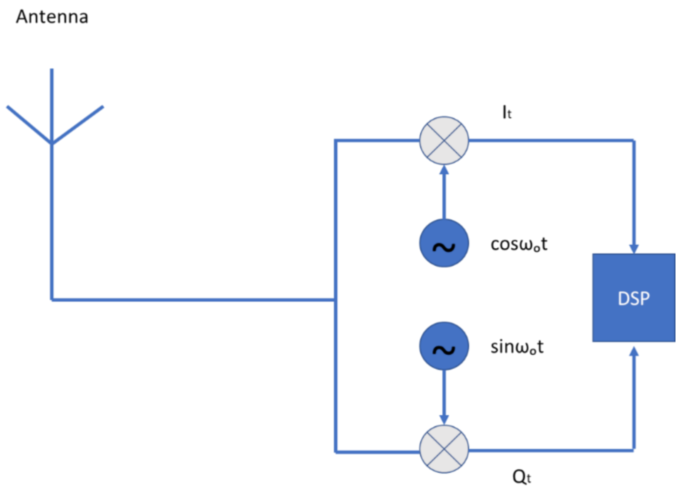


Figure 47 – The components which would form part of a basic SDR Receiver

In Figure 47 ω can be equated to $2\pi f$ where f is the frequency from the local oscillator. With respect to time, I and Q components have the same phase relationship as $sinx$ being 90 degrees different from $cosx$. Utilising I and Q components allows signals of different frequencies above and below the local oscillation frequency to be separated. There are other advantages as these vectors provide more information for a Fast Fourier Transform (FFT) than a single scaler. Further, it will produce the same result with half the sampling rate so a wider bandwidth can be achieved [353]. I and Q components can be plotted with Matplotlib to show various different graphical signal representations presented in section 3.2.

Figure 48 to Figure 53 show the 6 classes of GPS jamming signals graphically represented as both raw constellation scatter plots and histograms. Note that as with the PSD the x and y axis's are both limited to provide consistency between the classes. On the raw constellation scatter plot the y axis is limited from -8000 to 8000 using the Matplotlib ylim() function and the x axis is limited from -7500 to 30000 using Matplotlib xlim() function. For the histogram the y axis is limited from 0 to 1200 using the Matplotlib ylim() function and the x axis is limited from -8000 to 10000 using Matplotlib xlim() function.

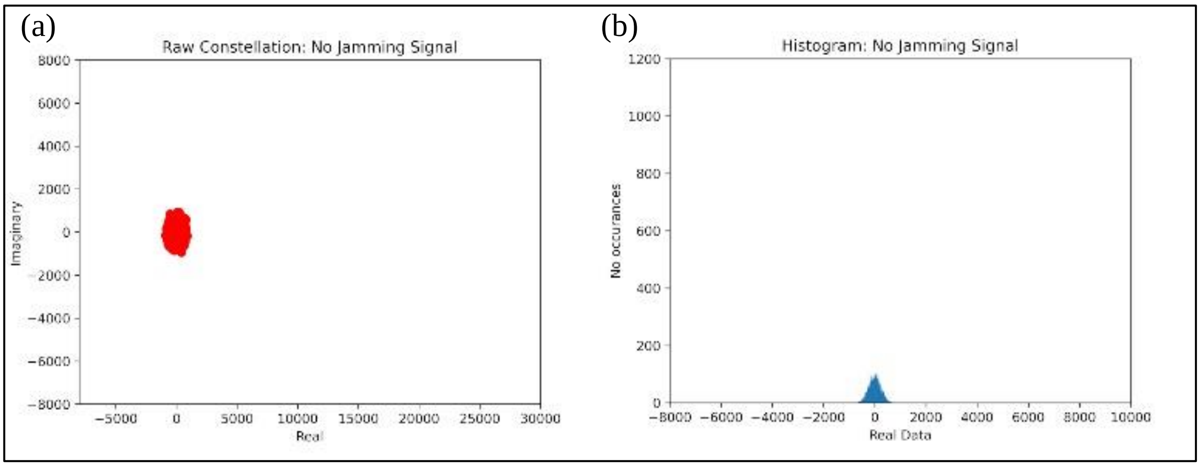


Figure 48 - Raw IQ Plot (a) & Histogram (b) for the class No Jamming Present

Figure 48 shows the raw IQ plot which is a scatter diagram (a) and the histogram plot (b), both plotted using Matplotlib. The no jamming signal class is a representation of any background noise, in the case of the GPS Jamming dataset described in section 3.1.1 this is the AWGN produced with a uniform distribution whereby C/N0 is between 25-50dBHz and JSR between 40-80dB. In Figure 48 a small cluster of red is observed in the lower left quadrant of the raw IQ scatter plot (a) and on the histogram (b) a small peak at the centre of the distribution.

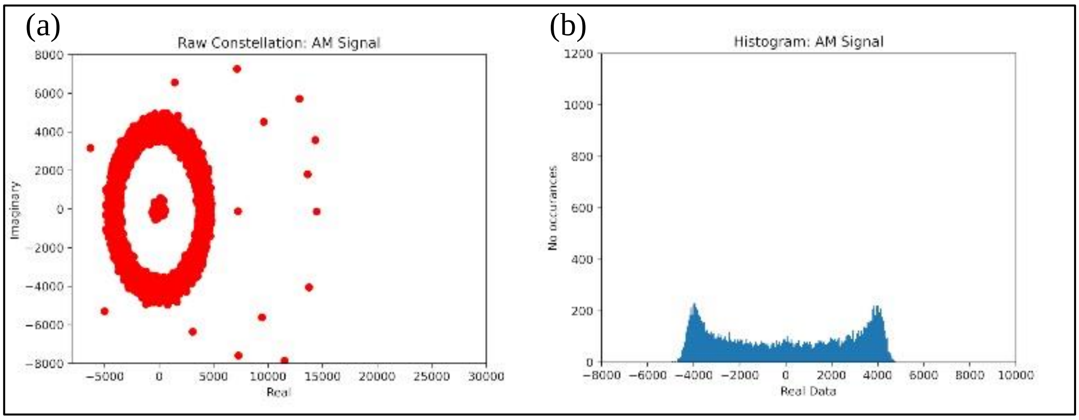


Figure 49 - Raw IQ Plot (a) & Histogram (b) for the class AM Jamming Signal

Figure 49 shows the raw IQ plot (a) and histogram (b) for the AM jamming signal. It can be observed compared to Figure 48 and the no jamming class, that the AM signal fills a greater proportion of both of the plots. The pattern is very distinctive in the raw IQ (a) presenting itself almost as an ovate bullseye and the histogram (b) showing peaks at each end of the real component distribution. Visually there is a clear difference between Figure 48 for no jamming signal present and Figure 49 for the presence of an AM jamming signal.

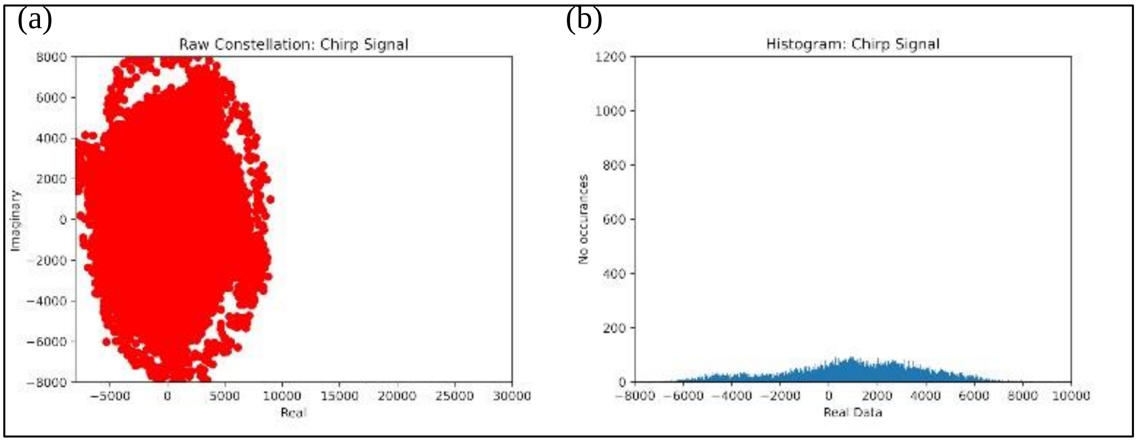


Figure 50 - Raw IQ Plot (a) & Histogram (b) for the class Chirp Jamming Signal

Figure 50 shows the raw IQ plot (a) and histogram (b) for the Chirp jamming signal. Compared with the AM signal in Figure 49 and no jamming signal in Figure 48, the chirp jammer presents itself as a larger concentration again of red in the raw IQ plot (a) and as a wider and more evenly spread distribution on the histogram (b).

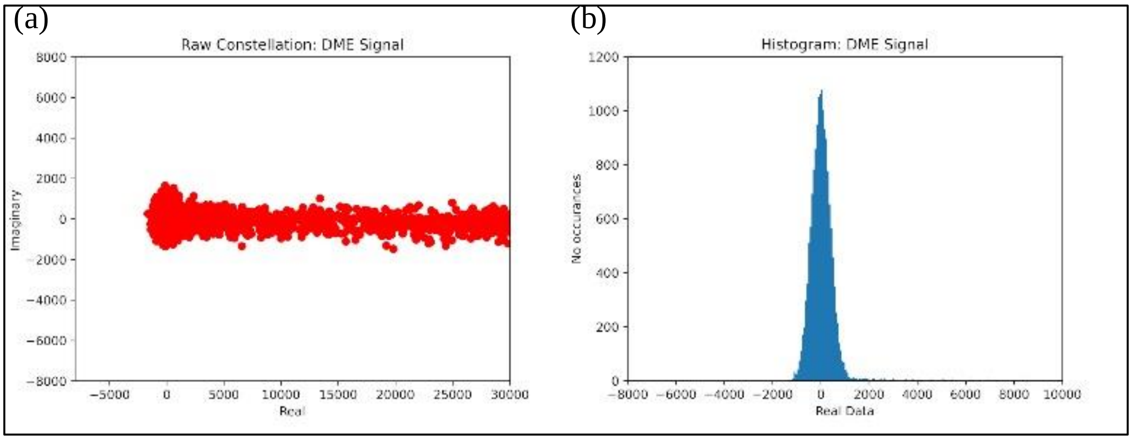


Figure 51 - Raw IQ Plot (a) & Histogram (b) for the class DME Jamming Signal

Figure 51 shows the raw IQ plot (a) and histogram (b) for the DME or pulse jamming signal. The pulse is clearly observed as a red concentration in the form of a line with a circular head in the raw IQ plot (a). On the histogram (b) it presents itself as a vertical distribution which extends upwards to the top of the y axis range. Comparing the patterns to the classes seen so far; no jammer in Figure 48, the AM jammer in Figure 49 and the chirp jammer in Figure 50, there is again a clear visible difference in the pattern of each different jamming signal type.

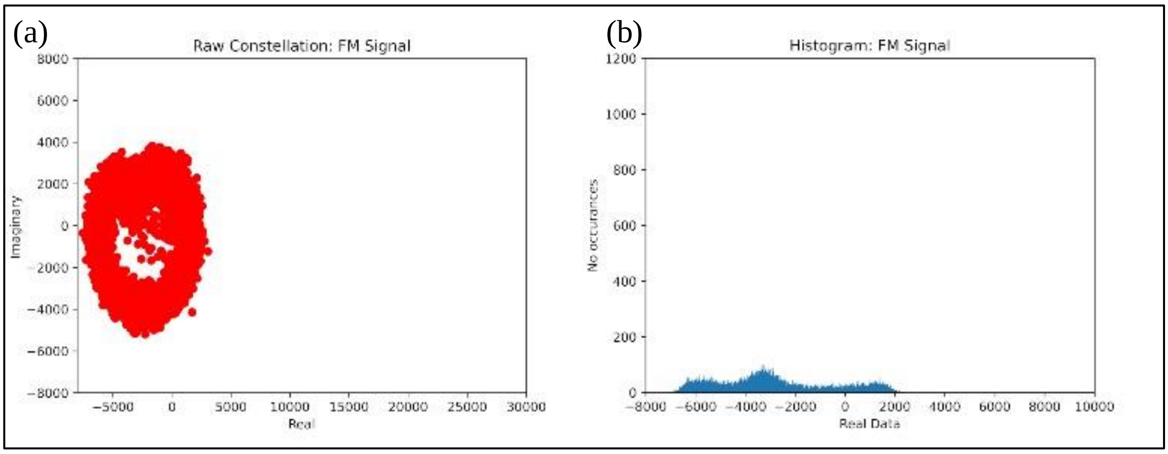


Figure 52 - Raw IQ Plot (a) & Histogram (b) for the class FM Jamming Signal

Figure 52 shows the raw IQ plot (a) and histogram (b) for the FM jamming signal. The FM signal comprises of several jamming signals at different frequencies being sent out at once. On the raw IQ (a) this presents itself as a circular pattern which is mostly filled in compared to the hollow bullseye pattern that the AM signal produced in Figure 49. On the histogram (b) 3 peaks can be observed on the distribution which correlates to the jamming signals on different frequencies which make up the FM signal.

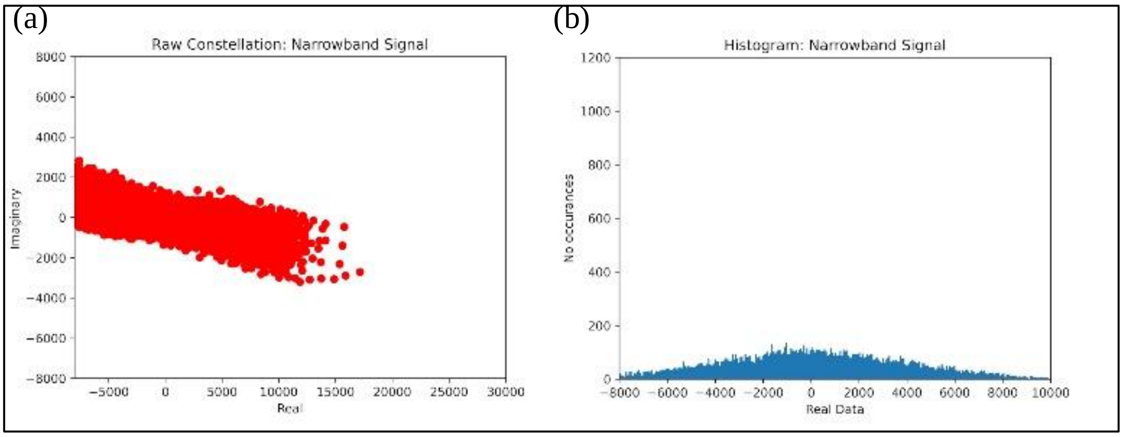


Figure 53 - Raw IQ Plot (a) & Histogram (b) for the class Narrowband Jamming Signal

Figure 52 shows the raw IQ plot (a) and histogram (b) for the Narrowband jamming signal. While the raw IQ plot (a) looks similar in its horizontal spread to the DME pulse jammer in

Figure 51, the histogram (b) looks significantly different, displaying a wide spread across the whole distribution in contrast to the vertical peak of the DME jammer. Comparing the narrowband jammer in Figure 52 to the no jamming class in Figure 48, it is visually clear that a jamming signal is present. When all the graphical signal representations for raw IQ (a) and histogram (b) are compared for all the classes from Figure 48 through to Figure 53, they are all visually different to the eye.

3.2.4 Concatenation

The block process for taking the raw data, producing the graphs and then concatenating them into one image is shown as a block diagram in in Figure 54.

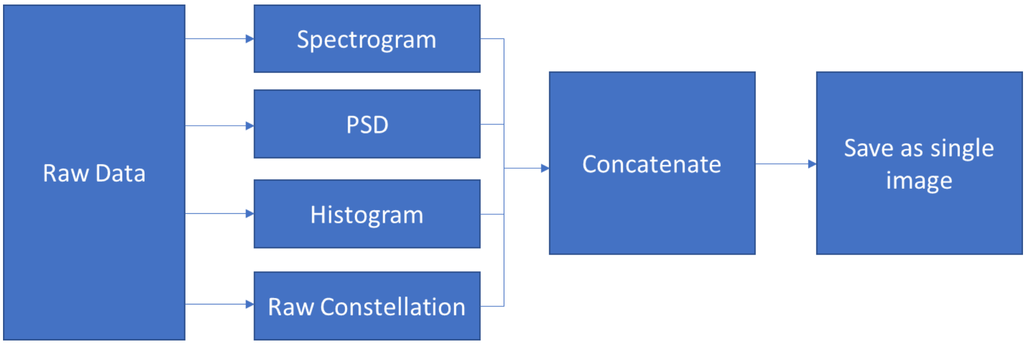


Figure 54 - Block diagram of concatenation process from raw data to an image ready for the CNN feature extraction

As described in sections 3.2.1 to 0, Python3 library Matplotlib was used to generate spectrogram, PSD, histogram, and constellation colour plots. Two functions were defined in python for vertical and horizontal concatenation. Essentially the images are stitched together to form a larger image and then the new image can be resized to 224x244 pixels which is the size required for the CNN. The following concatenated images are produced from the DroneRF dataset detailed in section 3.1.3. The frequency range covered is from 2.402GHz – 2.482GHz (Ch 1 – Ch 13 Wi-Fi bands with the exception of the first and last 1 MHz). In the figures below 0Hz represents the centre of the captured spectrum 2.442GHz.

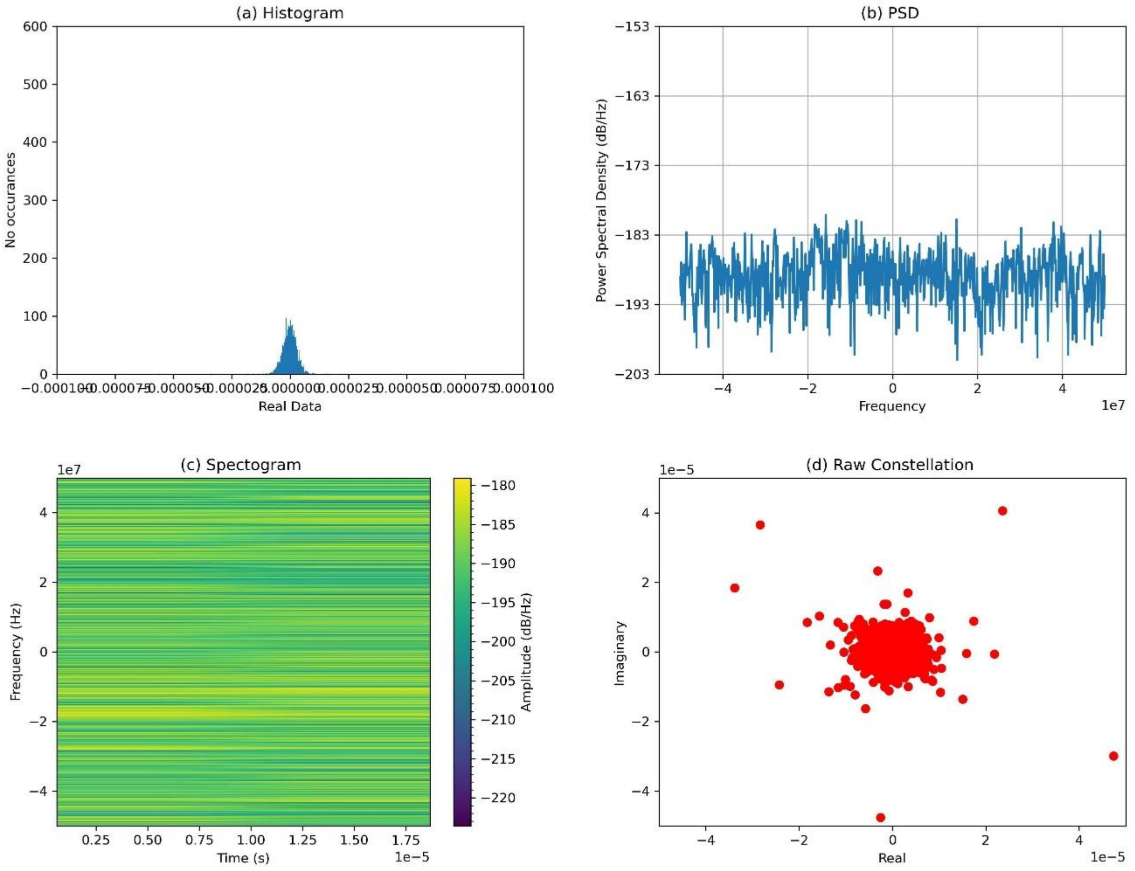


Figure 55 - DroneRF dataset in graphical signal representations (a) Histogram, (b) PSD, (c) Raw Constellation, (d) Spectrogram for class No UAS Present

Figure 55 shows the different signal representations when there is no UAS present. What we are looking at here is the background noise or other signals present in the frequency band at the time of signal capture. Figure 56 shows the Bebop in mode 1 – switched on and connected to the controller. It is clear on the PSD (b) that there is some activity in the higher end of the spectrum (2.44-2.48GHz). As there is no video transmission in this mode, it is likely that this is the command and control signal between the UAS and controller.

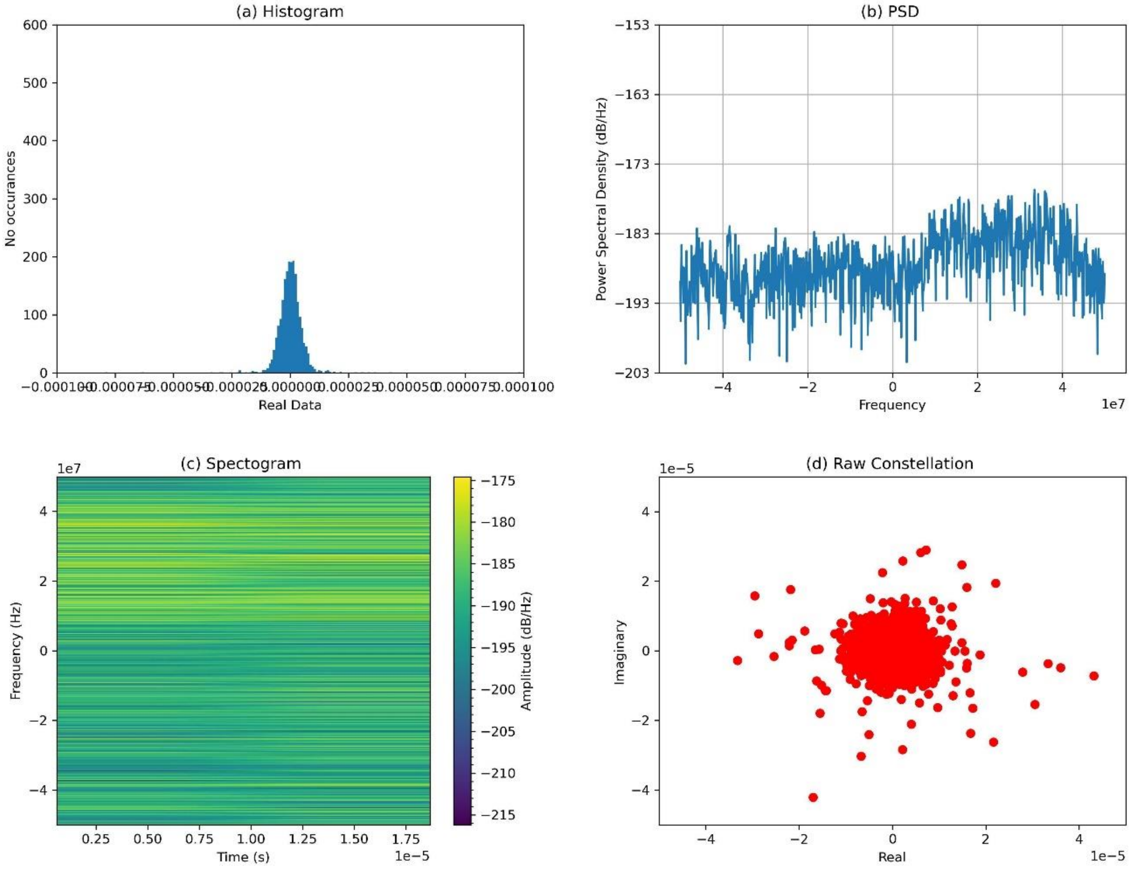


Figure 56 - DroneRF dataset in graphical signal representations (a) Histogram, (b) PSD, (c) Raw Constellation, (d) Spectrogram for class Bebop Mode 1 - switched on

In Figure 57 the Bebop is hovering in automatic mode. What we can see is that there is an even spread of activity across the entire band. In particular if we compare the PSD (b) to the no UAS class. If we compare the spectrogram in Figure 57 where the platform is hovering in an automatic mode (with no active communication with the controller) to **Error! Reference source not found.** (no UAS present), we see a decrease in power in the higher end of the spectrum, again indicating this to be a command and control signal.

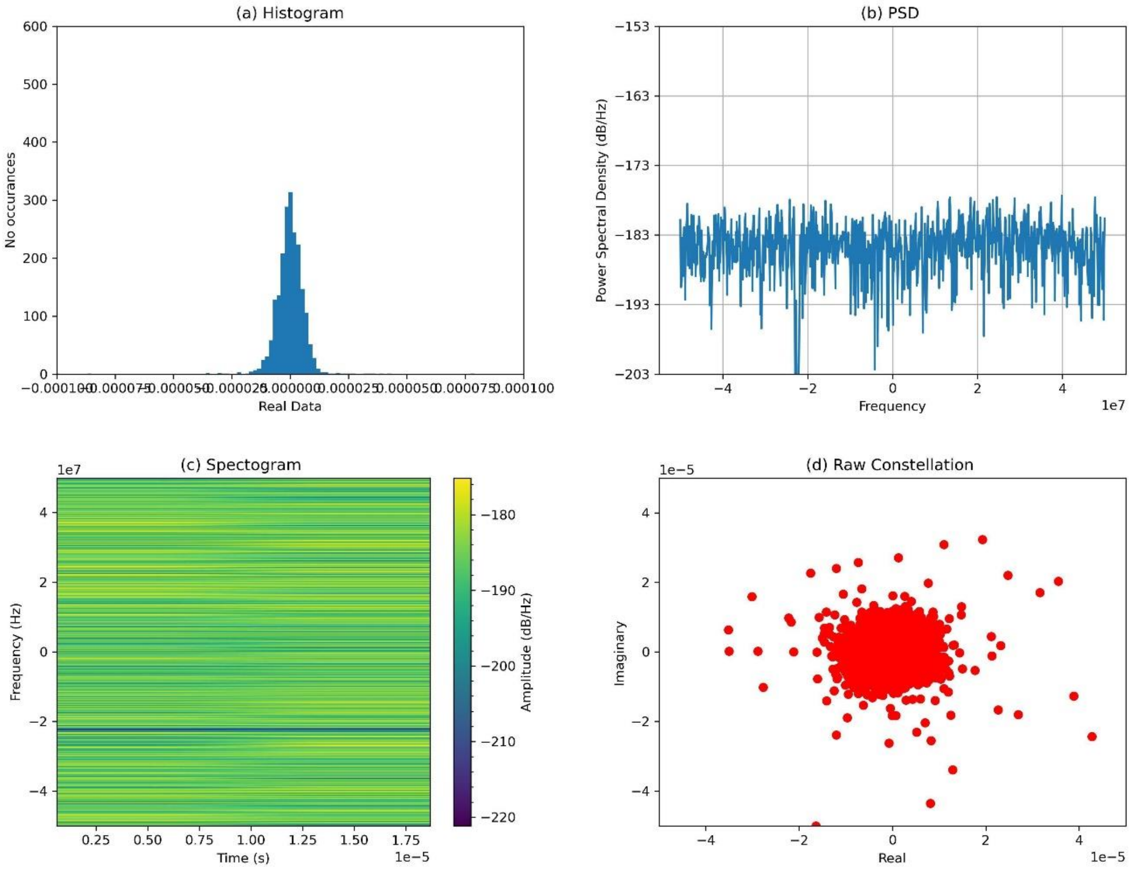


Figure 57 – DroneRF dataset in graphical signal representations (a) Histogram, (b) PSD, (c) Raw Constellation, (d) Spectrogram for class Bebop Mode 2 - hovering

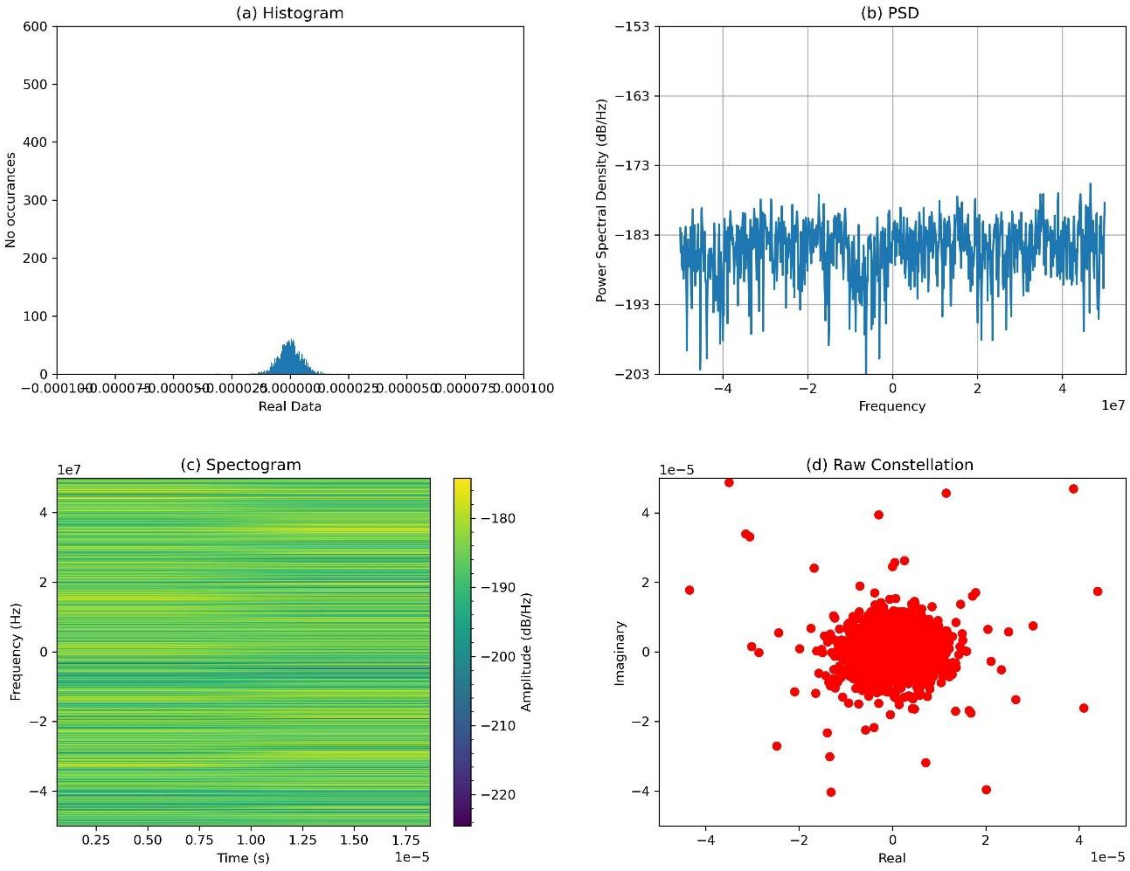


Figure 58 - DroneRF dataset in graphical signal representations (a) Histogram, (b) PSD, (c) Raw Constellation, (d) Spectrogram for class Bebop Mode 3 - flying without video

Figure 58 shows the Bebop flying without video and Figure 59 with video. The histogram (a) indicates an increase in activity with the video transmitting.

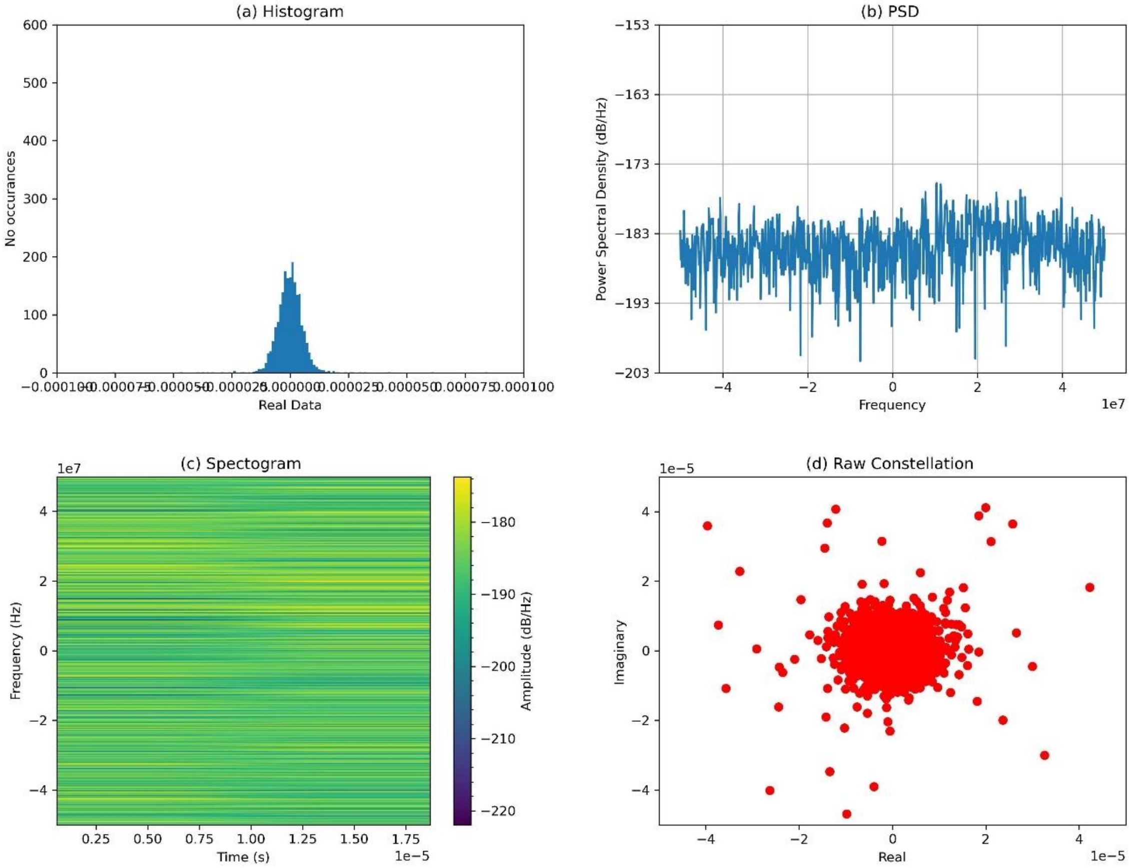


Figure 59 - DroneRF dataset in graphical signal representations (a) Histogram, (b) PSD, (c) Raw Constellation, (d) Spectrogram for class Bebop Mode 4 - flying with video

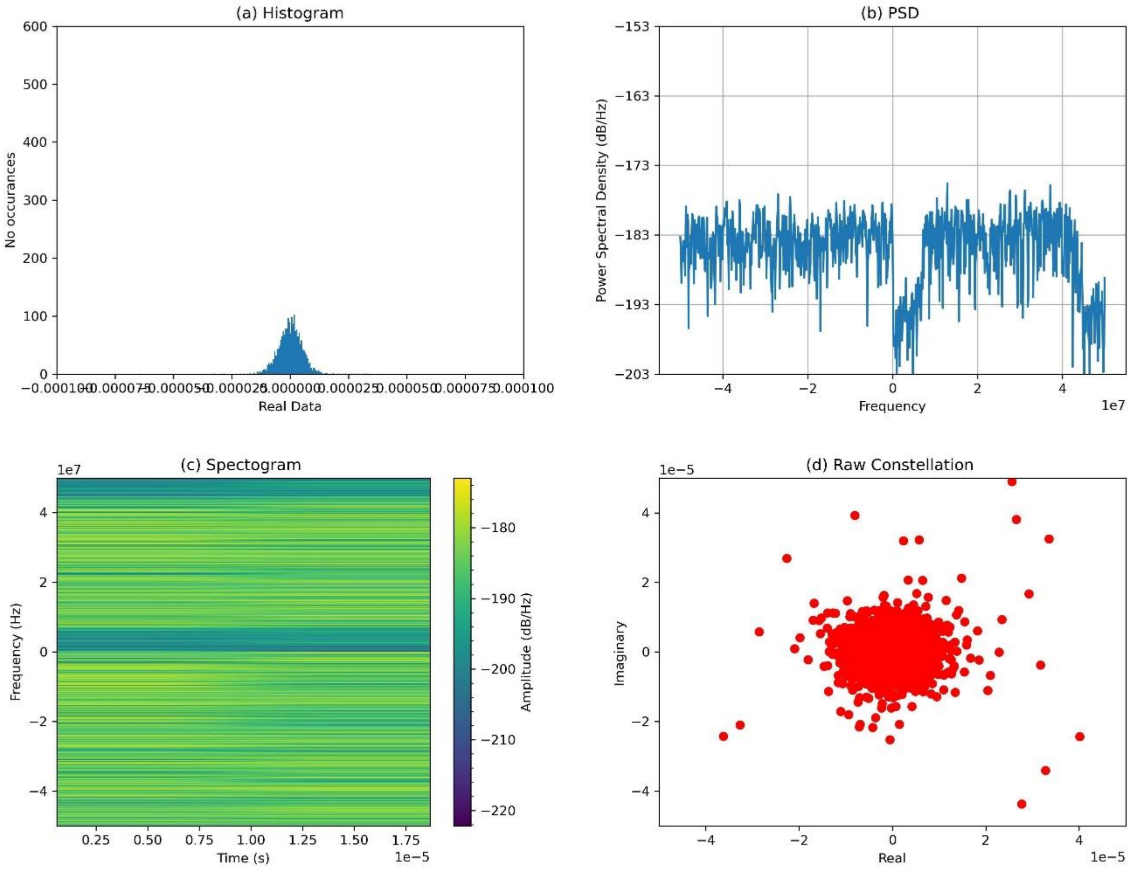


Figure 60 - DroneRF dataset in graphical signal representations (a) Histogram, (b) PSD, (c) Raw Constellation, (d) Spectrogram for class AR Mode 1 - switched on

Figure 60 shows the AR switched on and connected to the controller. The spectrogram (d) and PSD (b) show two clear bands of activity in the spectrum, one above and one below the centre frequency. Figure 61 shows the spectrum when the AR is hovering. If we compare this to Figure 57 where the Bebop is also hovering we can see a similar constant spread of activity across the entire spectrum but with a drop at the centre frequency.

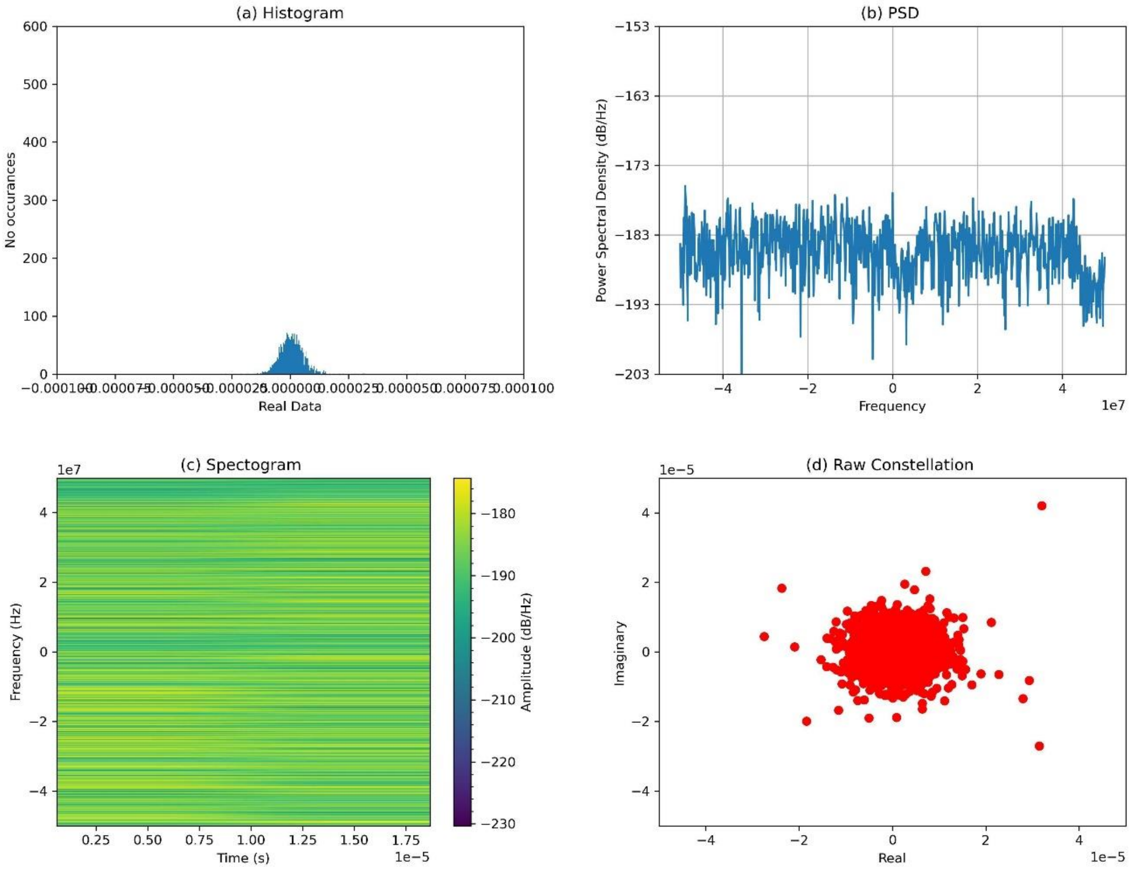


Figure 61 - DroneRF dataset in graphical signal representations (a) Histogram, (b) PSD, (c) Raw Constellation, (d) Spectrogram for class AR Mode 2 - hovering

Figure 62 and Figure 63 show the AR flying without and with video respectively. We can see in Figure 63 on the PSD (b) that the higher end of the spectrum increases in power by around 3dB when the video is present. We can also see a clear rise in the histogram (a) representation when the video feed is turned on.

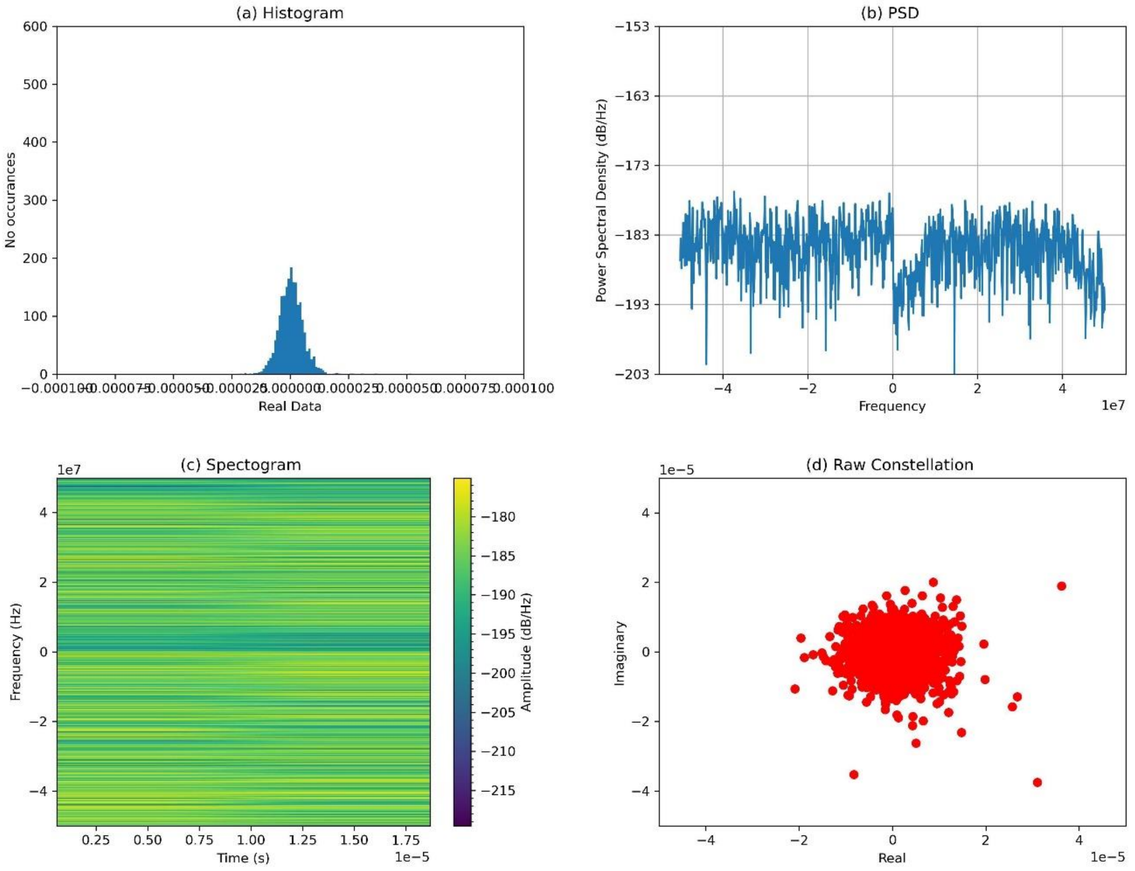


Figure 62 - DroneRF dataset in graphical signal representations (a) Histogram, (b) PSD, (c) Raw Constellation, (d) Spectrogram for class AR Mode 3 - flying without video

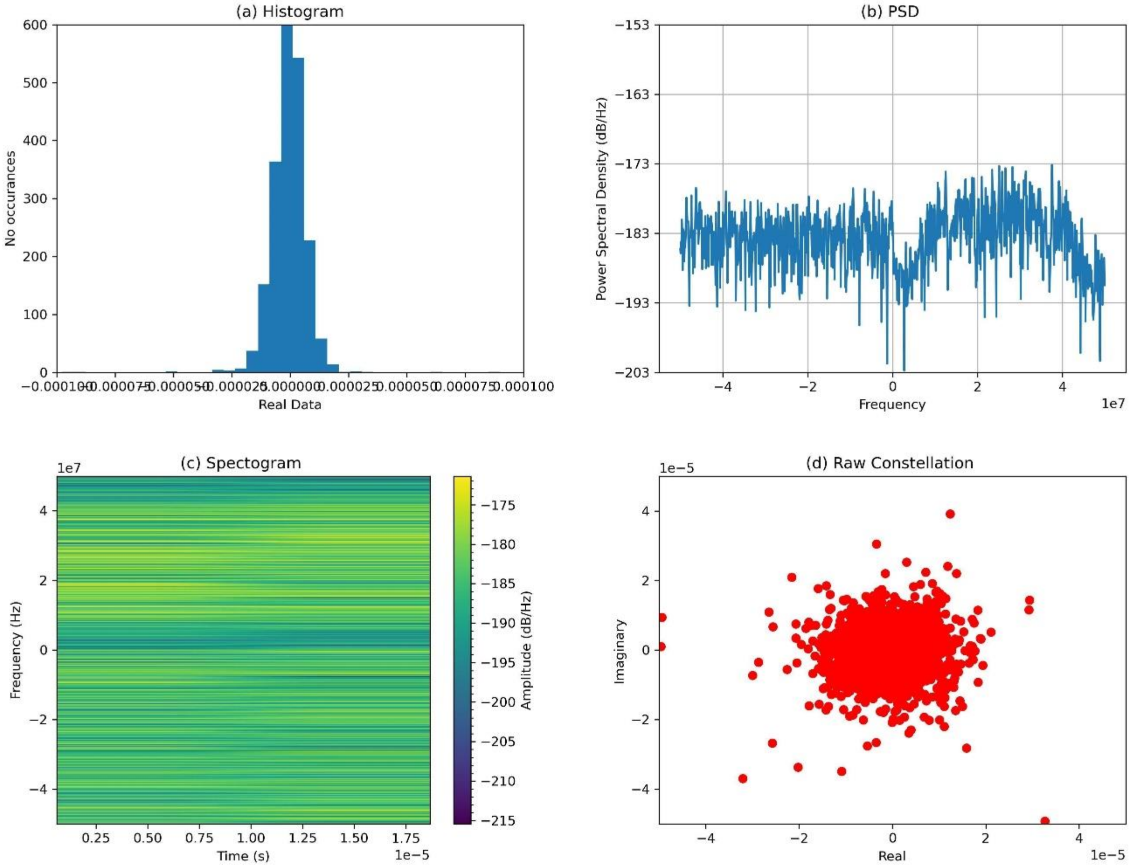


Figure 63 - DroneRF dataset in graphical signal representations (a) Histogram, (b) PSD, (c) Raw Constellation, (d) Spectrogram for class AR Mode 4 – Flying with video

Figure 64 shows the DJI Phantom turned on and connected to the controller. Comparing this to the Bebop in Figure 56 and the AR in Figure 60 we can observe that the Phantom has a more even spread of power across the entire spectrum.

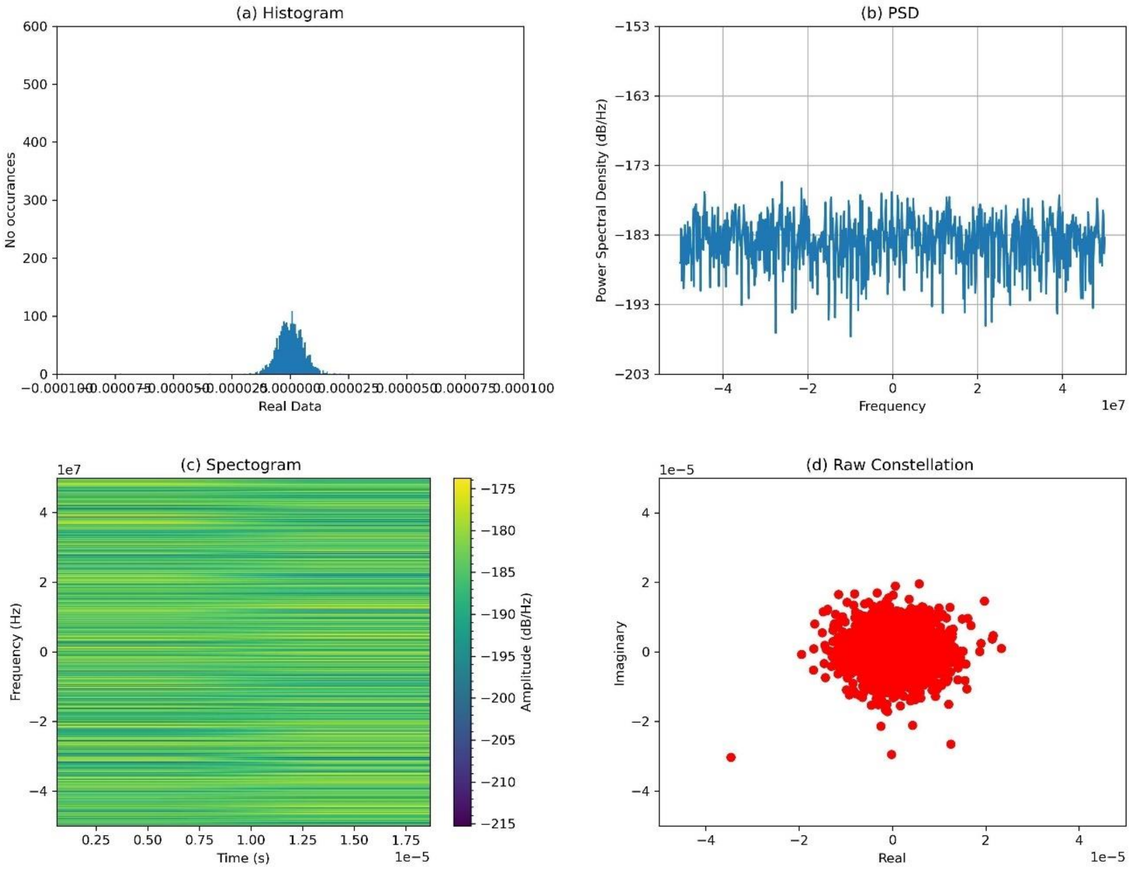


Figure 64 - DroneRF dataset in graphical signal representations (a) Histogram, (b) PSD, (c) Raw Constellation, (d) Spectrogram for class Phantom Mode 3 - switched on

In all of the figures it is hard to see any real pattern in the raw data changes, this may be due to the fact that we looking at so much of the frequency range at once. We need to consider the whole frequency range as we can’t be sure if a UAS will hop to a random Wi-Fi channel due to interference during operation.

Figure 65 shows the concatenated images as they are given to the CNN for feature extraction. Figure 65 shows an example of a concatenated image for each of the 6 classes of GPS Jammer types, marked (a) through to (f).

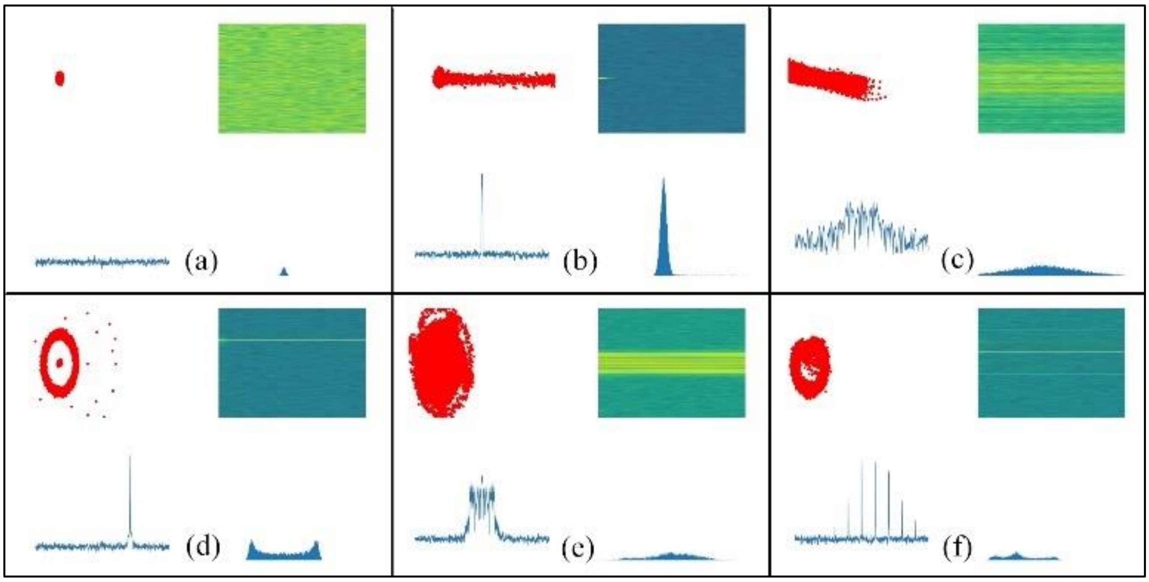


Figure 65 - Concatenated Dataset; (a) no jamming; (b) DME; (c) narrowband; (d) AM; (e) Chirp; (f) FM.

Figure 65 (a) shows the input to the CNN for the class no jamming signal present. Figure 65 (b) shows the input to the CNN for the class DME jammer. The graphs were created with no axis or label information, just the graph itself and then the 4 representations – spectrogram, histogram, PSD and raw constellation were stitched together using the functions we created for concatenation.

3.3 Wireless Interference

3.3.1 UAS Signals

For the UAS, signals were collected in the presence of Bluetooth and Wi-Fi signals to represent an environment whereby other signals were present in the same frequency band. The process by which this was achieved is described in section 3.1.4. Below the graphical signal representations will be considered for the spectrogram and the PSD.

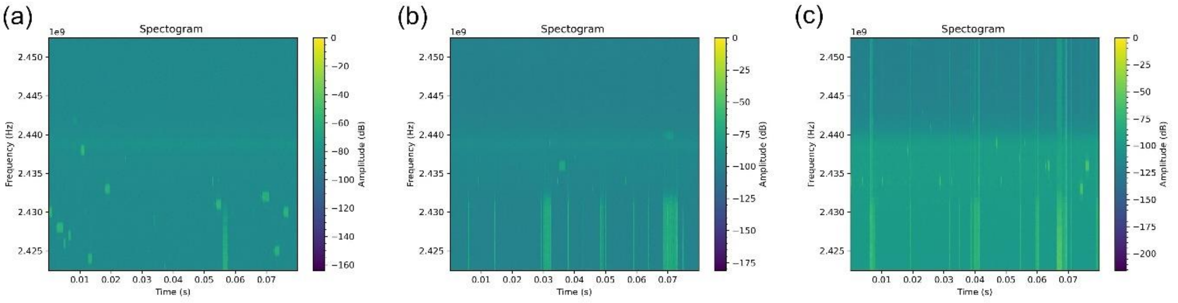


Figure 66 - Spectrogram Graphical Signal Representation for the classes (a) No UAS, (b) Parrot Disco Switched on and connected to the controller and (c) Parrot Disco flying with Interference

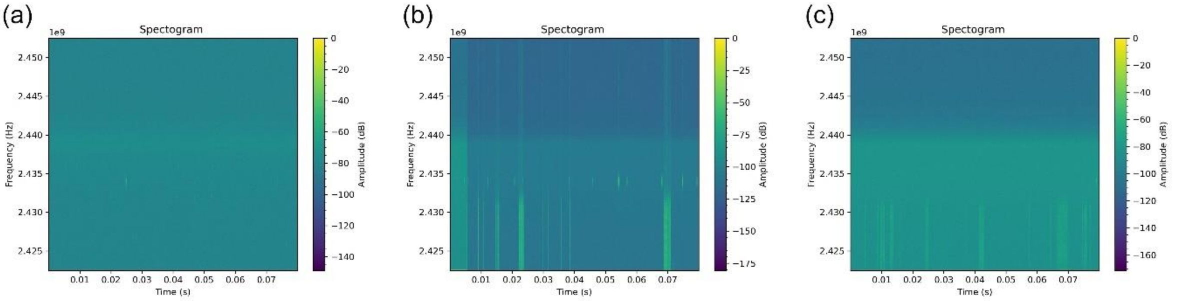


Figure 67 - Spectrogram Graphical Signal Representation for the classes (a) No UAS, (b) Parrot Disco Switched on and connected to the controller and (c) Parrot Disco flying no Interference

Figure 66 shows the spectrogram graphical signal representation for the classes no UAS (a), parrot disco switched on but not yet taken off (b) and disco flying (c). If Figure 66 is compared to Figure 67 which shows the no UAS and disco classes in a clean noise environment, the Bluetooth and Wi-Fi interference is clear on Figure 66. The Bluetooth signal is observed as the smaller yellow horizontal bursts of activity and the Wi-Fi as the thin vertical bursts occurring at around 9ms and 11.5ms. When the class disco switched on is consulted (b), it is impossible to tell with the human eye whether the Wi-Fi bursts on the spectrogram are coming from the interference or from the disco as it uses a version of the Wi-Fi protocol to communicate. There are no strong Bluetooth interference signals present within the switched on class (b), this is entirely possible as it is a burst protocol and there may not have been any activity within that 20ms time frame. Lastly on the spectrogram for the disco flying class (c) in Figure 66 a Bluetooth interference signal can be observed but again it is impossible to know visually whether the Wi-Fi signal we are observing are the parrot disco or the interference.

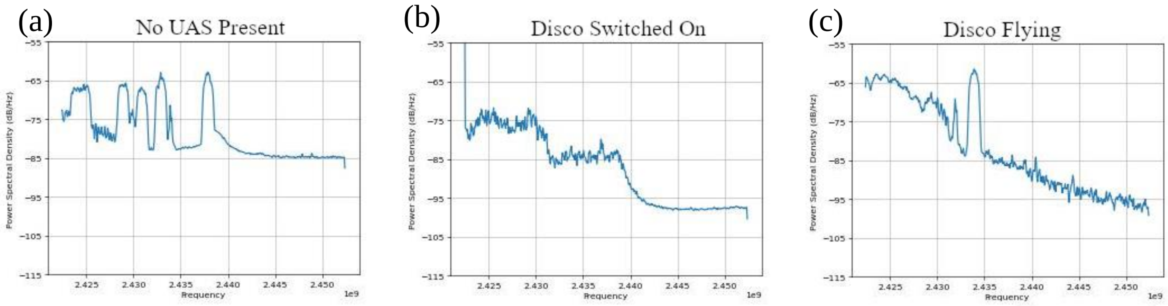


Figure 68 - PSD Graphical Signal Representation for the classes (a) No UAS, (b) Parrot Disco Switched on and connected to the controller and (c) Parrot Disco flying with Interference

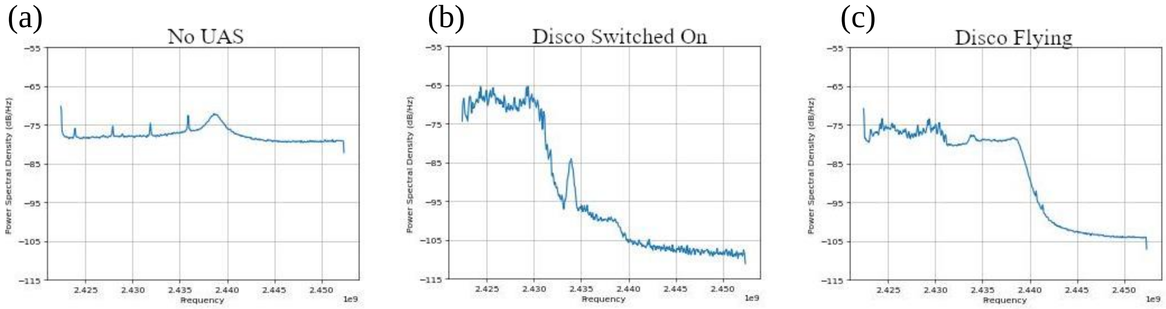


Figure 69 – PSD Graphical Signal Representation No UAS (a), Parrot Disco switched on and connected to the controller (b) and Parrot Disco Flying (c)

When Figure 68 is compared with Figure 69 we can visually observe the interference in the no UAS class (a) by the peaks present on the PSD in Figure 68. When comparing the disco switched on (b) and flying (c), both signal patterns are different, with a visual peak being displayed within Figure 68 for the flying class (c) around 2.434GHz. Figure 70 shows the spectrograms for the classes switched on (a), hovering (b) and flying (c) for the Air 2 S with interference from Bluetooth and Wi-Fi devices present. Compared to the clean environment in Figure 71 there is additional activity present in the bottom end of the spectrum for the class switched on (a) which visually appears as both Bluetooth and Wi-Fi. For the classes hovering (b) and flying (c) we observe additional activity especially in the lower end of the spectrum.

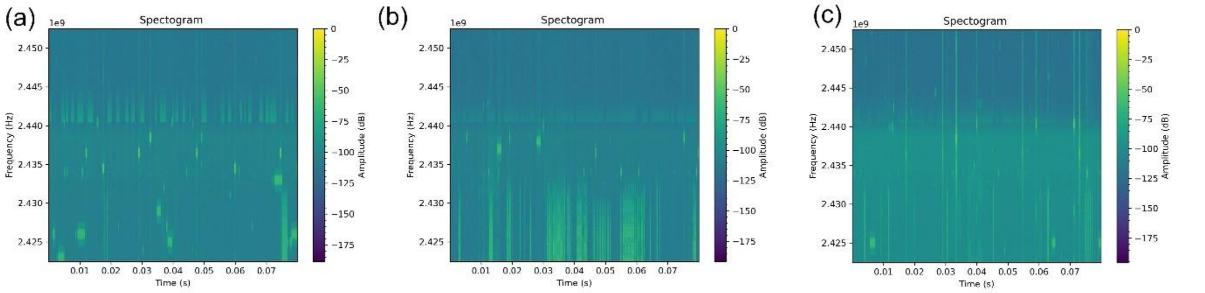


Figure 70 - Spectrogram Graphical Signal Representation for the classes (a) Air 2 S Switched on and connected to the controller, (b) Air 2 S hovering and (c) Air 2 S flying with Interference

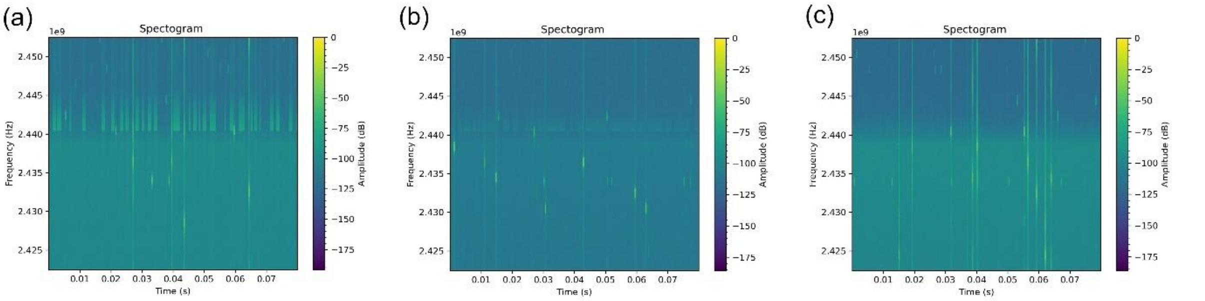


Figure 71 - Spectrogram Graphical Signal Representation for the classes (a) Air 2 S Switched on and connected to the controller, (b) Air 2 S hovering and (c) Air 2 S flying

Figure 72 shows the PSD for the Air 2 S classes switched on (a), hovering (b) and flying (c) from left to right respectively. Comparing Figure 72 to the clean environment in Figure 73 it is hard to visually distinguish the interference from the genuine UAS signal. However, the interference does produce a more jagged appearance of the PSD line in all three classes and the overall noise floor is increased.

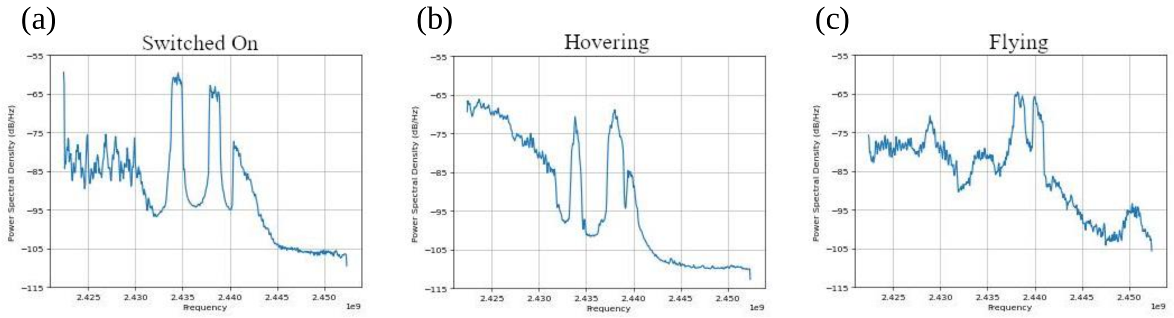


Figure 72 - PSD Graphical Signal Representation for the classes (a) Air 2 S Switched on and connected to the controller, (b) Air 2 S hovering and (c) Air 2 S flying with Interference

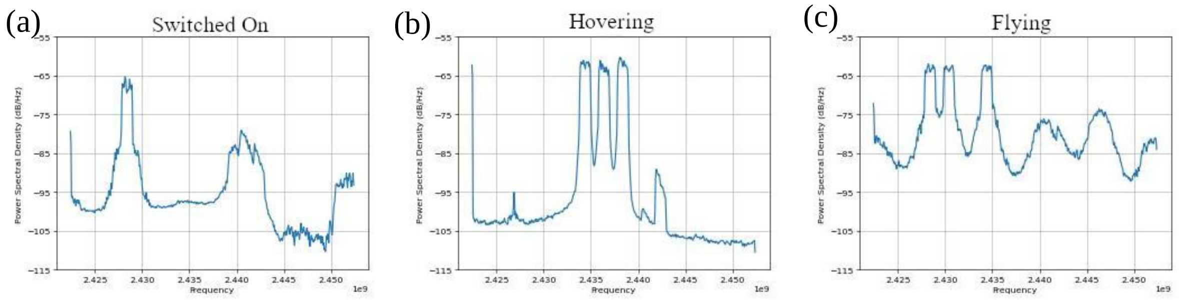


Figure 73 - PSD Graphical Signal Representation Air 2 S switched on and connected to the controller (a), Air 2 S hovering (b) and Air 2 S flying (c)

Figure 74 shows the PSD for the Inspire 2 for classes switched on (a), hovering (b) and flying (c). Compared with the clean environment in Figure 75 we observe an overall increase in noise, denoted by the background appearance of the spectrogram as greenish yellow in Figure 74 and more of a blue variant in Figure 75. While the platforms which operate with the Wi-Fi protocol are indistinguishable from the interference Wi-Fi, the Ocusync and Lightbridge based platform signals appear more like Bluetooth signals. In the classes hovering (b) and flying (c), it is impossible to say whether the Bluetooth type signals are the interference or the UAS when comparing Figure 74 with Figure 75.

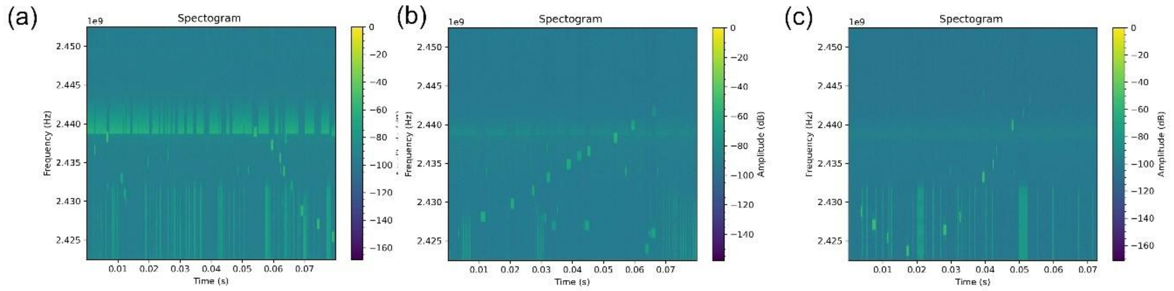


Figure 74 - Spectrogram Graphical Signal Representation for the classes (a) Inspire 2 Switched on and connected to the controller, (b) Inspire 2 hovering and (c) Inspire 2 flying with Interference

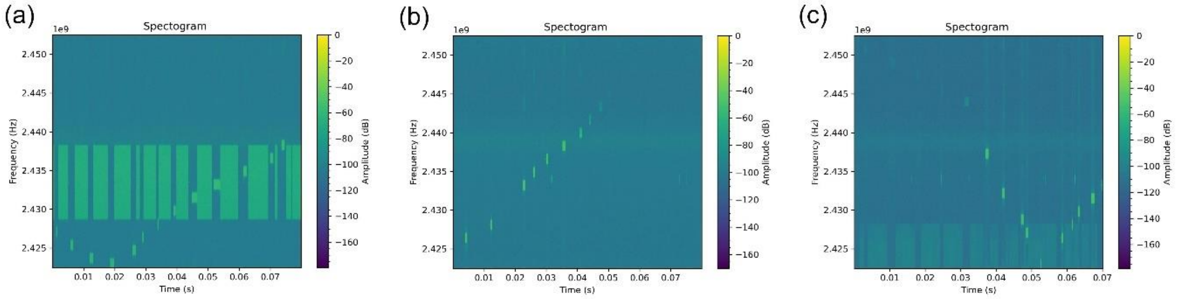


Figure 75 - Spectrogram Graphical Signal Representation for the classes (a) Inspire 2 Switched on and connected to the controller, (b) Inspire 2 hovering and (c) Inspire 2 flying

Figure 76 shows the PSD representation for the Inspire 2 switched on (a), hovering (b) and flying (c) in the presence of Bluetooth and Wi-Fi interference. When compared to the PSD without interference in Figure 77, again the noise floor is raised and the signals appear to have a jagged appearance as an indication of the interference being present.

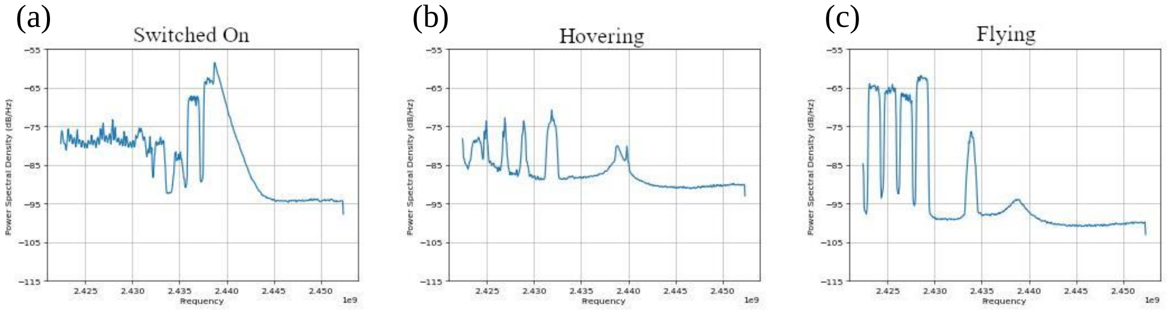


Figure 76 - PSD Graphical Signal Representation for the classes (a) Inspire 2 Switched on and connected to the controller, (b) Inspire 2 hovering and (c) Inspire 2 flying with Interference

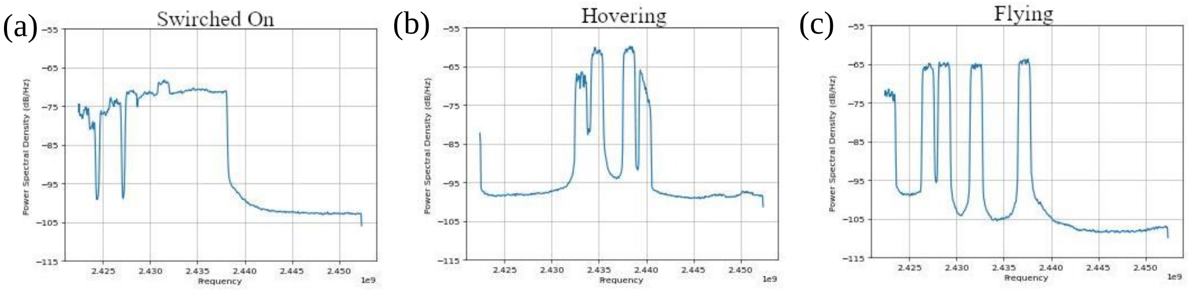


Figure 77 - PSD Graphical Signal Representation Inspire 2 switched on and connected to the controller (a), Inspire 2 hovering (b) and Inspire 2 flying (c)

Figure 78 shows the spectrogram for the Mavic Pro with interference for the classes switched on (a), hovering (b) and flying (c). When compared to the clean environment in Figure 79 it would be impossible to say visually which of the yellow bursts were interference sources and which were the UAS itself. However, just because the patterns and difference between the two cannot be easily distinguished with the human eye does not mean they are not there.

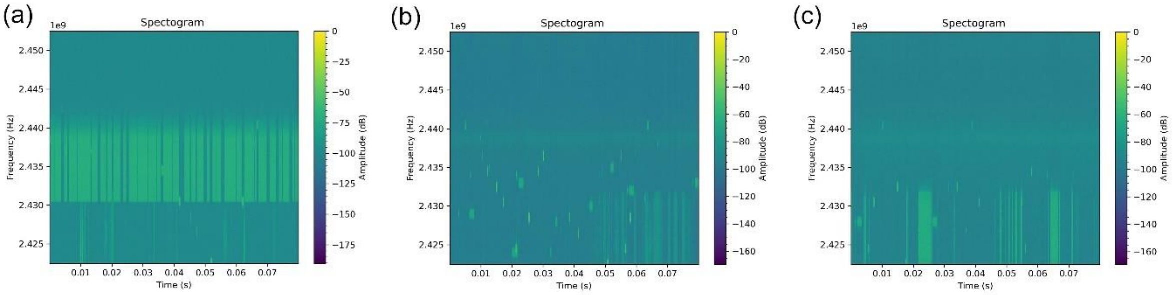


Figure 78 - Spectrogram Graphical Signal Representation for the classes (a) Mavic Pro Switched on and connected to the controller, (b) Mavic Pro hovering and (c) Mavic Pro flying with Interference

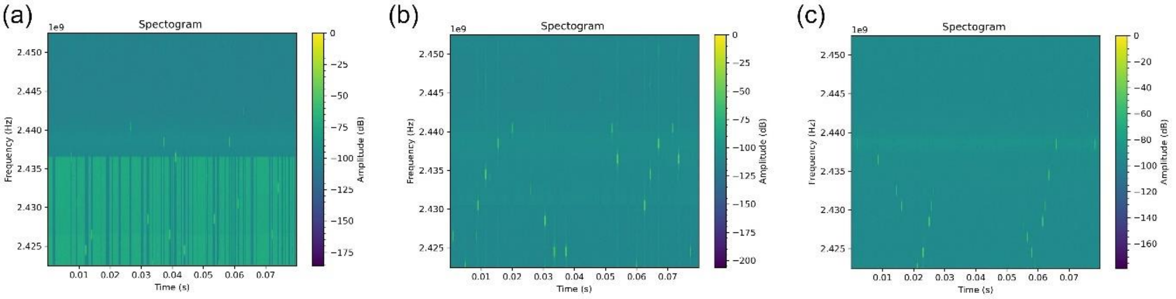


Figure 79 - Spectrogram Graphical Signal Representation for the classes (a) Mavic Pro Switched on and connected to the controller, (b) Mavic Pro hovering and (c) Mavic Pro flying

Figure 80 shows the PSD for the Mavic Pro 2 with interference for the classes switched on (a), hovering (b) and flying (c). When compared to the clean environment in Figure 81 the interference can be clearly observed in all three classes with the increase in peaks across the frequency range. It again is not easily distinguishable with the human eye as to which peaks are caused by the interference and which are caused by the UAS. Even when we compare back to the no UAS class with interference in Figure 68

Figure 68 - PSD Graphical Signal Representation for the classes (a) No UAS, (b) Parrot Disco Switched on and connected to the controller and (c) Parrot Disco flying with Interference, the only visual indication of the interference is the increase in the amount of peaks across the frequency band.

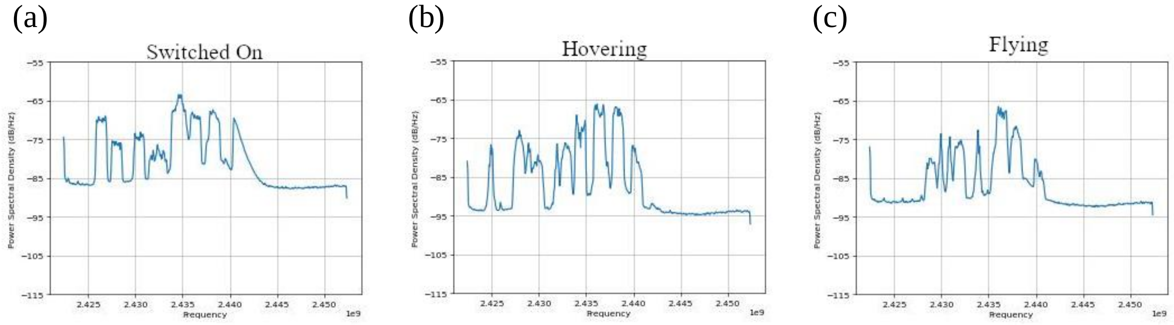


Figure 80 - PSD Graphical Signal Representation for the classes (a) Mavic Pro Switched on and connected to the controller, (b) Mavic Pro hovering and (c) Mavic Pro flying with Interference

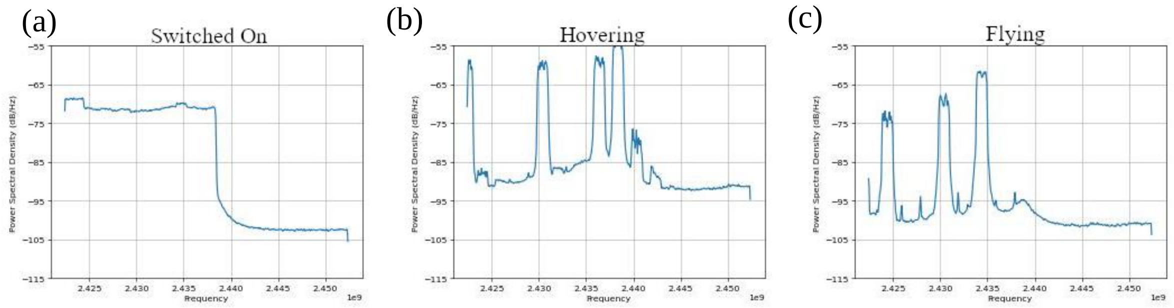


Figure 81 - PSD Graphical Signal Representation Mavic Pro switched on and connected to the controller (a), Mavic Pro hovering (b) and Mavic Pro flying (c)

Figure 82 shows the spectrogram for the Mavic Pro 2 with interference for the classes switched on (a), hovering (b) and flying (c). When compared to the clean environment in

Figure 83 it is difficult to visually detect the difference in pattern. There is a visible increase of activity within the hovering class (b) when the interference is added.

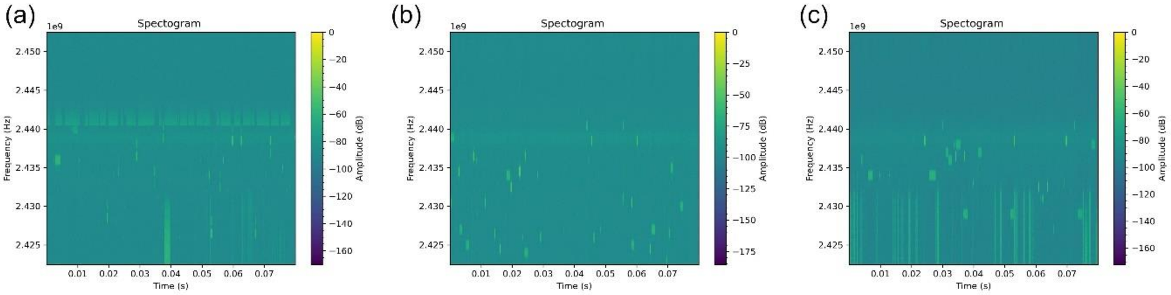


Figure 82 - Spectrogram Graphical Signal Representation for the classes (a) Mavic Pro 2 Switched on and connected to the controller, (b) Mavic Pro 2 hovering and (c) Mavic Pro 2 flying with Interference

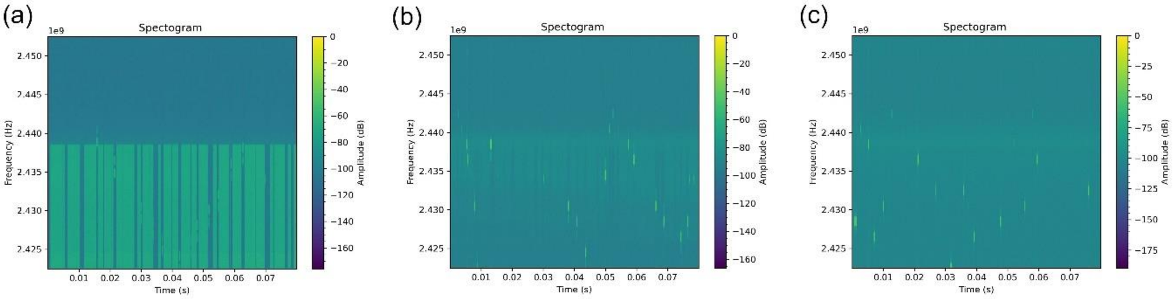


Figure 83 - Spectrogram Graphical Signal Representation for the classes (a) Mavic Pro 2 Switched on and connected to the controller, (b) Mavic Pro 2 hovering and (c) Mavic Pro 2 flying

Figure 84 shows the PSD for the Mavic Pro 2 with interference for the classes switched on (a), hovering (b) and flying (c). When compared to the clean environment in Figure 38 it is easier to observe that there is interference present by the increased number of peaks in all three classes. This is easier to observe in general visually when compared to the spectrograms in Figure 82 for each of the classes switched on (a), hovering (b) and flying (c).

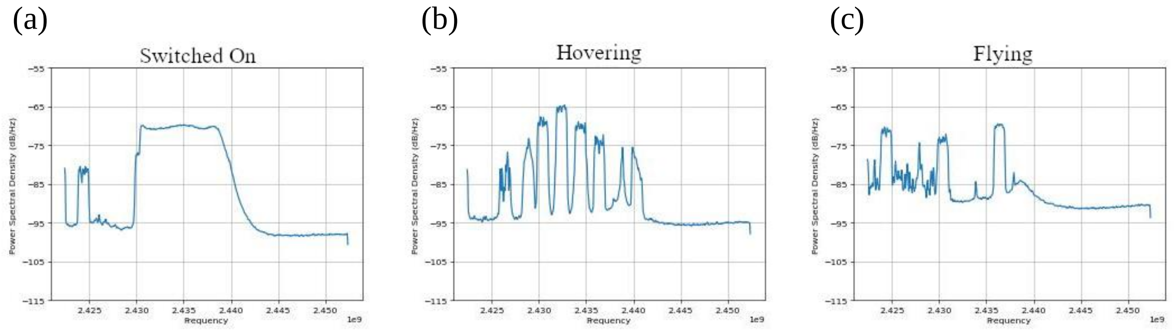


Figure 84 - PSD Graphical Signal Representation for the classes (a) Mavic Pro 2 Switched on and connected to the controller, (b) Mavic Pro 2 hovering and (c) Mavic Pro 2 flying with Interference

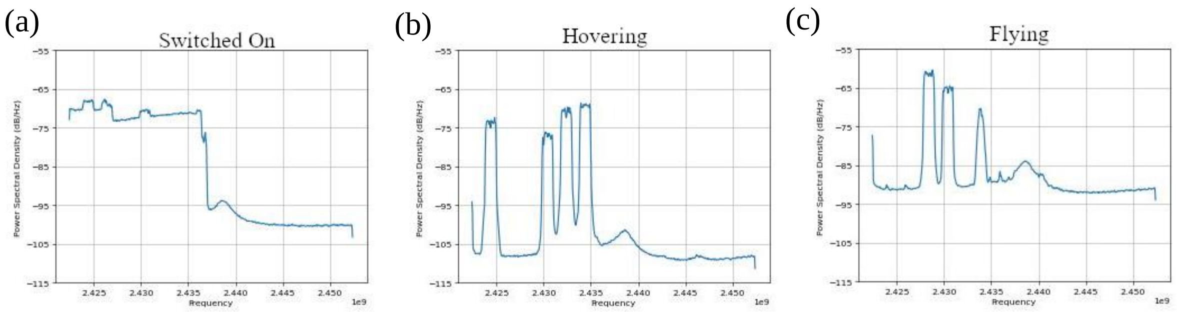


Figure 85 - PSD Graphical Signal Representation Mavic Pro 2 switched on and connected to the controller (a), Mavic Pro 2 hovering (b) and Mavic Pro 2 flying (c)

Figure 86 shows the Spectrogram for the Mavic Mini with interference for the classes switched on (a), hovering (b) and flying (c). When compared to the clean environment in Figure 87 the Bluetooth interference can be isolated visually in Figure 86 within all three classes but the Wi-Fi would be difficult to distinguish between interference and the Mavic Mini as it uses a Wi-Fi based protocol.

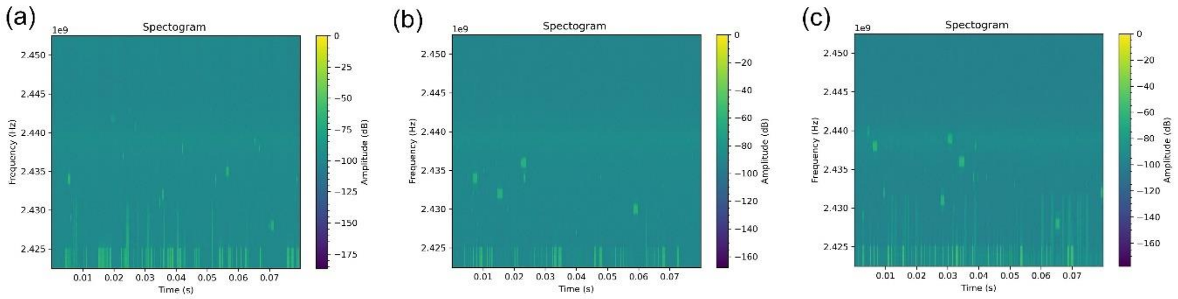


Figure 86 - Spectrogram Graphical Signal Representation for the classes (a) Mavic Mini Switched on and connected to the controller, (b) Mavic Mini hovering and (c) Mavic Mini flying with Interference

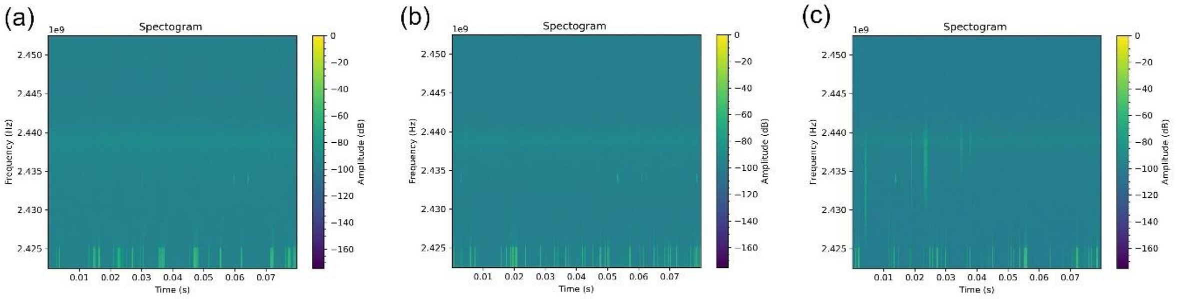


Figure 87 - Spectrogram Graphical Signal Representation for the classes (a) Mavic Mini Switched on and connected to the controller, (b) Mavic Mini hovering and (c) Mavic Mini flying

Figure 88 shows the PSD for the Mavic Mini with interference for the classes switched on (a), hovering (b) and flying (c). When compared to the clean environment in Figure 89 we can visually observe the difference the interference causes in each class by the increased number of peaks across the frequency range. In general terms both Figure 88 and Figure 89 show a general decreasing slope from left to right but with differing peaks occurring within the descent.

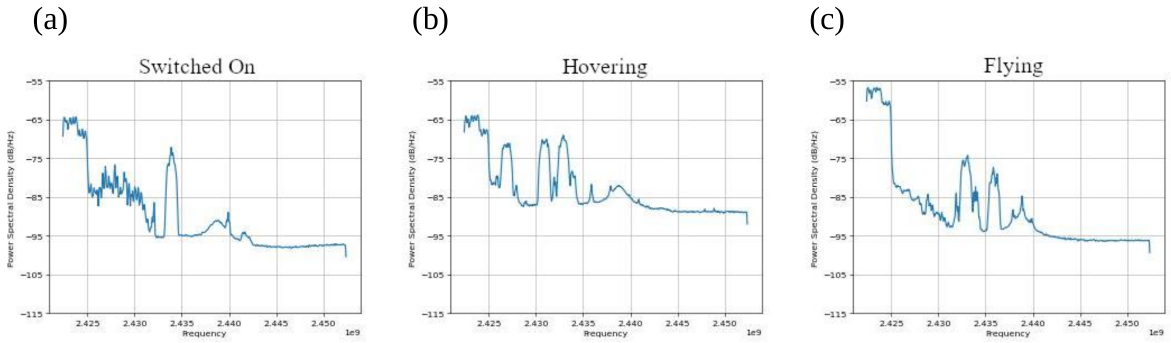


Figure 88 - PSD Graphical Signal Representation for the classes (a) Mavic Mini Switched on and connected to the controller, (b) Mavic Mini hovering and (c) Mavic Mini flying with Interference

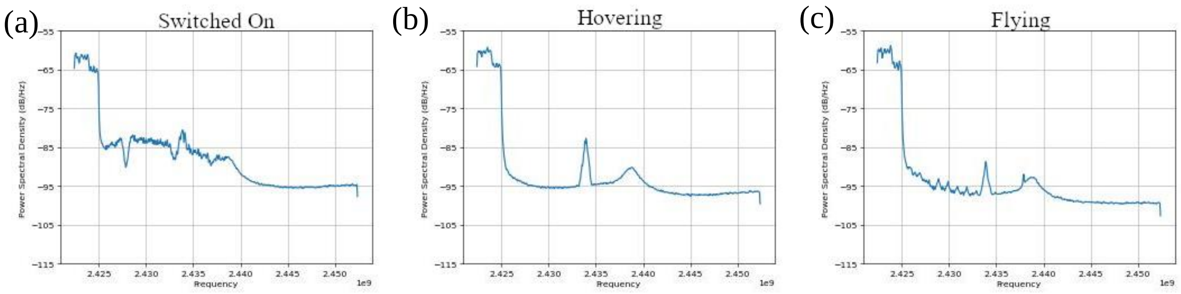


Figure 89 - PSD Graphical Signal Representation Mavic Mini switched on and connected to the controller (a), Mavic Mini hovering (b) and Mavic Mini flying (c)

Figure 90 shows the Spectrogram for the Phantom 4 with interference for the classes switched on (a), hovering (b) and flying (c). When compared to the clean environment Figure 91 we can visually observe the addition of Wi-Fi interference in Figure 90. However, as with the other Lightbridge and OcuSync platforms it is hard to tell which of the small horizontal bursts of yellow activity are Bluetooth interference and which are the Phantom 4. This is irrespective of the class switched on (a), hovering (b) or flying (c).

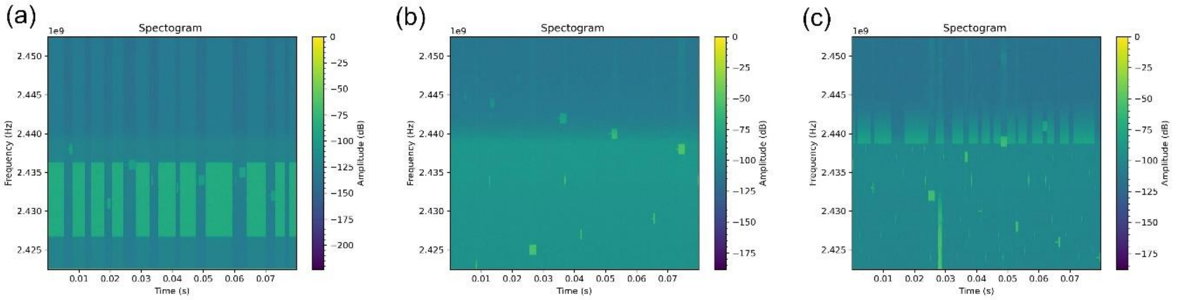


Figure 90 - Spectrogram Graphical Signal Representation for the classes (a) Phantom 4 Switched on and connected to the controller, (b) Phantom 4 hovering and (c) Phantom 4 flying with Interference

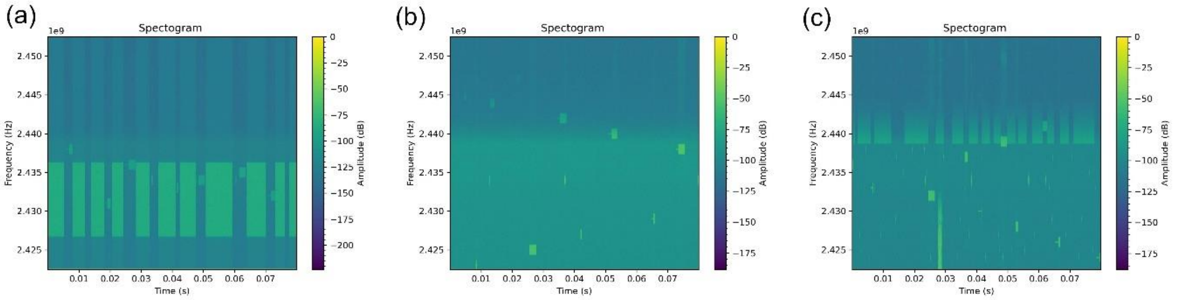


Figure 91 - Spectrogram Graphical Signal Representation for the classes (a) Phantom 4 Switched on and connected to the controller, (b) Phantom 4 hovering and (c) Phantom 4 flying

Lastly Figure 92 shows the PSD for the Phantom 4 with interference for the classes switched on (a), hovering (b) and flying (c). When compared to the clean environment Figure 93 the interference again displays itself as a general increase in the noise floor and with a jagged appearance to the signal in all three classes switched on (a); hovering (b) and flying (c).

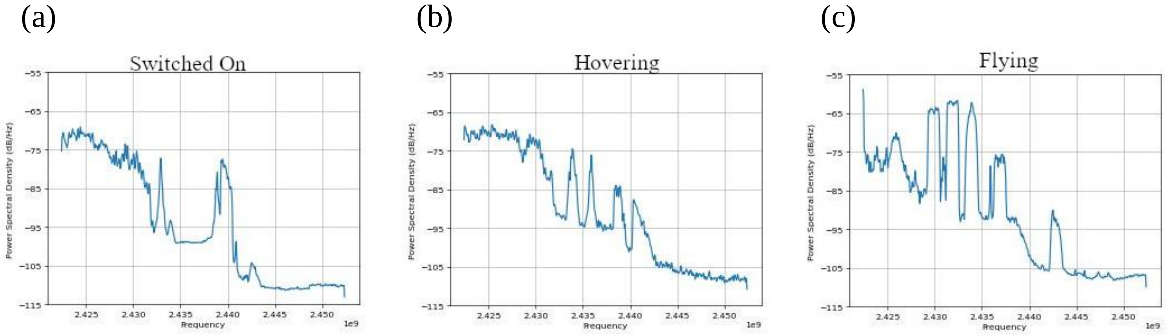


Figure 92 - PSD Graphical Signal Representation for the classes (a) Phantom 4 Switched on and connected to the controller, (b) Phantom 4 hovering and (c) Phantom 4 flying with Interference

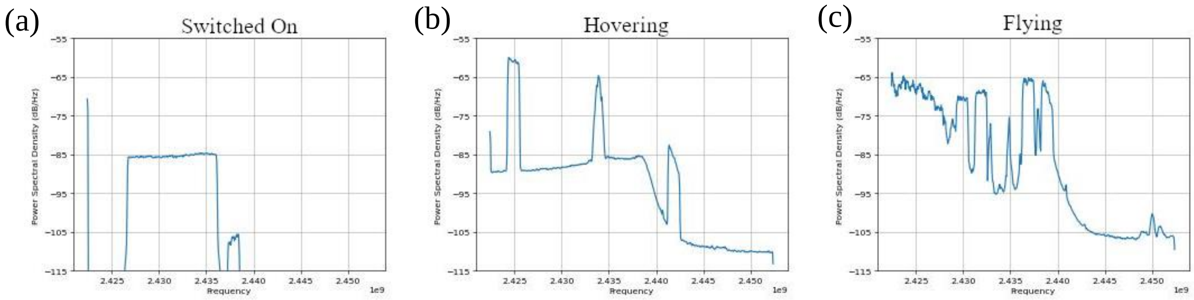


Figure 93 - PSD Graphical Signal Representation Phantom 4 switched on and connected to the controller (a), Phantom 4 hovering (b) and Phantom 4 flying (c)

3.3.2 GPS Jamming Signals

GPS Jamming Dataset

Figure 94 to Figure 97 shows the graphical signal representations for AM jamming signals for 30, 10, -10 and -20dB SNR respectively so the addition of the SNR changes can be observed.

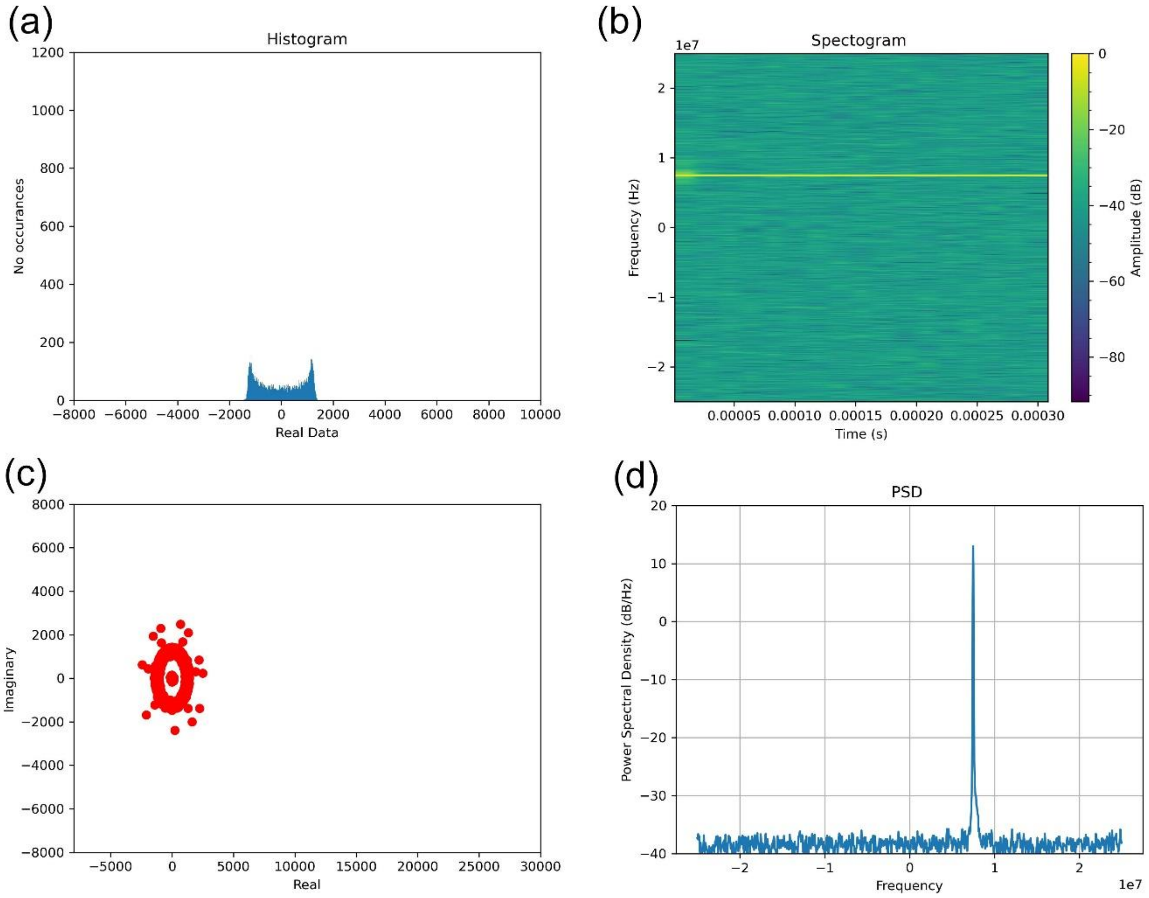


Figure 94 - AM jamming signal present at 30dB SNR; PSD (top left); spectrogram (top right); raw constellation (bottom left); histogram (bottom right).

In Figure 94 at 30dB SNR the AM jamming signal can be observed clearly in all signal representations by comparing the individual graphs with those seen in **Error! Reference source not found.**. If we compare Figure 94 with the corresponding PSD in Figure 44, spectrogram in Figure 32 and raw IQ scatter plot and histogram in Figure 49 for the GPS Jamming Dataset we can see the signals observe the same pattern but with less noise.

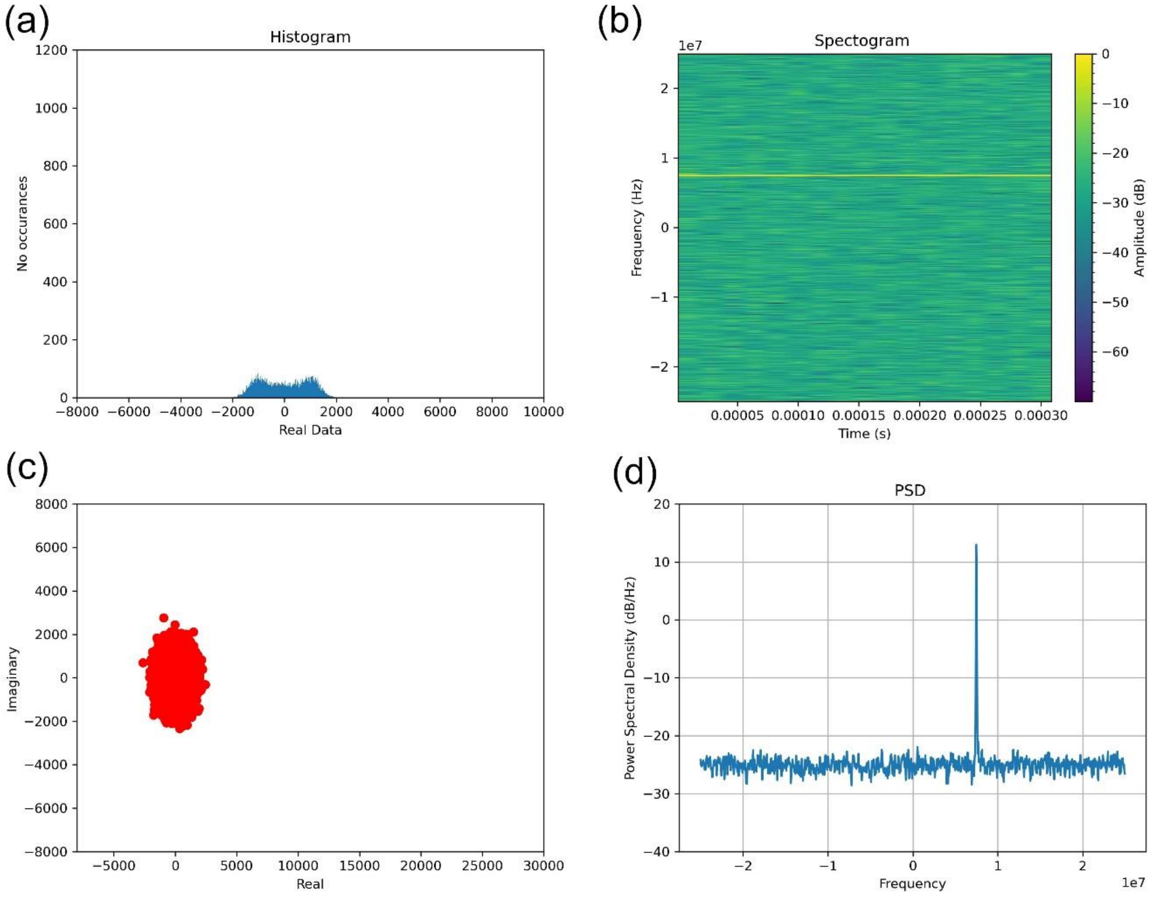


Figure 95 - AM jamming signal present at 10dB SNR; PSD (top left); spectrogram (top right); raw constellation (bottom left); histogram (bottom right).

In Figure 95 the SNR is decreased from 30 to 10dB. We observe the noise floor raised in the PSD on the top left and the noise increase in the spectrogram by the change in colour from blue to green/blue compared to Figure 94. On the raw constellation the red circle fills in and we see a slightly wider spread of real data on the histogram. If we also compare back with the corresponding PSD in Figure 44, spectrogram in Figure 32 and raw IQ scatter plot and histogram in Figure 49 for the GPS Jamming Dataset we can see the signals observe the same differences when comparing SNR 30dB in Figure 94.

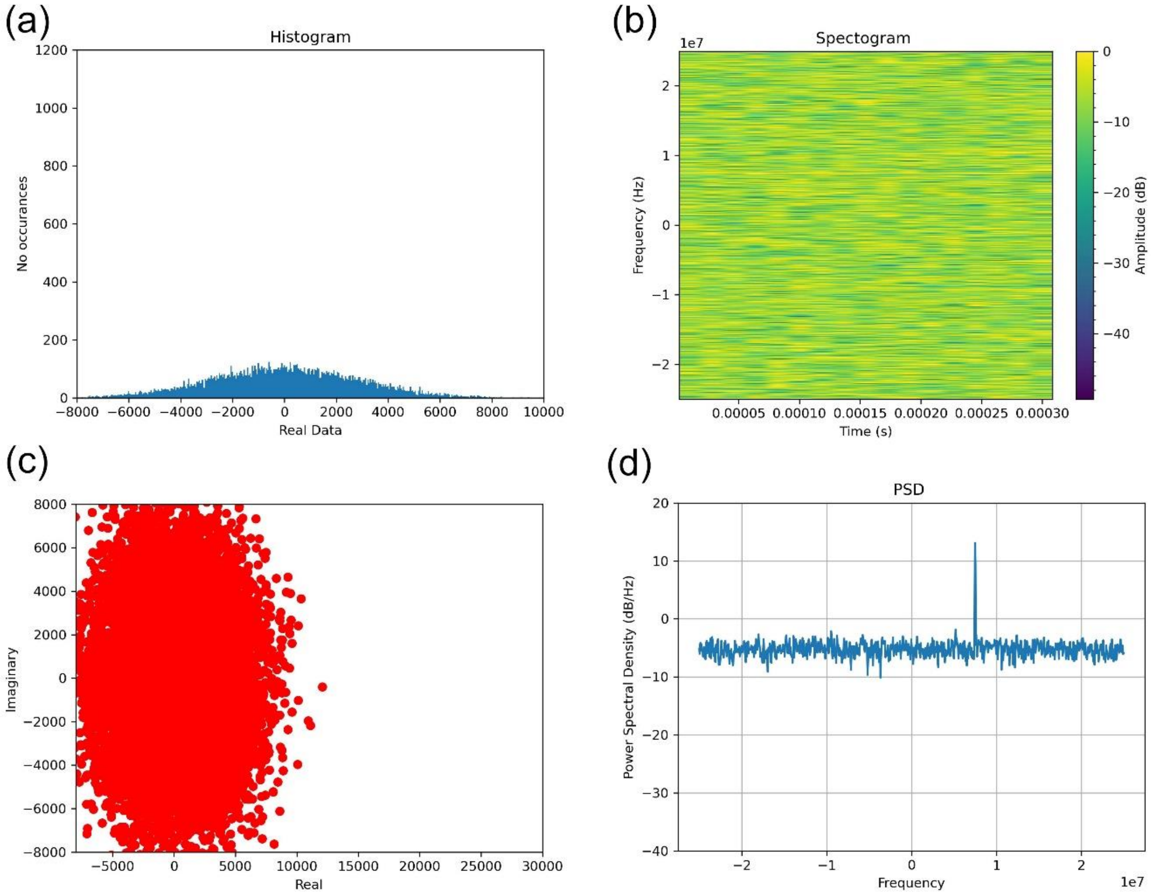


Figure 96 - AM jamming signal present at -10dB SNR; PSD (top left); spectrogram (top right); raw constellation (bottom left); histogram (bottom right).

In Figure 96 the SNR decreases to -10dB and we observe the noise floor increase from 18dB/Hz to -2dB/Hz on the PSD from Figure 95 to Figure 96. The background on the spectrogram turns more yellow to represent the increase in noise, more red is present on the raw constellation and again the spread of the histogram real data grows across the entire spectrum range. We are starting to lose the signals within the noise. However the signal pattern can still be observed clearly in the PSD with the peak remaining around 18dB/Hz higher than the noise floor. The signal can be seen visually in the spectrogram but it is faint in comparison to the background noise level. The pattern of the original raw IQ plot (oval shape) and the pattern in the original histogram is visually lost within the noise.

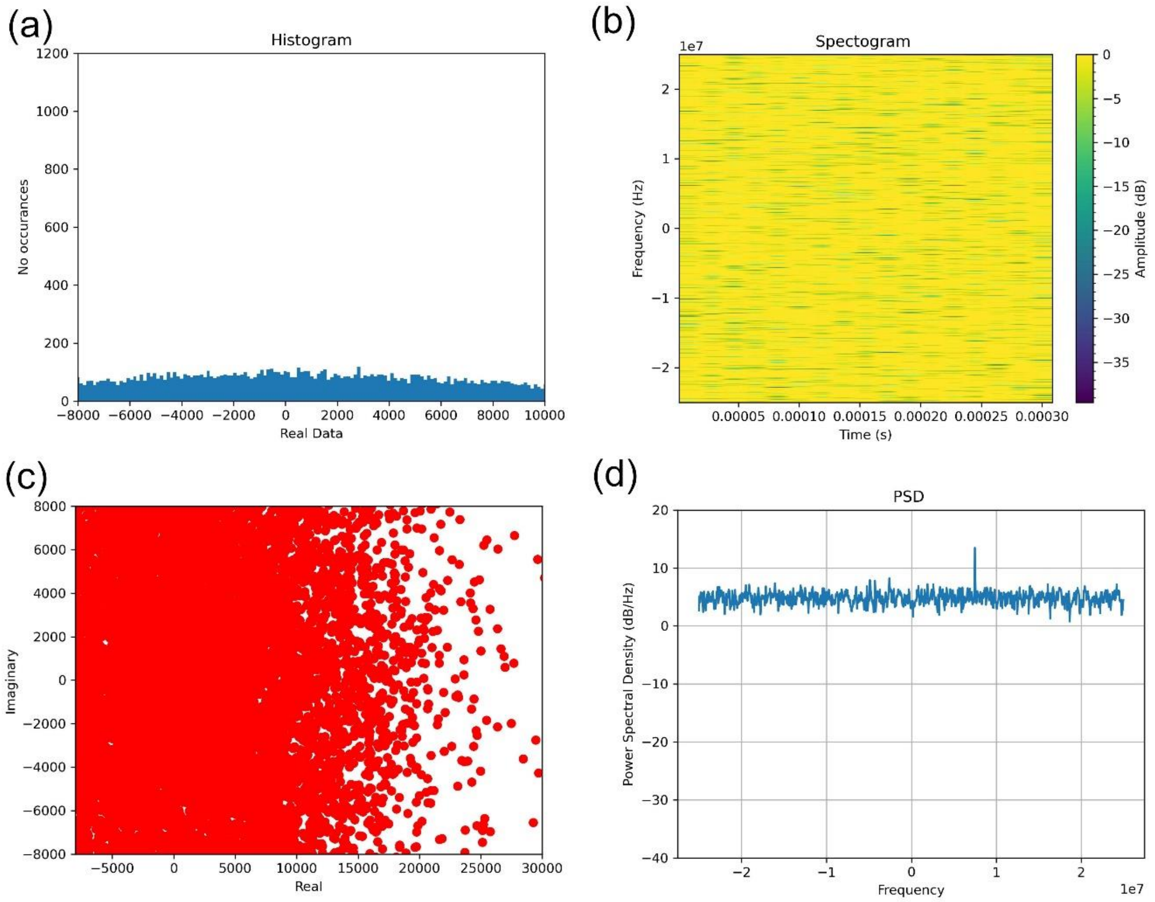


Figure 97 - AM jamming signal present at -20dB SNR; PSD (top left); spectrogram (top right); raw constellation (bottom left); histogram (bottom right).

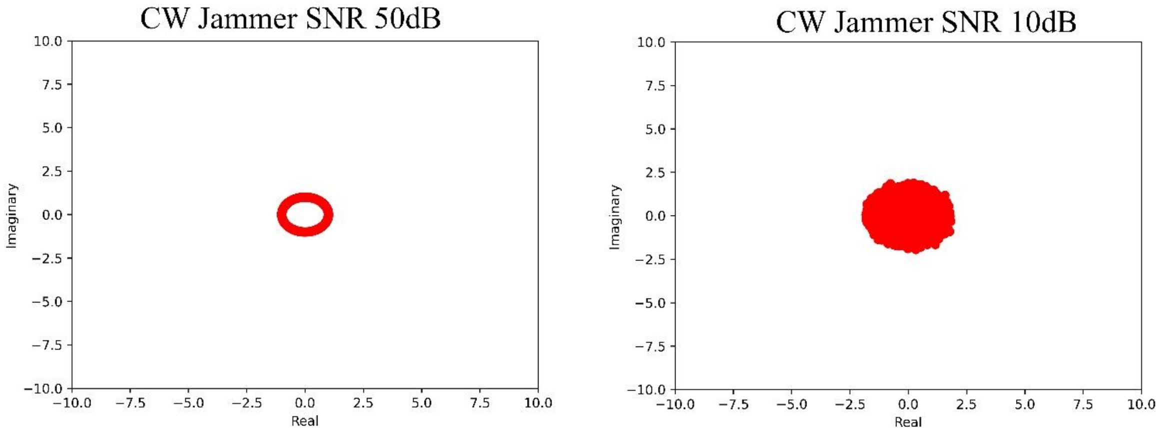
In Figure 97 the SNR decreases to -20dB and we observe the noise floor increase to 10dB/Hz on the PSD from Figure 96 to Figure 97. The background on the spectrogram turns more yellow to the point where our AM signal cannot be visually detected with the eye. The raw constellation fills in with a greater proportion of red representing the noise and the spread of the histogram becomes even. The signal is lost within the noise and is not visible within the spectrogram, raw IQ plot and within the histogram. The only plot to preserve the signal visually is the PSD where we can observe the signal peaking up at around 8-9dB/Hz above the noise floor. So that the effect of SNR could be tested fully, a dataset was produced for

SNR levels 30dB, 10dB, -10dB and -20dB where each class was made up of 1000 images 224x224 pixels.

JamDetect Dataset [23]

For the interference experiments using the JamDetect dataset, image datasets were created for SNR 50dB, 30dB, 10dB, -10dB and -20dB to understand the effect on classification accuracy. A PSD was used to represent the signal in the frequency domain, displaying the power distribution over the frequency range. A spectrogram was used to consider the signal in the time domain. Both representations were plotted in Matplotlib Python 3 with an FFT size of 1024 and a Hanning windowing function. A raw constellation is plotted with real and imaginary parts of the signal on the x and y axis respectively and a histogram representation of the real part of the signal over 500 bins. Lastly a concatenated representation onto one image is created using python to include a dataset which contains all 4 graphical signal representations. Datasets were created of 1000 images of size 224x224.

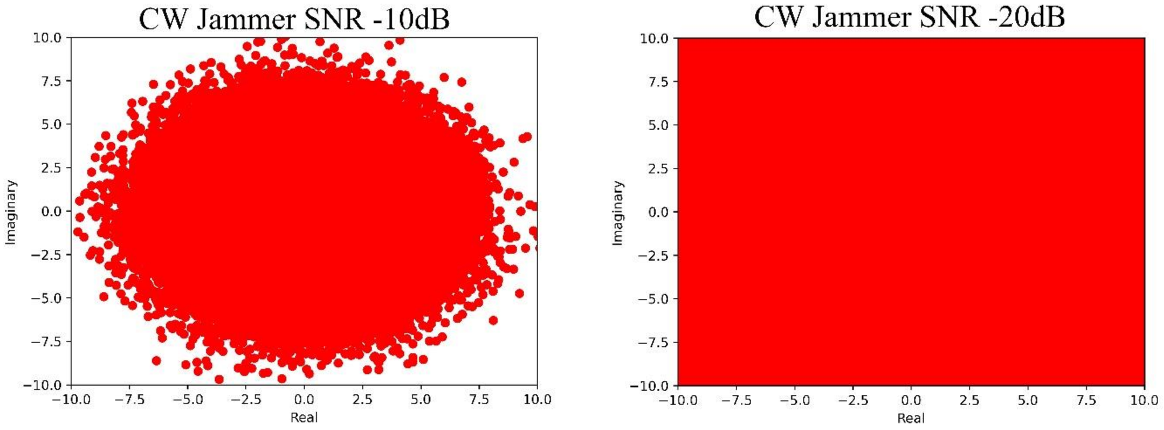
First the CW jamming signal at a SNR of 50dB are observed in Figure 98 (left). The jamming signal is clearly present on the raw constellation graphical representation.


CW Jammer SNR 50dB

CW Jammer SNR 10dB

Figure 98 - JamDetect CW Jammer SNR 50dB (left) SNR 10dB (right).

Figure 98 (right) shows the same CW jamming signal but after dropping the SNR to 10dB. The constellation in red fills in and increases in size compared to Figure 98 (left).


CW Jammer SNR -10dB

CW Jammer SNR -20dB

Figure 99 - JamDetect CW Jammer SNR -10dB (left) SNR -20dB (right).

In Figure 99 (left) the SNR ratio is dropped to -10dB and again the red constellation increases in size. In Figure 99 (right) the CW jamming signal with a SNR of -20dB is observed and the whole plot is covered in red due to the increase of noise.

Chapter 4 - Supervised Learning

The overall process followed for the supervised learning is shown in Figure 100 below. The previous section described the process for dataset use/generation, representing the signal as an image in the form of a graphical signal representation and the introduction of interference in the frequency band.

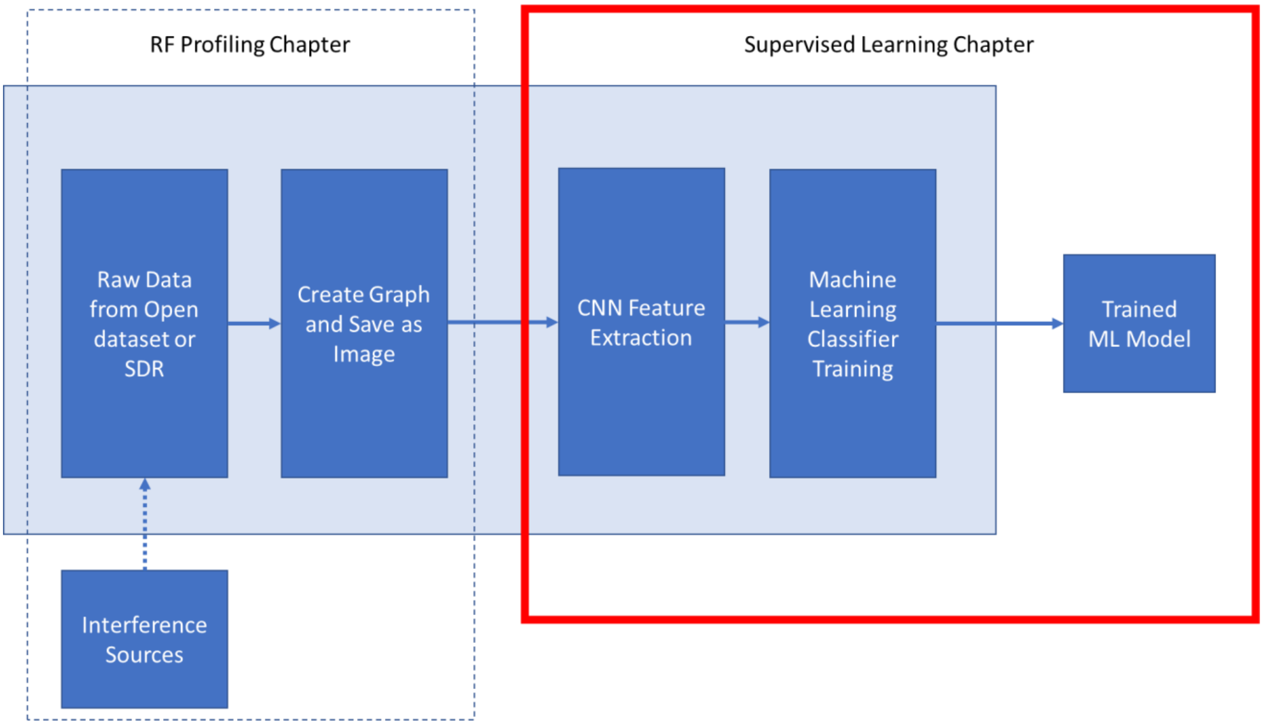

RF Profiling Chapter
Supervised Learning Chapter
Raw Data from Open dataset or SDR
Create Graph and Save as Image
CNN Feature Extraction
Machine Learning Classifier Training
Trained ML Model
Interference Sources

Figure 100 - Supervised Learning Chapter Explanation

As highlighted by the red box in Figure 100, this chapter will describe the CNN feature extraction, machine learning classifiers and will provide the supervised learning experiments and results.

4.1 CNN Feature Extraction [25] [27] [23]

4.1.1 VGG-16

CNNs are often used for object detection but through a process called transfer learning, a pre-trained CNN can be used for other purposes such as detecting medical condition through brain scan or eye scan images [300], [314]. A VGG-16 is a type of CNN with 16 layers, produced by Oxford Visual Geometry Group and commonly utilised in a pre-trained manner using a 14 million image database called ImageNet [287]. To utilise the VGG-16 in our research for feature extraction forward propagation is stopped at the last pooling layer to enable features to be saved. Table 8 shows the structure of the VGG-16 CNN used.

Table 8 - VGG-16 Architecture.

Layer Type	Shape
Input Layer)	224x224x3
Convolutional 2D Layer	112x112x128
Convolutional 2D Layer	112x112x128
Max Pooling 2D Layer	112x112x128
Convolutional 2D Layer	56x56x256
Convolutional 2D Layer	56x56x256
Max Pooling 2D Layer	56x56x256
Convolutional 2D Layer	28x28x512
Convolutional 2D Layer	28x28x512
Convolutional 2D Layer	28x28x512

Max Pooling 2D Layer	28x28x512
Convolutional 2D Layer	14x14x512
Convolutional 2D Layer	14x14x512
Convolutional 2D Layer	14x14x512
Max Pooling 2D Layer	7x7x512
Convolutional 2D Layer	7x7x512
Convolutional 2D Layer	7x7x512
Convolutional 2D Layer	7x7x512
Max Pooling 2D Layer	7x7x512

The last layer of shape 7x7x512 produces a feature vector of 25,088 values when flattened.

4.1.2 ResNet 50

ResNet [288] allows for deep neural networks to be trained using a technique called skip connection which take the output from an earlier layer and combines it with the a later layer. This technique overcame prior difficulties with training very deep neural networks whereby gradients would vanish due to repeated multiplication. ResNet50 has been commonly used for transfer learning research with a large scale image recognition database of over 14 million images called ImageNet [287]. Training the weights in a neural network from scratch can take a very long time and needs a large amount of training data, for example the 14 million images that trained the weights for ImageNet. Transfer Learning allows other domains to benefit from the use of pre-trained weights for a new purpose. ResNet50 is 50 layers deep, consisting of 48 convolution layers, 1 max pooling and 1 average pooling layer. The last layer will have an output shape of 7x7x2048. For feature extraction this gives us a feature vector of 100,352 values when the shape is flattened.

4.1.3 Feature Maps

4.1.3.1 UAS Signals

To try to understand the features that are being chosen by the CNN we plot the features maps for the Parrot AR from the DroneRF dataset for the class switched on and connecter to the controller. In particular we consider the PSD for the class Parrot AR switched on and connected to the controller. Figure 102 to Figure 104 shows the output of convolutional layers 0, 20 and 48. We have restricted the depth in the maps to 64 for consistency in the comparison but it should be noted that the depth is greater in deeper layers. PSD was chosen for the feature map visualisation as it produces the highest accuracy in Section 4.4.1. Figure 101 shows the input image given to the ResNet50 model for feature visualisation.

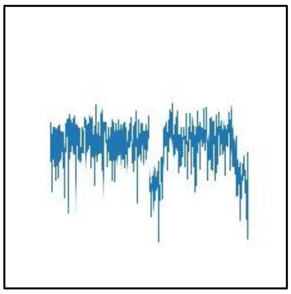


Figure 101 - PSD Graphical Signal Representation for the Parrott AR Class Switched On

Figure 102 shows the output of the first convolutional layer (layer 0) and we can clearly see the detail of the PSD for both input images AR Mode 1 – switched on and connected to the controller.

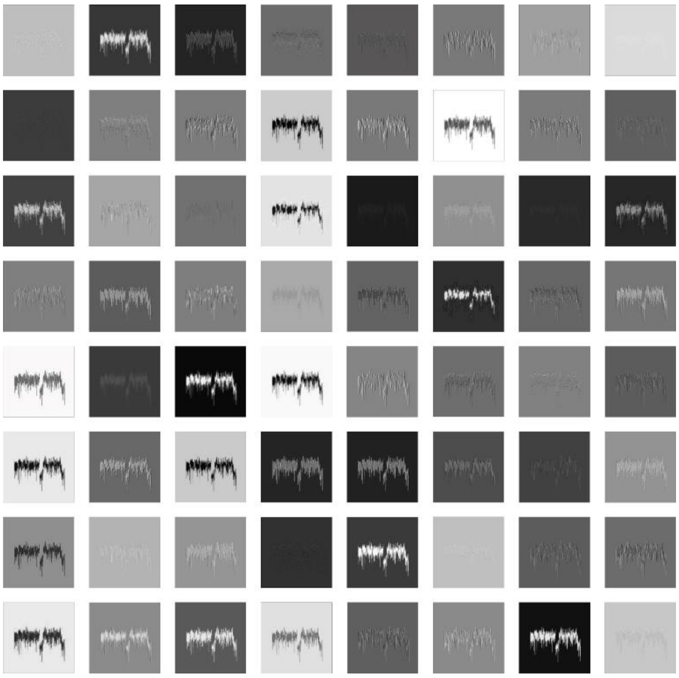


Figure 102 - Feature Map Extraction Convolutional Layer 0

As we move to convolutional layer 20, Figure 103 shows that although we can still see the outline and depth of the PSD, we start to lose some detail. This happens because the CNN starts to pick up on generic concepts rather than specific detail.

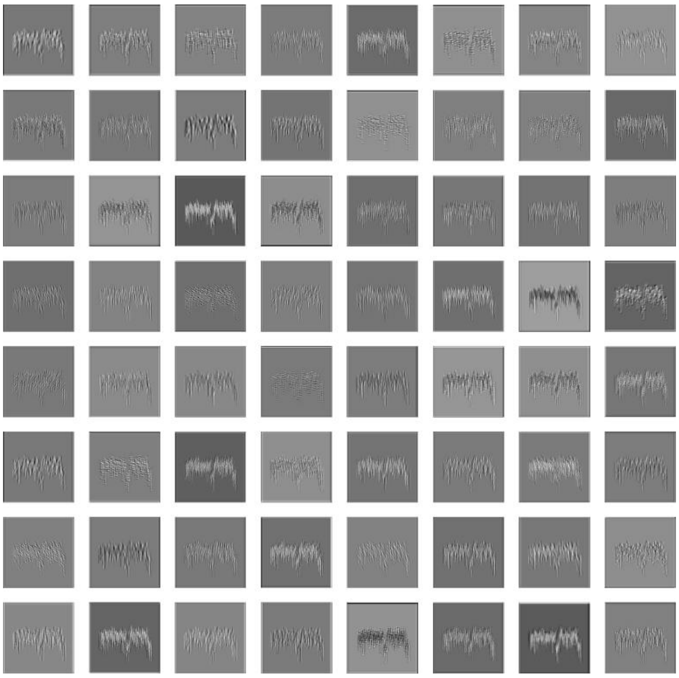


Figure 103 - Feature Map Extraction Convolutional Layer 20

Figure 104 shows the output of the last convolutional layer 48 and we can see that it is difficult now to determine with the human eye what the features are that the CNN is distinguishing.

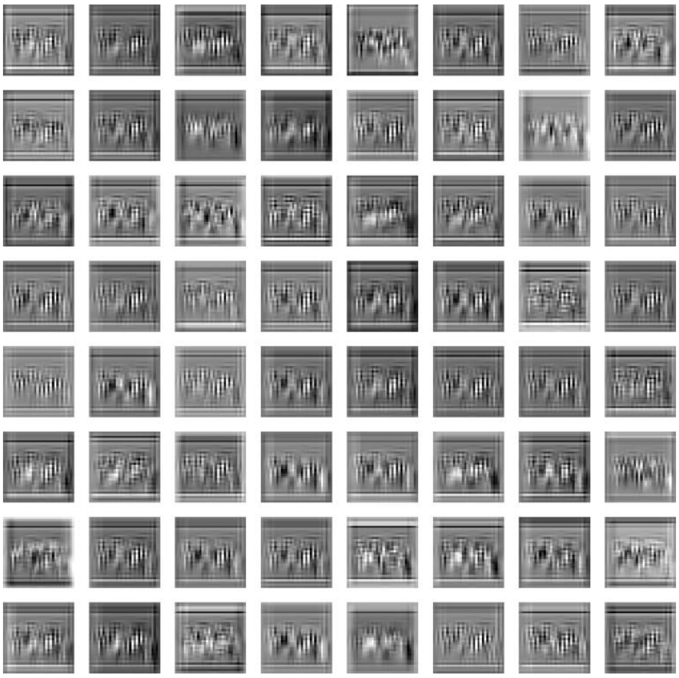


Figure 104 - Feature Map Extraction Convolutional Layer 48

4.1.3.2 GPS Jamming Signals [27]

For the GPS Jamming Dataset detailed in section 3.1.1, to help visualise which features the CNN was choosing we created feature map outputs from 3 of the main blocks that end in pooling layers, shown in Figure 105 below.

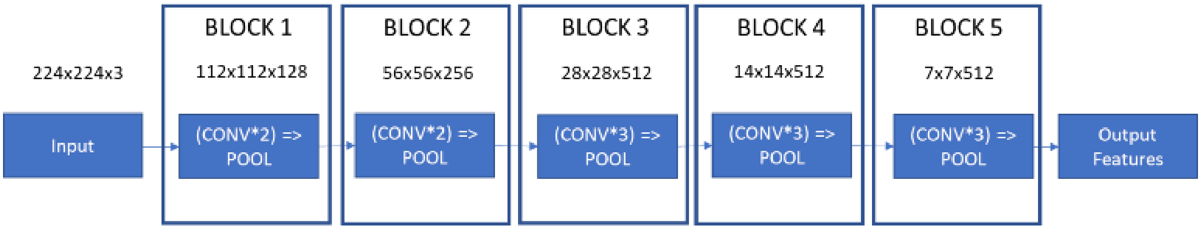


Figure 105 - CNN Model Structure: VGG16 architecture

Although the depth in deeper layers is greater than 64 we set a limit of 64 for comparison consistency.

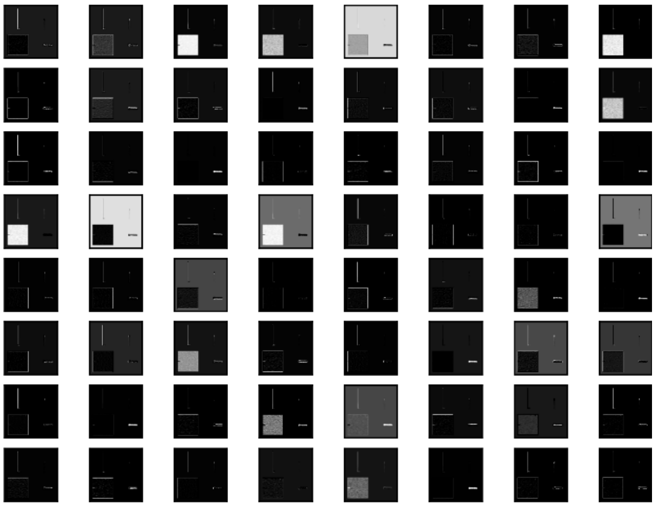


Figure 106 - Feature Maps Extracted from Block 1 (DME Signal)

We can see clearly that Figure 106 displays fine detail. The original DME input image 1 is shown in the left of Figure 109.

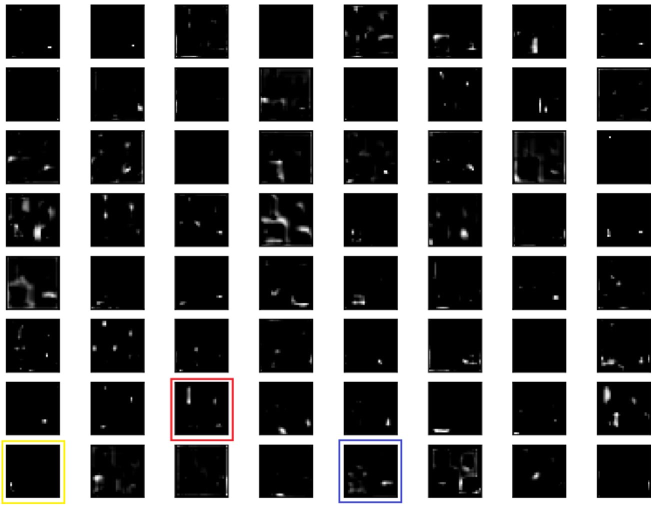


Figure 107 - Feature Maps Extracted from Block 4 (DME Signal)

As we move into block 4 in Figure 107 we lose more detail from the feature maps. This is normal as the model starts to interpret more generic concepts. We interpret the red box as the peak of the histogram and the blue box as potentially the concentration of raw IQ samples. Our interpretation of the yellow box is the DME pulse on the spectrogram image.

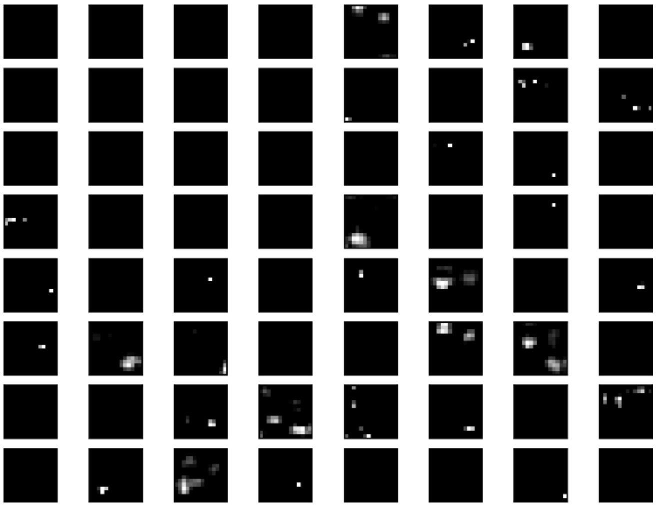


Figure 108 - Feature Maps Extracted from Block 5 (DME Signal)

In Figure 108 we can see how difficult it is to determine what the CNN is picking as features. It is not uncommon with CNNs for a human eye to be unable interpret deeper feature maps.

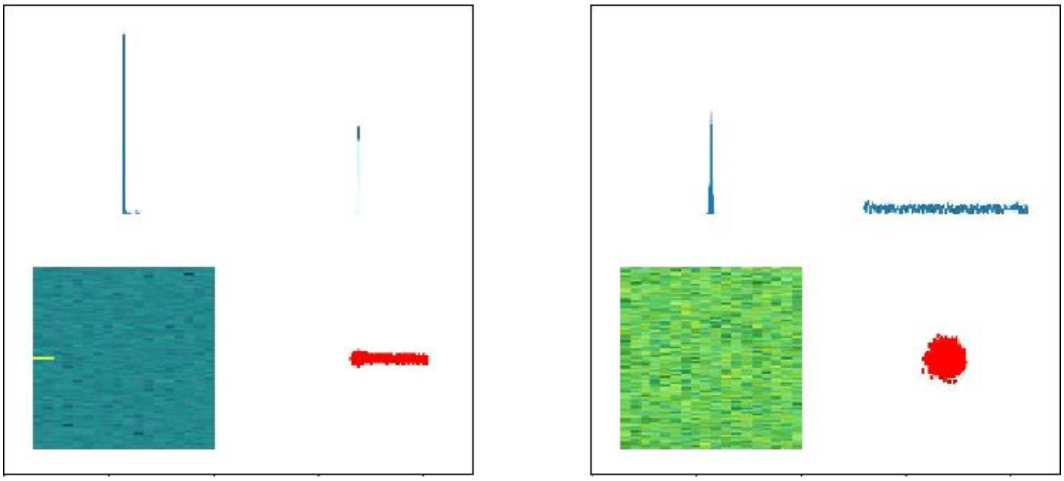


Figure 109 - Input Image 1 DME (left), Input Image 2 No Jamming signal (right)

For comparison Figure 110 shows block 4 when input image 2 (no jamming signal present) from Figure 109 is used as the input. The original input image can be seen in the right hand image in Figure 109.

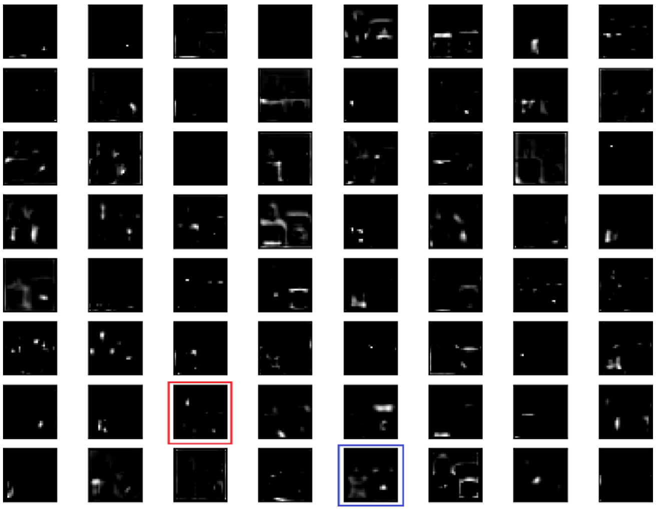


Figure 110 - Feature Maps Extracted from Block 4 (No Jamming Signal)

Comparing the features we interpreted from Figure 107, the red box potentially represents the histogram peak. We have highlighted the same square in Figure 110 which shows a slightly lower peak, correlating with the difference in the input images seen in Figure 109. The blue box in Figure 110 again correlates with Figure 107 and highlights the raw constellation.

4.2 Machine Learning Classifiers [25] [27] [24]

4.2.1 Logistic Regression

LR is a machine learning model which has a fixed number of parameters based on the number of features in the input. The output of LR is categorical and uses a sigmoidal curve. The equation for a sigmoid can be seen in Equation (14).

$$h = \frac{e^{x}}{(1 + e^{-x})} \tag{14}$$

The output of Equation (14) will always be between 0 and 1 so if we define a threshold for example of 0.5, then any values below 0.5 will return 0 and above 1. x represents the input features and to initialise Θ it is multiplied by a random value Θ. When there are multiple features this makes the equation seen in Equation (15).

$$h = \Theta_0 + \Theta_1 X_1 + \Theta_2 X_2 +.. \tag{15}$$

The algorithm in Equation (15) updates Θ and eventually will establish a relationship between the features and the output through updating Θ. For a situation where we have multiple classes, the sigmoid is generalised and this is called the Softmax function. The Softmax function takes a input vector and then plots it to a probability distribution between 0 and 1. In Equation (16) we describe the Softmax function for vector z with k dimensions or classes [354].

$$softmax(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{zj}} \tag{16}$$

LR was implemented in Python 3 through Sklearn. Ridge Regression was used as the penalty for the loss function and Limited memory Broyden–Fletcher–Goldfarb–Shanno (LGBFS) was used as the solver. Values for regularisation (‘C’) were optimised using SkLearn GridSearchCV.

4.2.2 Linear SVM

A Linear SVM was also considered in some of the experiments. Mete and Ensari [355] show the use of SVM as a classifier following CNN feature extraction for flower species as superior to other ML models. The SVM support vector classification function in Sklearn is used to implement this model. For hyper-parameter selection values for regularisation (‘C’) were chosen in a range of [10, 1.0, 0.1, 0.01, 0.001].

4.2.3 Random Forest

A RF Classifier is evaluated which works by creating a number of decision trees and outputting the prediction of the tree using averaging to improve accuracy. The number of trees in the forest was set to 1000 for the experiments. RF has cross validation/bootstrapping built in during training so is not necessary to be included in the hyper-parameter optimisation.

4.2.4 K-Nearest Neighbour (kNN)

kNN uses the dataset to find the closet point to the input point. The classifier then works by using a majority vote of neighbours. The Minkowski distance was utilised as part of our experiments and can be calculated as show in Equation (17).

$$sum(|x-y|^{p})^{1/p} \tag{17}$$

The k-nearest neighbour of the particular data point is then found and assigned to the class that has the highest probability.

$$P(y = j \mid X = x) = \frac{1}{K}\sum_{i \epsilon A} I(y^{(i)} = j) \tag{18}$$

Equation (18) shows the probability of an input x being assigned to the class that has the highest probability [356]. For each of the models, fivefold cross validation was used to indicate whether the model was overfitting. Hyper-parameter regularisation for LR and the number of neighbours for kNN were optimised using threefold nested cross-validation.

4.3 Performance Metrics [25] [27]

4.3.1 Cross Validation

In machine learning, models need to be able to make accurate predictions on data not seen before and this is known as generalisation [357]. Traditional models split the data into training and test, the training data is used to train the model and some test data is kept separate to validate the results. This technique does not work well with limited datasets as it can produce high bias. Cross validation is a method for assessing how a machine learning model will generalise to a data set. The technique aims to estimate whether the model will make accurate predictions in practice and can highlight problems such as overfitting or bias [358]. K-Fold cross validation is a method which is considered to produce a model with less bias with a small dataset. The data is partitioned randomly into k subsets, the model trained using k-1 subsets and the last subset is then used as the test data to validate the model. The performance of the model is measured by averaging the score for each subset. To validate the machine learning models in our experiments the dataset (800 samples per class) was divided into 10 folds using the K-Fold cross-validation through Python SkLearn. StratifiedKfold was used which keeps the different class percentage the same for each subset so the model is equally distributed [359].

There is a trade off in terms of bias and variance when it comes to choosing a value for k. The statistical community tend to agree on a value of either k = 5 or k = 10 as they have proved not to be susceptible to either high bias or high variance [360]. A value of 10 was evaluated in these experiments and the full model (including feature extraction) was included within the cross validation. Hyper-parameter optimsation was conducted using nested k-fold cross validation through SkLearn GridSearchCV. Hyper-parameters are values which control the learning process and are set, rather than being learned. Table 9 shows the hyper-parameters chosen for the experiments in this thesis.

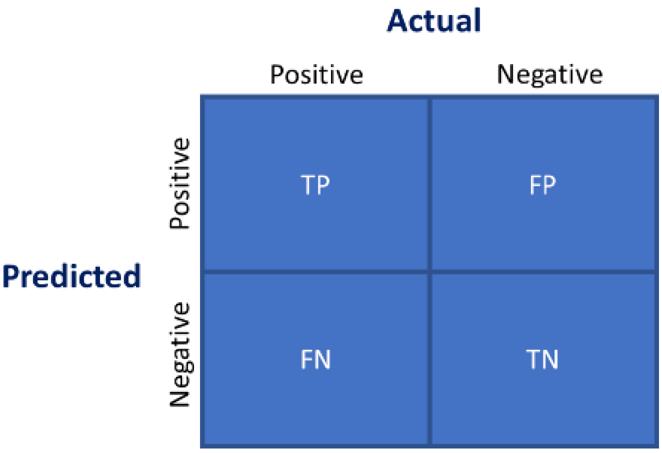
Table 9 - Hyper-parameter Optimisation

	Regularisation ('C')
LR	[100, 10, 1.0, 0.1, 0.01]
SVM	[10, 1.0, 0.1, 0.01, 0.001]

Choosing an optimal set is an important part of the machine learning process. Gridsearch searches through subsets of the hyper-paramaters for every combination and evaluates their performance using cross validation. Grid search was chosen over Random search, an alternative which randomly selects the combinations to find the best one, because the goal of the work was to look at the accuracy. Although random search is less computationally heavy and takes less time, it won't always produce the most accurate result. Other optimisation methods exist such as Bayesian but they were not considered in this paper. Features were extracted from the dataset and then predictions were made using the best classification models (determined by the hyper-parameter optimisation).

4.3.2 Metrics

There are different metrics we could use to evaluate performance. A confusion matrix helps us to calculate and visualise indictors of performance such as accuracy. Figure 111 shows the components that make up a confusion matrix.



		Actual	
		Positive	Negative
Predicted	Positive	TP	FP
	Negative	FN	TN

Figure 111 - Confusion Matrix Pictoral Representation

True Positive (TP) tells us that the prediction was correct and it was true to what was predicted. True Negative (TN) is where we have predicted something was incorrect and it was incorrect. False Positive (FP) is where we have predicted something was correct but it was not. False Negative (FN) is where we predicted incorrect when it was correct. The values help us to define accuracy and F1-score. Accuracy shows us how often the model was right in its predictions, shown in Equation (19).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{19}$$

To calculate F1-Score we need to understand how to calculate both recall and precision. Precision is calculated by dividing TP by TP + FP. This shows how many predicted positives were actually positive. Recall is calculated by dividing TP by TP + FN. It shows the fraction of positives that were correctly predicted.

Equation (20) shows the formula for precision showing how many positive predictions were correct.

$$Precision = \frac{TP}{TP + FP} \tag{20}$$

Equation (21) shows the equation for recall and considers the correctly predicted positives as a fraction.

$$Recall = \frac{TP}{TP + FN} \tag{21}$$

F1-Score is often used as a performance metric as it takes into consideration both recall and precision as seen in Equation (22).

$$F1\ Score = \frac{2\ (Precision * Recall)}{Precision + Recall} \tag{22}$$

4.4 Classification Results & Discussion

The classification results and discussion have been broken down by dataset. Table 10 gives an overview of the differences between the experiments including CNN type, ML classifiers, image types tested, number of classes, SNR and the dataset used. The process described in sections Chapter 4 to 0 was used for all experiments unless otherwise detailed in the results below.

Table 10 - Experiments Overview

UAS Classification Experiments							
Experiment	CNN Types	ML Classifiers	Image Types	Classes	SNR	Dataset	Reference
1	VGG-16	LR, SVM, RF	PSD, Spec	2,4,10	No	DroneRF	[28]
2	ResNet-50	LR	PSD, Spec, Raw IQ, Hist	10	No	DroneRF	[25]
3	VGG-16	LR, kNN	PSD, Spec	2,4,10	Yes	DroneDetect (4 UAS)	[26]
4	VGG-16	LR, kNN	PSD, Spec	2,8,21	Yes	DroneDetect (8 UAS)	[24]
GPS Jamming Signal Classification							
5	VGG-16	SVM, LR, RF	PSD, Spec, Raw IQ, Hist, Concat	6	No	GPS Jamming Dataset	[27]

6	ResNet-50	SVM, RF	PSD, Spec, Raw IQ, Hist, Concat	6	Yes	GPS Jamming Dataset	
7	VGG-16	LR	PSD, Spec, Raw IQ, Hist, Concat	6	Yes	JamDetect	[23]

4.4.1 DroneRF Dataset Experiments [28] [25]

The DroneRF dataset as described in section 3.1.3 and produced by Al-S’ad *et al.* [204] covers 80MHz of the 2.4GHz frequency band using 2x SDRs. The dataset includes the Parrot Bebop, Parrot AR (Elite 2.0), the DJI Phantom 3 and the class no UAS present.

4.4.1.1 Experiment 1

Overview

Experiment 1 uses the open DroneRF dataset. The experiments are considered in three sets, 2 class – UAS, No UAS; 4 class – No UAS, Bebop, AR, Phantom; and for 10 classes we include flight modes, mode 1 – switched on and connected to controller, mode 2 – hovering automatically, mode 3 – flying without video, mode 4 – flying with video. The 10 classes include; No UAS; Bebop Mode 1; Bebop Mode 2; Bebop Mode 3; Bebop Mode 4; AR Mode 1; AR Mode 2; AR Mode 3; AR Mode 4 and Phantom Mode 1.

Signals were represented in the time domain as a spectrogram and in the frequency domain as a PSD. This would let us evaluate whether viewing the data in the time or frequency domain produced higher accuracy results for classification. Python 3 Matplotlib was used to plot the spectrogram and PSD images both with 1024 FFT size, Hanning windowing functions and centre frequency 2.442GHz covering a range of 2.402GHz – 2.482GHz. Images were saved as size 224x224 pixels. For our experiments datasets were created for spectrograms and PSD with 1000 samples per class. This was further split into 80% for k fold cross validation training/test data and 20% as a separate evaluation dataset not used in training. A VGG-16 is used for CNN feature extraction and machine learning classifiers LR, SVM and RF are evaluated for classification.

Results

Table 11 shows that representing the signal in the time domain via spectrogram is approximately 20% less accurate than PSD. LR performs the best out of the machine learning classifiers producing 87.5% (+/- 0.5%) accuracy for PSD with 10 class flight mode classification, 89.2% (+/- 0.9%) with 4 class UAS type classification and 100% (+/- 0.0%) for 2 class UAS detection. Although LR outperformed SVM and RF, the actual difference is marginal. In the 2 class detection for PSD both LR and SVM produce 100% accuracy and RF 99%. As we increase the classifications the gap increases but it is still small, less than 1% at 10 classifications between LR and SVM and 2.5% difference between RF and LR for 4 classifications. Al-S’ad *et al*. in [240] showed that accuracy decreased with the increase of classes, showing a 15% drop in accuracy as the classes increased from 2 to 4 and a further 37.7% drop as 4 was increased to 10 classes. Our results show an 11% decrease in accuracy as we increase from 2 to 4 classes and only a marginal difference of 1% when we increase 4 to 10 classes (LR & PSD).

Table 11 - DroneRF UAS Classification Accuracy (%) & F1-Score (%)

		2 Class		**4 Class**		**10 Class**	
		Spec	**PSD**	**Spec**	**PSD**	**Spec**	**PSD**
SVM	**Acc**	80.8 (+/-1.0%)	100 (+/-0.0%)	68.6 (+/-1.0%)	88.5 (+/-0.7%)	67.3 (+/-0.6%)	86.9 (+/-0.5%)
	F1	80.8 (+/-1.0%)	100 (+/-0.0%)	68.4 (+/-1.0%)	88.5 (+/-0.7%)	67.0 (+/-0.5%)	86.7 (+/-0.5%)
LR	**Acc**	82.7 (+/-0.8%)	100 (+/-0.0%)	70.3 (+/-0.9%)	89.2 (+/-0.9%)	67.4 (+/-1.2%)	87.5 (+/-0.5%)
	F1	82.7 (+/-0.8%)	100 (+/-0.0%)	70.3 (+/-0.9%)	89.2 (+/-0.9%)	67.4 (+/-1.1%)	87.5 (+/-0.5%)
RF	**Acc**	77.0 (+/-0.4%)	99.9 (+/-0.1%)	62.0 (+/-0.8%)	90.0 (+/-0.8%)	54.9 (+/-1.6%)	84.5 (+/-0.8%)
	F1	76.6 (+/-0.4%)	99.9 (+/-0.1%)	61.6 (+/-0.7%)	89.8 (+/-0.8%)	53.3 (+/-1.6%)	84.4 (+/-0.8%)

Figure 112, Figure 113 and Figure 114 show the confusion matrix for LR, RF and SVM respectively using PSD results from the hold-out evaluation data set. This data was not used as part of cross validation training and hyper-parameter optimisation. Figure 112 shows that when we want to simply detect whether there is a UAS present or not that we can achieve 100% accuracy with this methodology.

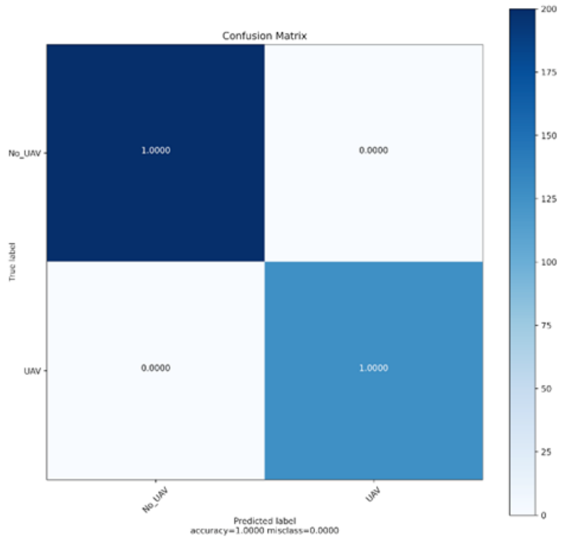
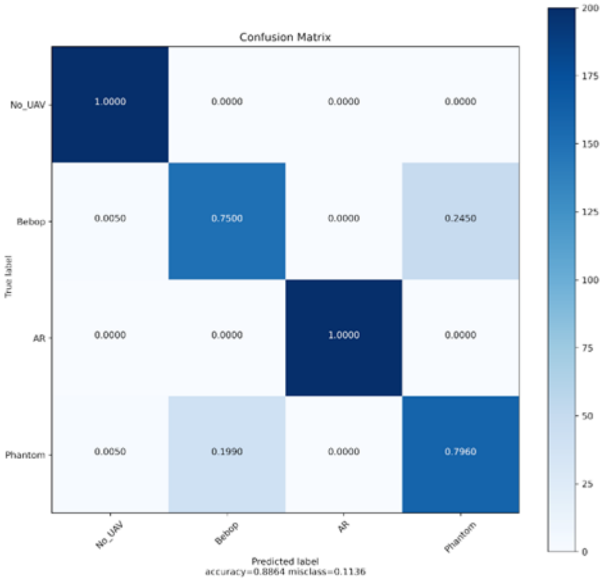


Figure 112 - Confusion Matrix LR 2 Class PSD Graphical Signal Representation

Figure 113 shows the confusion matrix when we are considering not only detecting the UAS but classifying it in terms of type – No UAS, Bebop, AR or Phantom. We can see that accuracy reduces to 88.6%.

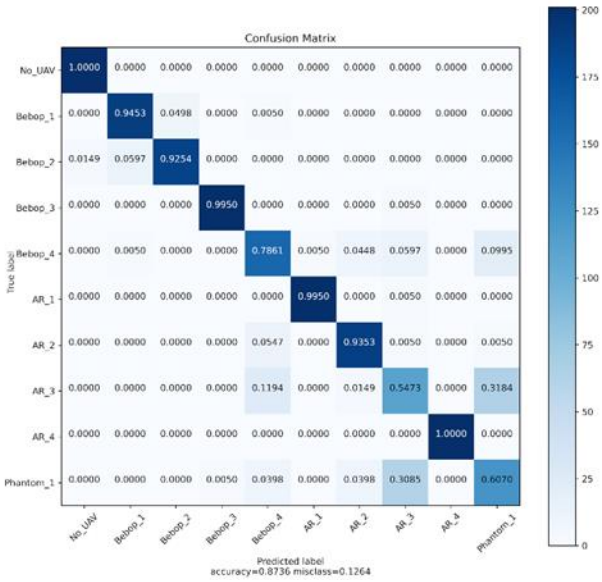


	No_UAV	Bebop	AR	Phantom
No_UAV	1.0000	0.0000	0.0000	0.0000
Bebop	0.0050	0.7500	0.0000	0.2450
AR	0.0000	0.0000	1.0000	0.0000
Phantom	0.0050	0.1990	0.0000	0.7960

Figure 113 - Confusion Matrix LR 4 Class PSD Graphical Signal Representation

Lastly Figure 114 shows the confusion matrix for all 10 classes and flight modes. LR with PSD signal representation produces 87.36% accuracy which is a 40% increase compared directly to the work conducted by Al-S’ad *et al*. [240]. We can see that the classifier has problems distinguishing between the Phantom 3 when switched on and connected to the controller and the Parrot AR when flying without video. The spectrum in the frequency domain must look similar for both those classes, making it harder for the classifier to distinguish between the two. A potential way of increasing the accuracy would be to add more SDRs to monitor and capture the 5GHz spectrum. The Parrot AR does not operate in that space but the Phantom 3 does, this information would undoubtedly help the classifiers in distinguishing between the two.

)



	No_UAV	Bebop_1	Bebop_2	Bebop_3	Bebop_4	AR_1	AR_2	AR_3	AR_4	Phantom_1
No_UAV	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Bebop_1	0.0000	0.9453	0.0498	0.0000	0.0050	0.0000	0.0000	0.0000	0.0000	0.0000
Bebop_2	0.0149	0.0597	0.9254	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Bebop_3	0.0000	0.0000	0.0000	0.9950	0.0000	0.0000	0.0000	0.0050	0.0000	0.0000
Bebop_4	0.0000	0.0050	0.0000	0.0000	0.7861	0.0050	0.0448	0.0597	0.0000	0.0995
AR_1	0.0000	0.0000	0.0000	0.0000	0.0000	0.9950	0.0000	0.0050	0.0000	0.0000
AR_2	0.0000	0.0000	0.0000	0.0000	0.0547	0.0000	0.9353	0.0050	0.0000	0.0050
AR_3	0.0000	0.0000	0.0000	0.0000	0.1194	0.0000	0.0149	0.5473	0.0000	0.3184
AR_4	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
Phantom_1	0.0000	0.0000	0.0000	0.0050	0.0398	0.0000	0.0398	0.3085	0.0000	0.6070

Figure 114 - Confusion Matrix LR 10 Class PSD Graphical Signal Representation

Discussion

In Experiment 1 it was shown that LR had the highest performing classifier but the difference between SVM and RF is small. PSD representation in the frequency domain produces higher accuracy than the use of time domain spectrograms for all experiments. LR in conjunction with PSD produced 100% accuracy for 2 class UAS detection and 88.6% accuracy for 4 class UAS type classification. For 10 classifications including flight modes LR produced 87% accuracy which is a 40% increase on prior work in the field by Al-Sa’d *et al.* They also showed the performance of the classifiers to decrease as the classification number increased. They found a 37.7% drop in accuracy when increasing the class number from 4 to 10. Our results show only a 1% decrease in accuracy when increasing the classifications from 4 to 10. In the context of using this model within an early warning system, the accuracy levels required would be part of the system requirements. For example, while one system may have requirements for a high accuracy results, another system may have other sensor data which could be combined to produce that high level indication and therefore a single model may not require results which are as high in performance. This would be laid out in the early warning system requirements phase.

Incorporating the 5GHz spectrum could help the classifier increase accuracy in distinguishing between flight modes and UAS types. It would be beneficial to evaluate the performance of these algorithms in a congested spectrum. For example, testing in a built up city environment where there will be a lot of activity in the Wi-Fi spectrum. Overall experiment 1 has introduced an approach using transfer learning through CNN feature extraction and machine learning classification that has high accuracy even when classifying flight modes and outperforming prior work using the same DroneRF dataset.

4.4.1.2 Experiment 2

Overview

Samples from the DroneRF dataset were plotted in each of the 4 signal representations using MatPlotlib and saved as images with 300 DPI. Separate datasets of images for raw constellation, spectrogram, PSD and histogram were created. Each class within each dataset contained 1000 image representations resized to 224x224 pixels. The databases were split so there were 8,000 images for use with k-fold cross validation and 2000 images with the evaluation set. A ResNet-50 was used for the CNN feature extraction and machine learning classifier LR for the classification of 10 classes of UAS flight modes. The 10 classes include the Parrot Bebop and the Parrot AR flight modes of switched on and connected to controller; hovering automatically with no input from the controller; flying with video transmission; and flying without video. The DJI Phantom 3 was the 10th class but only provided in the DroneRF dataset in the first mode - switched on and connected to controller.

Results

The extracted feature maps for these experiments can be seen in section Chapter 4.

Cross Validation Training/Test Data

It can be seen from Table 12 that PSD outperforms raw constellation, spectrogram and histogram representations for 10 class UAS flight mode classification.

Table 12 - ResNet 10 Class UAS Flight Mode Classification Accuracy and F1-Score

Metric	Raw IQ	Spectrogram	PSD	Histogram
Accuracy (%)	45.3 (+/- 1.1)	83.8 (+/- 1.1)	92.3 (+/- 0.3)	37.0 (+/- 0.2)
F1 Score (%)	45.1 (+/- 1.1)	83.7 (+/- 1.2)	92.3 (+/- 0.3)	36.8 (+/- 0.2)

Spectrogram representation was approximately 10% less accurate than PSD, with histogram performing the worst out of all the representations. PSD produced 92.3 (+/- 0.3%) accuracy and F1-score which is an increase of over 45% from previous published work.

Table 13 - ResNet 10 Class UAS Flight Mode Individual Classification LR F1-Score

Mode	Raw IQ (%)	Spectrogram (%)	PSD (%)	Histogram (%)
No UAS	51	97	100	49
Bebop Switched on	26	88	97	26
Bebop Hovering	29	83	97	18

Bebop Flying	90	79	100	79
AR Switched on	23	92	100	21
AR Hovering	31	86	94	14
AR Flying	20	69	71	18
Phantom 3 Flying	35	64	71	25

Table 13 shows the individual representations and their F1-score performance for each individual class. PSD outperforms the other representations in all classes. Overall Table 13 shows that PSD is the most accurate way to classify UAS signals across 80MHz of the Wi-Fi band using transfer learning with ResNet50 CNN feature extraction and LR.

Hold-Out Evaluation Results

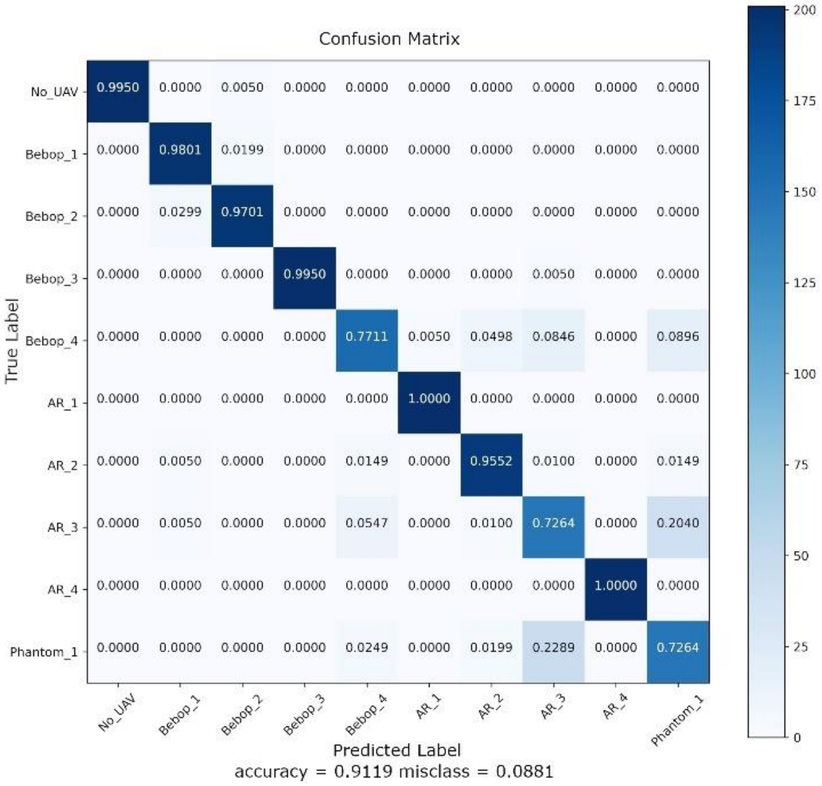
2000 images as an evaluation set were kept entirely separate from the images used with the cross validation training and testing process. This was done to ensure there wasn’t any overfitting occurring in the results.

The evaluation data set results in terms of accuracy and F1-score can be seen in Table 14. These results confirm the cross validation results in Table 12 with PSD producing the highest accuracy and F1-score. Table 14 shows that PSD is over 10% more effective than spectrograms and over 40% more accurate than raw constellation and histograms.

Table 14 - ResNet 10 Class UAS Flight Mode Evaluation Data Accuracy & F1-Score

Metric	Raw IQ	Spectrogram	PSD	Histogram
Accuracy (%)	43.1	81.5	91.2	36.7
F1 Score (%)	42.9	81.7	91.2	36.6

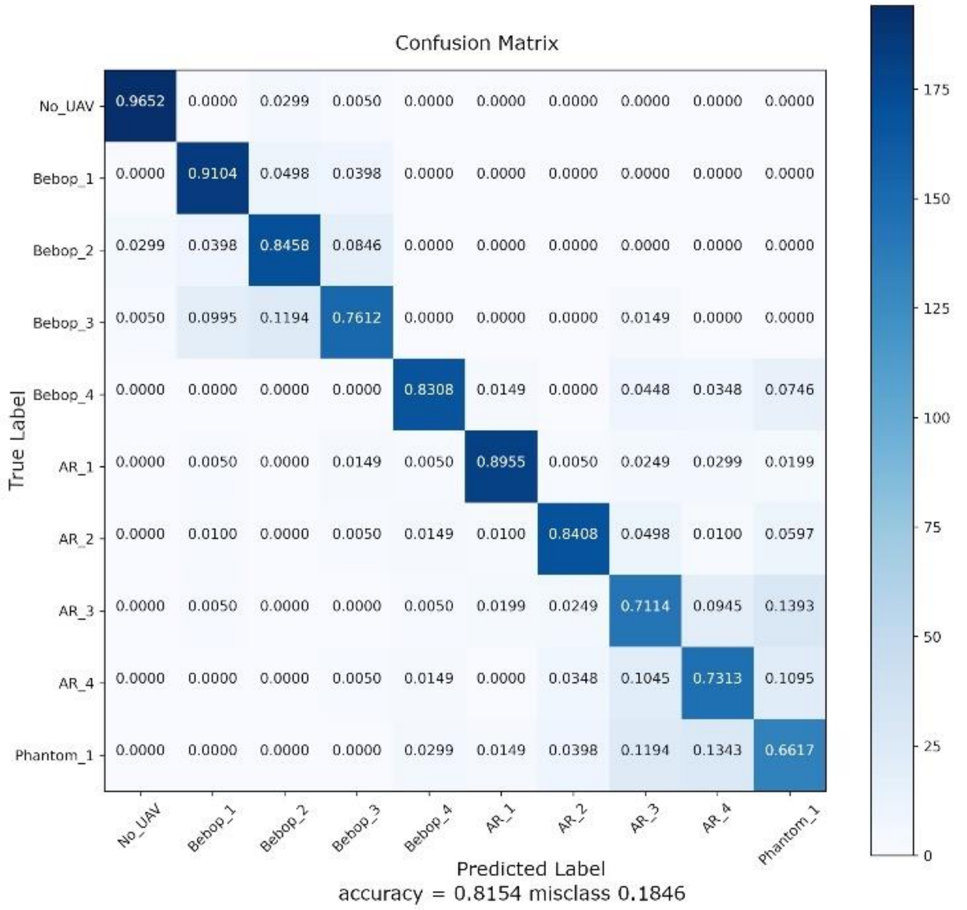
Figure 115 and Figure 116 show the confusion matrix for PSD and spectrogram representations respectively. Both are able to detect whether a UAS is present or not with an accuracy of 96% or over, with PSD performing at 99.7%.



	No_UAV	Bebop_1	Bebop_2	Bebop_3	Bebop_4	AR_1	AR_2	AR_3	AR_4	Phantom_1
No_UAV	0.9950	0.0000	0.0050	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Bebop_1	0.0000	0.9801	0.0199	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Bebop_2	0.0000	0.0299	0.9701	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Bebop_3	0.0000	0.0000	0.0000	0.9950	0.0000	0.0000	0.0000	0.0050	0.0000	0.0000
Bebop_4	0.0000	0.0000	0.0000	0.0000	0.7711	0.0050	0.0498	0.0846	0.0000	0.0896
AR_1	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000
AR_2	0.0000	0.0050	0.0000	0.0000	0.0149	0.0000	0.9552	0.0100	0.0000	0.0149
AR_3	0.0000	0.0050	0.0000	0.0000	0.0547	0.0000	0.0100	0.7264	0.0000	0.2040
AR_4	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
Phantom_1	0.0000	0.0000	0.0000	0.0000	0.0249	0.0000	0.0199	0.2289	0.0000	0.7264

Figure 115 - ResNet 10 Class UAS Flight Mode Confusion Matrix PSD Graphical Signal Representation

Figure 115 and Figure 116 confirm that both representations were worst at classifying the AR in mode 3 (flying without video) and the Phantom 3 when switched on and connected to the controller. The reason for the Phantom 3 could be the fact that it will hop Wi-Fi channels based on interference. Without monitoring the spectrum separately we can’t be sure whether this is happening. A larger training dataset could increase accuracy of the Phantom 3 by capturing more configurations of the frequency hopping. Further, the parrot AR when flying without video must look similar in terms of features in the frequency domain to the Phantom 3 when switched on and connected to the controller.



	No_UAV	Bebop_1	Bebop_2	Bebop_3	Bebop_4	AR_1	AR_2	AR_3	AR_4	Phantom_1
No_UAV	0.9652	0.0000	0.0299	0.0050	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Bebop_1	0.0000	0.9104	0.0498	0.0398	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Bebop_2	0.0299	0.0398	0.8458	0.0846	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Bebop_3	0.0050	0.0995	0.1194	0.7612	0.0000	0.0000	0.0000	0.0149	0.0000	0.0000
Bebop_4	0.0000	0.0000	0.0000	0.0000	0.8308	0.0149	0.0000	0.0448	0.0348	0.0746
AR_1	0.0000	0.0050	0.0000	0.0149	0.0050	0.8955	0.0050	0.0249	0.0299	0.0199
AR_2	0.0000	0.0100	0.0000	0.0050	0.0149	0.0100	0.8408	0.0498	0.0100	0.0597
AR_3	0.0000	0.0050	0.0000	0.0000	0.0050	0.0199	0.0249	0.7114	0.0945	0.1393
AR_4	0.0000	0.0000	0.0000	0.0050	0.0149	0.0000	0.0348	0.1045	0.7313	0.1095
Phantom_1	0.0000	0.0000	0.0000	0.0000	0.0299	0.0149	0.0398	0.1194	0.1343	0.6617

Figure 116 - ResNet 10 Class UAS Flight Mode Confusion Matrix Spectrogram Graphical Signal Representation

Figure 115 and Figure 116 show the confusion matrix for PSD and Spectrogram representations. Both representations produce an overall accuracy of above 81% but PSD performs with higher overall accuracy above 91%. Due to the fact we produced our results with the same open DroneRF dataset used and produced by Al-Sa'd et al. [240], who achieved accuracy of 46.8% using a DNN across all 10 classes, we can directly compare our results. Our results show that LR with PSD, to achieve over 91% accuracy, a 45% increase compared with the prior work. We achieve this by viewing UAS classification as an image classification problem and utilising transfer learning from the field of imagery. Al-S'ad et al. found that when they increased the classification from detecting the presence of a UAS (2 class) to its type (4 class), to include flight modes (10 class) the accuracy decreased significantly. They put this down to similarities caused by two of the UASs (Bebop and AR) being manufactured by the same company. We have shown our approach using CNN feature extraction able to improve these results distinguishing between same manufacturer.

Discussion

Experiment 2 for the DroneRF ResNet-50 10 class flight mode UAS classification results have shown that PSD outperforms raw constellation, spectrogram and histogram representations for LR. PSD produced over 91% accuracy with cross validation results and the evaluation dataset. We achieve this by viewing UAS classification as an image classification problem, utilising transfer learning and presenting signal representations as graphical images to a deep CNN. If a system like this was employed in the real world it would likely need to be trained in the particular environment that it needed to work in. For example, a built up city area will have more background noise in the Wi-Fi bands than a rural area. This will also likely affect the accuracy of the classifier so field testing in city areas is paramount for this type of system. It may also help researchers understand how much frequency hopping occurs due to interference and whether this has an impact on detection and classification accuracy. Further, the issue of how often you would need to re-train the classifier, as RF bands are constantly changing, and how much change the classifier can cope with before accuracy starts being affected is an important question which would need further investigation.

Future work could also consider the employment of another SDR to capture the 5Ghz band to fully represent dual frequency band UASs such as the Bebop and Phantom 3. It is thought that this would further improve accuracy as it provides an increase of distinguishing features. The dataset could also be expanded to include more UAS platforms. This method which utilises signal representations as graphical images would require more processing power and therefore increased energy requirements compared with processing 1D data. Further work could look at hardware implementations such as FPGA, GPU and hardware accelerators such as Tensor processing unit [361] by Google to evaluate practical limitations for 2D data against the use of 1D compared with accuracy of the models. In conclusion experiment 3 has shown a novel approach by treating UAS classification as an imagery detection problem utilising the benefits of transfer learning and outperforming previous work in the field by over 45%.

4.4.2 DroneDetect Dataset Experiments [26] [24]

The DroneDetect dataset was produced as described in section 3.1.4 to firstly expand the number of UAS platforms and to allow for the introduction of Bluetooth and Wi-Fi interference upon signal collection. In comparison to the DroneRF dataset, the DroneDetect dataset captures a much smaller part of the 2.4GHz Wi-Fi frequency band 28MHz centred around 2.4375GHz. Where possible UAS platforms were restricted to the 2.4GHz band for these experiments. The uplink and downlink are not discriminated against, everything which is transmitting for the UAS flight modes switched on, hovering and flying is recorded within that 28MHz band as part of the collection.

4.4.2.1 Experiment 3

Overview

For experiment 3 four classes were taken from the DroneDetect dataset – DJI Inspire 2, DJI Mavic 2 Pro, DJI Mavic Mini and no UAS present. Datasets for PSD and spectrograms with no interference, Bluetooth interference and Wi-Fi interference were produced for 2 class UAS detection, 4 class UAS type classification and 10 class UAS flight mode classification. For UAS detection the UAS class was made up of an equal number of samples from Inspire Mode 3 – flying, Mavic Mode 3 – flying and Mini Mode 3 Flying. Mode 3 flying was also used to make the UAS type class for each UAS. VGG-16 was used as the CNN for feature extraction and machine learning classifiers LR and kNN for classification.

Results

Table 15 shows that accuracy and F1-score are unaffected for 2 class UAS detection when we consider LR with PSD. LR outperforms kNN for spectrogram representation and for PSD, both LR and kNN produce near 100% accuracy. We also observe that PSD frequency domain features are more robust than spectrogram time domain features when Bluetooth and Wi-Fi interference is introduced. Bluetooth interference effects the spectrogram representation accuracy more than Wi-Fi signals.

Table 15 - DroneDetect 2 Class UAS Detection Accuracy and F1-Score with Interference

Classifier	Image Type	Metric	Clean	Bluetooth	Wi-Fi
LR	PSD	Accuracy (%)	100(+/- 0.0)	100(+/- 0.0)	100(+/- 0.0)
LR	PSD	F1 Score (%)	100(+/- 0.0)	100(+/- 0.0)	100(+/- 0.0)
LR	Spec	Accuracy (%)	99.1(+/- 0.3)	91.9(+/- 1.9)	96.8(+/- 0.7)
LR	Spec	F1 Score (%)	99.1(+/- 0.3)	91.9(+/- 1.9)	96.8(+/- 0.7)
kNN	PSD	Accuracy (%)	100(+/- 0.0)	99.9(+/- 0.2)	100(+/- 0.0)
kNN	PSD	F1 Score (%)	100(+/- 0.0)	99.9(+/- 0.2)	100(+/- 0.0)
kNN	Spec	Accuracy (%)	95.5(+/- 1.0)	86.8(+/- 1.4)	90.2(+/- 1.6)
kNN	Spec	F1 Score (%)	95.5(+/- 1.0)	86.7(+/- 1.5)	90.1(+/- 1.7)

Table 16 shows the results with 5 fold cross validation for 4 class UAS type classification. Again we note that PSD outperforms spectrogram graphical signal representation and that LR outperforms kNN as the machine learning classifier. As with Table 15 we note that frequency domain features are more robust to the introduction of interference than time domain features. LR with PSD maintains near 100% accuracy for 4 class UAS type detection.

Table 16 - DroneDetect 4 Class UAS Type Classification Accuracy and F1-Score with Interference

Classifier	Image Type	Metric	Clean	Bluetooth	Wi-Fi
LR	PSD	Accuracy (%)	100(+/- 0.0)	99.9(+/- 0.1)	100(+/- 0.0)
LR	PSD	F1 Score (%)	100(+/- 0.0)	99.9(+/- 0.1)	100(+/- 0.0)
LR	Spec	Accuracy (%)	99.8(+/- 0.1)	91.5 (+/-1.6)	95.8(+/-0.5)

LR	Spec	F1 Score (%)	99.8(+/- 0.1)	91.5 (+/-1.6)	95.8(+/-0.5)
kNN	PSD	Accuracy (%)	100(+/- 0.0)	98.3(+/- 0.3)	98.9(+/- 0.1)
kNN	PSD	F1 Score (%)	100(+/- 0.0)	98.3(+/- 0.3)	98.9(+/- 0.1)
kNN	Spec	Accuracy (%)	93.9(+/- 0.9)	83.8(+/- 1.4)	87.5 (+/-1.3)
kNN	Spec	F1 Score (%)	93.9(+/- 1.0)	83.8(+/- 1.5)	87.6 (+/-1.3)

Table 17 shows the results for 10 class flight mode classification. LR with PSD representation produces the highest accuracy with no interference, Bluetooth interference and Wi-Fi interference, maintaining over 95% accuracy in all scenarios. Again frequency domain features are shown to be more robust to interference from Bluetooth and Wi-Fi.

Table 17 - DroneDetect 10 Class UAS Flight Mode Classification Accuracy and F1-Score with Interference

Classifier	Image Type	Metric	Clean	Bluetooth	Wi-Fi
LR	PSD	Accuracy (%)	98.0(+/-0.4)	96.4(+/- 0.5)	97.2(+/- 0.3)
LR	PSD	F1 Score (%)	98.0(+/-0.4)	96.4(+/- 0.5)	97.2(+/- 0.3)
LR	Spec	Accuracy (%)	88.6(+/-0.8)	84.9(+/- 0.8)	82.0(+/- 0.7)
LR	Spec	F1 Score (%)	88.6(+/-0.8)	84.9(+/- 0.8)	81.9(+/- 0.7)
kNN	PSD	Accuracy (%)	93.5(+/-0.4)	89.5(+/- 0.8)	92.0(+/- 0.1)
kNN	PSD	F1 Score (%)	93.5(+/-0.4)	89.5(+/- 0.8)	92.0(+/- 0.1)
kNN	Spec	Accuracy (%)	79.7(+/-0.8)	72.0(+/- 0.5)	71.4(+/-0.4)
kNN	Spec	F1 Score (%)	79.7(+/-0.8)	72.0(+/- 0.5)	71.5(+/-0.3)

The results also indicate that the use of the BladeRF and Palm Tree Vivaldi Ultra-wideband Antenna is worth further investigation for low cost UAS detection due to the overall accuracy achieved.

Discussion

In experiment 3 UAS detection and classification in the presence of real world interference from Bluetooth and Wi-Fi signals for 4 UAS classes from the DroneDetecet dataset. Results showed that CNN feature extraction for frequency domain features are more robust than time domain features when interference is introduced. If we only consider UAS detection then accuracy can be maintained at 100% as we introduce Bluetooth and Wi-Fi interference for PSD representation and LR as the classifier. Classification accuracy for 4 class UAS type can be maintained at over 99% again with PSD representation and LR. Accuracy drops by 1.6% when Bluetooth interference is present for 10 class flight mode classification and only 0.7% in the presence of Wi-Fi interference, indicating that Bluetooth signals are more likely to interfere with detection and classification accuracy than Wi-Fi signals.

Further work could include increasing the number of UAS types in the dataset and experimenting with multiple interference sources at once, for example Bluetooth and Wi-Fi together. Detection distance could also be evaluated with a trained model. Lastly, the results in experiment 3 suggest further investigation should be considered using the BladeRF or other low cost SDRs as an option for UAS detection and classification systems.

4.4.2.2 Experiment 4

Overview

In experiment 4 the effect of real-world interference from Bluetooth and Wi-Fi signals concurrently on the full DroneDetect dataset including 7 types of UAS; Air 2 S, Parrot Disco, DJI Inspire 2, Mavic Pro, Mavic Pro 2, Mavic Mini and the Phantom 4. This allows for the assessment of multiple UASs that operate using different transmission systems: Wi-Fi, Lightbridge 2.0, OcuSync 1.0, OcuSync 2.0 and the OcuSync 3.0. 2 class UAS detection, 8 class UAS type classification and 21 class UAS flight mode classification are considered in experiment 4 and with interference from both Bluetooth and Wi-Fi signals concurrently. The CNN considered is a VGG-16 with machine learning classifiers LR and kNN for graphical signal representations PSD and spectrograms.

Results

Table 18 shows that in the presence of concurrent real-world interference Bluetooth and Wi-Fi signals, we can still detect a UAS with 100% accuracy and 100% F1-score using LR and PSD graphical representation. UAS-type classification produces 98.1 (+/-0.4)% accuracy and a 98.1 (+/-0.4)% F1-score, with PSD and LR and UAS flight mode classification achieving 95.4 (+/-0.3)% accuracy and a 95.4 (+/-0.3)% F1-score.

Table 18 - DroneDetect Detection, UAS Type & Flight Mode Classification Accuracy and F1-Score with Interference Results accuracy and F1-score

Classifier	Image Type	Metric	Detection	Type	Flight Mode
LR	PSD	Accuracy (%)	100(+/-0.0)	98.1 (+/-0.4)	95.4 (+/-0.3)
LR	PSD	F1 Score (%)	100(+/-0.0)	98.1 (+/-0.4)	95.4 (+/-0.3)
LR	Spec	Accuracy (%)	96.7 (+/-1.5)	90.5 (+/-0.8)	87.3 (+/-0.4)
LR	Spec	F1 Score (%)	96.7 (+/-1.5)	90.5 (+/-0.9)	87.3 (+/-0.4)
kNN	PSD	Accuracy (%)	96.6 (+/-0.2)	93.5 (+/-0.6)	86.5 (+/-0.5)
kNN	PSD	F1 Score (%)	96.6 (+/-0.2)	93.4 (+/-0.7)	86.3 (+/-0.5)
kNN	Spec	Accuracy (%)	88.0 (+/-1.3)	75.1 (+/-1.5)	64.6 (+/-0.9)
kNN	Spec	F1 Score (%)	87.9 (+/-1.4)	75.3 (+/-1.5)	64.8 (+/-0.8)

Accuracy and F1-score decrease as the classes increase, but high accuracy is maintained for flight mode classification. The table shows that LR outperforms kNN in all the experiments. Time domain features from the spectrogram graphical signal representations are less robust to the interference than frequency domain PSD features are.

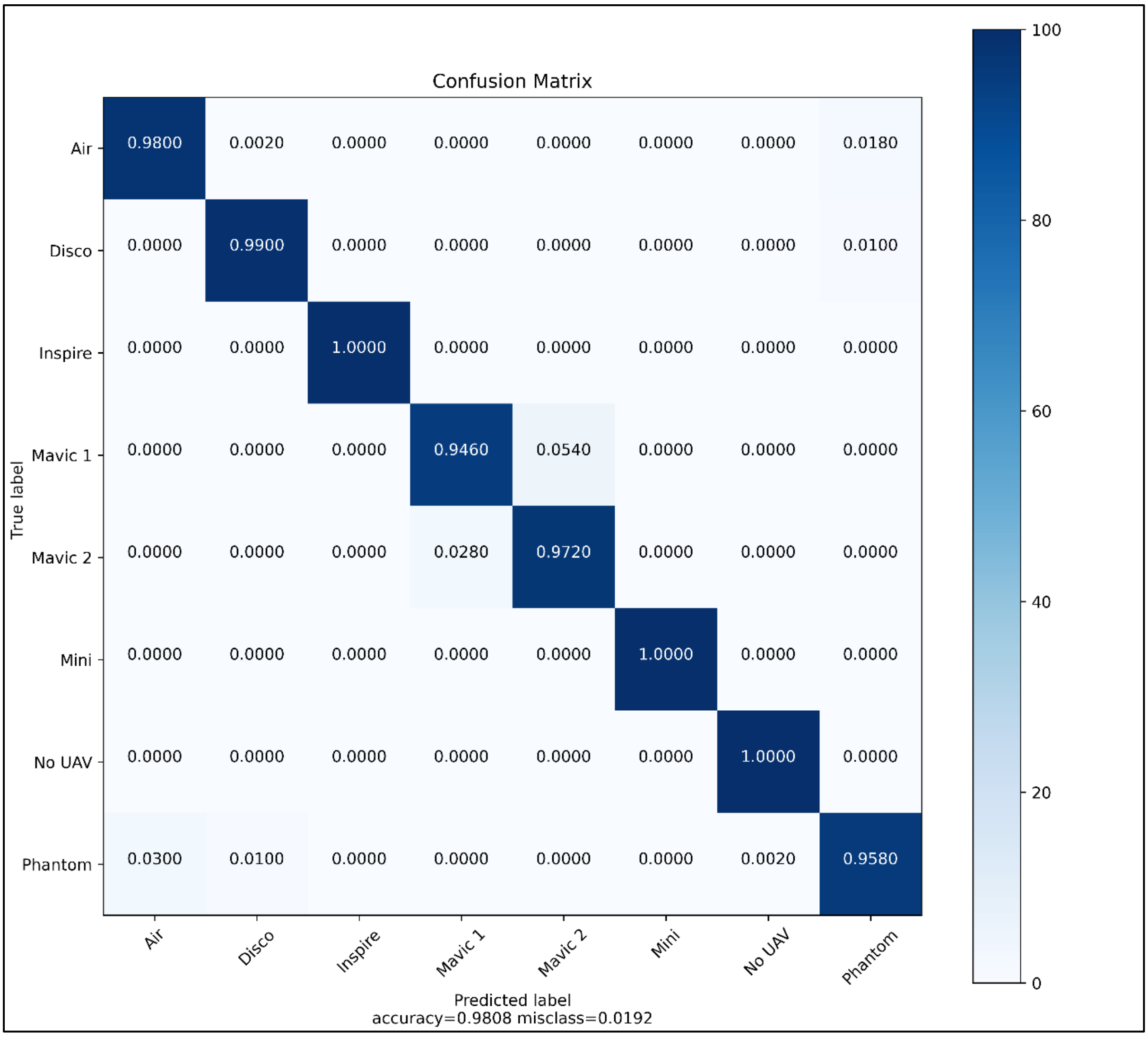
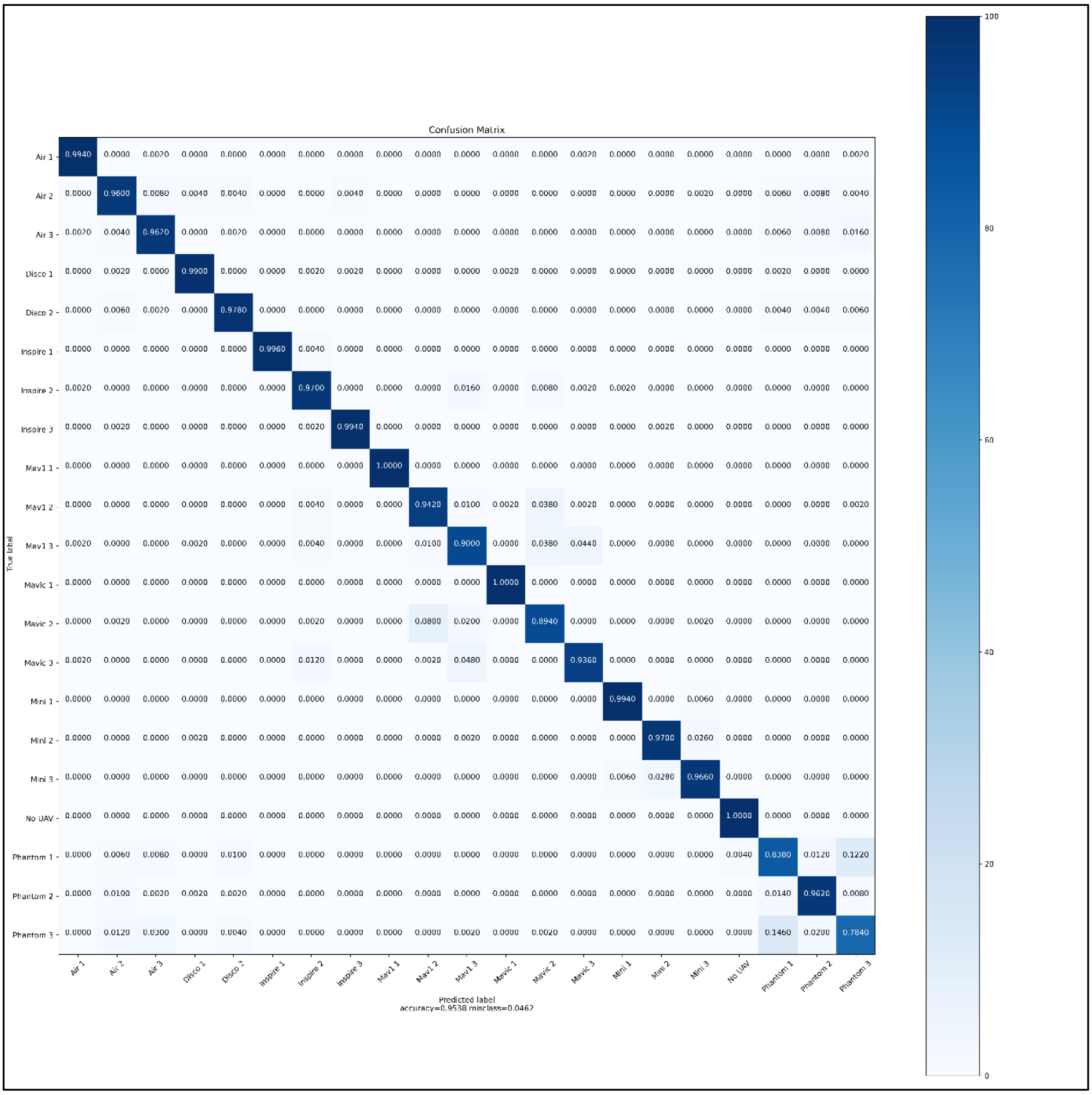


Figure 117 - DroneDetect Interference UAS type classification Confusion Matrix Graphical Signal Representation

Figure 117 shows the confusion matrix for UAS-type classification. We can observe that the classifier has some misclassification between the Mavic Pro and the Mavic 2 Pro. If we go back to Table 18, we can see that the Mavic Pro uses OcuSync 1.0 and that the Mavic 2 Pro uses OcuSync 2.0. The main difference between the two transmission systems is that OcuSync 2.0 utilises both the 2.4 and 5.8GHz frequency bands. The misclassification is likely due to the similar nature of the systems in the 2.4GHz band.



	Air 1	Air 2	Air 3	Disco 1	Disco 2	Inspire 1	Inspire 2	Inspire 3	Mav1 1	Mav1 2	Mav1 3	Mavic 1	Mavic 2	Mavic 3	Mini 1	Mini 2	Mini 3	No UAV	Phantom 1	Phantom 2	Phantom 3
Air 1	0.9940	0.0000	0.0020	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0020	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0020
Air 2	0.0000	0.9600	0.0080	0.0040	0.0040	0.0000	0.0000	0.0040	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0020	0.0000	0.0060	0.0080	0.0040
Air 3	0.0020	0.0040	0.9620	0.0000	0.0020	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0060	0.0080	0.0160
Disco 1	0.0000	0.0020	0.0000	0.9900	0.0000	0.0000	0.0020	0.0020	0.0000	0.0000	0.0000	0.0020	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0020	0.0000	0.0000
Disco 2	0.0000	0.0060	0.0020	0.0000	0.9780	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0040	0.0040	0.0060
Inspire 1	0.0000	0.0000	0.0000	0.0000	0.0000	0.9960	0.0040	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Inspire 2	0.0020	0.0000	0.0000	0.0000	0.0000	0.0000	0.9700	0.0000	0.0000	0.0000	0.0160	0.0000	0.0080	0.0020	0.0020	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Inspire 3	0.0000	0.0020	0.0000	0.0000	0.0000	0.0000	0.0020	0.9940	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0020	0.0000	0.0000	0.0000	0.0000	0.0000
Mav1 1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Mav1 2	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0040	0.0000	0.0000	0.9420	0.0100	0.0020	0.0380	0.0020	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0020
Mav1 3	0.0020	0.0000	0.0000	0.0020	0.0000	0.0000	0.0040	0.0000	0.0000	0.0100	0.9000	0.0000	0.0380	0.0440	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Mavic 1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Mavic 2	0.0000	0.0020	0.0000	0.0000	0.0000	0.0000	0.0020	0.0000	0.0000	0.0800	0.0200	0.0000	0.8940	0.0000	0.0000	0.0000	0.0020	0.0000	0.0000	0.0000	0.0000
Mavic 3	0.0020	0.0000	0.0000	0.0000	0.0000	0.0000	0.0120	0.0000	0.0000	0.0020	0.0480	0.0000	0.0000	0.9360	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Mini 1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.9940	0.0000	0.0060	0.0000	0.0000	0.0000	0.0000
Mini 2	0.0000	0.0000	0.0000	0.0020	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0020	0.0000	0.0000	0.0000	0.0000	0.9700	0.0260	0.0000	0.0000	0.0000	0.0000
Mini 3	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0060	0.0280	0.9660	0.0000	0.0000	0.0000	0.0000
No UAV	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
Phantom 1	0.0000	0.0060	0.0080	0.0000	0.0100	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0040	0.8380	0.0120	0.1220
Phantom 2	0.0000	0.0100	0.0020	0.0020	0.0020	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0140	0.9620	0.0080
Phantom 3	0.0000	0.0120	0.0300	0.0000	0.0040	0.0000	0.0000	0.0000	0.0000	0.0000	0.0020	0.0000	0.0020	0.0000	0.0000	0.0000	0.0000	0.0000	0.1460	0.0200	0.7840

Figure 118 - DroneDetect Interference UAS flight mode classification Confusion Matrix Graphical Signal Representation

Figure 118 shows the confusion matrix for 21 class UAS flight mode classification. We can observe that the misclassification occurs again between the classes of Mavic Pro and Mavic Pro 2, which we can put down to the similarities between OcuSync 1.0 and 2.0. The second area of misclassification occurs between Phantom 4 switched on and Phantom 4 flying. However, the misclassification is small.

Discussion

Overall, experiment 4 has shown that although UASs can prove a serious security challenge, especially in airfield scenarios, detection and classification can be achieved amongst real-world interference. Using CNN feature extraction with transfer learning and machine learning classifiers, UASs operating with the same transmission systems can be distinguished amongst concurrent Bluetooth and Wi-Fi signals. For UAS detection, 100% accuracy can be achieved, and for UAS types and flight mode classification, values of 98.1% (+/-0.4%) and 95.4% (+/-0.3%), respectively, are achieved.

Future work should consider more than one UAS of the same type entering the airspace to evaluate how specific the neural network feature extraction is. Further to this, metrics such as detection distance and detection time can be applied for a trained model to detect and classify a UAS in real time to understand real-world feasibility.

4.4.3 GPS Jamming Dataset Experiments [27]

The next two experiments use the GPS Jamming Dataset described in section 3.1.1 and produced in [31].

4.4.3.1 Experiment 5

Overview

Experiment 5 uses the GPS Jamming Dataset containing the classes AM, FM, DME, Chirp, Narrowband and no jammer present. Transfer learning is applied through feature extraction using a VGG16 CNN pretrained on the ImageNet dataset and machine learning classifiers SVM, LR and RF. To date, prior research in this field has concentrated on spectrogram representation but experiment 5 shows that the novel concatenation of signal representations (PSD, spectrogram, raw constellation and histogram) is more effective, allowing the CNN to benefit from the strengths of each individual representation.

A dataset containing 1000 image representations for each class was constructed for raw constellation; spectrogram; PSD; histogram; and a concatenation of all four. The databases were split 80/20 giving a total of 4,800 images for use with k-fold cross validation and a hold-out evaluation set of 1,200 images. Images were saved with 100 DPI and re-sized to 224x224 pixels. Features were extracted from the dataset and then predictions were made using the best classification models (determined by the hyper-parameter optimisation). This evaluation dataset was not used to train the classification model so provides a final estimate of the performance of the model following training and validation.

Results

Feature maps for the CNN extraction for these results can be seen in section Chapter 4. Table 19 shows results from the nested cross validation (800 images per class). The results show SVM, LR and RF models to perform very closely in accuracy and F1-score compared with each other. The use of all 3 models for classification with the concatenation image database outperforms individual image signal representations. Concatenation of the 4 signal representations (PSD, histogram, spectrogram and raw constellation), classified using SVM and LR achieves 98% (+/- 0.5%) accuracy, RF 96.3% (+/- 0.6%) F1-score with 10 fold cross validation. This is an increase of 3.1% accuracy compared with prior research in this field by Ferre *et al.* which concentrated on spectrogram images.

Table 19 - GPS Jamming Dataset Classification Results Accuracy & F1-Score

		Raw	**Spec**	**PSD**	**Hist**	**Concat**
SVM	***Accuracy (%)***	75.7 (+/2.2)	83.7 (+/1.3)	91.9 (+/1.2)	78.3 (+/1.5)	98.1 (+/- 0.5)
	F1-Score (%)	75.6 (+/2.1)	83.7 (+/1.3)	91.8 (+/1.2)	78.5 (+/1.5)	98.1 (+/- 0.5)
LR	***Accuracy (%)***	75.9 (+/1.5)	84.4 (+/1.8)	92.2 (+/1.3)	77.3 (+/1.1)	98.1 (+/- 0.5)
	F1-Score (%)	76.1 (+/1.5)	84.4 (+/1.7)	92.2 (+/1.3)	77.4 (+/1.1)	98.1 (+/- 0.5)
RF	***Accuracy (%)***	73.9 (+/1.4)	80.8 (+/- 1.7)	91.3 (+/- 1.5)	74.8 (+/- 0.8)	96.3 (+/- 0.6)

	F1-Score (%)	74.0 (+/1.4)	80.6 (+/- 1.7)	91.2 (+/- 1.5)	75.3 (+/- 0.8)	96.2 (+/- 0.6)

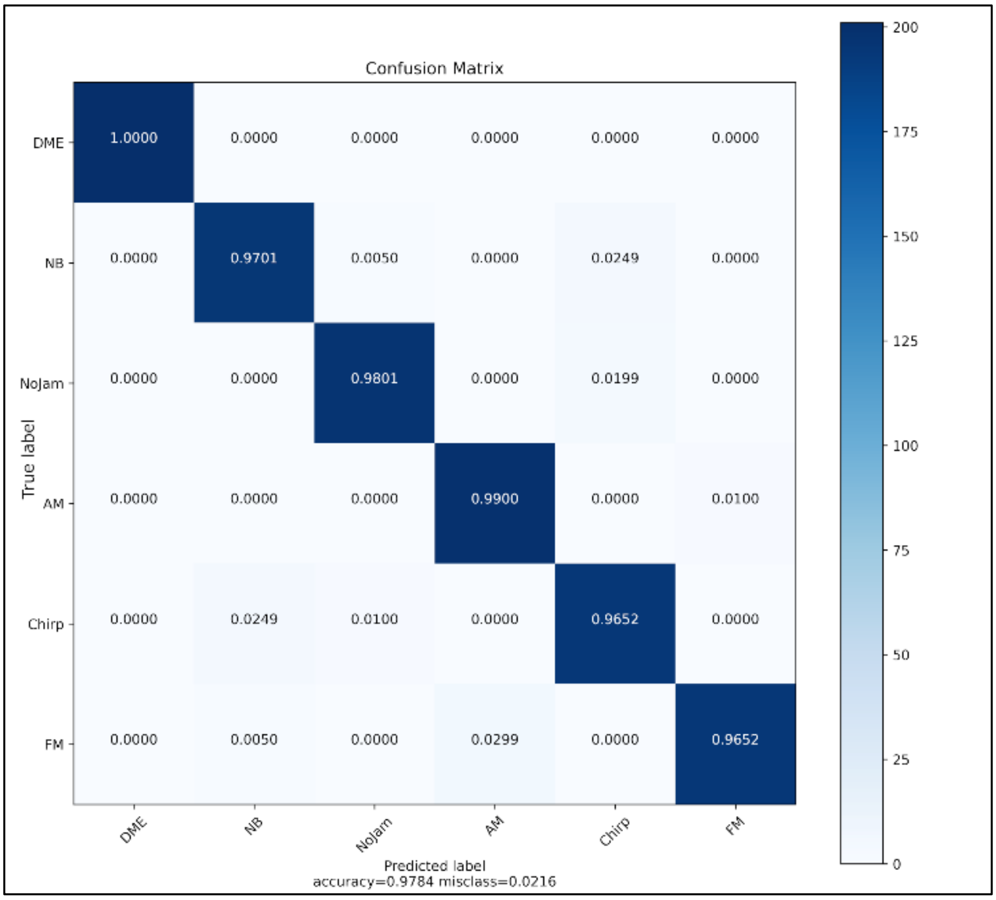
Table 20 shows the mean F1-Scores for individual jamming types using LR as the classifier. The concatenation of the images outperforms the individual representations for all jamming types. It can be seen that different representations can produce higher accuracy results for classifying different individual types of jamming signals (see underlined values). Raw constellation is best at classifying narrowband while spectrograms are best for chirp signals. PSD for classifying no jamming signals, FM and AM, while histogram is best for DME. This helps to understand why concatenating the representations into one image produces the highest results. The concatenation uses the strengths of each individual signal representation.

Table 20 - GPS Jamming Dataset Individual Jamming Types LR F1-Score %

	Raw	**Spec**	**PSD**	**Hist**	**Concat**
DME	98	97	99	99	100
NB	95	86	81	61	97
No Jam	62	91	99	71	98
Single AM	50	73	98	69	98
Single Chirp	58	86	82	77	96
Single FM	85	74	97	94	98

Figure 119 shows the Confusion Matrix for the concatenation dataset within the evaluation data for LR.

The evaluation data results back up the cross validation scores which indicates that the model does generalize to new data and is not overfitting. LR performs the best out of the 3 models at 97.8% accuracy but the difference is marginal (0.3% difference SVM and 2.3% difference RF). Figure 119 shows that the model can detect whether a signal is present or not with an accuracy of over 98%. However, considering only detection of a signal, RF produced the best accuracy of 99%.



	DME	NB	NoJam	AM	Chirp	FM
DME	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
NB	0.0000	0.9701	0.0050	0.0000	0.0249	0.0000
NoJam	0.0000	0.0000	0.9801	0.0000	0.0199	0.0000
AM	0.0000	0.0000	0.0000	0.9900	0.0000	0.0100
Chirp	0.0000	0.0249	0.0100	0.0000	0.9652	0.0000
FM	0.0000	0.0050	0.0000	0.0299	0.0000	0.9652

Figure 119 - GPS Jamming Dataset LR Confusion Matrix

Discussion

In experiment 5 we demonstrated a novel approach to the classification of GNSS jamming signals by considering a concatenation of PSD, spectrogram, raw constellation and histogram representations of the signal as an image classification problem. Each signal representation was shown to have a particular strength, for example raw constellation was best for classifying narrowband signals while spectrograms were best for DME signals. The concatenation uses the strengths of each individual signal representation to produce the highest accuracy in the results.

Using a transfer learning approach with a VGG16 CNN pre-trained with ImageNet weights we have shown the concatenation approach achieves a mean classification accuracy of 98% (+/-0.5%). This outperforms previous research in the field which has concentrated on spectrogram image representation. Further, the process of transfer learning allows us to achieve high accuracy without needing a very large dataset of signals. Evidence from our experiments demonstrates that PSD produces higher classification accuracy (over 8%) than spectrograms when used as individual signal representation images. Our research was validated using 10 fold cross validation, with 3 fold nested cross validation for hyper-parameter optimisation. A holdout evaluation dataset, not used in training and cross validation, confirmed the concatenation dataset to perform with accuracy at 97.8% for LR, 97.5% for SVM and 95.4% for RF.

Further work could include field testing with real GNSS jammers to understand how useful synthetic training sets are. Other types of transfer learning such as fine tuning a CNN VGG16 which has been pretrained with ImageNet or other databases such as ResNet could also be considered. Overall the results in experiment 5 have shown a novel approach using transfer learning which has robustness and high accuracy in GNSS jamming signal classification and outperforms previous work in the field.

4.4.3.2 Experiment 6

Overview

In experiment 6 the GPS Jamming dataset was produced for SNR levels 30dB, 10dB, -10dB and -20dB where each class was made up of 1000 images 224x224 pixels. Each class was further split 800 for use with k fold cross validation and 200 kept entirely separate to validate the model. The ResNet-50 was the CNN used for feature extraction . Graphical signal representations considered include PSD, spectrogram, raw IQ constellation, histogram and a concatenation of all four.

The dataset of graphical signal representations are fed to a CNN for feature extraction. The features are then presented to machine learning classifiers Random Forest and SVM which were . SVM was employed with the hyper-parameter optimisation for regularisation values between 10 and 0.001 and Random Forest was set to 100 trees for the experiments using the built in bootstrapping process for hyper-parameter optimisation.

Results

Table 21 below shows the results with 5 fold cross validation. We can observe each individual signal representation for varying levels of SNR from 30dB down to a very low SNR of -20dB. We can observe that for all types of signal representation; raw, spectrogram; PSD, histogram and concatenation, we see that the accuracy goes down as the SNR decreases. SVM outperforms Random Forest in all instances. However the difference is marginal when the concatenated graphical signal representation is used. SVM is considerably slower to train that Random Forest so the advantages should be weighed up depending on the priorities of the system as Random Forest still produces high accuracy at low SNR with the correct input.

Table 21 - GPS Jamming Dataset Classification Accuracy at set SNR Levels

	SNR (dB)	**Raw**	**Spec**	**PSD**	**Hist**	**Concat**
SVM	***30***	97.2 % (+/-0.8)	99.6 % (+/-0.0)	99.6 % (+/-0.1)	99.6 % (+/-0.2)	99.9 % (+/-0.1)
	10	97.2 % (+/-0.3)	99.3 % (+/-0.1)	99.6 % (+/-0.2)	99.4 % (+/-0.3)	99.9 % (+/-0.1)
	-10	90.9 % (+/-0.6)	79.4 % (+/-1.5)	86.9 % (+/-0.3)	87.2 % (+/-0.8)	97.2 % (+/-0.7)
	-20	50.1 % (+/-1.2)	40.6 % (+/-1.3)	67.1 % (+/-1.3)	50.2 % (+/-1.8)	72.8 % (+/-1.2)
RF	***30***	96.2 % (+/-0.1)	98.7 % (+/-0.3)	99.7 % (+/-0.2)	98.9 % (+/-0.1)	99.9 % (+/-0.2)
	10	97.2 % (+/-0.3)	97.7 % (+/-0.4)	99.5 % (+/-0.2)	99.0 % (+/-0.5)	99.9 % (+/-0.1)
	-10	84.4 % (+/-1.3)	74.6 % (+/-1.5)	84.5 % (+/-0.7)	81.6 % (+/-0.9)	94.0 % (+/-1.5)
	-20	46.7 % (+/-0.6)	41.9 % (+/-1.7)	65.7 % (+/-0.5)	45.7 % (+/-1.7)	66.9 % (+/-1.6)

In order to consider generalisation and be sure that the model was not over fitting a separate dataset of 200 images per class was reserved to validate the system post cross validation test and training. When we compare the results in Table 22 to Table 21 we can see that the accuracy measures are roughly the same. This means that the model is not over fitting, it is able to generalise and works on data it hasn’t seen before.

Table 22 - GPS Jamming Dataset Validation Results Classification at set SNR Levels

	SNR (dB)	**Raw**	**Spec**	**PSD**	**Hist**	**Concat**
SVM	***30***	96.8 %	99.7 %	99.8 %	99.6 %	99.9 %
	10	97.4 %	99.5 %	99.7 %	99.5 %	100 %
	-10	90.7 %	79.2 %	87.1 %	89.0 %	97.3 %
	-20	52.5 %	42.3 %	65.6 %	50.7 %	74.7 %
RF	***30***	96.6 %	98.7 %	99.5 %	98.5 %	100 %
	10	97.6 %	97.4 %	99.2 %	99.0 %	99.9 %
	-10	86.5 %	74.1 %	83.6 %	82.6 %	95.1 %
	-20	48.3 %	41.7 %	63.8 %	44.8 %	68 %

Figure 120 and Figure 121 show the confusion matrix for machine learning classifier SVM using the concatenation of graphical signal representations at -10dB SNR and -20dB SNR respectively. We can observe at -10dB the classifier is starting to have very minor misclassifications between NB and chirp signals, and also FM and AM signals.

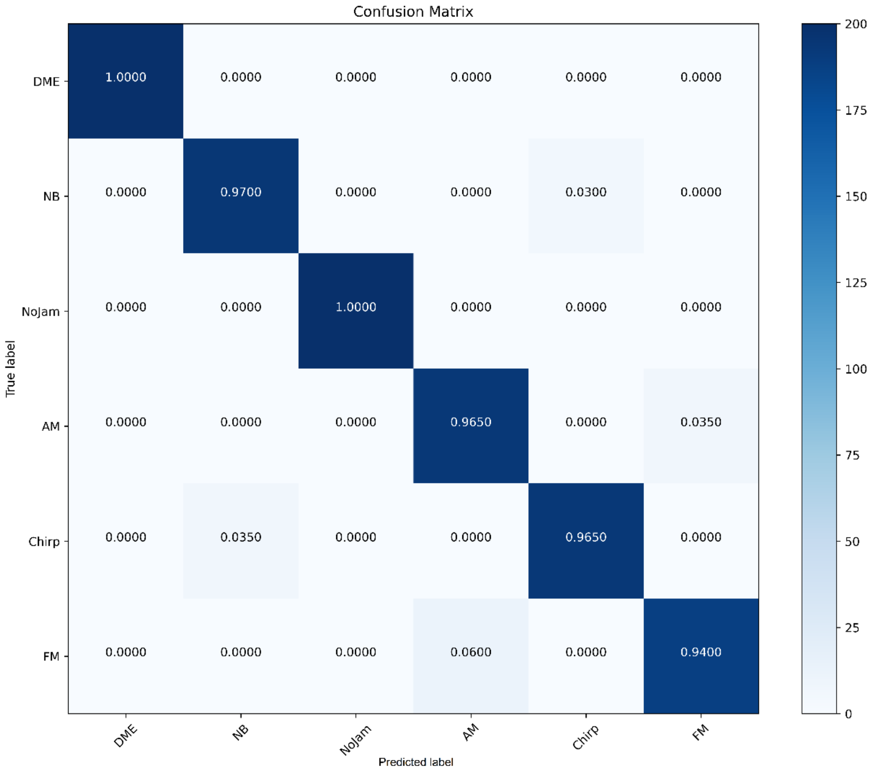


Figure 120 - GPS Jamming Dataset Confusion Matrix SVM Concatenation 10dB SNR

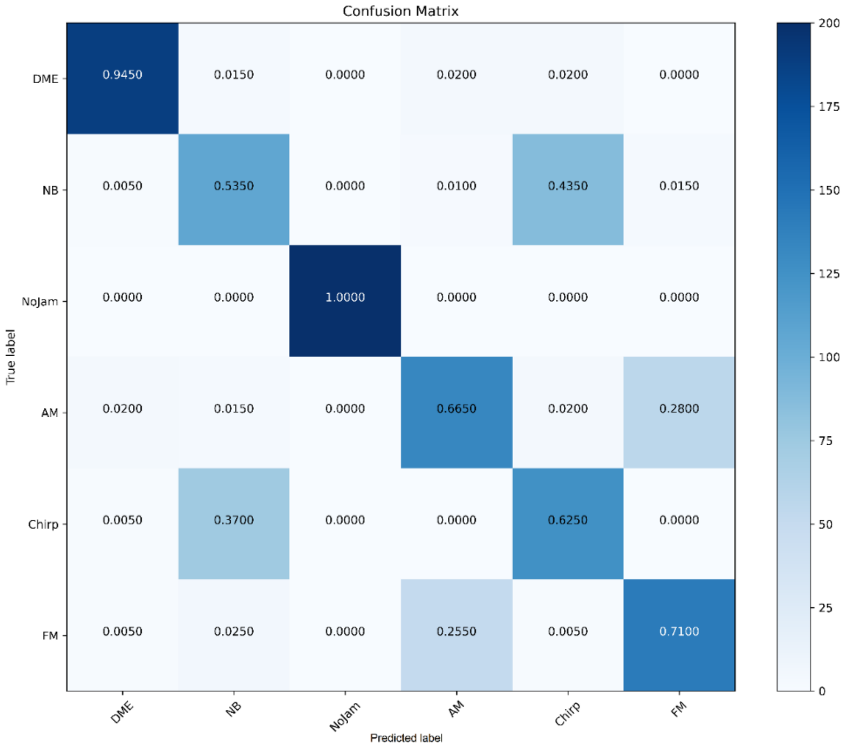


Figure 121 - GPS Jamming Dataset Confusion Matrix SVM Concatenation -20dB SNR

Figure 121 shows that as the SNR decreases to -20dB those mis-classifications become larger between NB and chirp and also between FM and AM jamming signal types. To understand why the concatenation of signal representation produces higher accuracy results we have shown individual accuracy scores in Table 23 for machine learning classifier SVM at -20dB SNR. It can be seen that a spectrogram produces the highest accuracy for DME jamming signals and raw constellation produces the highest accuracy for FM jamming signal types. By concatenating the signal representations into one image which is then fed to the CNN for feature extraction, the CNN is able to utilise all the strengths of the individual signal representations.

Table 23 - GPS Jamming Dataset Invidiual Accuracy -20dB SNR

	Raw	**Spec**	**PSD**	**Hist**	**Concat**
DME	40 %	**95** %	84 %	36 %	95 %
NB	32 %	28 %	**45** %	38 %	55 %
No Jam	**100** %	32 %	**100** %	**100** %	100 %
AM	45 %	47 %	**61** %	30 %	68 %
Chirp	30 %	27 %	**47** %	34 %	59 %
FM	**70** %	27 %	56 %	64 %	71 %

If the results from a high SNR environment of 30dB in Table 21 are compared with previous work in [27] for the concatenation of signal representations, a deeper neural network Resnet50 produces higher accuracy results. In [27] the VGG-16 CNN was used for feature extraction and produced 98% (+/- 0.5%) while this research has shown the ResNet50 to increase that to 99.9% (+/-0.1%). Further, if this research is compared to [23] it shows that machine learning classifiers SVM and Random Forest are able to utilise the strengths of each individual representation of the signal through concatenation and continue to perform well on data the model hasn't seen before.

Discussion

Overall experiment 6 shows that the use of a deep CNN architecture and transfer learning for feature extraction, in conjunction with machine learning classifiers SVM and Random Forest, can produce high accuracy results in low SNR environments. The highest accuracy results are produced when graphical signal representations spectrogram, histogram, PSD and raw constellation are concatenated together so the CNN can benefit from the strengths of each individual representation. This is more apparent in low SNR environments where the concatenation representation can produce 75% accuracy at -20dB SNR. A separate dataset used to validate the model shows that there was no overfitting present in the results.

Future work could look to expand the jamming classifications and look to test the model with real signals and detection/classification distances. However, this is problematic in some countries for example in the UK where legalities would prohibit these trials. In today's economy GPS technology is only increasing in dependencies from 5G to driverless cars so the issue of jamming detection and classification remains of the upmost importance.

4.4.4 JamDetect Dataset Experiments [23]

4.4.4.1 Experiment 7

Overview

The JamDetect dataset is produced containing 6 different types of commercial jamming signals including chirp, continuous wave (CW), barrage, narrowband, pulse and protocol aware jammers. It is produced with set levels of SNR and details of this process are described in section 3.1.2. Experiment 7 considers a VGG-16 as the CNN utilising transfer learning for feature extraction and machine learning classifier LR for classification. Graphical signal representations considered include PSD, spectrogram, raw IQ constellation, histogram and a concatenation of all four. Datasets were created for each graphical signal representations and the concatenated images at SNRs 50dB, 30dB, 10dB, -10dB and -20dB to understand the effect on classification accuracy. Datasets were created of 1000 images of size 224x224 for use with 5-fold cross validation. 5-fold cross validation was chosen due to the work of [360] who found that either k=5 or k=10 both showed empirically errors rates which displayed neither high variance or high bias. K=5 is less computationally heavy than k=10 so it was the choice for the experiments.

Results

Table 24 shows the training/test 5-fold cross validation results. PSD produces the highest accuracy when SNR is reduced. It can be seen that as the SNR reduces from 50 downwards, but a decrease in accuracy is not seen until the SNR drops below 10dB. This allows us to make the assumption that evaluating the signal in the frequency domain is less susceptible to noise and able to still identify and classify the signal. When the time domain via the spectrogram graphical representation is considered it can be seen that it maintains accuracy levels up to -10dB SNR but significantly decreases below this, this is a significant drop when compared to the PSD graphical signal representation. The raw constellation is shown to be the most susceptible to noise and has the lowest performing accuracy scores. This indicates that the raw constellation data may produce higher accuracy results with some pre-processing which could include processes such as filtering. Concatenating the different signal representations seems to also inherit the low accuracy levels seen for the raw constellation and histogram representations at lower SNR levels which is logical.

Table 24 - JamDetect Classification Results Set Levels SNR Accuracy (%)

SNR (dB)	PSD Accuracy (%)	Spectrogram Accuracy (%)	Raw Accuracy (%)	Hist Accuracy (%)	Concat Accuracy (%)
50	100	100	83.3 (+/- 0.1)	100	100
30	100	100	66.7 (+/- 0.1)	99.9 (+/-0.1)	100
10	100	100	76.8 (+/- 1.5)	94.3 (+/-0.2)	100
-10	100	99.6 (+/- 0.3)	50.2 (+/- 1.2)	50.7 (+/-0.7)	100
-20	82.7 (+/-0.7)	40.0 (+/-0.6)	25.8 (+/-1.3)	22.6 (+/-0.8)	74.0 (+/- 1.5)

Table 25 shows the validation results. The most significant finding from the validation results is with respect to the concatenated signal representations. Although effective at SNR levels of 10dB and above, the validation scores show that the concatenation of the signal representation presents overfitting at -10dB and below. This means that the model was learning the training data so well that the model did not generalise when it was given new samples in the validation set. A slight increase at the mid SNR level is also seen which indicates that the model produces higher accuracy with some level of noise present. This indicates that the model can generalise well with a certain amount of noise present.

Table 25 - JamDetect Validation Classification Results Set Levels SNR Accuracy (%)

SNR (dB)	PSD Accuracy (%)	Spectrogram Accuracy (%)	Raw Accuracy (%)	Hist Accuracy (%)	Concat Accuracy (%)
50	100	100	83.4	94.1	96.1
30	100	100	66.8	100	84.1
10	100	100	78.5	94.2	100
-10	100	99.7	51.1	50.1	55.8
-20	81.7	40.1	27.1	23.9	19.4

The validation results also confirm the training and test results that the PSD produces the highest accuracy scores in low SNR environments and therefore is the least susceptible to noise.

Discussion

Overall our results have shown that the PSD graphical signal representation is the least susceptible to noise and produces the highest accuracy in low SNR environments. While our previous work showed that concatenating various graphical signal representations together as the input for the CNN, this extension of that work has shown this to be effective only in environments with SNR levels higher than -10dB. This is significant for congested environments where it is vital to know GPS jamming is being attempted. In low SNR environments PSD should be utilised for GPS jamming classification.

Further work could include exploring deeper architectures and types of neural networks for feature extraction. Experimentations with real GPS jamming signals which includes distance of detection and classification should also be considered for future exploration. Overall experiment 7 has shown that PSD graphical signal representation provides the highest accuracy for GPS jamming classification in low SNR environments with CNN feature extraction and machine learning classifier logistic regression.

Chapter 5 - Unsupervised Learning

This section will cover the unsupervised algorithm experiments. How this work fits into the other chapters discussed so far is highlighted by the red box in Figure 122 below. The work in this chapter and the results presented in section 5.5.2 have been published early access with IEEE Transactions on Intelligent Transportation Systems [19].

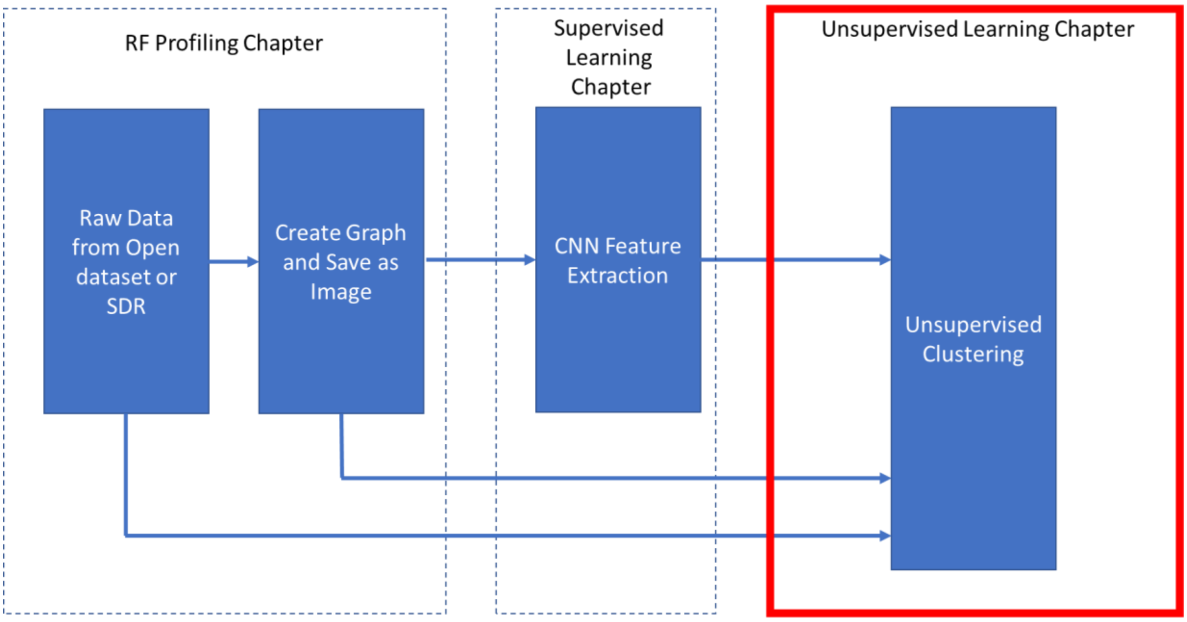

RF Profiling Chapter
Supervised Learning Chapter
Unsupervised Learning Chapter
Raw Data from Open dataset or SDR
Create Graph and Save as Image
CNN Feature Extraction
Unsupervised Clustering

Figure 122 - Unsupervised Learning Chapter within larger Thesis Construct

5.1 Clustering

Human beings cluster objects on a daily basis and from a very early age. For example the CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) determines whether the user is a human by presenting it with a challenge response problem such as identifying pictures with chimneys in them. In a similar manner, clustering uses patterns found in the data to group data objects. The same set of data can be clustered in different ways depending on the numbers of clusters which are defined and depending on how the type of clustering algorithm used. The main two types of clustering are hierarchical and partitional.

As the name suggests hierarchical clustering produces a cluster hierarchy, also known as a dendrogram. Within the structure clusters also contain sub-clusters. These sub clusters can be determined either with a bottom up approach using similarity (assuming all the data points are individual) or through a top down approach which starts with one clusters and with each step the cluster is split down further [362]. The issue with hierarchical clustering lies with the sheer amount of data which is available to process for different problems, there are so many possibilities that locating a optimum by investigation all of them is very difficult [363].

Partitional clustering or non-hierarchical clustering gives a way of dealing with this issue using a heuristic approach that takes a global optimum and performs the optimization from there. It doesn’t follow the hierarchical structure that can resemble a tree but rather the data is grouped by a criteria such as similarity but with no hierarchical connections. K-means clustering is a popular type of partitional clustering.

5.2 K-Means Clustering

Unsupervised machine learning algorithm K-means clustering does not need training data labels. Instead unsupervised learning works by letting the algorithm work out the underlying inherent patterns in the data. K-means was chosen as in comparison to other clustering models as it is fast and has been proven robust [364]. K-means clustering uses a metric of similarity to group the data and then that group is represented using a centroid. When the model is presented with new information it will assign the data to its closest centroid and therefore group. Similarity is found in these experiments by using the Euclidean distance. Equation (23) below shows the formula for calculating the Euclidean distance d_{ij} [365].

$$d_{ij} = \sqrt{\sum_{k=1}^{n}(x_{ik} - x_{jk})^2} \tag{23}$$

In equation (23) n represents the number of vectors and x_{ik} and x_{jk} are the two data points being compared. There are three parts to using k-means clustering. First a value must be selected for k which defined the number of initial centroids or cluster means. These initial values can be chosen at random or by other means which will be discussed in section 5.3. When those initial centroids are chosen from the data, the rest of the points are then assigned to each one depending on the similarity measure discussed in equation (23). The k-means algorithm then calculates new centroids based on the new mean of each other clusters. The process it iterative from there on, with the new centroid value calculation being followed by the data points being again assigned to their closest centroid. The process repeats until the centroids no longer change from one iteration to the next and they therefore achieve convergence.

5.3 K-Means Initialisation

One of the key factors which has been shown to determine how well K-means performs is to do with how the centroids are initialised [366]. This experiments in this thesis will consider the most common initialisations which include k-means++, principal component analysis (PCA) dimensionality reduction and random.

K-means++ uses the probability proportional to the squared distance after selecting the first centroid randomly. The effect of this is that the centroids are moved as far away as they can be from each other [367]. Initialisation which is random works by choosing random points from the dataset and using an average distance between the centroid and the point [368].

PCA is a method of reducing dimensionality but keeping the feature information which is deemed important [369] [370]. Due to the dimensionality being reduced, PCA can produce very quick results compared with the other initialisations. PCA was tested by Ding and He [371] who found that PCA reduced data contained the same information which produced the same centroids for K-means. We are using PCA to reduce the dimensionality by projecting the data into an n dimensional space (where n is equal to the number of known classes). Then we are using those components of the PCA as the initialisation method which is deterministic. The method has also been tested by Li and Li [372] who compare PCA initialisation to k-means++ and random initialisation using the handwritten digits dataset [373]. Li and Li showed that PCA gave the same cluster centroids, similar accuracy to k-means++ and random initialisations, but performed in a much quicker time.

K-means was chosen as the clustering method due to its speed and we can use the value of k to help indicate new platforms when other known systems are in use on an airfield. This is due to the fact that UASs are being used increasingly for legitimate purposes. For example on airfields small UASs are being used for security, being deployed quickly to go and check a fence perimeter if an alarm is triggered [374]. They are used for the detection of runway debris, building inspections and controlling wildlife [375]. Early warning systems for UASs are normally made up of numerous subsystems and potentially different sensor types. So if for example, we knew that we had 4 different UASs operating on an airfield and a radar identified a potential UAS flying near the runway, an unsupervised learning algorithm could be set to $k + 1$, so in this instance $k = 5$, to provide a quick indication and confirmation of whether the radar was picking up something of concern, i.e. a new cluster. That could then trigger a supervised learning algorithm of higher accuracy which will however take longer to determine a result. The clustering will have provided the vital information, that it is highly likely there is a new platform operating in a restricted airspace. The value comes from how quickly the clustering can give a result as an indicator and this is the reason why k-means clustering has been chosen for these experiments.

5.4 Performance Metrics

As stated previously the point of unsupervised learning is to allow the algorithm to find inherent patterns in the data by not giving any label information. However, if label information is available it can be used to check and understand how well the model has performed. For our datasets we have the label information so this can be used to check how well the model is performing against our ground truth labels by using clustering quality metrics. The first is called v-measure score and it is a harmonic of homogeneity score and completeness score. Homogeneity looks of whether the clusters only contain members of a single class and completeness looks at whether all points in a class belong to the same cluster [376]. A 1 represents perfect scoring between the label and the prediction, and equation (24) shows how the v-measure score is calculated from the two.

$$v - measure = \frac{(1 + \beta)(homogeniety)(completeness)}{\beta(homogeneity + completeness)} \tag{24}$$

When β is less than 1 more weight is assigned to homogeneity, when greater than q more weight is assigned to completeness. A perfect score is defined as $v = 1$. The Adjusted Rand Index (ARI) and the Adjusted Mutual Information (AMI) consider the difference between the label and the sample cluster with AMI normalising against chance. A 0 would indicate random labels and 1 indicating perfect matches.

5.5 Results

Supervised learning techniques applied to RF signals have been considered for the classification of UAS type with high accuracy but due to labelled data assume the UAS signal is already known. Unsupervised learning algorithms such as K-means clustering provide a potential for clustering small UAS signals which have not been seen before for this problem set. The use of transfer learning and CNN feature extraction with spectrogram graphical signal representations have been successfully used in a supervised manner for medical diagnosis and audio classification. This research is the first application of transfer learning and CNN feature extraction as a pre-cursor to an unsupervised learning algorithm for use with UAS RF signals (or GPS jamming signals). Table 26 shows a key for all the experiments in this section with experiments 1 representing the raw images – either in PSD or spectrogram form being presented for k-means clustering. For experiments 2 and 3 the images have been pre-processed with a pre-trained VGG-16 and ResNet-50 respectively for feature extraction. Then the features are presented for k-means clustering.

Table 26 – K-Means Experiments Key

Experiment	Graph Type	Method
1	PSD or Spectrogram	Raw Image
2	PSD or Spectrogram	VGG-16 Feature extraction
3	PSD or Spectrogram	ResNet-50 Feature extraction

5.5.1 DroneRF Dataset

5.5.1.1 Overview

The DroneRF dataset as described in section 3.1.3 and produced by Al-S’ad *et al.* [204] covers 80MHz of the 2.4GHz frequency band using 2x SDRs. The dataset includes the Parrot AR (Elite 2.0), Parrot Bebop, no UAS present and the DJI Phantom 3. On Figure 127 to Figure 130 the cluster numbers 0-3 represents the AR, Bebop, No UAS and Phantom 3 respectively. On Figure 123 to Figure 126 cluster 0 represents no UAS present and cluster 1 represented a UAS being active in the area. Signals were represented in the time domain as a spectrogram and in the frequency domain as a Power Spectral Density (PSD). This would let us evaluate whether viewing the data in the time or frequency domain produced higher accuracy results for classification. Python 3 Matplotlib was used to plot the spectrogram and PSD images both with 1024 FFT size, Hanning windowing functions and centre frequency 2.442GHz covering a range of 2.402GHz – 2.482GHz. Images were saved as size 224x224 pixels with 800 images per class. For the VGG-16 feature vectors of 25, 0888 values were produced and for the ResNet-50 features vectors of 100,352. Experiment 1 uses the raw images with K-means clustering and experiments 2 and 3 VGG-16 and ResNet-50 feature extraction respectively.

5.5.1.2 Results – Clustering (UAS Detection)

Table 26 shows a key for understanding the experiments presented in tables Table 31 and Table 32. Experiments 1 represent raw images, experiments 2 VGG-16 features extraction and experiments 3 ResNet-50 features extraction.

Table 27 - DroneRF K-Means Clustering Detection Results UAS

Exp.	Image	Init	Time (s)	homo	compl	v-measure	ARI	AMI
1	PSD	kmeans++	0.423	0.982	0.982	0.982	0.992	0.982
1	PSD	random	0.104	0.982	0.982	0.982	0.992	0.982
1	PSD	PCA	0.126	0.982	0.982	0.982	0.992	0.982
1	SPEC	kmeans++	0.441	0.001	0.001	0.001	0.001	0.001
1	SPEC	random	0.155	0.001	0.001	0.001	0.001	0.000
1	SPEC	PCA	0.147	0.001	0.001	0.001	0.001	0.001
2	PSD	kmeans++	2.664	0.970	0.970	0.970	0.987	0.970
2	PSD	random	2.086	0.970	0.970	0.970	0.987	0.970
2	PSD	PCA	1.424	0.970	0.970	0.970	0.987	0.970
2	SPEC	kmeans++	2.930	0.187	0.255	0.145	0.216	0.100
2	SPEC	random	3.483	0.210	0.272	0.175	0.236	0.098
2	SPEC	PCA	1.745	0.206	0.267	0.173	0.233	0.099
3	PSD	kmeans++	10.785	0.986	0.986	0.986	0.995	0.986
3	PSD	random	9.470	0.986	0.986	0.986	0.995	0.986
3	PSD	PCA	6.031	0.986	0.986	0.986	0.995	0.986
3	SPEC	kmeans++	17.291	0.026	0.026	0.026	0.035	0.025
3	SPEC	random	13.115	0.028	0.028	0.028	0.028	0.028
3	SPEC	PCA	9.256	0.024	0.024	0.024	0.024	0.023

Table 27 shows the results for the clustering of UAS detection for experiment 1 - raw images (PSD and spectrogram) and experiments 2 and 3 whereby the PSD and spectrograms were preprocessed using a VGG-16 and ResNet-50 feature extraction respectively. Table 27 shows that v-measure scores for PSD significantly outperform spectrograms – for raw images and both types of CNN feature extractors. Random initialisation outperforms k++ and PCA generally for inference time, while for v-measure PCA and random produce the highest scores. ARI and AMI mimic the trends seen with v-measure scores.

Overall the best performing combination of initialisation and image with or without CNN feature extraction was PSD graphical signal representation with a REsNet-50 feature extractor, producing a 0.986 v-measure score. Initialisation did not affect v-measure score but if the initialisation method was to be chosen by quickest inference time then PCA was fastest at 6.031 seconds. However, PSD raw images without feature extraction performed nearly as high on v-measure at 0.982, again with all v-measure scores for each initialisation begin the same. The quickest inference time was seen with the random initialisation at 0.104 of a second, significantly quicker than using the ResNet-50 feature extractor.

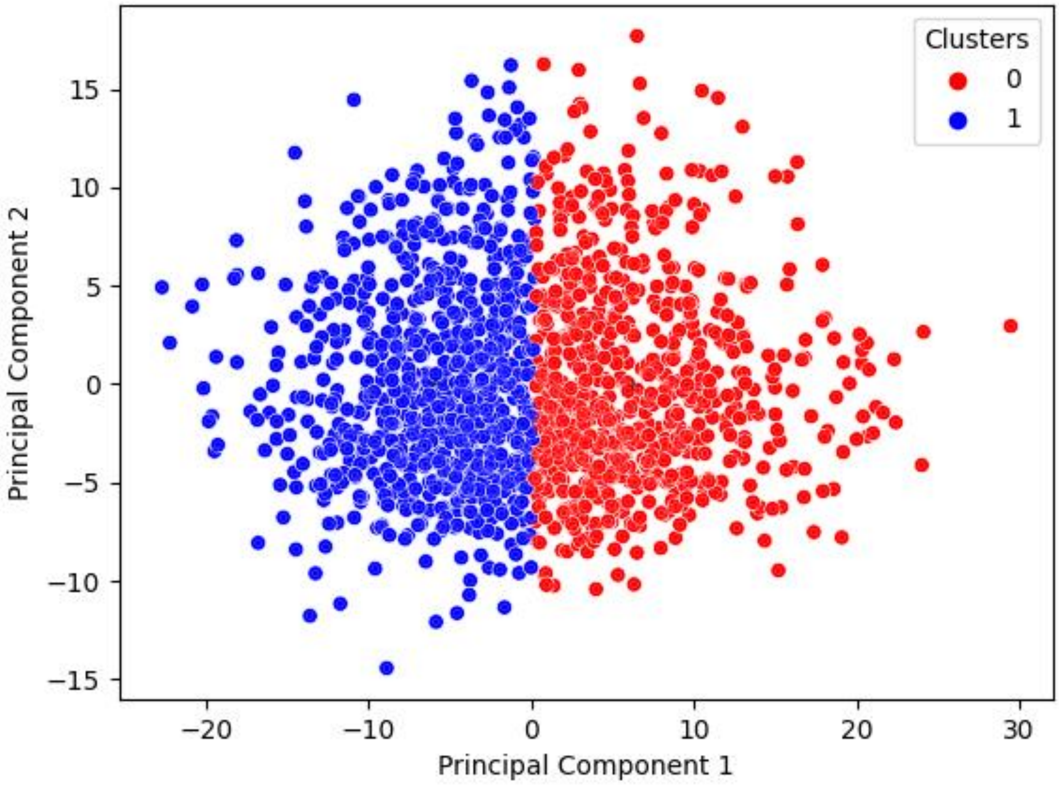


Figure 123 - DroneRF K-Means Detect Spectrogram Graphical Signal Representation Raw Image

Figure 123 shows the first two principal components plotted on a scatter graph for spectrogram raw images with PCA initialisation of K-means. This produced a very low v-measure score of 0.001 indicating that k-means was unable to cluster between UAS and no UAS. This is also seen visually in Figure 123 were the overlap between cluster 0 and cluster 1 can be observed as they meet in the middle of the graph.

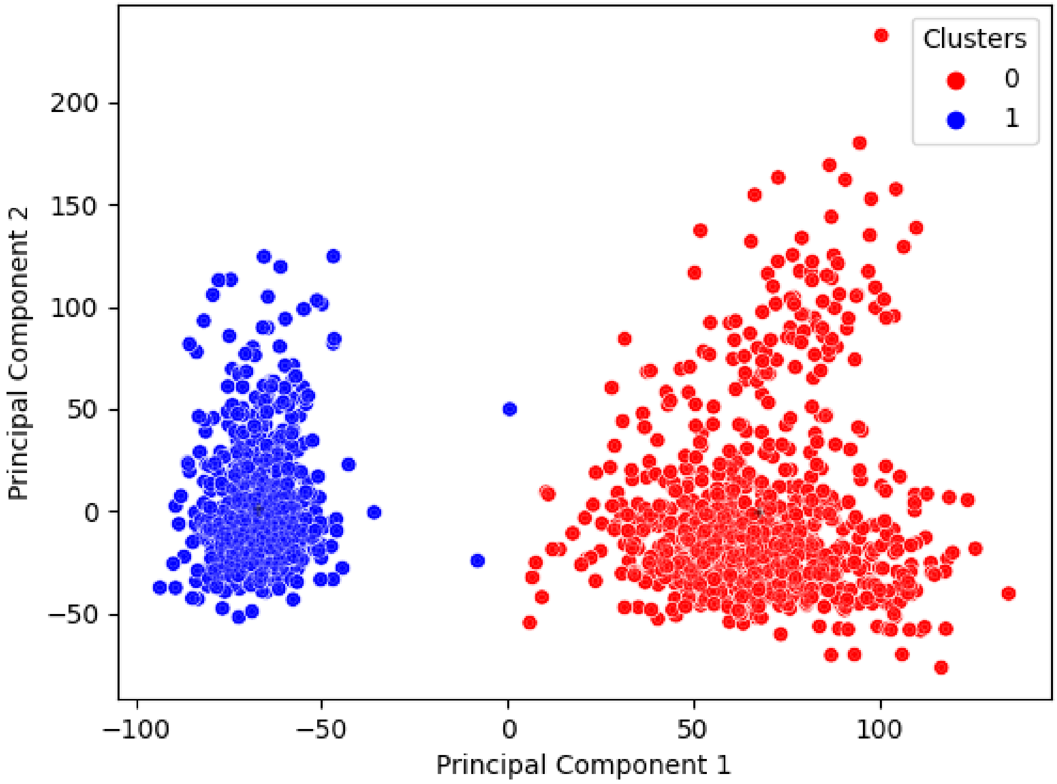


Figure 124 - DroneRF K-Means Detect PSD Graphical Signal Representation ResNet-50 Feature Extraction

A clear difference can be seen in Figure 124 when it is compared to Figure 123. Figure 124 shows principle component 1 and 2 for PSD graphical signal representations pre-processed with ResNet-50 feature extraction. There is good separation between no UAS (cluster 0) and UAS (cluster 1), with the UAS signals having a higher concentration of grouping. The separation is reflected in the v-measure score of 0.986, the highest performing combination which was tested.

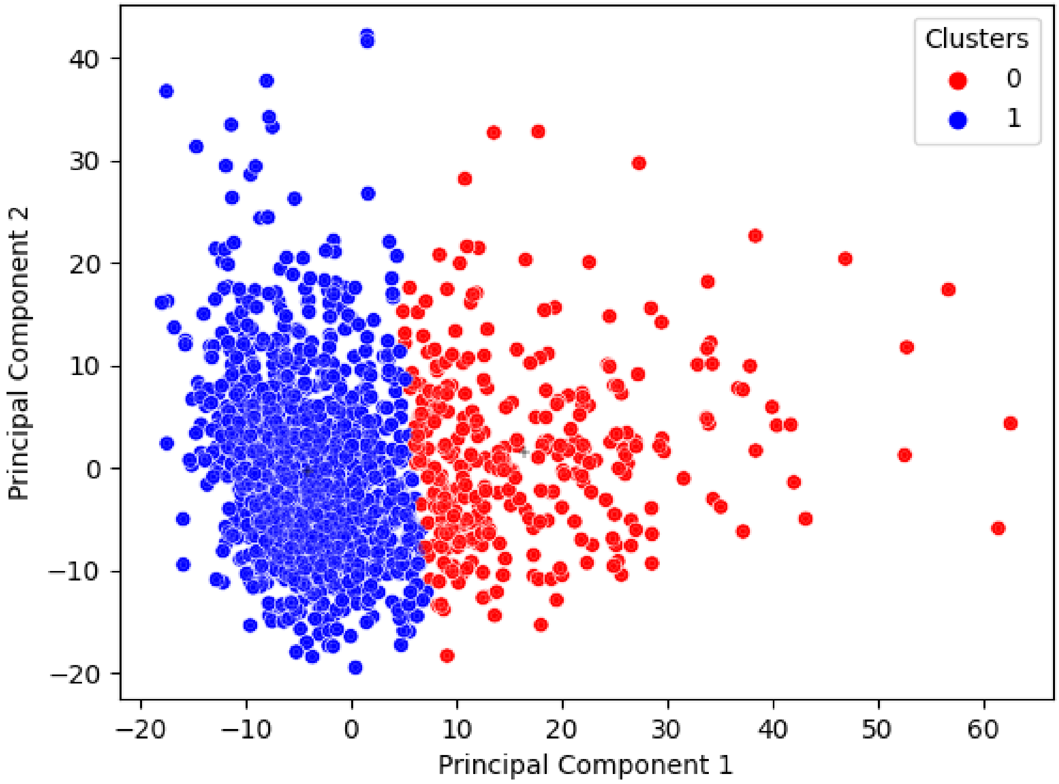


Figure 125 - DroneRF K-Means Detect Spectrogram Graphical Signal Representation VGG-16 Feature Extraction

Figure 125 shows the spectrogram signal representations with VGG-16 feature extraction. As with Figure 123 which displays the raw spectrograms, there is little separation in either Figure 123 or Figure 125 between the clusters. If the centre of the graph is observed closely it can be seen that the no UAS (cluster 0) and UAS (cluster 1) overlap in the red and blue. This correlates with the low v-measure scores seen in Table 27.

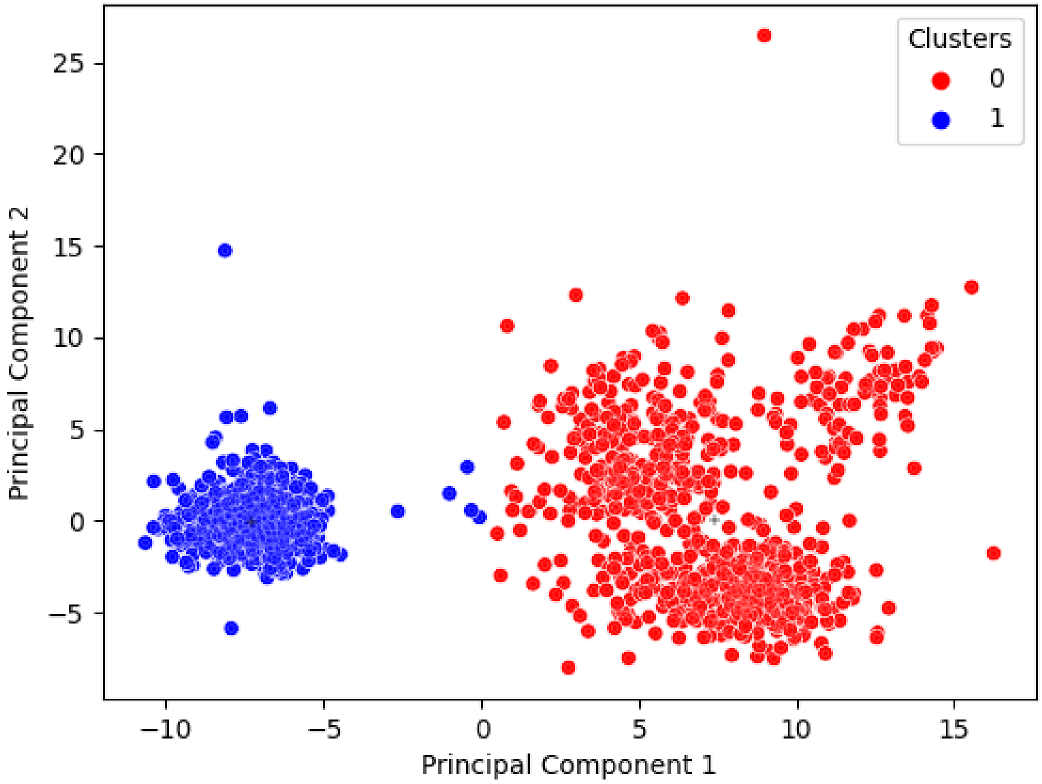


Figure 126 - DroneRF K-Means Detect PSD Graphical Signal Representation Raw Image

Lastly Figure 126 shows the first two principle components for raw PSD graphical signal representations with PCA initialisation of k-means clustering. Figure 126 shows a clear separation between no UAS (cluster 0) and UAS present (cluster 1) with the UAS cluster showing a higher concentrated cluster. This visual separation is correlated by the high v-measure score of 0.982 described in Table 27.

5.5.1.3 Results – Clustering (UAS Type)

Table 28 shows the results for the K-means clustering. In general PSD images – whether raw or used with VGG-16 or ResNet-50 features extraction, significantly outperform spectrogram images, showing that frequency domain representation produces higher v-measure scores for clustering than time domain. Initialising the k-means clustering with PCA not only produces the highest v-measure score but it has the quickest inference time for raw images, VGG-16 and ResNet-50 feature extraction.

Table 28 - DroneRF K-Means Clustering Type Results UAS

Exp.	Image	Init	Time (s)	homo	compl	v-measure	ARI	AMI
1	PSD	kmeans++	0.889	0.594	0.825	0.691	0.541	0.690
1	PSD	random	0.301	0.714	0.835	0.770	0.636	0.770
1	PSD	PCA	0.189	0.715	0.836	0.771	0.637	0.770
1	SPEC	kmeans++	0.942	0.170	0.173	0.171	0.149	0.171
1	SPEC	random	0.337	0.172	0.175	0.173	0.148	0.173
1	SPEC	PCA	0.220	0.172	0.175	0.173	0.149	0.173
2	PSD	kmeans++	8.771	0.443	0.776	0.564	0.354	0.564
2	PSD	random	6.771	0.514	0.671	0.582	0.383	0.581
2	PSD	PCA	4.682	0.683	0.773	0.725	0.632	0.725
2	SPEC	kmeans++	7.609	0.210	0.219	0.214	0.158	0.214
2	SPEC	random	5.661	0.209	0.215	0.212	0.155	0.211
2	SPEC	PCA	3.750	0.211	0.222	0.216	0.162	0.215
3	PSD	kmeans++	30.623	0.559	0.688	0.617	0.431	0.616
3	PSD	random	24.855	0.455	0.648	0.534	0.375	0.534
3	PSD	PCA	15.489	0.570	0.692	0.625	0.447	0.625
3	SPEC	kmeans++	36.033	0.195	0.203	0.199	0.151	0.198
3	SPEC	random	30.847	0.193	0.202	0.198	0.149	0.197
3	SPEC	PCA	20.150	0.193	0.202	0.198	0.149	0.197

The highest v-measure score of 0.771 is produced by raw image - PSD graphical signal representation and PCA initialisation. This completed in a time of 0.189 seconds. Raw PSD images with PCA initialisation showed a higher v-measure score than feature extraction with PCA initialisation which was 0.725 for VGG-16 feature extraction and 0.625 for ResNet-50 feature extraction.

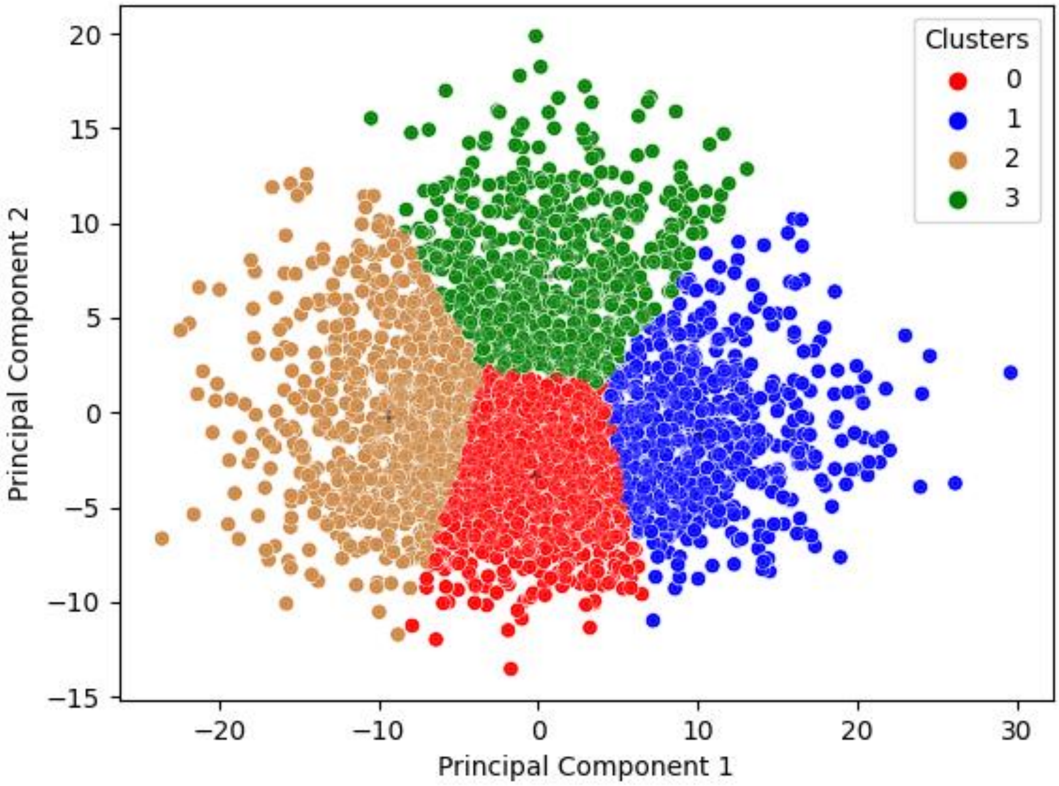


Figure 127 - DroneRF K-Means Spectrogram Graphical Signal Representation Raw Image

Figure 127 shows the first two principle component for spectrogram raw image graphical signal representation. It can be seen that the class no UAS (cluster 2) is not clearly separated from the UAS signals AR, Bebop and Phantom. There is a not a clear divide between any of the clusters defined.

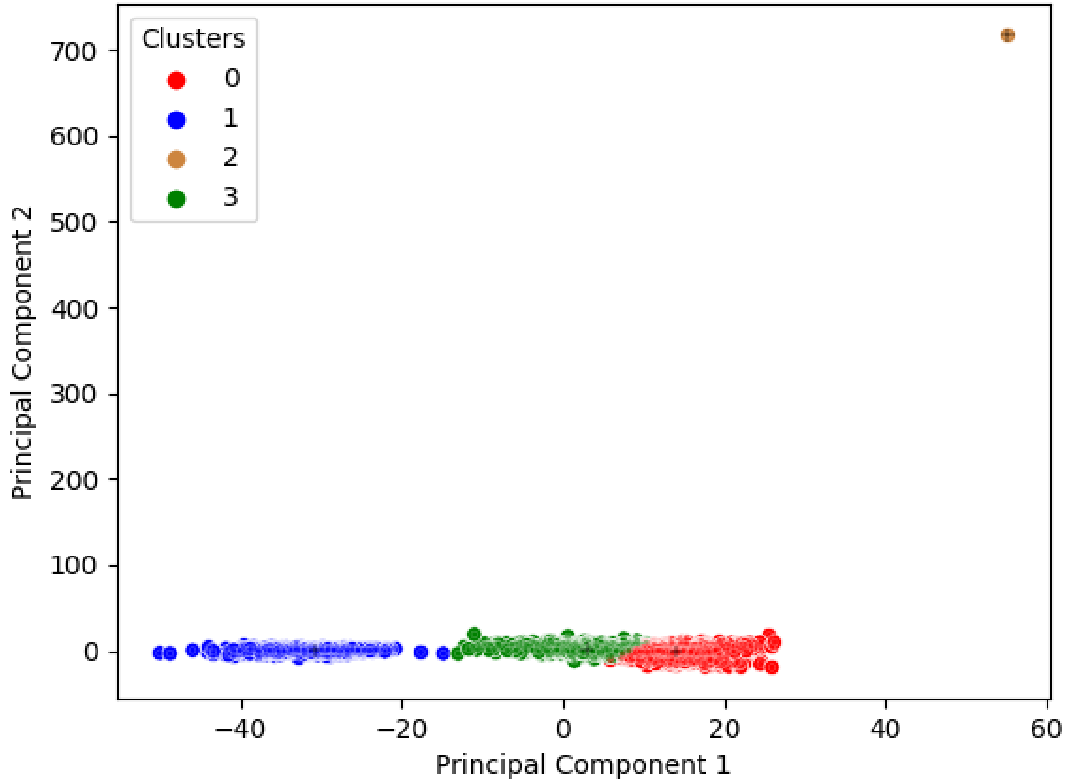


Figure 128 - DroneRF K-Means PSD Graphical Signal Representation VGG-16 Feature Extraction

Figure 128 shows the scatter plot of principle component 1 and 2 from VGG-16 feature extraction of the PSD with PCA initialisation. The class no UAS (cluster 2) is here clearly separated from the UAS clusters. The Bebop (cluster 1) also shows a higher visual separation from the AR and Phantom 3.

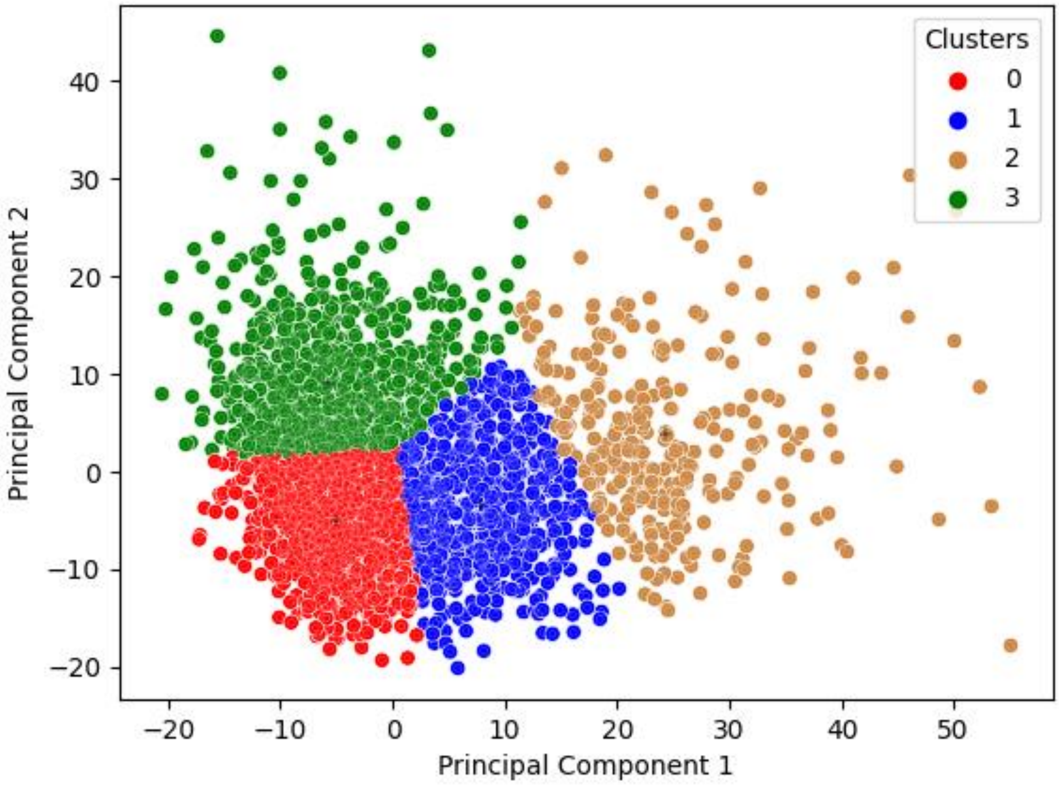


Figure 129 - DroneRF K-Means Spectrogram Graphical Signal Representation VGG-16 Feature Extraction

Figure 129 shows the VGG-16 feature extraction with spectrogram. It is not as clearly separated as the PSD representation in Figure 128. It also visually depicts more outliers and a wider spread than the spectrogram images which are used without feature extraction in Figure 127. The feature extraction with spectrograms did produce a slightly higher v-measure score of 0.216 compared to 0.173 with raw spectrograms in Table 28.

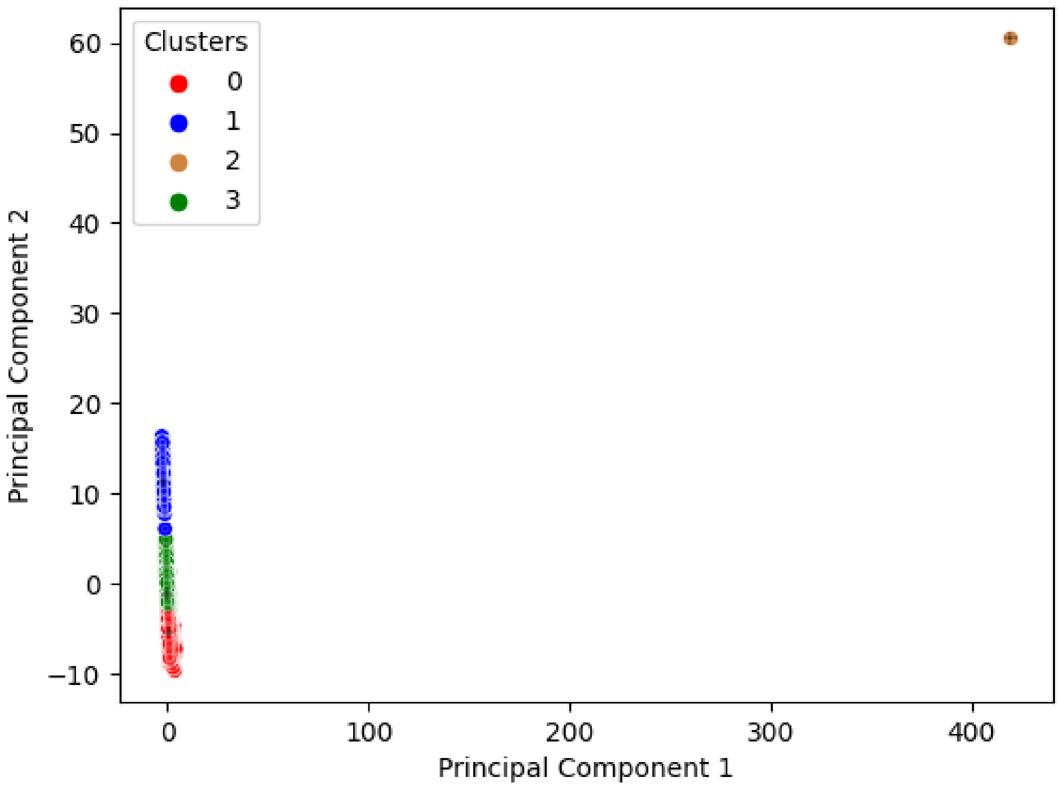


Figure 130 - DroneRF K-Means PSD Graphical Signal Representation Raw Image

Figure 130 shows the first two principle components plotted on a scatter graph for PSD Raw images. This produced the highest v-measure score of 0.771 and completed in under 0.2 seconds. Cluster 2 on Figure 130 represents the class no UAS and it can be clearly seen there is a distinct separation between a UAS being present and not present. This reflects Figure 126 which shows good separation for clustering no UAS and UAS signals using PSD graphical signal representations and PCA initialisation for k-means.

5.5.1.4 Discussion

UAS detection results shows that PSD graphical signal representations significantly outperformed spectrograms for k-means clustering regardless of the initialisation means or whether feature extraction had occurred. For inference time, random initialisation outperformed k++ and PCA and for v-measure PCA and random initialisation produced the highest scores. Considering both inference time and v-measure score, PSD raw images without feature extraction produced a v-measure of 0.982 in 0.104 of a second, significantly quicker than using the ResNet-50 feature extractor which did have a marginally higher v-measure score. Visually it could be seen from Figure 126 that good separation was achieved reflecting the high v-measure score for PSD graphical signal representation with PCA initialisation.

PSD images significantly outperform spectrogram images for raw image use and for feature extraction. This indicates that representing the signal in the frequency domain produces higher clustering results than representing the signal in the time domain. Initialising k-means with PCA produces the highest v-measure scores and the fastest inference times. There is no real advantage to applying feature extraction compared to raw images. The feature extraction adds time to the process and does not increase the v-measure score compared with raw images. The highest v-measure score of 0.771 which completed in under 0.2 seconds is using raw PSD graphical signal representations with PCA initialisation.

5.5.2 DroneDetect Dataset

5.5.2.1 Overview

The experiments use the 'DroneDetect' Dataset and the following eight UAS; DJI Air 2 S (OcuSync 3.0), Parrot Disco (Wi-Fi), DJI Inspire 2 Pro (Lightbridge 2.0), DJI Mavic Pro (OcuSync 1.0), DJI Mavic Pro 2 (OcuSync 2.0), DJI Mavic Mini (Wi-Fi), DJI Phantom 4 (Lightbridge 2.0) and No UAS present. The data is presented to the k-means clustering as an image of the graphical signal representations spectrogram and PSD; following CNN feature extraction using a VGG-16 pre-trained on ImageNet and following CNN feature extraction using a ResNet-50 pre-trained on ImageNet. Images for PSD and spectrogram datasets were saved as 224x224 pixels with 400 images per class.

5.5.2.2 Results – Clustering (UAS Detection)

Table 26 shows a key for understanding the experiments presented in tables Table 29 and Table 30. Experiments 1 represent raw images, experiments 2 VGG-16 features extraction and experiments 3 ResNet-50 features extraction.

Table 29 - DroneDetect K Means Clustering Results UAS Detect Clean

Exp.	Image	Init	Time (s)	homo	compl	v-measure	ARI	AMI
1	PSD	kmeans++	0.237	0.965	0.965	0.965	0.985	0.965
1	PSD	random	0.133	0.655	0.668	0.661	0.693	0.661
1	PSD	PCA	0.122	0.760	0.764	0.762	0.819	0.762

1	SPEC	kmeans++	0.288	0.259	0.263	0.261	0.325	0.260
1	SPEC	random	0.059	0.259	0.263	0.261	0.325	0.260
1	SPEC	PCA	0.060	0.259	0.263	0.261	0.325	0.260
2	PSD	kmeans++	1.618	1.000	1.000	1.000	1.000	1.000
2	PSD	random	1.270	1.000	1.000	1.000	1.000	1.000
2	PSD	PCA	0.753	1.000	1.000	1.000	1.000	1.000
2	SPEC	kmeans++	1.858	0.708	0.712	0.710	0.783	0.710
2	SPEC	random	1.506	0.767	0.768	0.768	0.842	0.767
2	SPEC	PCA	0.847	0.712	0.716	0.714	0.788	0.714
3	PSD	kmeans++	8.203	1.000	1.000	1.000	1.000	1.000
3	PSD	random	7.751	1.000	1.000	1.000	1.000	1.000
3	PSD	PCA	3.832	1.000	1.000	1.000	1.000	1.000
3	SPEC	kmeans++	6.162	0.782	0.783	0.782	0.856	0.782
3	SPEC	random	7.395	0.782	0.783	0.782	0.856	0.782
3	SPEC	PCA	3.712	0.793	0.794	0.794	0.860	0.793

Table 29 shows K-means clustering results for UAS detection in a clean environment using the DroneDetect dataset. In general PSD graphical signal representation outperform spectrograms. The highest v-measure was produced from using the PSD image with either the VGG-16 or the ResNet-50 feature extractor, both producing a perfect score of 1.000. Inference time for the ResNet-50 was 3.832 seconds and 0.753 seconds for the VGG-16, showing that the VGG-16 produces higher v-measure scores. An even faster inference time can be seen with only a small reduction in v-measure by employing the PSD raw images with a k++ initialisation. This produces a v-measure of 0.965 with an inference time of 0.237 seconds.

Table 30 - DroneDetect K Means Clustering Results UAS Detect Intereference

Exp.	Image	Init	Time (s)	homo	compl	v-measure	ARI	AMI
1	PSD	kmeans++	0.312	0.177	0.282	0.217	0.094	0.216
1	PSD	random	0.108	0.177	0.282	0.217	0.094	0.216
1	PSD	PCA	0.138	0.177	0.282	0.217	0.094	0.216
1	SPEC	kmeans++	0.349	0.107	0.111	0.109	0.136	0.108
1	SPEC	random	0.137	0.107	0.111	0.109	0.136	0.108
1	SPEC	PCA	0.125	0.107	0.111	0.109	0.136	0.108
2	PSD	kmeans++	1.849	0.179	0.283	0.219	0.096	0.281
2	PSD	random	1.661	0.284	0.361	0.318	0.212	0.317
2	PSD	PCA	0.959	0.284	0.361	0.318	0.212	0.317
2	SPEC	kmeans++	2.599	0.343	0.343	0.343	0.435	0.342
2	SPEC	random	2.001	0.157	0.266	0.197	0.075	0.075
2	SPEC	PCA	1.126	0.000	0.000	0.000	-0.001	-0.001
3	PSD	kmeans++	8.711	0.544	0.569	0.556	0.554	0.556
3	PSD	random	7.279	0.291	0.367	0.325	0.222	0.324
3	PSD	PCA	4.933	0.291	0.367	0.325	0.222	0.324
3	SPEC	kmeans++	8.754	0.011	0.012	0.012	0.013	0.011
3	SPEC	random	7.394	0.153	0.264	0.194	0.073	0.193
3	SPEC	PCA	4.949	0.008	0.008	0.008	0.009	0.009

Table 30 shows the same K-means clustering results for UAS detection but in an environment where both Bluetooth and Wi-Fi interference signals are present. Information about these signals can be found in section 3.1.4 and section 0. As with the clean environment in Table 29, PSD graphical signal representation significantly outperforms spectrograms. However, the interference effects performance in terms of v-measure considerably. Comparing the top performing combination in Table 29 of PSD with VGG-16 feature extraction producing a v-measure of 1.000, in Table 30 PSD with VGG-16 feature extraction produces the highest v-measure of 0.318 for PCA and random initialisation. This shows that interference does affect the ability of k-means to perform clustering. However, the highest performing combination was seen with the ResNet-50 feature extractor, PSD images and k++ initialisation. This produced a v-measure of 0.556 with an inference time of 8.711. This indicates that the deeper CNN architecture may be more resilient to noise when determining UAS features.

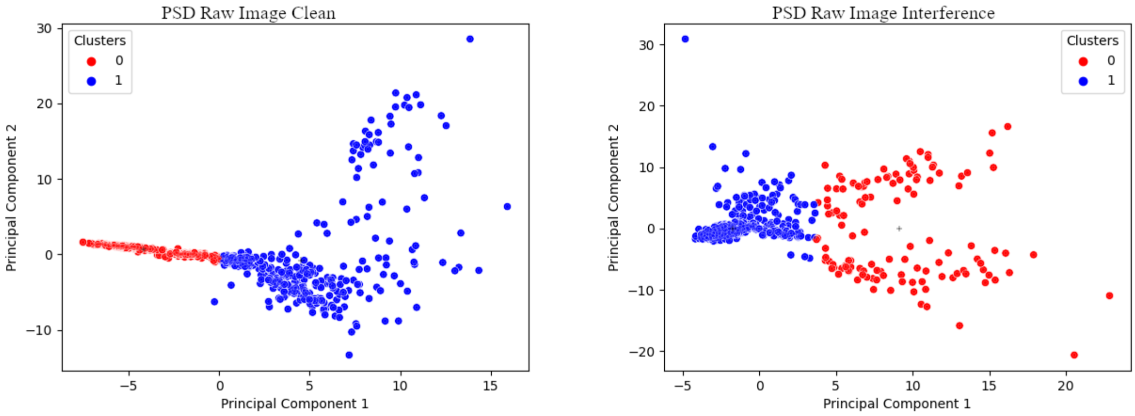


Figure 131 - DroneDetect PSD Graphical Signal Representation Raw Image Clean and Interference Detection

Figure 131 shows the first two principle components from the combination of PSD raw images with PCA initialisation for k-means clustering. On the left the clean environment is observed and on the right the interference is introduced. Table 29 and Table 30 show that the v-measure score drops from 0.762 to 0.217 when the interference is present with the UAS or no UAS signal. The left hand side of Figure 131 a higher concentration of cluster 0 which represents no UAS, especially compared to the interference on the right hand side where the red points for no UAS are actually quite far away from the centroid assigned to them (marked by the black cross).

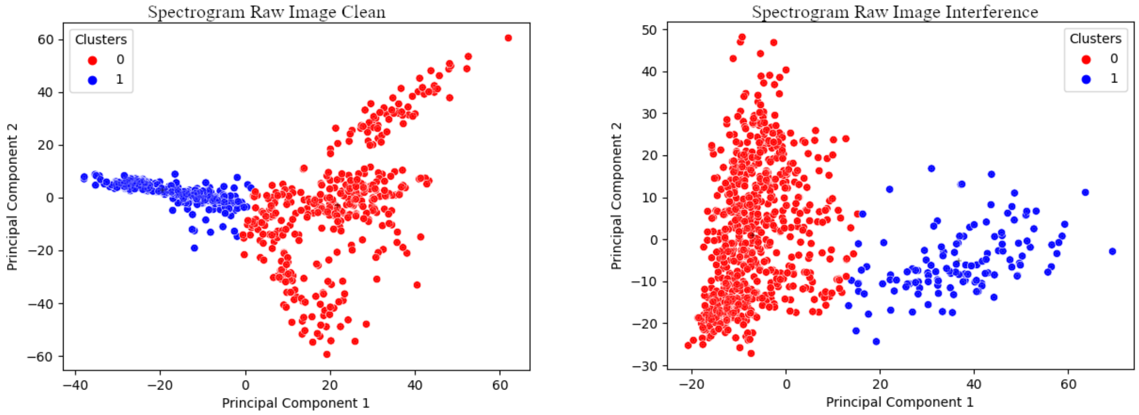


Figure 132 - DroneDetect Spectrogram Graphical Signal Representation Raw Image Clean and Interference Detection

Figure 132 shows the spectrogram raw images utilising PCA initialisation for k-means in a clean environment on the left and in the presence of interference on the right. Corresponding v-measure scores are 0.261 for the clean environment and 0.109 for the interference presence, presented in Table 29 and Table 30 respectively. Visually there is a higher level of concentration, especially for the UAS cluster, in the clean environment.

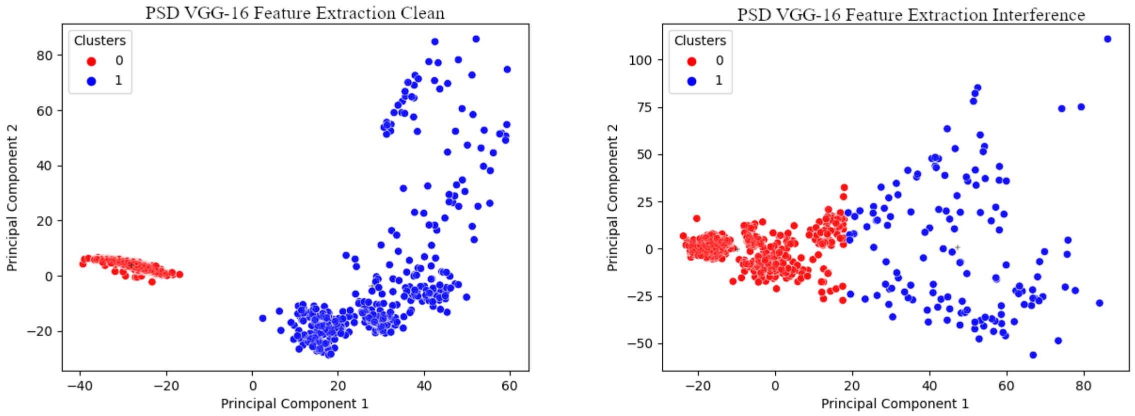


Figure 133 - DroneDetect PSD Graphical Signal Representation VGG-16 Feature Extraction Clean and Interference Detection

Figure 133 shows the principle component 1 and 2 plot of VGG-16 feature extraction with PSD images and PCA initialisation for k-means. In the clean environment on the left the separation of the two clusters is clear and corresponds with a v-measure score of 1.000. On the right hand side the separation is less vivid and again this is reflected with a significant drop in v-measure to 0.325 as seen in Table 29 and Table 30.

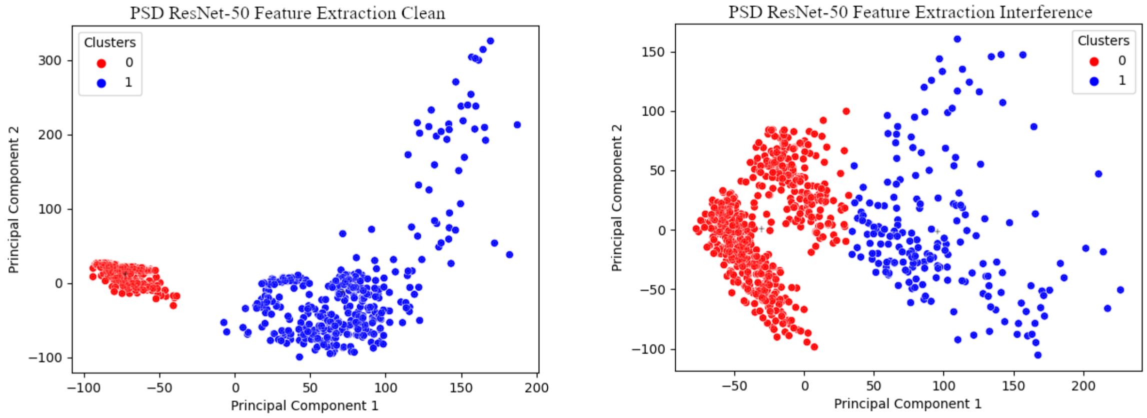


Figure 134 - DroneDetect PSD Graphical Signal Representation ResNet-50 Feature Extraction Clean and Interference Detection

Lastly Figure 134 shows the clean and interference environments with a combination which uses the ResNet-50 for feature extraction with PSD images. The highest performing combination for the interference environment used k++ for initialisation and showed a 0.556 v-measure score. Figure 134 shows PCA initialisation which produced a 0.325 v-measure score. On the left hand side in the clean environment on Figure 134 the separation between cluster 0 (no UAS present) and cluster 1 (UAS present) is clear with a higher concentration of grouping around the centroid (black cross) for no UAS present (cluster 0).

5.5.2.3 Results – Clustering (UAS Type)

Table 26 shows a key for understanding the experiments presented in tables Table 31 and Table 32. Experiments 1 represents raw images, experiment 2 shows the VGG-16 features extraction results and experiments 3 considers ResNet-50 feature extraction.

Table 31 – DroneDetect K Means Clustering Results UAS Type Clean

Exp.	Image	Init	Time (s)	homo	compl	v-measure	ARI	AMI
1	PSD	kmeans++	1.014	0.392	0.585	0.469	0.197	0.467
1	PSD	random	0.703	0.586	0.684	0.631	0.413	0.630
1	PSD	PCA	0.280	0.580	0.668	0.620	0.428	0.619
1	SPEC	kmeans++	1.207	0.271	0.314	0.291	0.148	0.288
1	SPEC	random	0.895	0.260	0.279	0.269	0.144	0.266
1	SPEC	PCA	0.401	0.295	0.319	0.307	0.159	0.304
2	PSD	kmeans++	13.914	0.629	0.731	0.676	0.488	0.675
2	PSD	random	9.999	0.631	0.700	0.664	0.490	0.663
2	PSD	PCA	4.927	0.651	0.701	0.675	0.525	0.674
2	SPEC	kmeans++	17.061	0.694	0.865	0.770	0.579	0.770
2	SPEC	random	13.947	0.684	0.773	0.725	0.555	0.724
2	SPEC	PCA	6.083	0.807	0.821	0.814	0.731	0.814
3	PSD	kmeans++	68.375	0.698	0.724	0.711	0.581	0.709
3	PSD	random	50.335	0.737	0.776	0.756	0.628	0.755
3	PSD	PCA	24.509	0.711	0.731	0.721	0.617	0.720
3	SPEC	kmeans++	67.166	0.717	0.815	0.762	0.583	0.762
3	SPEC	random	56.466	0.678	0.765	0.719	0.533	0.718
3	SPEC	PCA	24.564	0.720	0.749	0.734	0.586	0.733

Table 31 highlights the performance metrics for presenting the k-means clustering on PSD and spectrogram images and Feature Extraction (FE) using VGG-16 and ResNet-50 for clean signals. PCA dimensionality reduction speeds up the clustering time compared to random and k-means++ initialisations. If we consider time only then the fastest performance comes from using the images on their own with no feature extraction and PCA dimensionality reduction. The highest v-measure score is 0.814 which is using spectrogram images with the VGG-16 FE and PCA dimensionality reduction. ARI and AMI scores mimic the v-measure in terms of the highest performing. However, the highest v-measure score comes at a cost of time taking 6.083 seconds to complete. It is also interesting to note that the v-measure scores remain similar between experiment 1, 2 and 3 for PSD images, the CNN feature extraction does not increase v-measure significantly. While with the spectrogram images, the v-measure score doubles. For example raw spectrogram images using k++ initialisation produce 0.291 v-measure, while after the CNN feature extraction the v-measure increases to 0.770. It can also be seen that the deeper CNN architecture in experiment 3 does increase the v-measure scores for PSD but not for spectrogram.

Table 32 - DroneDetect K Means Clustering Results UAS Type Interference

Exp.	Image	Init	Time (s)	homo	compl	v-measure	ARI	AMI
1	PSD	kmeans++	1.009	0.363	0.521	0.428	0.275	0.426
1	PSD	random	0.675	0.473	0.575	0.519	0.356	0.517
1	PSD	PCA	0.193	0.584	0.617	0.600	0.475	0.599
1	SPEC	kmeans++	0.945	0.274	0.290	0.282	0.144	0.279
1	SPEC	random	0.798	0.276	0.292	0.284	0.145	0.281
1	SPEC	PCA	0.321	0.276	0.292	0.284	0.145	0.281
2	PSD	kmeans++	17.207	0.709	0.821	0.760	0.620	0.760
2	PSD	random	11.010	0.674	0.786	0.726	0.595	0.725
2	PSD	PCA	5.232	0.678	0.723	0.699	0.554	0.698

2	SPEC	kmeans++	19.534	0.440	0.519	0.476	0.342	0.474
2	SPEC	random	14.621	0.450	0.544	0.493	0.351	0.491
2	SPEC	PCA	7.045	0.496	0.513	0.505	0.370	0.503
3	PSD	kmeans++	72.722	0.684	0.702	0.693	0.566	0.692
3	PSD	random	55.690	0.617	0.668	0.641	0.525	0.640
3	PSD	PCA	24.284	0.633	0.653	0.643	0.513	0.642
3	SPEC	kmeans++	79.026	0.448	0.511	0.477	0.318	0.475
3	SPEC	random	78.479	0.441	0.490	0.464	0.319	0.462
3	SPEC	PCA	31.298	0.463	0.491	0.477	0.335	0.475

Table 32 highlights the performance metrics for presenting the k-means clustering on PSD and spectrogram images and feature extraction using VGG-16 and ResNet-50 for signals in the presence of interference. Again, PCA dimensionality reduction speeds up the clustering time compared to random and k-means++ initialisations. As with the clean signals, if we consider time only then the fastest performance comes from using the images on their own with no feature extraction and PCA dimensionality reduction. The highest v-measure score is 0.760 which is using PSD images with the VGG-16 FE. This indicates that PSD is more robust to noise. ARI and AMI scores mimic the v-measure in terms of the highest performing. Again the highest v-measure score comes at a cost of time at 17 seconds.

To understand which implementation is best we really must consider the desired application of the system for it to be used within. For example if the system needs to display as high an accuracy as possible but time is not urgent then the correct choice would be a spectrogram with VGG-16 FE and PCA initialisation which takes 6 seconds and produces a 0.814 v-measure score in a clean environment. In the presence of interference the PSD outperforms the spectrogram, still with the VGG-16 and k++ initialisation which takes 17 seconds and produces a 0.760 v-measure score. However, if the system needs to be highly time sensitive but can cope with a slightly lower accuracy then the best option would be image PSD with PCA which can complete in 0.280 seconds with a v-measure of 0.620 in a clean environment and 0.193 seconds with a v-measure of 0.60 in the presence of interference. In this instance the unsupervised learning could be used for a very timely indication that a UAS might be in the vicinity to cue another system, even a supervised algorithm with much higher accuracy.

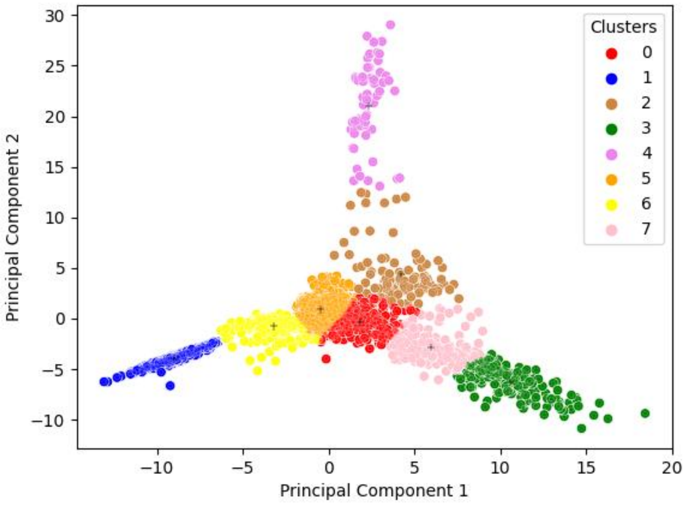


Figure 135 - DroneDetect K-Means PSD Graphical Signal Representation Image PCA Initialisation Clean

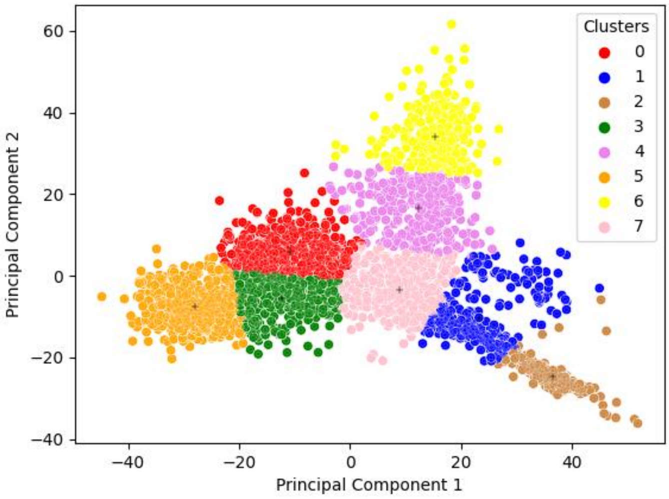


Figure 136 - DroneDetect K-Means Spectrogram Graphical Signal Representation Image VGG-16 FE PCA Initialisation Clean

Figure 135 shows the k-means clustering for PSD images using PCA dimensionality reduction in a clean environment with no interference from Bluetooth or Wi-Fi signals. Fig. Figure 136 shows spectrogram images which have been through FE with VGG-16 and PCA initialisation in a clean environment. Comparing

Figure 135 to Figure 136 it can be seen that the centroids in

Figure 135 have greater separation corresponding to a higher v-measure score, ARI and AMI.

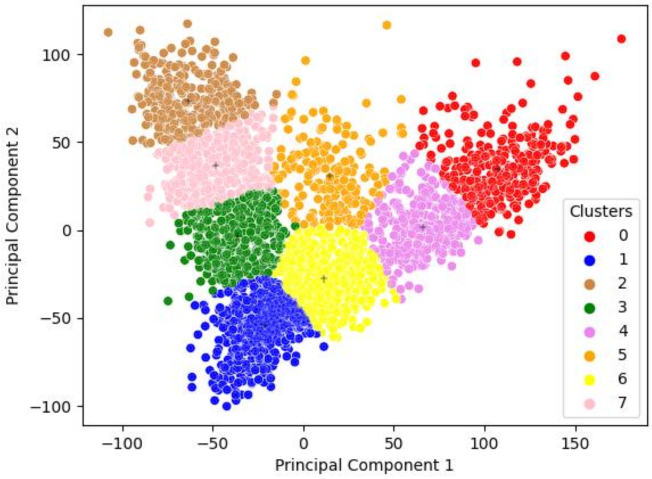


Figure 137 - DroneDetect k-Means Spectrogram Graphical Signal Representation Image ResNet-50 FE PCA Initialisation with interference

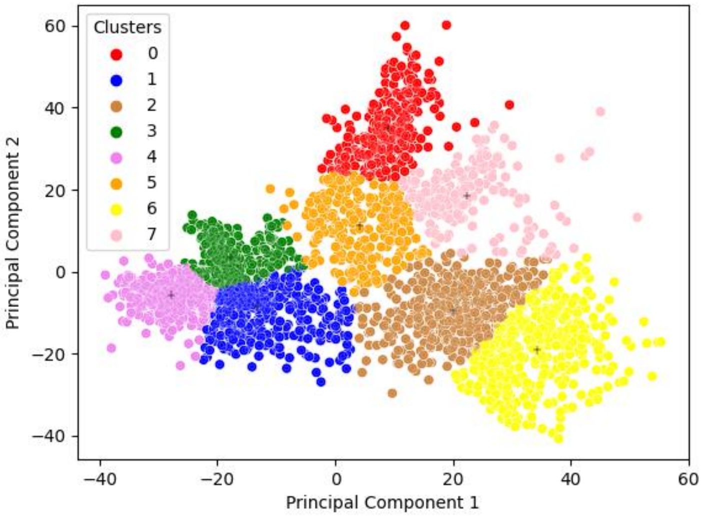


Figure 138 - DroneDetect K-Means PSD Graphical Signal Representation Image VGG-16 FE PCA Initialisation with Interference.

Figure 137 shows the k-means clustering for spectrogram images using PCA dimensionality reduction and ResNet-50 FE in the presence of interference whereby the centroids are marked by black crosses. We can see that the clusters boundaries are quite close together. This produces the slightly lower v-measure score of 0.495 and a longer resolution time of 30 seconds.

Figure 138 shows the higher accuracy implementation of the PSD image VGG-16 FE with PCA initialisation producing a v-measure of 0.699 in the presence of interference but completing in a slower time of 5.2 seconds.

5.5.2.4 Discussion

Overall these experiments have shown that unsupervised learning algorithms such as K-means clustering provide a potential for clustering small UAS signals which have not been seen before. In the future this could be extended for other datalink type and not exclusive to UAS signals. For UAS type, clustering graphical representations of the signal and utilising CNN feature extraction with transfer learning produces the highest v-measure score but at a cost of time, 6 seconds in a clean environment with spectrograms and 17 seconds with PSD in the presence of interference. With small UASs being capable of traveling at speeds of 45 mph, timely detection is essential in many use cases. Utilising a PSD image with PCA dimensionality and accepting a reduction of 0.2 v-measure in a clean environment allows clustering time to complete in under 0.3 second even in environments with interference from Bluetooth and Wi-Fi in the same band. Ultimately it should be dependent on the mission of the system. If a timely result is prudent then PSD images implemented with PCA initialisation would provide effective early warning to instigate the cueing of a secondary sensor or supervised algorithm with higher classification accuracy. However, employing transfer learning with CNN FE gives a higher v-measure score of 0.814 when employed with PCA initialisation and producing the clustering in 6 seconds.

For detection PSD graphical signal representation significantly outperformed spectrogram graphical signal representations. The interference from Bluetooth and Wi-Fi signals introduced in the frequency band decreased v-measure scores significantly. Comparing the highest performing clean combination of PSD, VGG-16 and any initialisation saw a 0.7 drop in the same combination with interference. This shows that interference does affect the ability of k-means to perform clustering. However, the ResNet-50 feature extractor was able to achieve a v-measure score of 0.556 even in the presence of the interference with PSD images and k++ initialisation. This experiment indicates that the deeper CNN architecture may be more resilient to noise when determining UAS features.

Future work could consider a larger number of UAS types in the dataset. Other types of unsupervised clustering algorithms and dimensionality reduction techniques could also be compared to extend this work. Another piece of valuable future work should include the collection of datasets at set distances away from the detection system. This would allow for a thorough evaluation of the correlation between clustering results and distance of the UASs from the detection equipment. Datasets could also be collected in a city based environment to understand the effect that a highly congested EM spectrum would have on the clustering results. The reason for this is that built up areas will contain other signals in the same band, for example Wi-Fi signals. These signals may introduce ‘confusion’ as noise within the band and inhibit the ability of the model to correctly cluster the results. This has the potential to be especially detrimental in built up areas but it would need tested to understand whether the clustering was able to distinguish between UAS signals and ordinary Wi-Fi signals in the same band. Future work should also include the testing of more than one of the same UAS type to see whether it would be pointed to the same cluster or whether the algorithm would define a new cluster for a second UAS model. If the second UAS of the same model type was to be placed in the same cluster then this would be a limitation of the system for early warning as an adversary operating a same ‘type’ UAS would not show up as a new cluster. Hence this would need further investigation. However, these experiment so far have shown the ability of unsupervised clustering for the timely early warning of malicious activity in restricted airspace which remains paramount in an era of ever-increasing dependencies on small UASs.

5.5.3 GPS Jamming Dataset

5.5.3.1 Overview

For the experiments the data was considered in 3 formats. For a baseline a dataset was kept in its raw IQ form. Secondly, datasets were created of images, as described and shown above for PSD and spectrogram. Lastly the images were fed to a pre-trained CNN for feature extraction to understand if the process of transfer learning was able to increase accuracy. VGG-16 and ResNet-50 were both considered for CNN feature extraction. Section 3.1.1 describes the dataset and section the 3.2 graphical signal representations as images. Section Chapter 4 describes the CNN feature extraction.

5.5.3.2 Results

Table 33 to Table 36 respectively highlight the performance metrics for presenting the k-means clustering on raw IQ data, images in the form of PSD, spectrogram, histogram, raw IQ plot and concatenation; and feature extraction using VGG-16 and ResNet-50 for PSD, spectrogram, histogram, raw IQ plot and concatenation. In terms of time, dimensionality reduction through PCA speeds up the clustering process when the input data is images. However, although the fastest time for completion was 0.295 seconds, accuracy was impacted. If we consider v-means as a way of representing homogeneity and completeness, the highest v-measure of 0.811 was using ResNet-50, the deeper neural network FE which took 57 seconds to complete.

Table 33 - GPS Jamming Dataset K-Means Clustering Results Raw

	Init	Time (s)	homo	compl	v-measure	ARI	AMI
RAW	kmeans++	11.542	0.418	0.784	0.545	0.24	0.544
RAW	random	4.978	0.189	0.614	0.288	0.071	0.287
RAW	PCA	3.239	0.485	0.681	0.567	0.245	0.566

Using the raw data straight from the SDR without any pre-processing can be seen in Table 33. The highest v-measure score is seen from using PCA initialisation, giving a score of 0.567 and a 3.24 second completion time. This highlights the fact that pre-processing is necessary for higher v-measure scores and can in fact speed up the processing time too.

Table 34 - GPS Jamming Dataset K-Means Clustering Results Images

	Init	Time (s)	homo	compl	v-measure	ARI	AMI
PSD	kmeans++	1.474	0.158	0.177	0.167	0.099	0.166
PSD	random	0.825	0.149	0.15	0.15	0.09	0.148
PSD	PCA	0.317	0.164	0.166	0.165	0.109	0.164
Spectrogram	kmeans++	1.759	0.525	0.527	0.526	0.387	0.525
Spectrogram	random	1.607	0.525	0.527	0.526	0.387	0.525
Spectrogram	PCA	0.728	0.546	0.582	0.563	0.39	0.563
Histogram	kmeans++	1.172	0.139	0.516	0.219	0.071	0.218

Histogram	random	0.542	0.494	0.637	0.556	0.396	0.556
Histogram	PCA	0.295	0.570	0.672	0.616	0.429	0.616
Raw Scatter	kmeans++	1.241	0.181	0.536	0.271	0.072	0.270
Raw Scatter	random	0.579	0.657	0.685	0.671	0.558	0.670
Raw Scatter	PCA	0.295	0.735	0.736	0.736	0.663	0.735
Concatenation	kmeans++	1.167	0.616	0.705	0.657	0.512	0.657
Concatenation	random	0.722	0.552	0.712	0.622	0.490	0.621
Concatenation	PCA	0.485	0.648	0.654	0.651	0.507	0.651

The result with the highest v-measure score for providing the k-mean algorithm with images of the graphical signal representation came from using the raw IQ scatter plot of the data with PCA initialisation. This gave a v-measure score of 0.736 and completed in less than 0.3seconds.

Table 35 - GPS Jamming Dataset K-Means Clustering Results VGG-16 Feature Extraction

	Init	Time (s)	homo	compl	v-measure	ARI	AMI
PSD	kmeans++	17.332	0.456	0.548	0.498	0.311	0.497
PSD	random	10.461	0.376	0.512	0.433	0.253	0.432
PSD	PCA	5.246	0.522	0.551	0.536	0.355	0.536
Spectrogram	kmeans++	20.139	0.709	0.877	0.784	0.643	0.784
Spectrogram	random	12.551	0.751	0.854	0.799	0.691	0.799
Spectrogram	PCA	6.501	0.751	0.854	0.799	0.691	0.799

Histogram	kmeans++	21.084	0.190	0.812	0.308	0.102	0.307
Histogram	random	14.164	0.653	0.743	0.695	0.505	0.695
Histogram	PCA	6.671	0.635	0.691	0.662	0.468	0.661
Raw Scatter	kmeans++	18.546	0.538	0.730	0.619	0.440	0.619
Raw Scatter	random	14.647	0.665	0.720	0.691	0.511	0.691
Raw Scatter	PCA	9.395	0.700	0.754	0.726	0.561	0.726
Concatenation	kmeans++	24.904	0.671	0.774	0.719	0.563	0.718
Concatenation	random	20.845	0.729	0.777	0.752	0.593	0.752
Concatenation	PCA	14.159	0.800	0.805	0.803	0.702	0.802

The highest v-measure scores were produced from the feature extraction using the deeper CNN architecture – the ResNet-50. The VGG-16 was not far behind giving the highest v-measure score as 0.803 with a concatenation of the signal representations and using PCA initialisation but at a time of 14.16 seconds. The ResNet-50 increased this v-measure score to 0.811 using spectrograms with a random initialisation. However the time increased significantly taking 57 seconds to complete.

Table 36 - GPS Jamming Dataset K-Means Clustering Results ResNet-50 Feature Extraction

PSD	kmeans++	80.892	0.721	0.789	0.754	0.6	0.753
PSD	random	52.339	0.689	0.796	0.739	0.624	0.738
PSD	PCA	24.94	0.727	0.762	0.744	0.626	0.744
Spectrogram	kmeans++	80.399	0.8	0.819	0.809	0.735	0.809
Spectrogram	random	57.263	0.801	0.820	0.811	0.735	0.81
Spectrogram	PCA	34.65	0.747	0.798	0.772	0.648	0.772
Histogram	kmeans++	70.327	0.420	0.657	0.512	0.310	0.512

Histogram	random	73.784	0.557	0.663	0.606	0.423	0.605
Histogram	PCA	42.847	0.643	0.694	0.667	0.494	0.667
Raw Scatter	kmeans++	73.204	0.491	0.727	0.586	0.345	0.586
Raw Scatter	random	62.328	0.710	0.765	0.736	0.572	0.736
Raw Scatter	PCA	39.219	0.740	0.746	0.743	0.657	0.743
Concatenation	kmeans++	90.479	0.699	0.748	0.722	0.546	0.722
Concatenation	random	76.189	0.642	0.720	0.679	0.498	0.678
Concatenation	PCA	46.679	0.723	0.766	0.744	0.604	0.743

If we consider both time and v-measure together then by accepting a 0.02 drop in v-measure (0.799) with VGG-16 FE, spectrograms and PCA dimensionality reduction, the time is reduced to 6.5 seconds. Figure 139 shows the clustering with FE using the VGG-16 neural network with spectrograms and employing PCA dimensionality reduction.

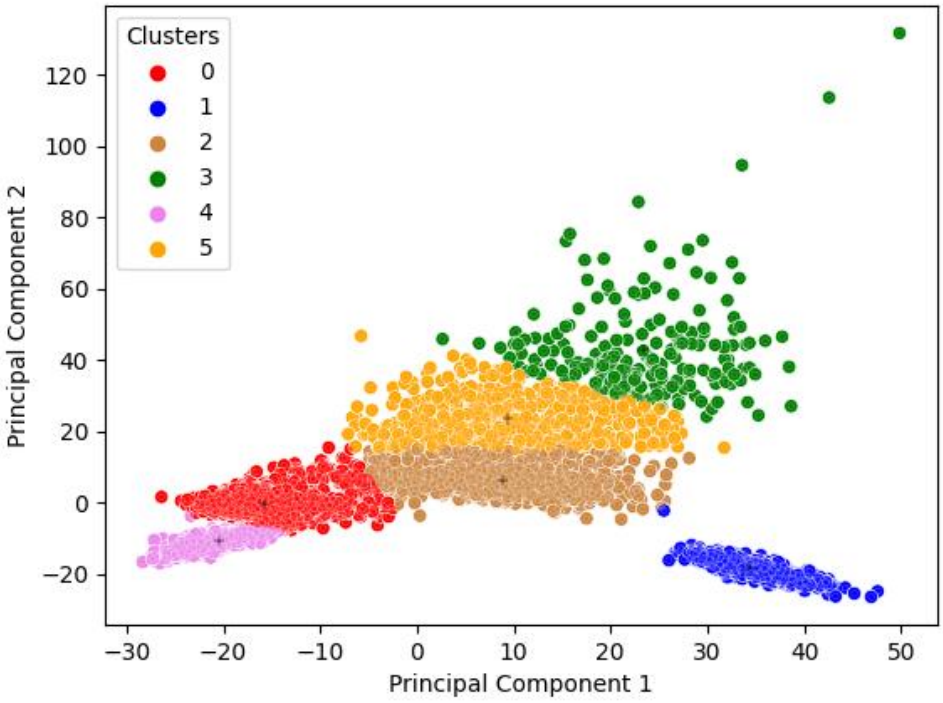


Figure 139 - FE VGG-16 Spectrogram Graphical Signal Representation K-means PCA initialisation

The VGG-16 gave a high v-measure score as 0.803 with a concatenation of the signal representations and using PCA initialisation but at a time of 14.16 seconds. This can be seen in Figure 140 where it can be seen that there are a few outliers but in general the classes have clustered onto the corresponding centroids.

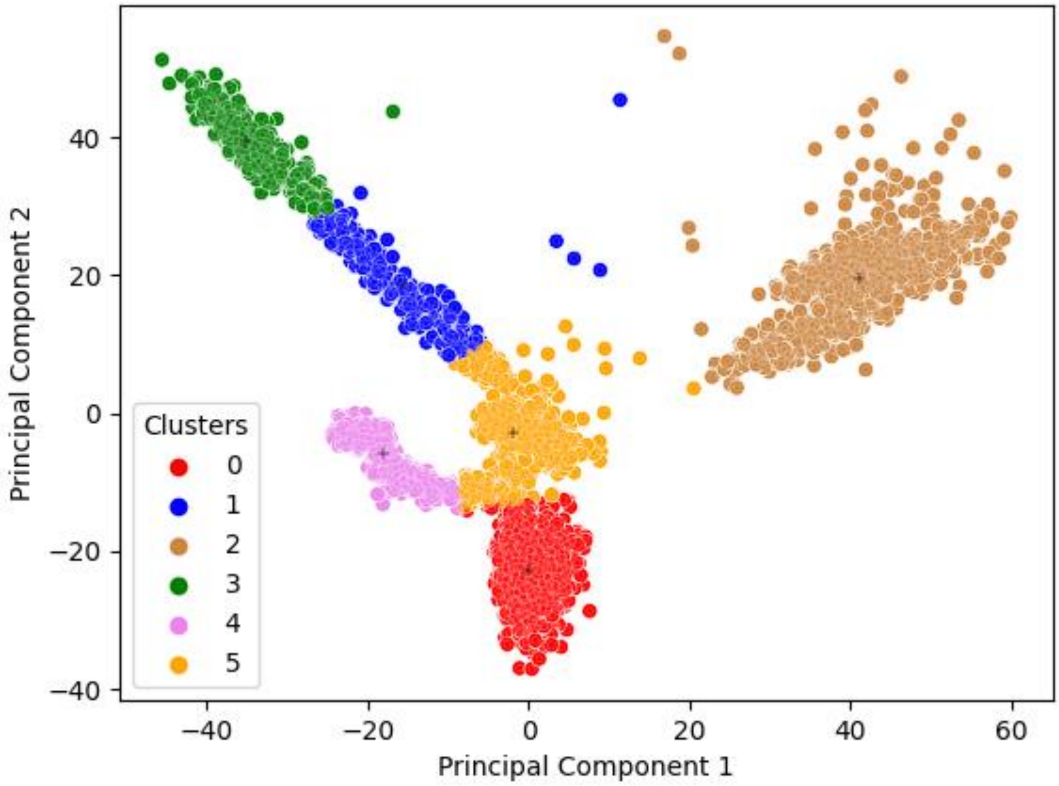


Figure 140 - FE VGG-16 Concatenation Graphical Signal Representation K-means PCA initialisation

However, the timely nature of the detection should be considered when applying these techniques. Clustering time can be further reduced to less than 0.3 second for a v-measure score of 0.736 with a raw IQ scatter plot and PCA dimensionality reduction. It is therefore essential for the application to consider what is important as there is a notable trade-off between accuracy and clustering time. ARI and AMI further emphasise the results shown through the v-measure, providing the same reflection of the accuracy of the clustering/grouping. Overall spectrograms outperformed PSD and by utilising PCA dimensionality reduction alongside FE with the VGG-16 we can produce high accuracy clustering of 0.8 in a short time frame. It can also be noted that comparing the highest scoring v-measure for raw IQ data is 0.24 less than using spectrograms with VGG-16 FE and PCA dimensionality reduction. The raw IQ scatter plot of the data with PCA initialisation gave the overall best performance with a v-measure score of 0.736 and completing in less than 0.3seconds.

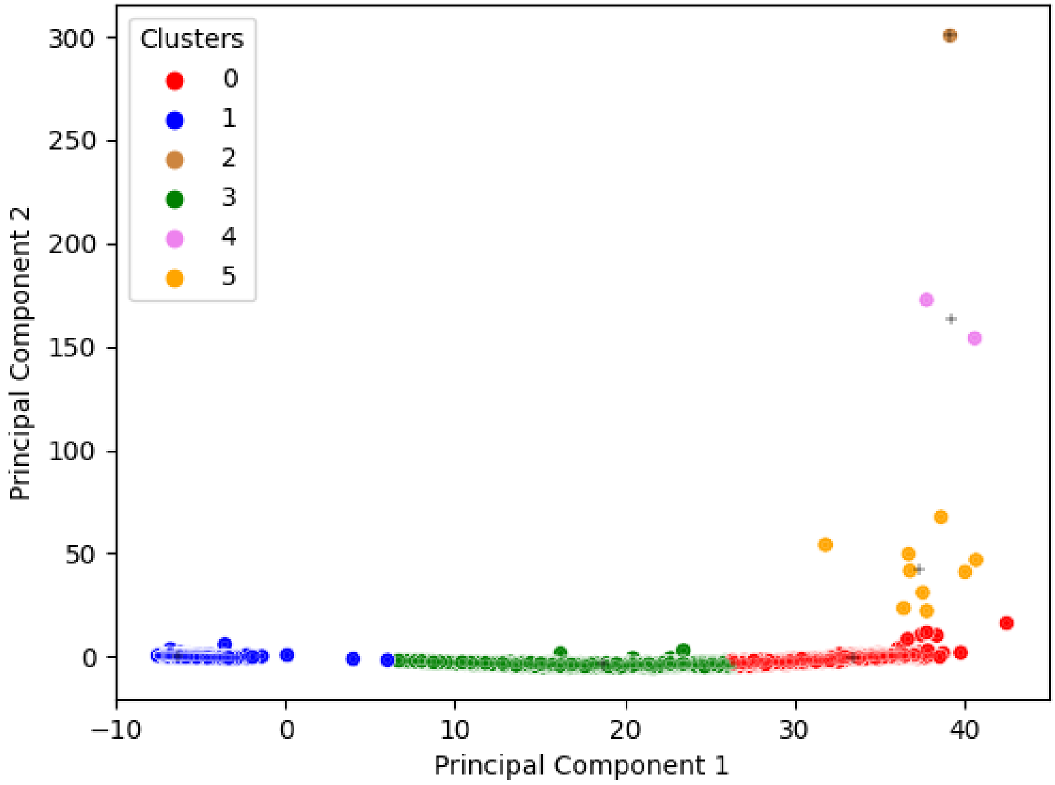


Figure 141 - Raw IQ Scatter Plot Image K-means PCA initialisation

5.5.3.3 Discussion

Overall, this experiment showed that employing the dimensionality reduction technique PCA, can reduce time and improve the accuracy (v-measure, ARI, AMI) of clustering. In general spectrogram images produced higher v-measure scores than PSD and raw data. The raw data scores highlighted the fact that pre-processing is necessary for higher v-measure scores and can in fact speed up the processing time too. The use of images allows for a quicker clustering result to be calculated compared to raw data. When using CNN FE with transfer learning it is observed that v-measure, ARI and AMI scores are generally in the region of 25% higher. Utilising FE with PCA dimensionality reduction is calculated in a similar time frame to using raw data but with a substantial increase in v-measure, ARI and AMI. Utilising transfer learning with CNN FE is highly significant when limited data for training is available. Signal data is renowned for being computationally heavy and it can be limited due to legal sensitivities surrounding the collection of datasets. Therefore, transfer learning provides a valuable means to improve accuracy for small datasets. However, pre-processing the SDR data into an image, in particular the raw IQ scatter plot of the data with PCA initialisation gave the overall best performance with a v-measure score of 0.736 and completing in less than 0.3seconds.

Future work could consider a larger number of jammer classes and the system tested with real world data. However, there are legal constraints surrounding both the collection of signal data and broadcasting signals in many countries. However, the ability to cluster or classify signals such as those produced by GPS jammers remains paramount in an era of ever-increasing dependencies such as autonomous vehicles.

Chapter 6 - Early Warning including Unknown Signal Detection

Figure 142 shows the early warning system and what this chapter will be looking at. The low cost early warning system is made up of a BladeRF SDR and wideband antenna to capture the raw data and a Raspberry Pi to conduct all the processing required to make the graphs, images, to run the CNN feature extraction and the machine learning model predictor.

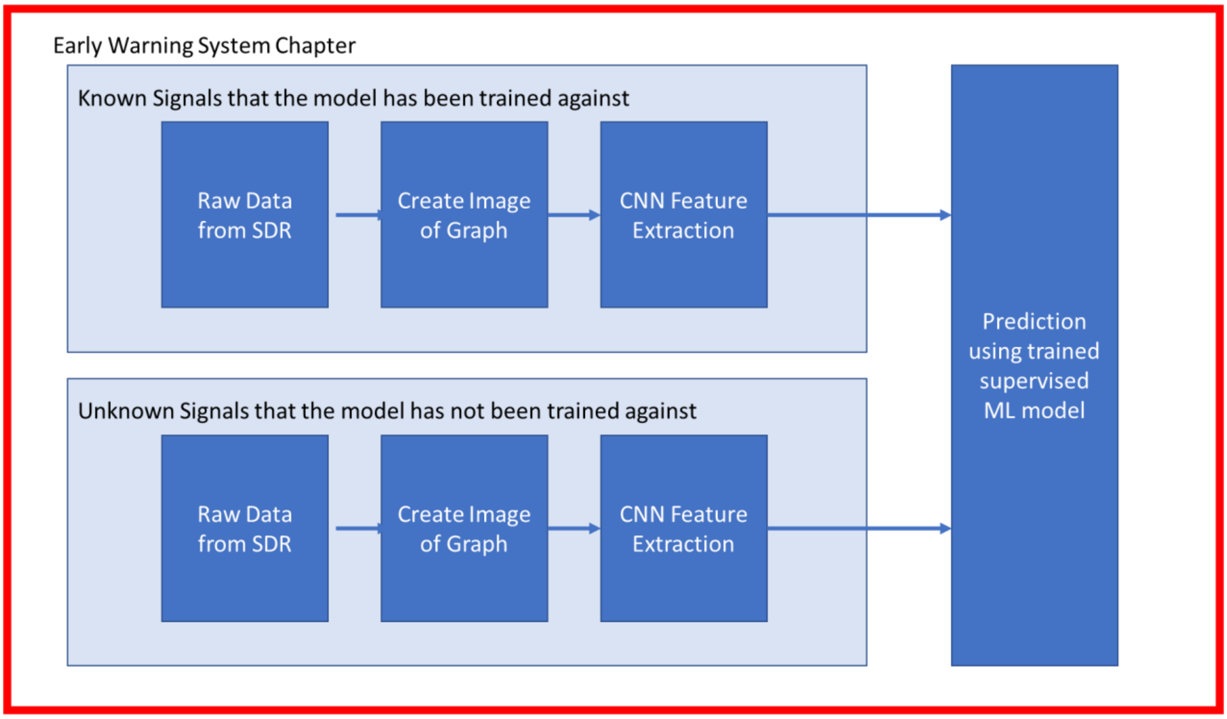

Early Warning System Chapter
Known Signals that the model has been trained against
Raw Data from SDR
Create Image of Graph
CNN Feature Extraction
Unknown Signals that the model has not been trained against
Raw Data from SDR
Create Image of Graph
CNN Feature Extraction
Prediction using trained supervised ML model

Figure 142 - Early Warning System Chapter with larger Thesis Construct

We have broken the chapter into sub sections. First we will discuss the creation of the detection and classification model. Then the implementation of the early warning system will be presented before presenting the results of both the model validation and then the prediction software running on the Raspberry Pi. Lastly the chapter will consider what will happen when the system is presented with signals it was not trained against. The information in this chapter has been published in a paper entitled ‘Low Cost Raspberry Pi based UAS Detection and Classification System using Machine Learning’ in a special edition MDPI Aerospace Journal which can be found in reference [20].

6.1 Early Warning System

6.1.1 Detection & Classification Model Creation

The training dataset used considered five platforms from the DroneDetect Dataset. Table 37 [377] shows the UAS considered in these experiments and the datalinks used for their transmission. For these experiments the platforms were restricted where possible to the 2.4GHz frequency range due to the use of one SDR. Future work to include a second SDR operating in the 5.8GHz range to cover both operating frequency ranges.

Table 37 - UAS Transmission Systems

Platform	Datalink	EIRP (2.4GHz)	Frequency Range (2.4GHz)
Air 2 S	OcuSync 3.0	20dBm	2.400-2.4835 GHz
Parrot Disco	Wi-Fi	19dBm	2.400-2.4835 GHz
Inspire 2	Lightbridge 2.0	17dBm	2.400-2.483 GHz
Mavic Pro 2	OcuSync 2.0	20dBm	2.400-2.4835 GHz
Mavic Mini	Wi-Fi	19dBm	2.400-2.4835 GHz

Each recording in the DroneDetect dataset consists of 1.2 x 10^8 complex samples equating to 2 seconds recording time in the form of a '.dat file'. In the experiments the recordings are split into samples equating to 80ms in length. The real and imaginary parts of the signal were added together and the samples processed in python using the Matplotlib API to produce spectrograms and power spectral density (PSD) graphs. The graphs were saved as images of 224x224 pixels to produce datasets of 250 samples per class whereby 200 were used to train the system and 50 to validate the results with k-fold cross validation. PSD and Spectrogram graphs were plotted using a 1024 FFT and a Hanning window with a 120 overlap.

Figure 143 and Figure 144 show a spectrogram and PSD respectively with a DJI Inspire flying. The platforms flew at a height of 20m in a 40m radius around the antenna with the pilot and controller approximately 4m from the detection/classification system. In both plots a wider concentrated band of larger bursts of activity can be observed in the higher end of the frequency band and also higher powered smaller bursts of activity, shown in a stronger yellow on the spectrogram, across a wider part of the spectrum.

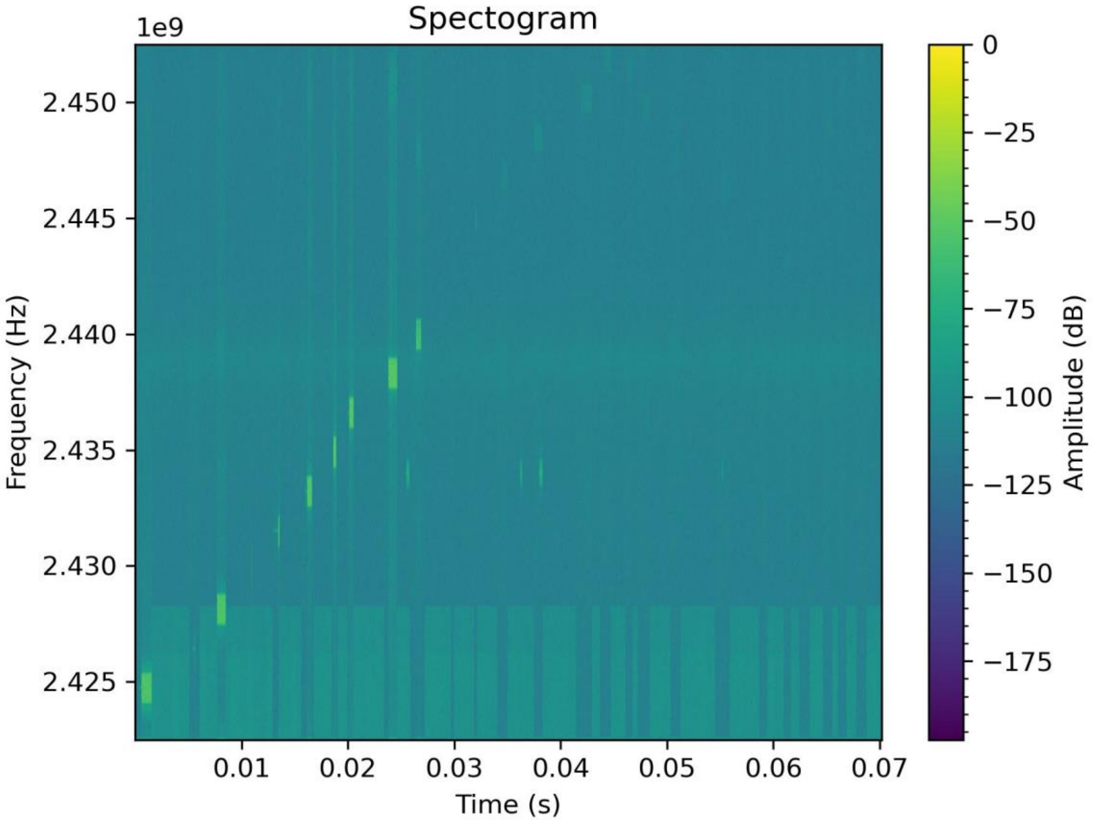


Figure 143 - Early Warning Spectrogram Graphical Signal Representation: DJI Inspire 2

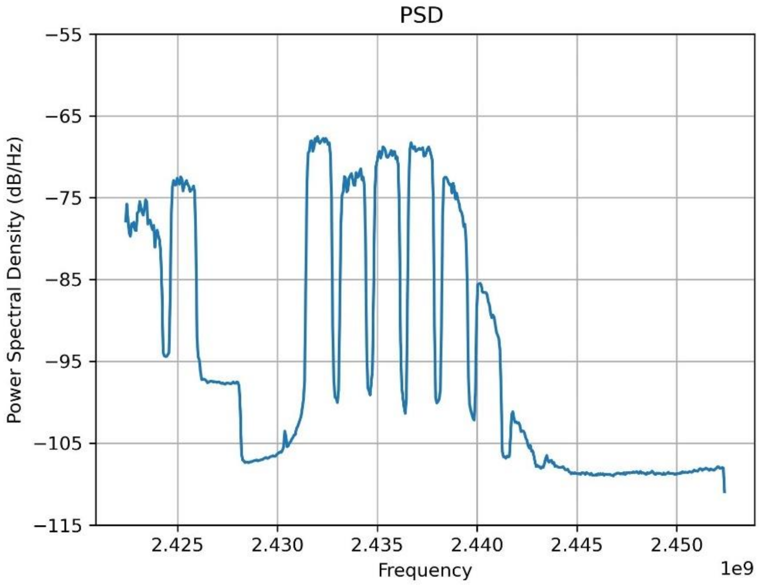


Figure 144 - Early Warning PSD Graphical Signal Representation: DJI Inspire 2

Figure 145 and Figure 146 show a spectrogram and PSD respectively with a DJI Mavic Mini. If Figure 143 and Figure 145 are compared, there is a clear difference visually between the transmission of the Mavic Mini and the Inspire. The same is observed when comparing the PSD graphs in Figure 144 and Figure 146.

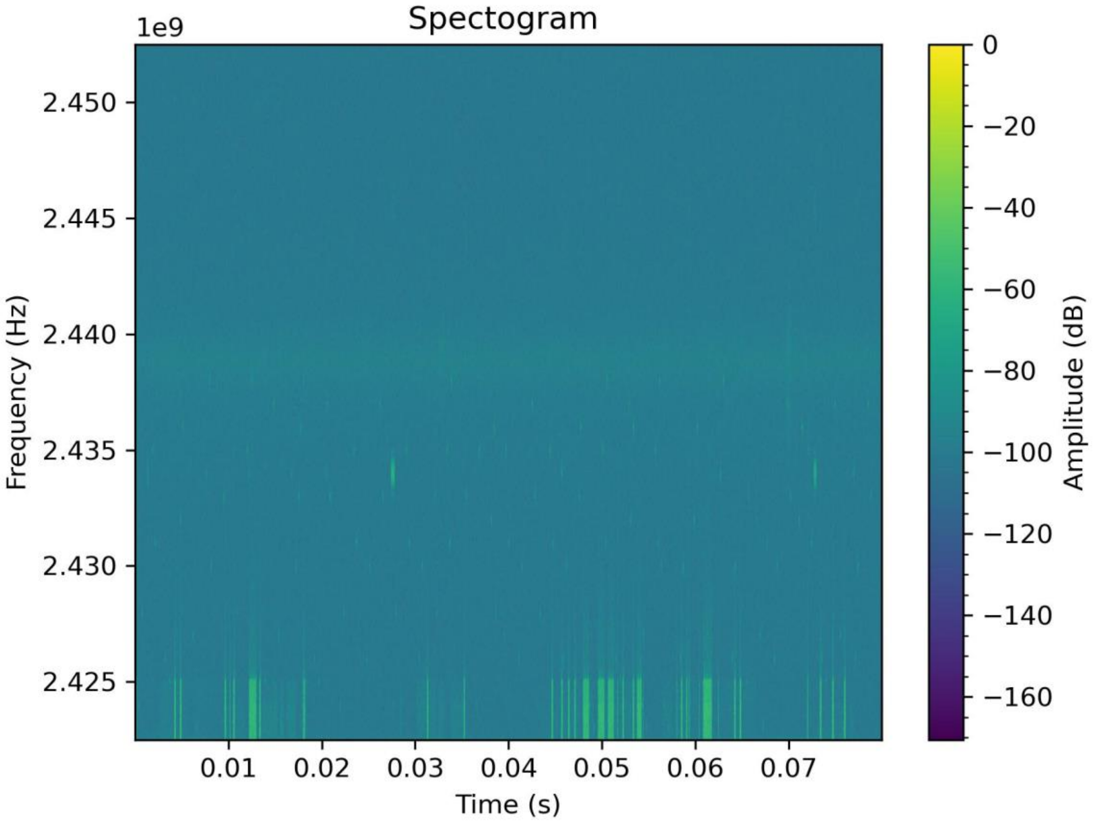


Figure 145 - Early Warning Spectrogram Graphical Signal Representation: DJI Mavic Mini

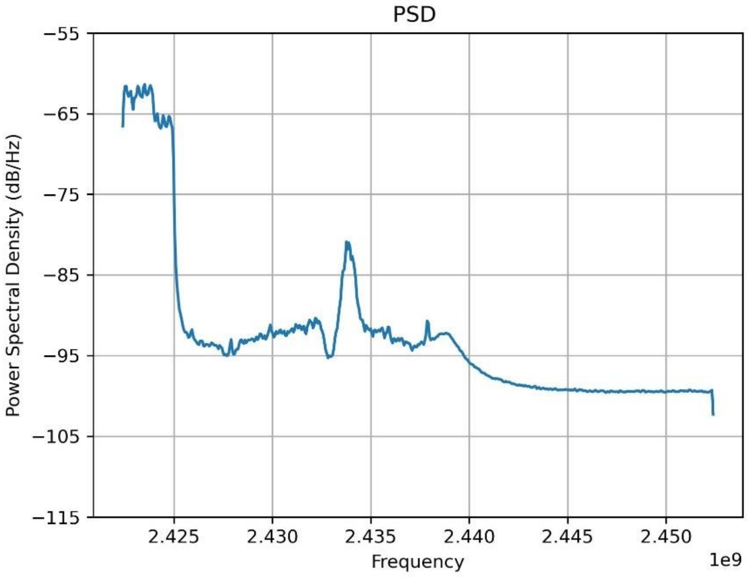


Figure 146 - Early Warning PSD Graphical Signal Representation: DJI Mavic Mini

Using the training datasets for spectrograms and PSD images a VGG-16 CNN with pre-trained weights on ImageNet, an object detection database of over 14 million images[289], was used as a feature extractor. During the training process, forward propagation is stopped at the last pooling layer and produces extracted features. The features were then used by machine learning classifiers LR and kNN to produce the classification model. Models were produced for spectrogram and PSD images, for classifiers LR and kNN and for 2 class detection and 6 class UAS type classification. Cross validation was used to try and highlight any overfitting with 5-fold and hyper-parameters were optimised using 3 fold nested cross validation for regularisation and the number of neighbours for kNN. Models were saved using the python pickle module.

6.1.2 Early Warning System Implementation

For these experiments the processing of the data to perform the classification was done on a low cost Raspberry Pi acting as an edge device. The Raspberry Pi can be purchased for $35 [378], the BladeRF SDR by Nuand at $480 [339] and the Palm Tree Vivaldi Antenna at $18.99 [342], making the cost of an edge device under $540. Figure 147 shows the configuration of an early warning system with 3 edge devices made up of an antenna, SDR and raspberry Pi and one control unit. For context the typical length of an airfield runway in the UK can vary between anywhere from 0.2 miles to 2.7 miles, with Gatwick Airport being 2 miles in length [181]. The length of a runway, along with the power of the antennas used would determine the amount of edge nodes required. It should also be noted that the approach to the runway must also be considered and not just the runway itself.

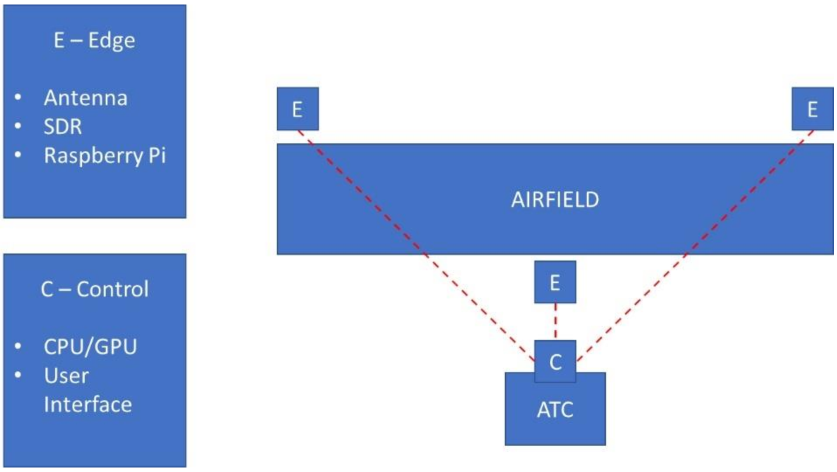
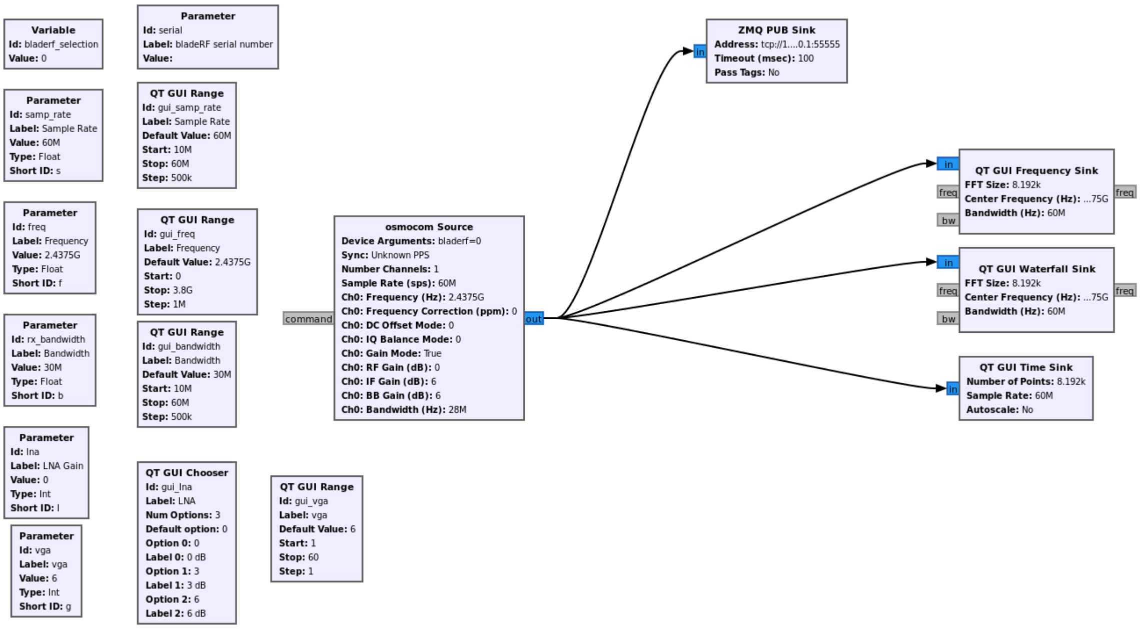


Figure 147 - Early Warning System Configuration

The control unit could simply be a laptop if the processing is happening on the edge devices. However, it could be a higher powered processor if the edge devices are used only to send the RF data back to the control unit to process there. For this reason we also considered ZeroMQ sockets to transmit the data between the edge and control units.

GNURadio was used to read the data from the BladeRF SDR and send it out through a ZeroMQ socket. ZeroMQ is an open source messaging and communications library which is asynchronous and fast. ZeroMQ has different types of sockets depending on the type of communication required for example request and reply is used when a reply is required for each message sent. In these experiments publish and subscribe is used. This is where a publisher can send data and multiple recipients can subscribe to receive it. This method was chosen as an early warning system may have two c2 nodes for redundancy which are both processing the data. Another methodology would be to do the processing at the receiving node but this would require more computational power at the end units. There are advantages and disadvantage of both approaches but ZeroMQ is capable of supporting either implementation with minor programming changes. ZeroMQ also supports pipelines for connected nodes and pairs for an exclusive connection.



Variable
Id: bladerf_selection
Value: 0

Parameter
Id: serial
Label: bladeRF serial number
Value:

Parameter
Id: samp_rate
Label: Sample Rate
Value: 60M
Type: Float
Short ID: s

QT GUI Range
Id: gui_samp_rate
Label: Sample Rate
Default Value: 60M
Start: 10M
Stop: 60M
Step: 500k

Parameter
Id: freq
Label: Frequency
Value: 2.4375G
Type: Float
Short ID: f

QT GUI Range
Id: gui_freq
Label: Frequency
Default Value: 2.4375G
Start: 0
Stop: 3.8G
Step: 1M

Parameter
Id: rx_bandwidth
Label: Bandwidth
Value: 30M
Type: Float
Short ID: b

QT GUI Range
Id: gui_bandwidth
Label: Bandwidth
Default Value: 30M
Start: 10M
Stop: 60M
Step: 500k

Parameter
Id: lna
Label: LNA Gain
Value: 0
Type: Int
Short ID: l

Parameter
Id: vga
Label: vga
Value: 6
Type: Int
Short ID: g

QT GUI Chooser
Id: gui_lna
Label: LNA
Num Options: 3
Default option: 0
Option 0: 0
Label 0: 0 dB
Option 1: 3
Label 1: 3 dB
Option 2: 6
Label 2: 6 dB

QT GUI Range
Id: gui_vga
Label: vga
Default Value: 6
Start: 1
Stop: 60
Step: 1

osmocom Source
Device Arguments: bladerf=0
Sync: Unknown PPS
Number Channels: 1
Sample Rate (sps): 60M
Ch0: Frequency (Hz): 2.4375G
Ch0: Frequency Correction (ppm): 0
Ch0: DC Offset Mode: 0
Ch0: IQ Balance Mode: 0
Ch0: Gain Mode: True
Ch0: RF Gain (dB): 0
Ch0: IF Gain (dB): 6
Ch0: BB Gain (dB): 6
Ch0: Bandwidth (Hz): 28M

ZMQ PUB Sink
Address: tcp://1....0.1:55555
Timeout (msec): 100
Pass Tags: No

QT GUI Frequency Sink
FFT Size: 8.192k
Center Frequency (Hz): ...75G
Bandwidth (Hz): 60M

QT GUI Waterfall Sink
FFT Size: 8.192k
Center Frequency (Hz): ...75G
Bandwidth (Hz): 60M

QT GUI Time Sink
Number of Points: 8.192k
Sample Rate: 60M
Autoscale: No

Figure 148 - GNURadio set up with ZeroMQ Socket

Figure 148 shows the GNURadio set up using a ZeroMQ socket which is publishing the BladeRF data. GNURadio can be run on a microcomputer such as a Raspberry Pi which when connected to an SDR provides a small footprint for an edge node in an early warning system.

On the c2 node a python script would then receive the information from the socket to produce a graphical representation of the signal as an image and run it through the prediction model. Within the experiments the socket information is received on the Raspberry Pi so that the time taken to made a prediction is evaluated using a low cost edge device. The inference time is recorded from when the libraries are loaded and the information is being received from the socket until the prediction is made. Another consideration is that the control unit would likely be taking inputs from other sensors. For example an early warning system may also include another sensor such as a video system or radar, which when activated would instigate the RF model to be run. The system could also include an unsupervised algorithm which may have less accuracy but would produce another indication marker in a very quick time scale.

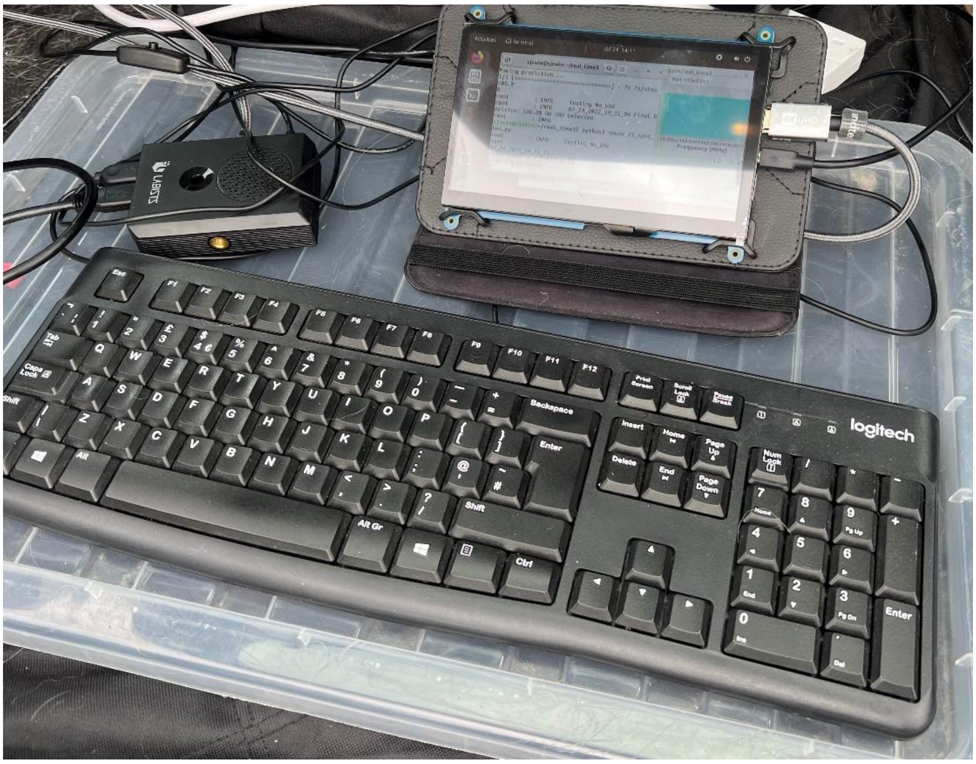


Figure 149 - Raspberry Pi based UAS Early Warning System

When we have the RF data and have produced the graphical signal representations, TensorFlow lite was used on the Raspberry Pi to load the previously trained model and make a prediction on the class. TensorFlow lite is a small version of the TensorFlow library specifically designed to run on embedded devices which are Linux based such as the Raspberry Pi. The Raspberry Pi in the experiments was loaded with Ubuntu 22.04, running python version 3.10.4 and TensorFlow version 2.9.0. Figure 149 shows a picture of the Raspberry Pi based UAS early warning system running the experiments on Ubuntu with python and TensorFlow.

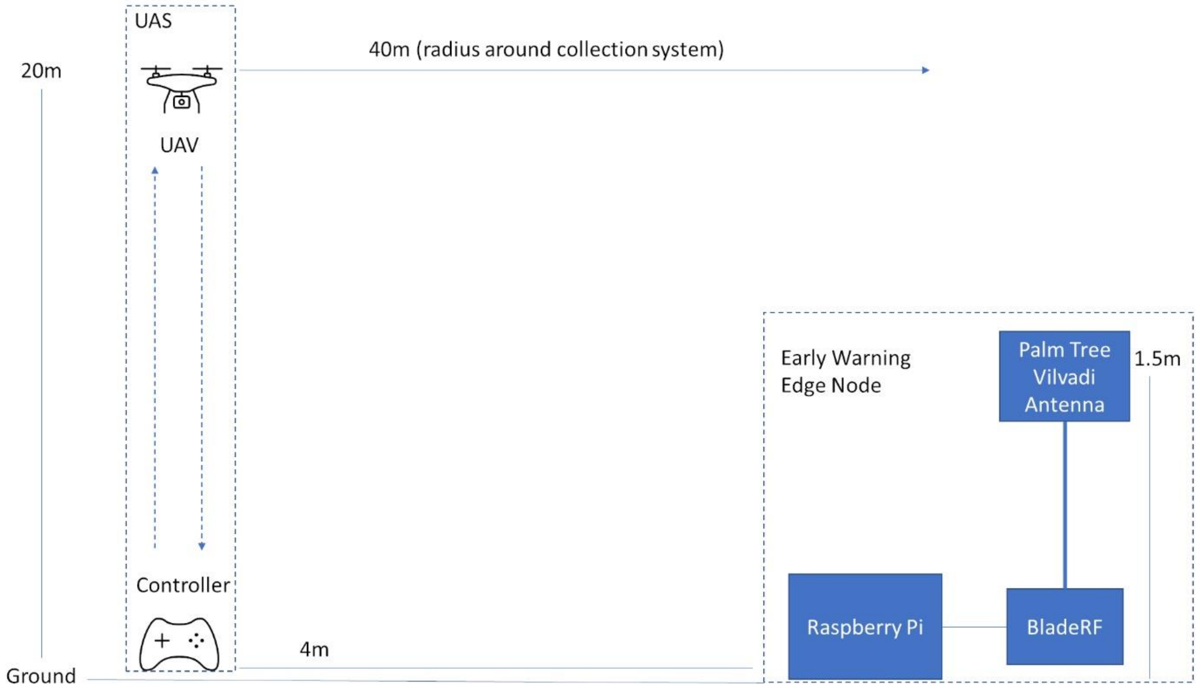


Figure 150 - Early Warning System Setup

Figure 150 shows the setup of the experiments to test the early warning system. The system is capturing any RF signals within a 28MHz bandwidth with a 2.4375GHz centre frequency. The BladeRF SDR was set with a sample rate of 60MBits/s and connected to a low cost antenna with a frequency range from 800MHz to 6GHz [340] [341]. GNURadio ran on the Raspberry Pi to visualise the spectrum and also to show the implementation of the ZMQ socket. A python script then ran in the terminal to receive the data and run the prediction using the previously trained models.

6.1.3 Early Warning System Results

6.1.3.1 Model Training & Validation

Before the early warning system is considered the results from training the model are evaluated using F1-score and accuracy as the performance metrics. The metric accuracy considers how many times the model was right, while F1-score also takes into account the metrics recall and precision. Recall calculates the fraction of positives predictions the model deemed to be correct while precision considers the amount of positive predictions which were in fact positive.

Table 38 - Early Warning Model Training Classification Results Accuracy & F1-Score

Classifier	Image	Metric	Detection	Type Classification
LR	PSD	Accuracy (%)	100(+/- 0.0)	99.3 (+/-0.6)
LR	PSD	F1-Score (%)	100(+/- 0.0)	99.2 (+/-0.6)
LR	SPEC	Accuracy (%)	99.6 (+/-0.3)	98.4 (+/-0.6)
LR	SPEC	F1-Score (%)	99.6 (+/-0.3)	98.4 (+/-0.6)
kNN	PSD	Accuracy (%)	100(+/- 0.0)	97.0 (+/-0.6)
kNN	PSD	F1-Score (%)	100(+/- 0.0)	97.0 (+/-0.6)
kNN	SPEC	Accuracy (%)	98.2 (+/-0.5)	95.7 (+/-1.5)
kNN	SPEC	F1-Score (%)	98.2 (+/-2.6)	95.6 (+/-1.5)

Table 38 gives the F1-score and accuracy scores for the different models trained for 2 class detection and 6 class UAS type classification. It can be observed that PSD graphical signal representations slightly outperform spectrograms and that the LR models again slightly outperform kNN but only marginally in both cases.

Table 39 - Early Warning Model Training Validation Results Accuracy & F1-Score

Classifier	Image	Metric	Detection	Type Classification
LR	PSD	Accuracy (%)	100	100
LR	PSD	F1-Score (%)	100	100
LR	SPEC	Accuracy (%)	98.6	98.5
LR	SPEC	F1-Score (%)	98.6	98.5
kNN	PSD	Accuracy (%)	99.3	97.7
kNN	PSD	F1-Score (%)	99.3	97.7
kNN	SPEC	Accuracy (%)	99.3	92.9
kNN	SPEC	F1-Score (%)	99.4	92.9

To ensure the models were not overfitting some data was held back for validation. Table 39 shows the validation results. When comparing Table 38 with Table 39 it can be seen that the models do not appear to be overfitting as the validation results do not drop significantly when the model is presented with new information.

6.1.3.2 Predictor Results

These models were then loaded onto the Raspberry Pi to be tested as part of the early warning system. Table 40 shows the results from running the 2 class detection system on the Raspberry Pi. It can be observed that running each model in the presence of no UAS flying, Mavic Mini and the Mavic Inspire produced the correct prediction results with 100% confidence each time. Inference time varied from 15 to 28 seconds which lends itself to the conclusion that edge processing on a Raspberry Pi should either be used in conjunction with other sensors which can produce a more timely result or the Raspberry Pi should act as a relay with the processing being done on a higher powered device on the control unit. Overall the 2-class detection system was correct with its prediction on whether a UAS was present or not with 100% accuracy and 100% confidence.

Table 40 - Early Warning 2-Class Detection Results

Classifier	Image	UAS Flying	Model Prediction	Prediction (%)	Time (s)
LR	PSD	No UAS	No UAS	100	28
LR	PSD	Mini	UAS Detected	100	26
LR	PSD	Inspire	UAS Detected	100	15
LR	SPEC	No UAS	No UAS	100	22
LR	SPEC	Mini	UAS Detected	100	23
LR	SPEC	Inspire	UAS Detected	100	19
kNN	PSD	No UAS	No UAS	100	24
kNN	PSD	Mini	UAS Detected	100	24
kNN	PSD	Inspire	UAS Detected	100	20
kNN	SPEC	No UAS	No UAS	100	24
kNN	SPEC	Mini	UAS Detected	100	26
kNN	SPEC	Inspire	UAS Detected	100	20

Table 41 shows the results from running the 6 class UAS type classification system on the Raspberry Pi. Comparing the inference times in Table 40 to Table 41, it can be observed that there is no real significant difference or penalty for performing a greater number of classes. 2-class detection inference times range from 15 to 28 seconds and for the 6-class UAS type classification system 18-28 seconds. In terms of prediction accuracy, the system is 100% correct and confident in its prediction for no UAS and for the Mavic Mini. For the Inspire the prediction model predicts correctly 2 out of 3 times but confidence in the prediction ranges from 50-66.67%.

Table 41 - Early Warning 6-Class UAS Type Classification Results

Classifier	Image	UAS Flying	Model Prediction	Prediction (%)	Time (s)
LR	PSD	No UAS	No UAS	100	22
LR	PSD	Mini	Mini	100	27
LR	PSD	Inspire	Inspire	50	18
LR	SPEC	No UAS	No UAS	100	22
LR	SPEC	Mini	Mini	100	24
LR	SPEC	Inspire	-	-	-
kNN	PSD	No UAS	No UAS	100	26
kNN	PSD	Mini	Mini	100	27
kNN	PSD	Inspire	Inspire	66.7	23
kNN	SPEC	No UAS	No UAS	100	27
kNN	SPEC	Mini	Mini	100	28
kNN	SPEC	Inspire	Air 2 S	60	21

Table 41 shows that the highest confidence results are seen from using the kNN classifier with PSD graphical signal representations. The overall accuracy was 90.9% for UAS type classification on the UASs tested. Further research would be needed to look at the reason why the Inspire produced lower confidence results, the original dataset may not have included all of the Lightbridge 2.0 activity. It is unlikely that the loss of confidence was due to environmental changes as the Mavic Mini and No UAS predictions were 100% correct with 100% confidence.

6.1.4 Discussion

The experiments showed that the Raspberry Pi 4 B connected to a BladeRF SDR and low cost antenna, is capable of running a CNN feature extractor and machine learning classifier as part of an early warning system for UASs. However, the inference times ranged from 15-28 seconds for 2-class UAS detection and 18-28 seconds for classification. This suggests that for systems which require timely results the Raspberry Pi would be better suited to act as a repeater of the raw SDR data. This would enable the production of the graphical signal representations and the machine learning model prediction to be completed on a higher powered central control unit. If time was not of a concern then the Raspberry Pi is capable of making predictions as an edge device and could easily form part of a larger system made up of other sensors capable of faster indications to trigger higher accuracy results such as these. The overall accuracy of the 2-class detection system was 100% and 90.9% for UAS type classification on the UAS tested and noting that 3 of the predictions for the classifier ranged from 50-66.67% in confidence.

Further research would be needed to understand the reason why the Inspire produced lower confidence results for 6-class UAS type classification. It is possible that the original dataset may not have included all of the potential Lightbridge 2.0 activity which may move around the frequency band dependent on other activity. It is unlikely that the loss of confidence was due to environmental changes as the Mavic Mini and No UAS predictions were 100% correct with 100% confidence. For these experiments the platforms were restricted to the 2.4GHz frequency range due to the use of one SDR. However, all of the platforms considered in these experiments are capable of auto switching between 2.4GHz and 5.8Ghz. A future piece of work would include a second SDR operating in the 5.8GHz range to cover both operating frequency ranges. Future testing should also include a wider range of sensors and utilising the capabilities of multiple edge SDRs to triangulate the signal.

6.2 Unknown Signal Classification

6.2.1 Overview

The purpose of this set of experiments was to understand whether a supervised machine learning model which had been trained using a number of different UAS signals, could pick up UAS signals it was not trained on. The machine learning model used was trained upon the signals detailed in Table 37 which contains the OcuSync 3.0, Lightbridge 2.0, OcuSync 2.0 and Wi-Fi transmission protocols.

As part of these experiments 3 different UAS are considered. The DJI Mavic 3 which uses the OcuSync 3+ as the for the transmission protocol. The model has been trained on the OcuSync 3 so the OcuSync 3+ is an evolution of the previous datalink. The eBee uses a proprietary protocol which is not detailed in any of the SenseFly documentation. However it does operate within the 2.400 to 2.4835 GHz and with a slightly lower EIRP than the Mavic 3. The DX8 is an 8 channel DSMX transmitter which uses CDMA and frequency hopping technology in the 2.4GHz frequency band.

The aim of the experiments is to understand whether a machine learning model which has been trained in a supervised manner against specific datalinks can recognise new datalinks which it wasn’t originally trained against. Both evolutionary datalink and entirely new ones will be considered as part of these experiments. Four different machine learning models will be considered as part of the experiments and are details by models 1 – 4 and the VGG-16 CNN, pre-trained on ImageNet, was used as the feature extractor for each one.

Table 42 - Unknown Signal Experiment Models

Model	Image Type	Machine Learning Classifier
1	PSD	LR
2	PSD	kNN
3	Spectrogram	LR
4	Spectrogram	kNN

Lastly in section 6.2.3.3 the supervised model using VGG-16 CNN feature extraction is trained to include the following signals – Air 2 S, Parrot Disco, eBee X, DJI Inspire 2, Mavic Pro, Mavic Pro 2, Mavic Mini, No UAS, DJI Phantom 4 and the DX8 DSMX transmitter. Machine learning classifiers LR and kNN are evaluated alongside graphical signal representations PSD and spectrogram for a total of 10 class UAS type classification. Now the three UAS used for the unknown signals will be considered and presented in more detail.

6.2.1.1 Mavic 3

Table 43 shows a comparison of the DJI Mavic 3 and the DJI Air 2 S. The Air 2 S operates on the OcuSync 3.0 datalink while the newer Mavic 3 uses the OcuSync 3.0+. From Table 43 it can be seen that the Mavic 3 has a higher EIRP of 33dBm compared to the Air 2 S of 26dBm and this is reflected with a longer maximum range of flight, 15km for the Mavic 3 and 12km for the Air 2 S.

Table 43 - DJI Mavic 3 [379] Comparison Air 2 S [380]

	DJI Mavic 3	DJI Air 2 S
EIRP	33dBm (FCC)	26dBm (FCC)
Operating Frequency	2.400-2.4835 GHz 5.725-5.850 GHz	2.400-2.4835 GHz 5.725-5.850 GHz
Max Distance (FCC)	15km	12km
Max download bit rate	O3+: 5.5MB/s (RC-N1) 15MB/s (DJI RC Pro) Wi-Fi 6: 80MB/s	44 Mbps (download bitrate) 16 Mbps (live video bitrate)

In terms of the downlink bit rate the Air 2 S has two variants, 44Mbps for download and 16Mbps for the live video transmission. The Mavic 3 on the other hand is different depending on the controller which is used with the platform. The bit rate varies from 5.5MB/s for the RC-N1 remote controller, 15MB/s with DJI RC Pro remote controller and for Wi-Fi 6 it can handle 80MB/s. For these experiments the RC-N1 remote controller is used but if a comprehensive dataset was to be made in the future all variants of the set up should be included within the training dataset.

6.2.1.2 eBee X

The eBee X made by SenseFly is a fixed wing UAS which is used for surveillance and surveying purposes, designed for the autonomous mapping of large areas. The eBee is fitted with a data link antenna which communicates with a USB ground modem plugged into a laptop and a piece of software called eMotion. The eMotion software is used for flight planning where the user chooses blocks such as aerial mapping and the software auto populates a flight plan. Although the eBee is classed as an autonomous UAS, autonomous with take off, in flight and landing, the datalink connection between the eBee and the ground modem is used for tracking position and monitoring flight progress. Commands can also be sent to the UAS in flight aswell.

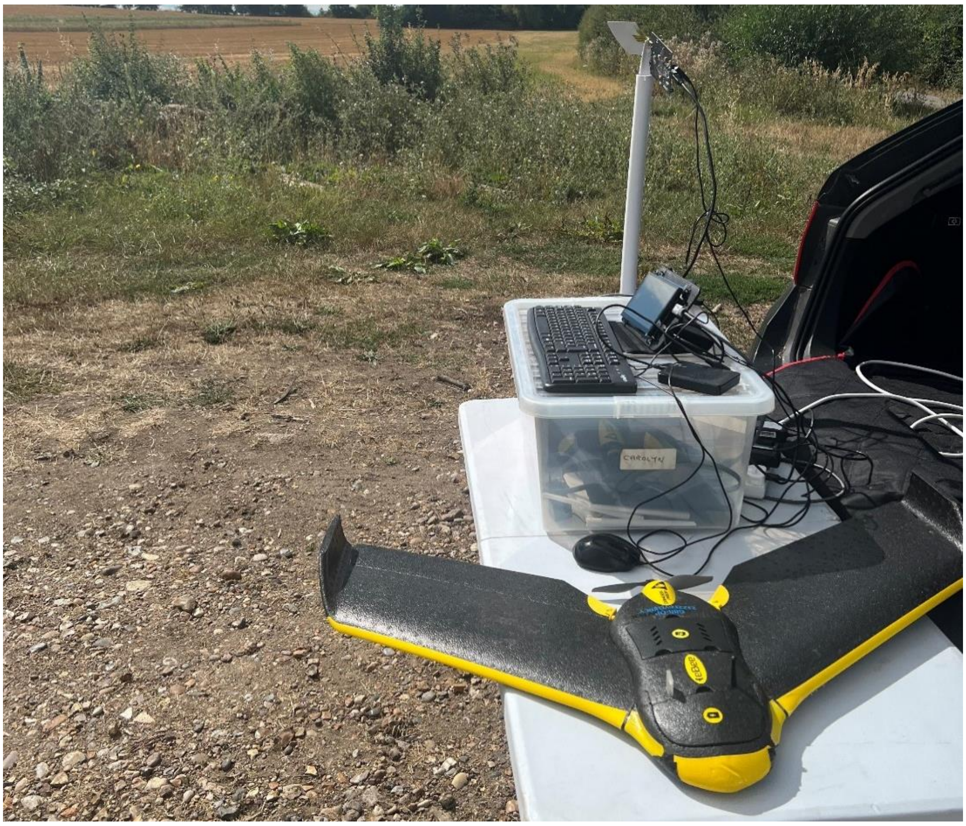


Figure 151 - eBee UAS Experiments for testing unknown non evolutionary signals

Figure 151 shows the eBee UAS and the detection software running on the raspberry Pi with the BladeRF SDR and Palm Tree Vivaldi wideband antenna. The Sensefly website does not give a detailed description of the datalink and no information regarding the protocol used for communication. The operating frequency is defined as 2.400 to 2.4835 GHz and the EIRP as 22.5dBm (FCC) and 20.0 dBm (CE/JP) [381].

6.2.1.3 DX8 DSMX Transmitter

The Spectrum radios have two transmitter protocols called DSM2 and DSMX which is an evolution of DSM2. DSM2 communicates using wideband Direct Sequence Spread Spectrum (DSSS). DSM2 keeps two frequency channels free so it can change between the two if the signal is lost due to interference. These channels are picked when the transmitter and the receiver are first turned on. The system works well unless interference occurs on both channels after that start up process is complete. If both channels experience interference which maybe wasn’t present when the channels were chosen at the pairing stage, DSM2 will fail.

DSMX was built to try and rectify this issue and it uses two layers of multi-access. DSMX communicates using Code Division Multiple Access (CDMA). CDMA is known for spreading itself out across a wide frequency range. It also uses pairs of transmitters and receivers which each have a particular hop sequence across that frequency range. This allows the pairs to communicate and not interfere with other pairs of transmitters and receivers. CDMA is known to be reliable for when there is interference in the same frequency band. DSMX uses a pseudo-random sequence which is decided upon between the transmitter and the receiver. The hop sequence will change thousands of times every single second across 23 different channels [382].

6.2.2 RF Profiling

6.2.2.1 Mavic 3

Figure 152 shows the spectrogram for the Mavic 3 which operates using the OcuSync 3+ datalink. It can be seen from observing Figure 152 that the bottom half of the spectrum has a higher general power level indicated by the yellow band. Then within the whole spectrum there are smaller bursts of yellow, some spanning a much larger portion of the spectrum and others occupying around 200-300MHz.

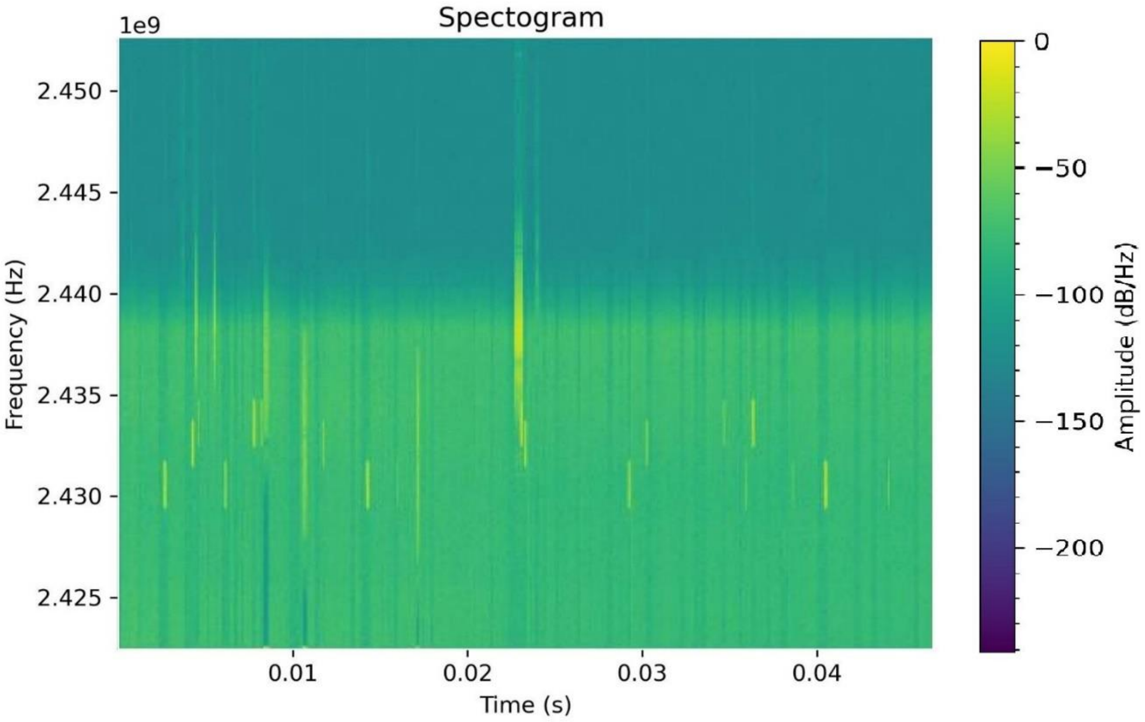


Figure 152 - Mavic 3 Spectrogram Graphical Signal Representation

For comparison Figure 153 shows the Air 2 S which operates using the OcuSync 3.0 datalink. Again the bottom half of the spectrum is displaying a great power output indicated by the yellow colour. Then bursts of yellow which span the whole frequency range but vary in intensity. Lastly smaller burst of yellow activity which span around 100MHz of the frequency band.

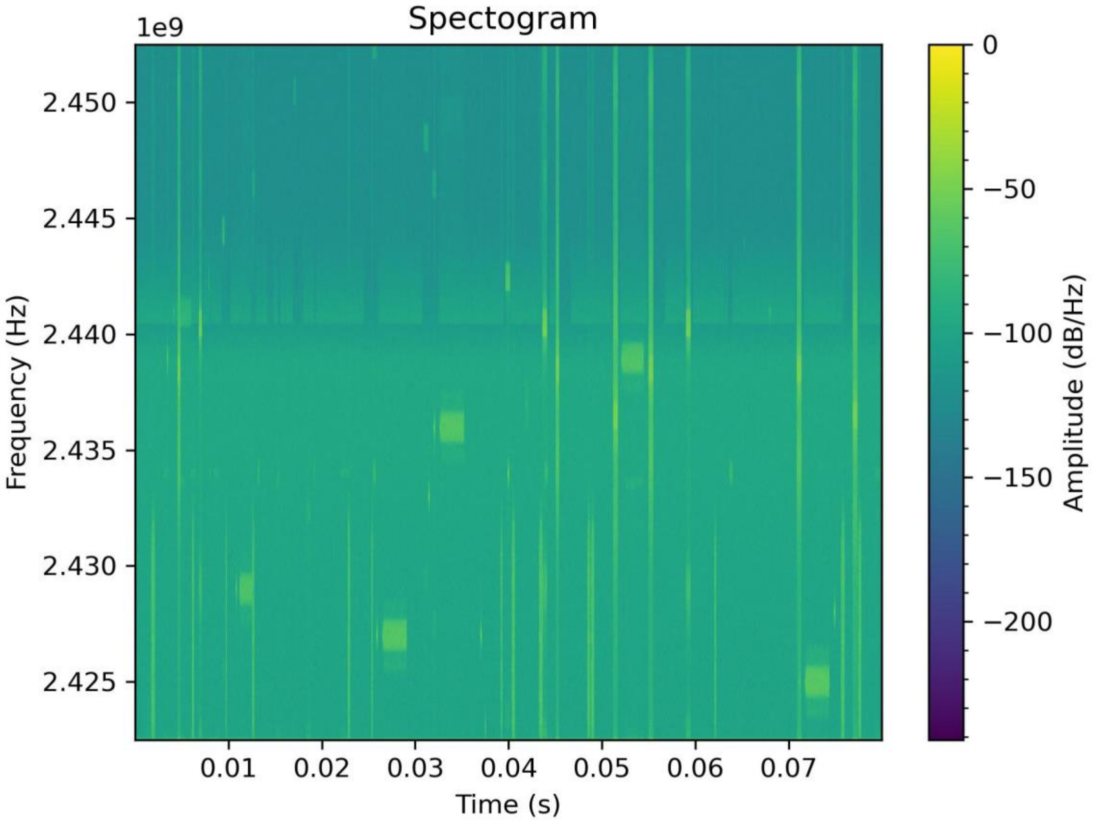


Figure 153 - Air 2 S Spectrogram Graphical Signal Representation

Comparing Figure 152 with Figure 153, the Mavic 3 displays more yellow activity. Visually the spectrum looks busier. The same increase in general power in the bottom half of the spectrum is observed in both Figure 152 and Figure 153. The Air 2 S in Figure 153 displays a consistent pattern visually compared with the Mavic 3. The visual inspection of the spectrogram for the OcuSync 3.0+ (Mavic 3) and the OcuSync 2.0 (Air 2 S) concludes that the images include both similarities and differences. If the CNN is including the similarities as features then the Mavic 3 should be detected as a UAS when the experiments are run.

6.2.2.2 DX8 DSMX Transmitter

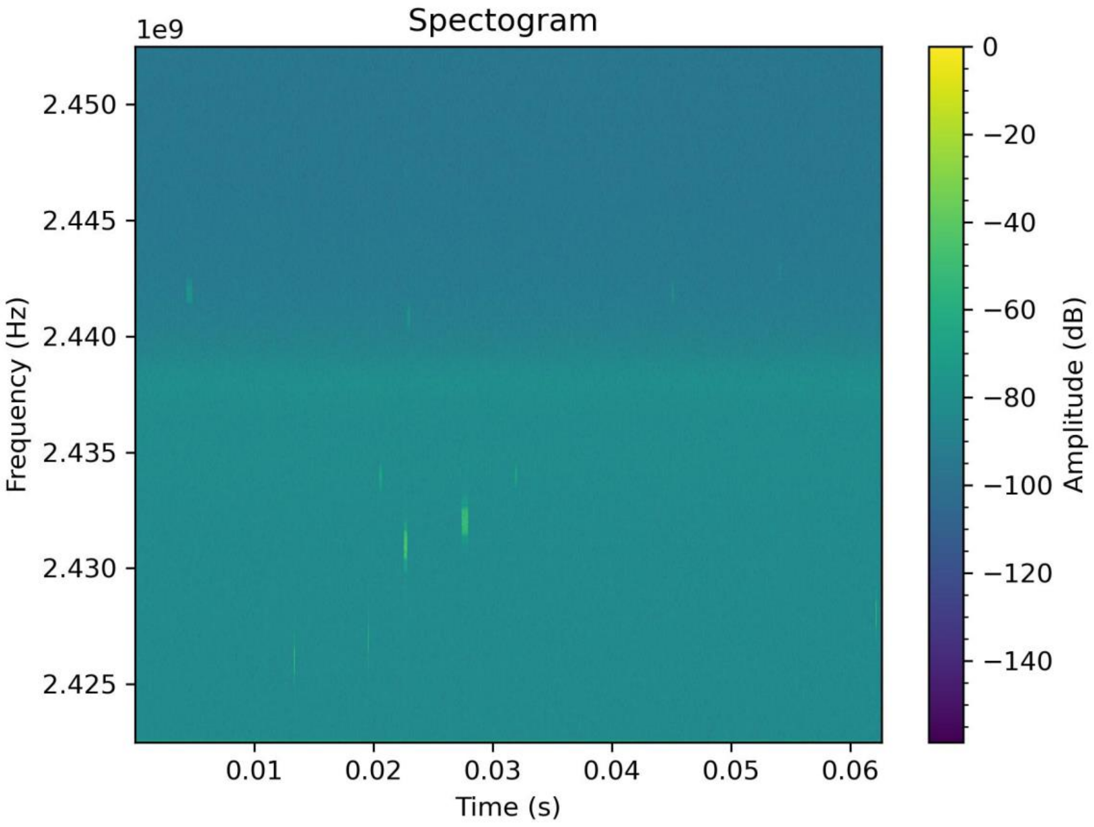


Figure 154 - DX8 Transmitter Spectrogram Graphical Signal Representation

Figure 154 shows the spectrogram for the DX8 transmitter which is operating with the DSMX protocol which utilises CDMA and frequency hopping across the entire band. The hopping happens at the rate of thousands per second so it is clear that although our experiments collect only 28MHz of the spectrum, it is still picking up the hopping due to the sheer frequency at which the transmission protocol hops.

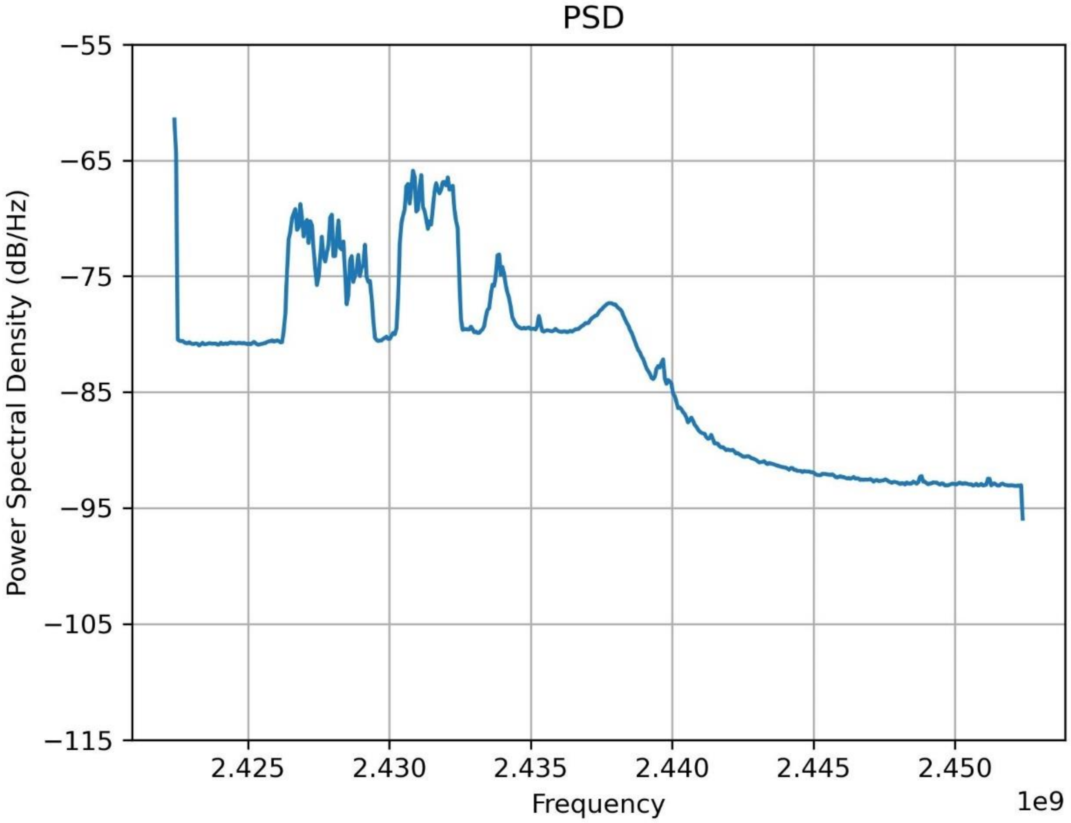


Figure 155 - DX8 Transmitter PSD Graphical Signal Representation

Figure 155 displays the PSD graphical signal representation of the DX8 transmitter. While we were unable to fly the platform, the signals were recorded whilst the joysticks were being moved around on the DX8 controller. Three main peaks can be observed in Figure 155 at around 2.426-2.429GHz, 2.431-2.433GHz and a slightly smaller one at 2.434GHz. Comparing back to a PSD for no UAS present in Figure 21 the DX8 transmitter signal is clearly visible in the spectrum.

6.2.2.3 eBee X

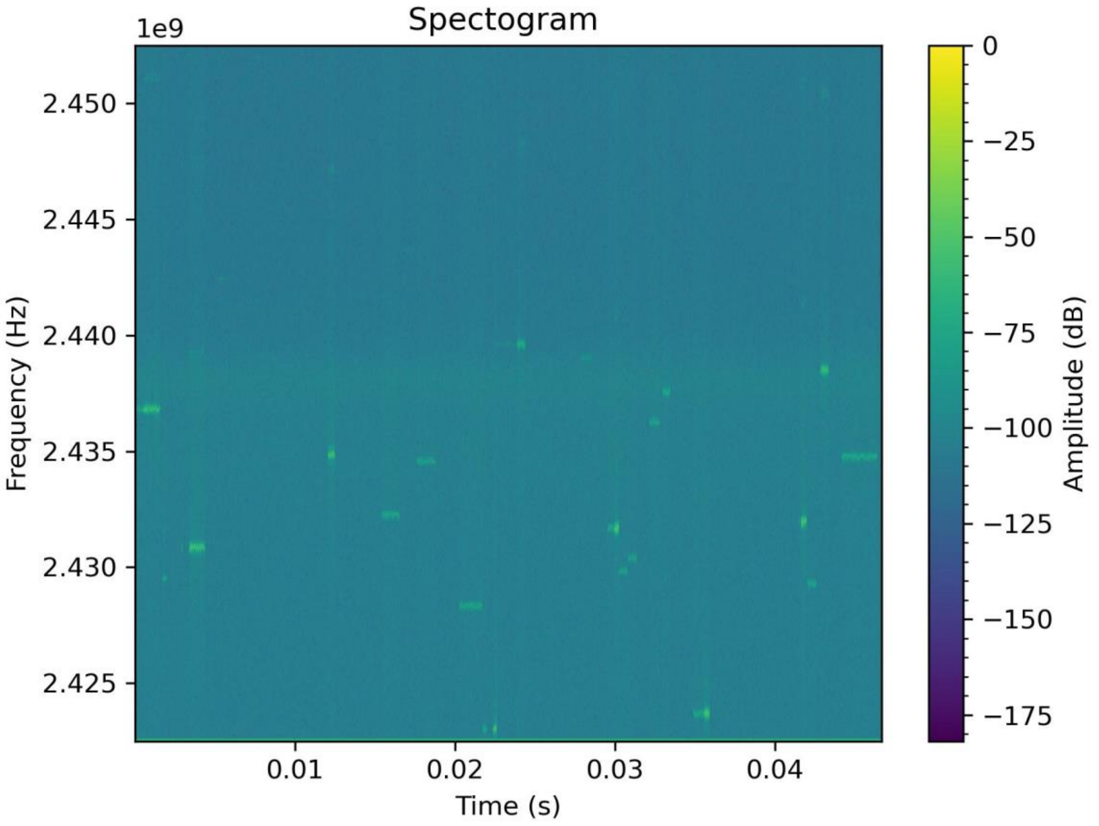


Figure 156 - eBee Switched on Spectrogram Graphical Signal Representation

Figure 156 displays the spectrogram when the eBee SenseFly surveillance UAS is switched on, connected to the ground modem but not taken off. The transmission between the UAS and the ground modem can be seen by observing the yellow bursts of activity, primarily in the lower end of the spectrum.

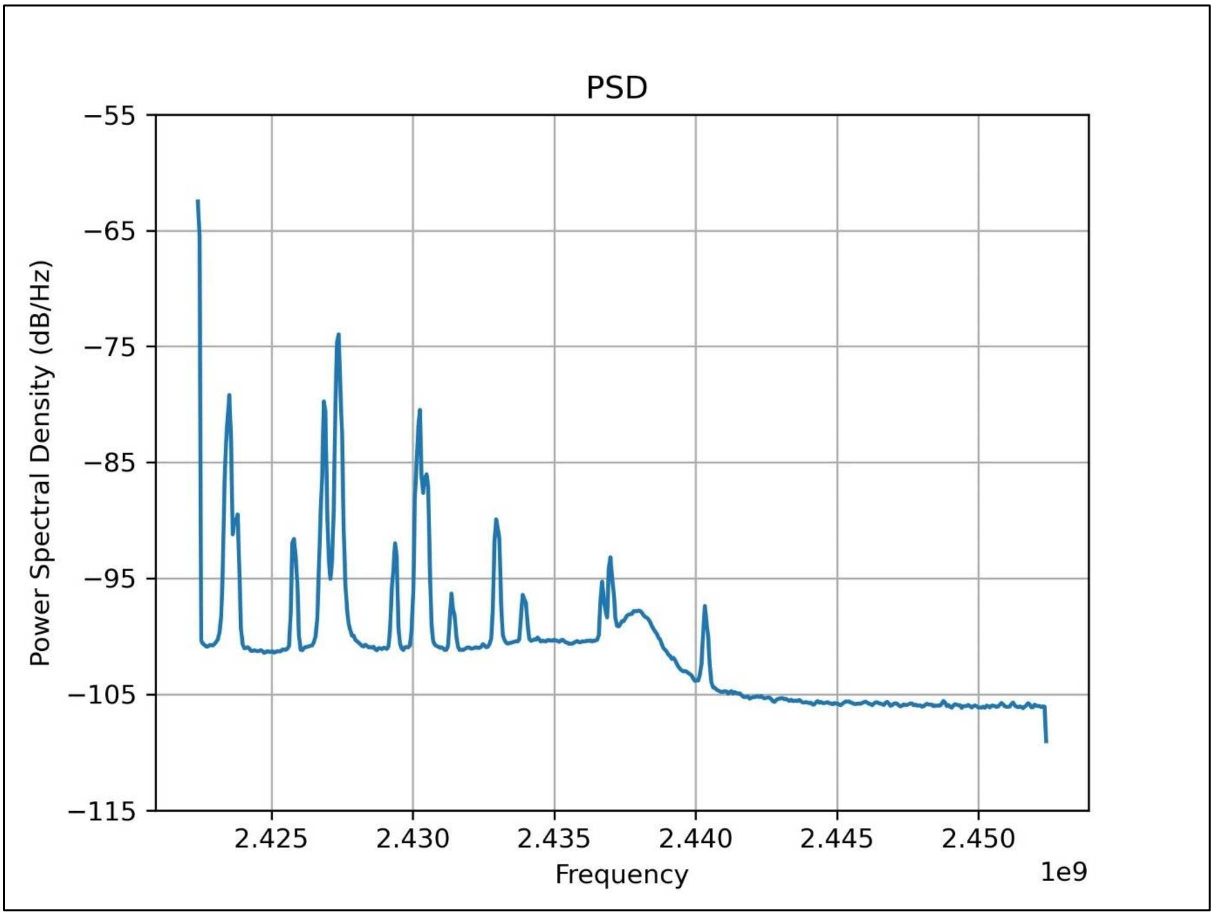


Figure 157 - eBee Switched on PSD Graphical Signal Representation

The yellow bursts of activity seen in the time domain in Figure 156 can be seen as spikes in frequency on the PSD in Figure 157. The peaks are frequent and occupy mainly the lower end of the frequency band. This correlates with the spectrogram in Figure 156 for the eBee switched on and connected to the ground modem.

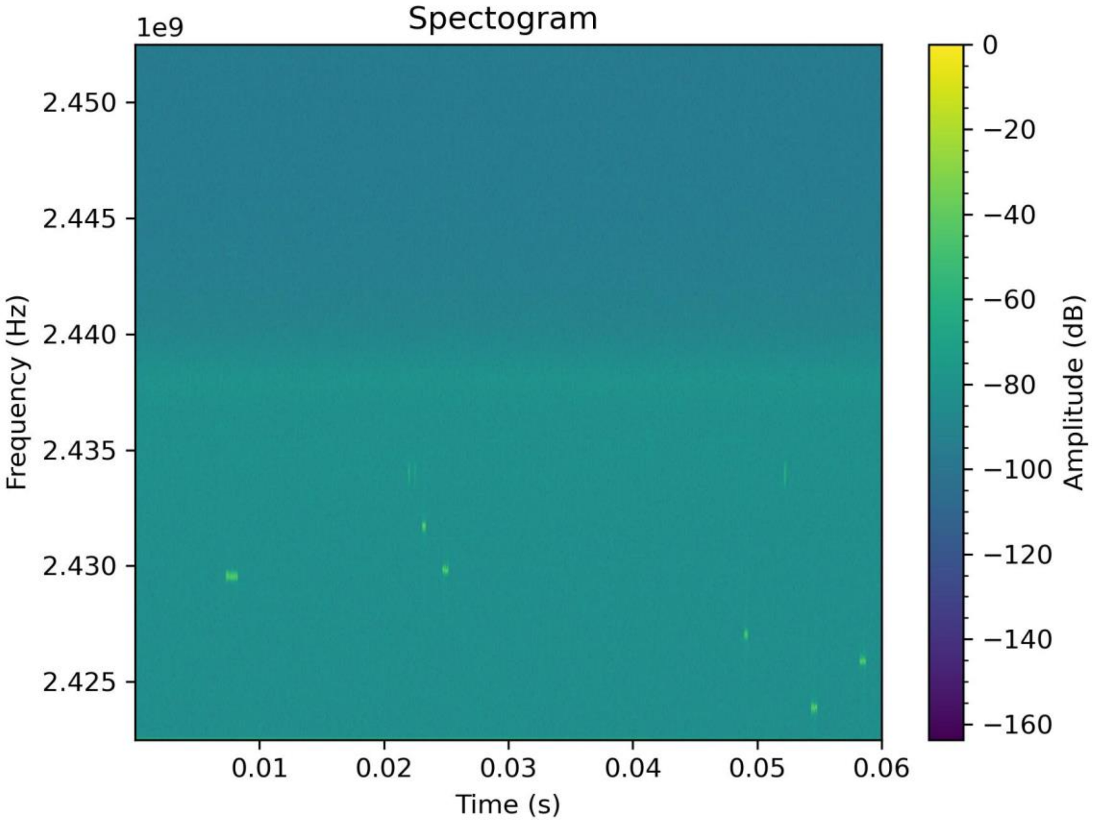


Figure 158 - eBee Flying Spectrogram Graphical Signal Representation

Figure 158 considers the eBee when it is flying and the graphical representation of that signal using a spectrogram for the time domain. The yellow bursts of activity can again be observed in the lower end of the frequency. Comparing this to the eBee switched on in Figure 156 it can be observed that the eBee flying produces more power generally in the lower end of the spectrum. This is observed by an increase in yellow background activity compared to the green in Figure 156. The other difference is the length and frequencies coverage of the bursts. In Figure 158 we see vertical bursts of activity which cover 1-2MHz and these bursts are not present when the platform is only switched on in Figure 156.

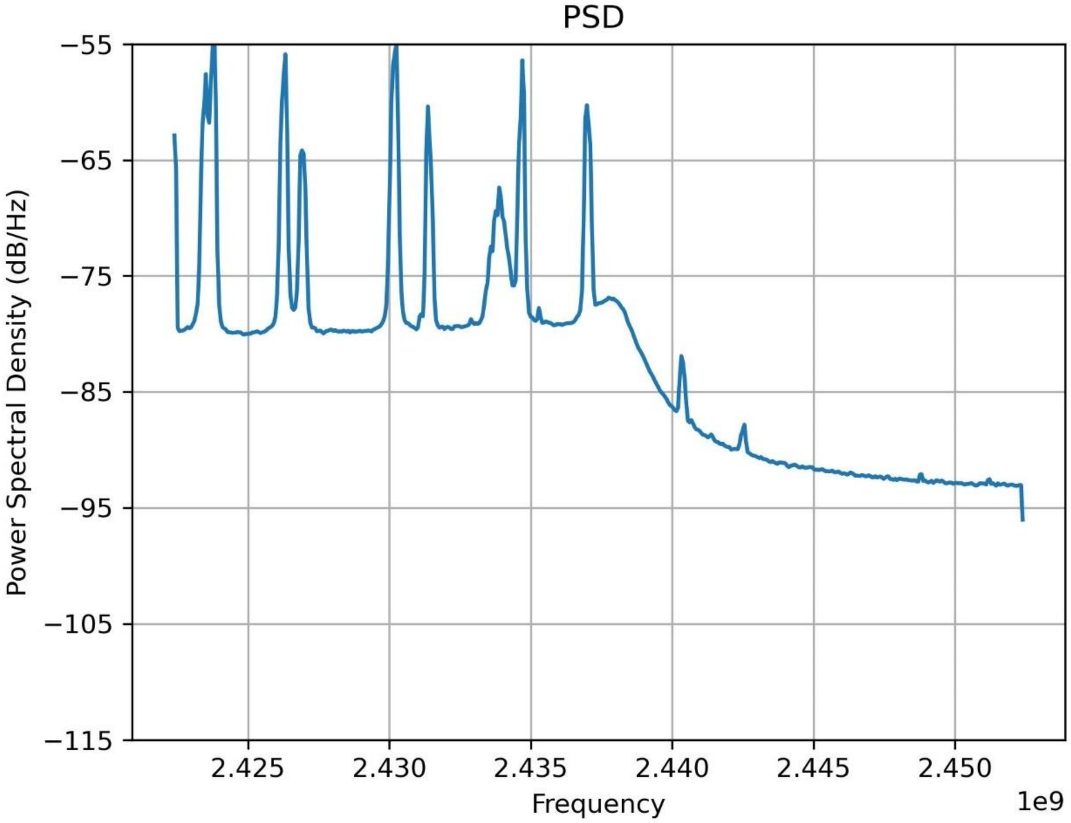


Figure 159 - eBee Flying PSD Graphical Signal Representation

Lastly Figure 159 shows the PSD for the eBee when it is flying. The entire signal is lifted by around 20dB/Hz compared to the signal observed for the platform only being switched on in Figure 157. In general the peaks which are present are higher than the ones observed in the switched on spectrogram.

6.2.3 Results

The results for understanding whether a signal which hasn’t been in the training data for a supervised learning model can still be predicted as a UAS is considered in two forms. Firstly the early warning system produced in section 6.1.2 is tested. Secondly to understand whether the positive or negative results from the early warning system are the signals are run through the machine learning model as evaluation sets giving a confusion matrix for the eBee X and the DX8 DSMX transmitter. A smaller set of images were collected for the Mavic 3 and these were also tested as an evaluation set. Lastly in section 6.2.3.3 the supervised model using VGG-16 CNN feature extraction is trained to include the following signals – Air 2 S, Parrot Disco, eBee X, DJI Inspire 2, Mavic Pro, Mavic Pro 2, Mavic Mini, No UAS, DJI Phantom 4 and the DX8 DSMX transmitter. Machine learning classifiers LR and kNN are evaluated alongside graphical signal representations PSD and spectrogram.

6.2.3.1 Early Warning

Table 44 shows the results from the low cost Raspberry Pi and Blade RF based early warning system when unknown signal types are presented to the machine learning model. Machine learning models for PSD graphical signal representations and spectrogram representations were combined with machine learning classifiers LR and kNN to test the results. The Mavic 3 which operated the OcuSync 3.0+ datalink is corrected classified a UAS for each model type. It can be assumed that the CNN feature extraction for the Air 2 S which operates the OcuSync 3.0 datalink are close enough to produce a positive result for the OcuSync 3.0+. The OcuSync 3.0+ is an evolution of the OcuSync 3.0. It was discussed earlier in Table 43 that the OcuSync 3.0 and the OcuSync 3.0+ have differing EIRP and bit rate transmissions. We can therefore assume that the basis for the CNN feature selection is not based on those two parameters but rather a pattern that the link is displaying is how it transmits. All four combinations of machine learning classifiers (LR of kNN) and graphical signal representation type (PSD or spectrogram) all produced positive results for a UAS detection when the Mavic 3 was flying.

Table 44 - Early Warning Predictions Unknown Signals

Classifier	Image	UAS Flying	Model Prediction	Prediction (%)	Time (s)
LR	PSD	Mavic 3	UAS Detected	100	37
LR	PSD	eBee X	No UAS Detected	99.86	16
LR	PSD	DX8	UAS Detected	100	22
LR	SPEC	Mavic 3	UAS Detected	100	21
LR	SPEC	eBee X	No UAS Detected	92.06	24
LR	SPEC	DX8	No UAS Detected	100	18
kNN	PSD	Mavic 3	UAS Detected	100	17
kNN	PSD	eBee X	No UAS Detected	100	24
kNN	PSD	DX8	No UAS Detected	100	21
kNN	SPEC	Mavic 3	UAS Detected	100	19
kNN	SPEC	eBee X	No UAS Detected	100	22
kNN	SPEC	DX8	No UAS Detected	100	20

Table 44 shows eBee X is not detected by any of the graphical signal representation and machine learning classifier combinations. This suggests that the CNN feature extraction is specific to the datalink type and that it would be pertinent to include a similar signal for positive identification in the future. More research would be needed to see how similar a signal would need to be to trigger a positive result. The results from the Mavic 3 however suggest this is possible and would cut down resources for training the dataset if less signal types were required. Table 44 shows that the DX8 DSMX transmitter is also in general classified as no UAS. There is one exception to this whereby the LR machine learning classifier with PSD graphical signal representation did produce a positive UAS detected result. This will be examined further in the next section 6.2.3.2 to understand whether this combination can consistently classify the DX8 DSMX transmitter correctly when the model was not trained on the signal to begin with.

6.2.3.2 Machine Learning Validation

Validation was performed using the machine learning models produced for classifying the Air 2 S, Parrot Disco, Inspire 2, Mavic Pro 2, Mavic Mini and No UAS. The UAS class included training samples from each of those platforms. For this experiment 20 samples of the class no UAS detected and 20 samples of the DX8 and eBee and 4 samples of the Mavic 3 were validated separately for the class UAS detected. Table 45 shows the results for the Mavic 3 for machine learning classifiers LR and kNN and for graphical signal representations PSD and spectrogram.

Table 45 - Mavic 3 Machine Learning Validation

Classifier	Image	Metric	Detection
LR	PSD	Accuracy (%)	100
LR	PSD	F1-Score (%)	100
LR	SPEC	Accuracy (%)	100
LR	SPEC	F1-Score (%)	100
kNN	PSD	Accuracy (%)	100

kNN	PSD	F1-Score (%)	100
kNN	SPEC	Accuracy (%)	96
kNN	SPEC	F1-Score (%)	96

Table 45 shows that the all machine learning classifiers and image types produce over 96% accuracy. Machine learning classifier LR outperforms kNN producing 100% accuracy and f1-score for both PSD and spectrogram graphical image representations.

Table 46 - eBee Machine Learning Validation

Classifier	Image	Metric	Detection
LR	PSD	Accuracy (%)	100
LR	PSD	F1-Score (%)	100
LR	SPEC	Accuracy (%)	74
LR	SPEC	F1-Score (%)	66
kNN	PSD	Accuracy (%)	60
kNN	PSD	F1-Score (%)	50
kNN	SPEC	Accuracy (%)	69
kNN	SPEC	F1-Score (%)	57

Table 46 shows the results for the eBee for machine learning classifiers LR and kNN and for graphical signal representations PSD and spectrogram. The highest performing combination is LR with PSD which producing 100% accuracy and 100% f1-score for classifying between UAS and no UAS using eBee signals. PSD outperforms spectrograms for the LR classifiers and spectrograms outperform PSD for kNN classifiers.

Table 47 - DX8 Machine Learning Validation

Classifier	Image	Metric	Detection
LR	PSD	Accuracy (%)	100
LR	PSD	F1-Score (%)	100
LR	SPEC	Accuracy (%)	82
LR	SPEC	F1-Score (%)	81
kNN	PSD	Accuracy (%)	89
kNN	PSD	F1-Score (%)	88
kNN	SPEC	Accuracy (%)	68
kNN	SPEC	F1-Score (%)	63

Table 47 shows the results for the DX8 DSMX transmitter for machine learning classifiers LR and kNN and for graphical signal representations PSD and spectrogram. The highest performing classifier again is LR with PSD images which achieves 100% accuracy and an f1-score of 100%. PSDs outperform spectrograms for both LR and kNN. In terms of machine learning classifiers LR outperforms kNN for both PSD and spectrogram graphical signal representations.

Figure 160 shows the confusion matrix for the Mavic 3 for PSD graphical signal representations. Machine learning classifier LR is shown on the left and kNN on the right. It can be observed from Figure 160 that both classifiers are able to distinguish between no UAS and UAS 100% of the time with no mis classifications. The confusion matrix for spectrograms with machine learning classifiers LR and kNN for the Mavic 3 is not shown as the confusion matrix present exactly the same with 100% accuracy and no misclassifications.

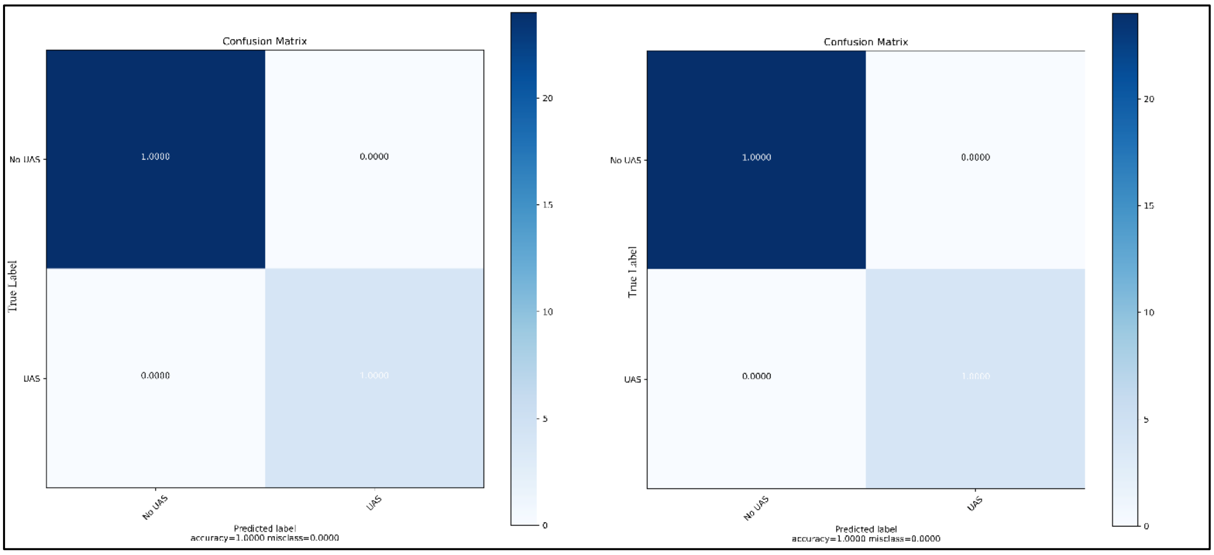


Figure 160 - Mavic 3 Confusion Matrix PSD Graphical Signal Representation LR (left) kNN (right)

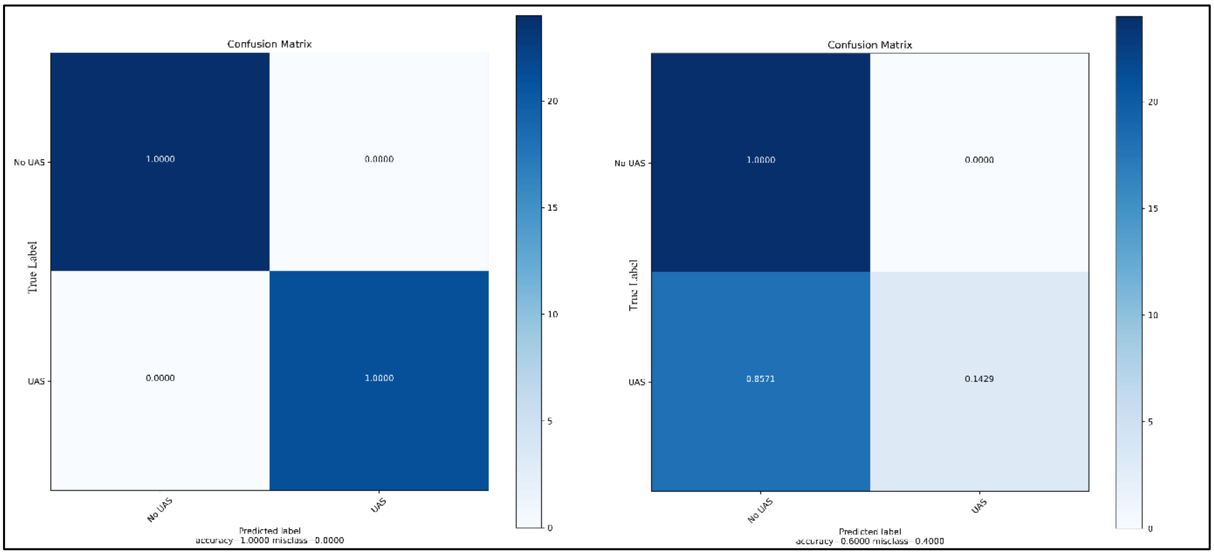


Figure 161 - eBee Confusion Matrix PSD Graphical Signal Representation LR (left) kNN (right)

Figure 161 shows the confusion matrix for PSD graphical signal representations and machine learning classifiers LR on the left and kNN on the right. It can be observed that the LR classifier is able to distinguish between UAS detected and no UAS detected 100% of the time with eBee PSDs. The kNN classifier on the right shows an overall misclassification of 40% indicating that PSD representations with LR is the highest performing combination for classifying eBee signals.

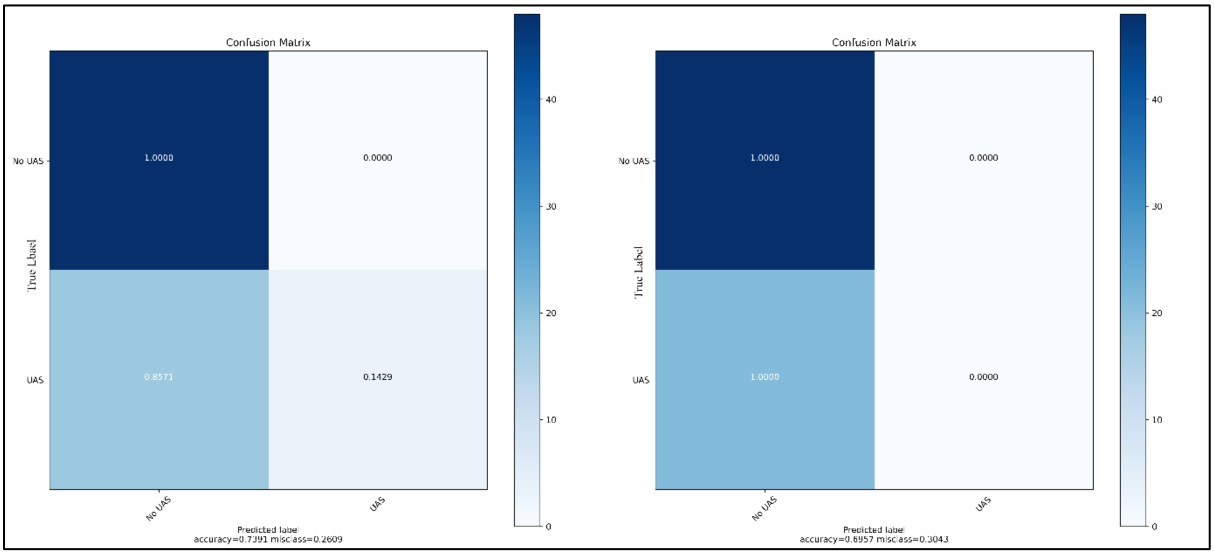


Figure 162 - eBee Confusion Matrix Spectrogram Graphical Signal Representation LR (left) kNN (right)

Figure 162shows the confusion matrix for the eBee X with spectrogram graphical signal representations and machine learning classifiers LR on the left and kNN on the right. The LR classifier shows an accuracy of 74% with 26% misclassifications and kNN shows 70% accuracy with 30% misclassification. Compared with the PSD representation in Figure 161 which could achieve 100% accuracy with no misclassifications for LR classifier, the performance of the spectrogram image is lower.

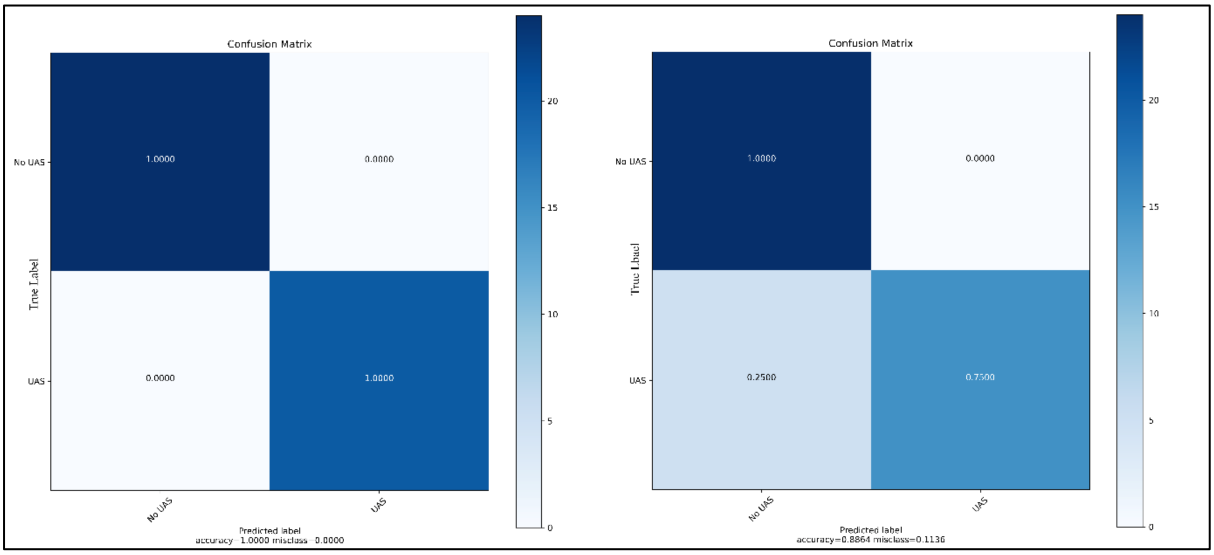


Figure 163 - DX8 Confusion Matrix PSD Graphical Signal Representation LR (left) kNN (right)

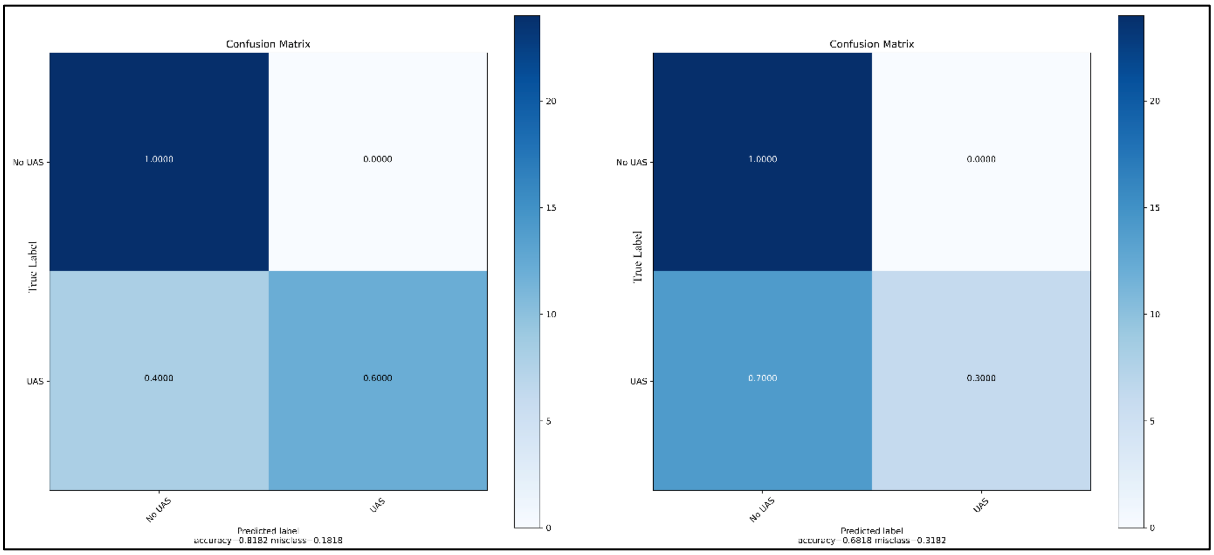


Figure 164 - DX8 Confusion Matrix Spectrogram LR (left) kNN (right)

Figure 163 shows the confusion matrix for the DX8 DSMX transmitter with PSD graphical signal representations and machine learning classifiers LR on the left and kNN on the right. Again the LR classifier is able to classify the DX8 signal as UAS and the no UAS signals as no UAS present 100% of the time with no misclassifications using PSD graphical signal representation. With the kNN classifier the accuracy drops to 89% with 11% misclassifications.

Lastly Figure 164 shows the confusion matrix for spectrogram graphical signal representations and machine learning classifiers LR on the left and kNN on the right for the Dx8 DSMX transmitter. It can be seen that the LR classifier can achieve 88% accuracy with 12% misclassifications. The kNN classifier does not perform as well as LR producing 68% accuracy with 32% misclassifications. Comparing the LR result of 88% accuracy for spectrogram with the PSD in Figure 163, representing the signal in the frequency domain with the PSD outperforms the spectrogram achieving 100% accuracy with no misclassifications.

6.2.3.3 Re-trained Supervised Models

The supervised model using VGG-16 CNN feature extraction were trained to include the following ten UAS signals classes – Air 2 S, Parrot Disco, eBee X, DJI Inspire 2, Mavic Pro, Mavic Pro 2, Mavic Mini, No UAS, DJI Phantom 4 and the DX8 DSMX transmitter. Machine learning classifiers LR and kNN were then evaluated alongside graphical signal representations PSD and spectrogram. Images for Air 2 S, Parrot Disco, DJI Inspire 2, Mavic Pro, Mavic Pro 2, Mavic Mini, No UAS and the DJI Phantom 4 all contained 250 images and were split 200 for use with 5 fold cross validation and 50 images were kept entirely separate to evaluate the system as a holdout dataset. eBee X and the DX8 DSMX contained 100 images were split 80/20 for 5 fold cross validation and a separate holdout dataset for evaluation to ensure the model was not overfitting.

Cross Validation Results

Table 48 shows the 5-fold cross validation results for the re-trained supervised model for the 10 classes of UAS type using VGG-16 CNN feature extraction followed by combinations of machine learning classifiers LR and kNN and graphical signal representations PSD and spectrogram. Table 48 shows that all combinations of the machine learning classifiers and graphical signal representations are able to produce accuracy and f1-scores of over 89.5% (+/-2.8%). The highest performing model is the LR machine learning classifier with PSD graphical signal representation. This produces an accuracy and f1-score of 98.7% (+/-0.4%) for the 10 class UAS type classifier. The spectrogram with LR was only marginally lower producing 98.4% (+/-0.4%) for both accuracy and f1-score.

Table 48 - Re-trained Supervised Model Cross Validation Results

Classifier	Image	Metric	Validation
LR	PSD	Accuracy (%)	98.7(+/-0.4)
LR	PSD	F1-Score (%)	98.7(+/-0.4)
LR	SPEC	Accuracy (%)	98.4(+/-0.4)
LR	SPEC	F1-Score (%)	98.4(+/-0.4)
kNN	PSD	Accuracy (%)	93.2 (+/-1.0)
kNN	PSD	F1-Score (%)	93.2 (+/-1.1)
kNN	SPEC	Accuracy (%)	89.5 (+/-2.8)
kNN	SPEC	F1-Score (%)	89.5 (+/-2.7)

Figure 167 shows the confusion matrix for the PSD using kNN machine learning classifier. It can be observed that the mix-classifications occur between the variants of the DJI datalinks

OcuSync and Lightbridge. No misclassification occurs with the Wi-Fi platforms Mavic Mini and the Parrot Disco. Also there are no mis-classifications seen with the DX8 and the eBee. Although misclassifications do occur between the OcuSync and Lightbridge variants, they are small. The overall accuracy is high at 93% with only 6.8% overall misclassifications.

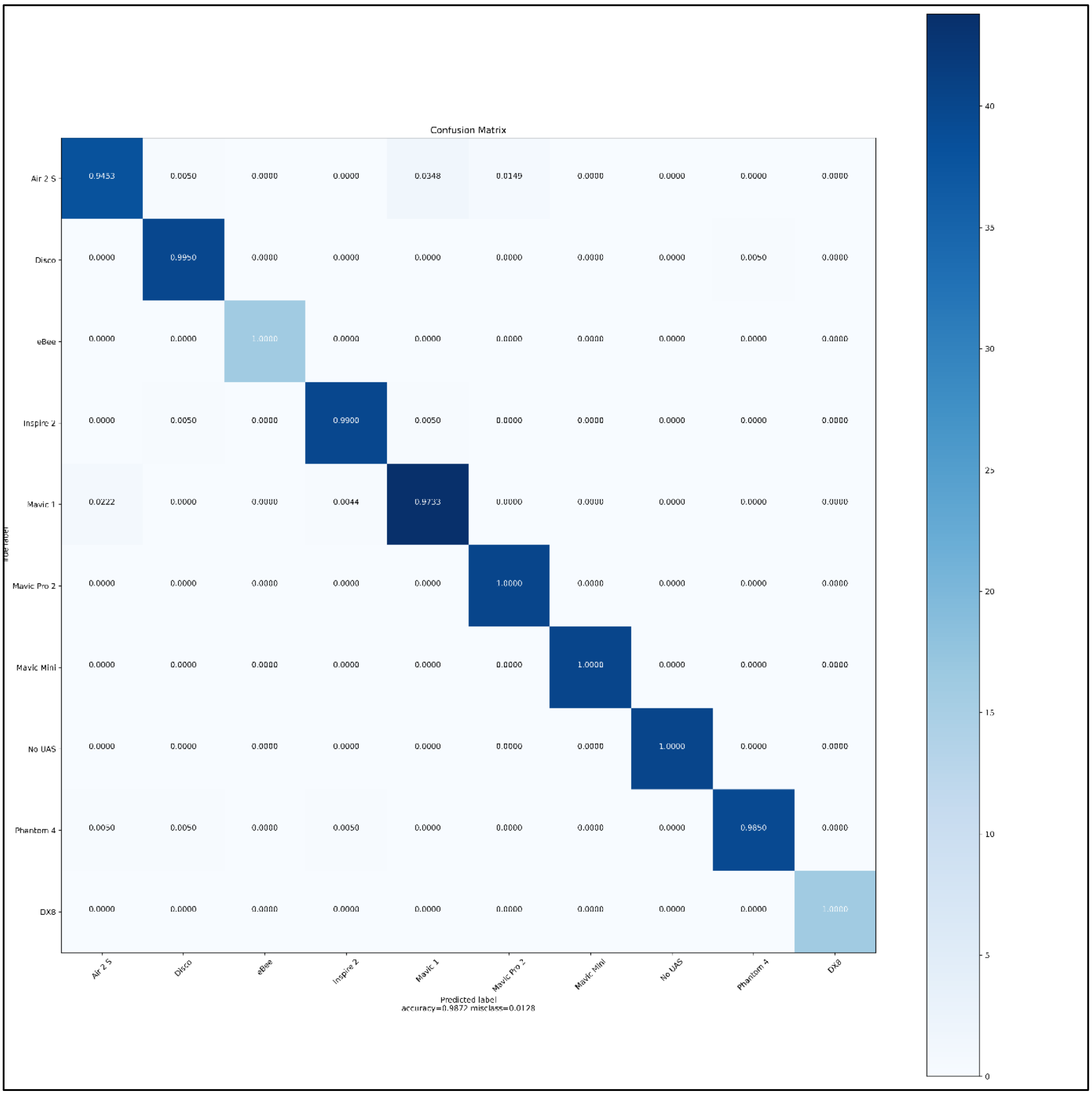
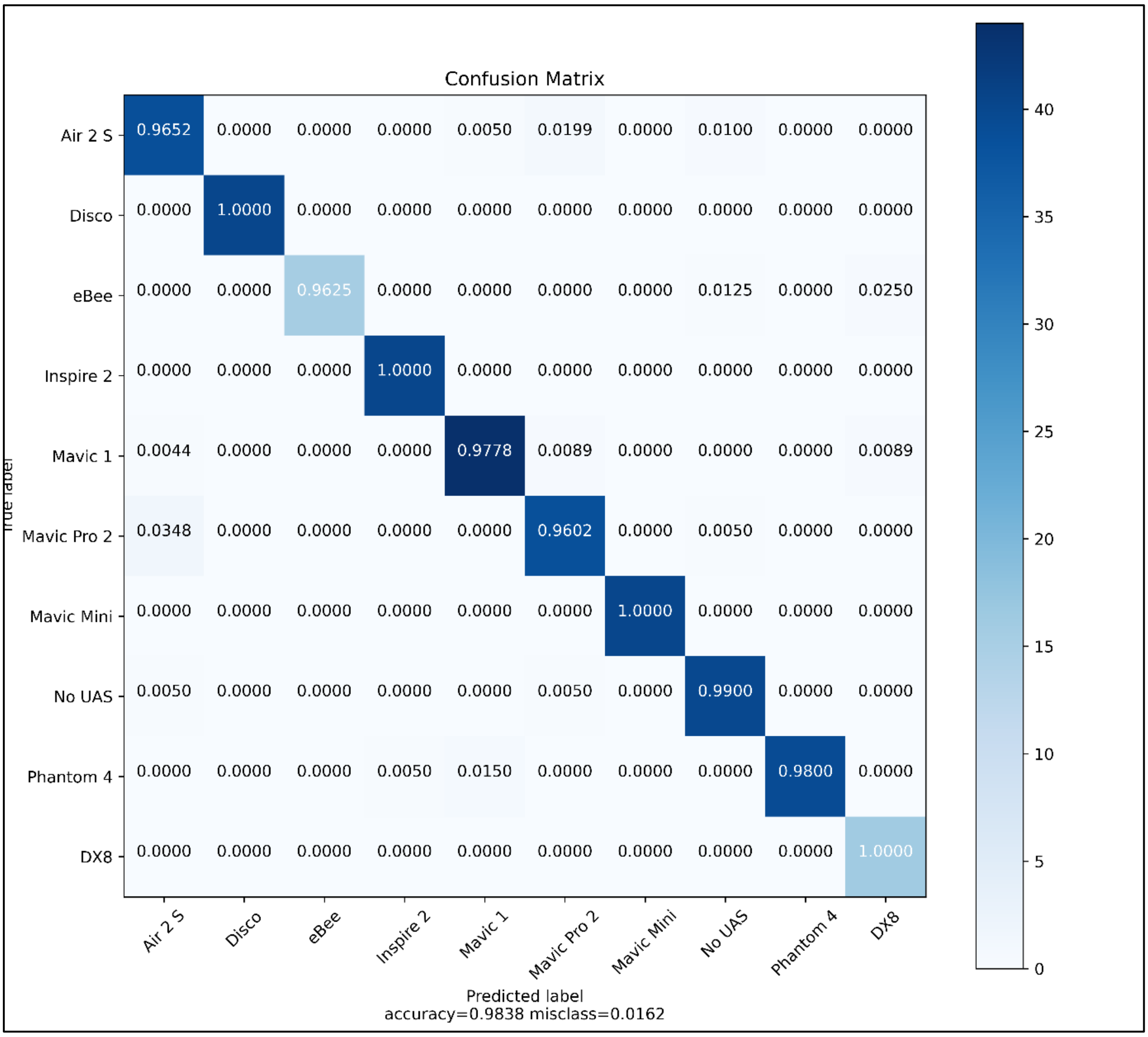


Figure 165 - Cross Validation Confusion Matrix PSD Graphical Signal Representation LR

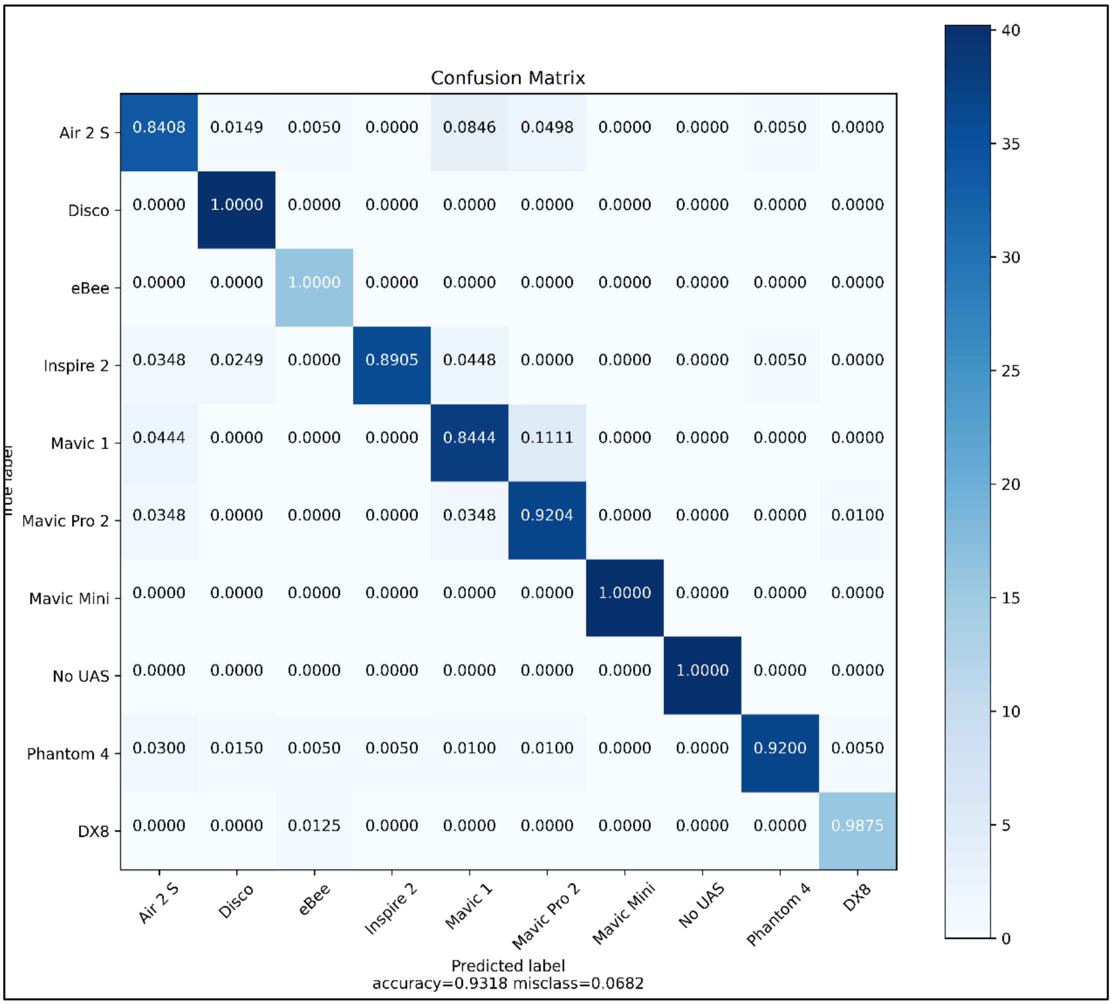
Figure 165 shows the cross validation confusion matrix for graphical signal representation PSD with machine learning classifier LR. It can be observed that there are some very minor misclassification but in total they amount to only 1.2%, giving the model an overall accuracy of 98.8%.



	Air 2 S	Disco	eBee	Inspire 2	Mavic 1	Mavic Pro 2	Mavic Mini	No UAS	Phantom 4	DX8
Air 2 S	0.9652	0.0000	0.0000	0.0000	0.0050	0.0199	0.0000	0.0100	0.0000	0.0000
Disco	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
eBee	0.0000	0.0000	0.9625	0.0000	0.0000	0.0000	0.0000	0.0125	0.0000	0.0250
Inspire 2	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Mavic 1	0.0044	0.0000	0.0000	0.0000	0.9778	0.0089	0.0000	0.0000	0.0000	0.0089
Mavic Pro 2	0.0348	0.0000	0.0000	0.0000	0.0000	0.9602	0.0000	0.0050	0.0000	0.0000
Mavic Mini	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
No UAS	0.0050	0.0000	0.0000	0.0000	0.0000	0.0050	0.0000	0.9900	0.0000	0.0000
Phantom 4	0.0000	0.0000	0.0000	0.0050	0.0150	0.0000	0.0000	0.0000	0.9800	0.0000
DX8	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000

Figure 166 - Cross Validation Confusion Matrix Spectrogram Graphical Signal Representation LR

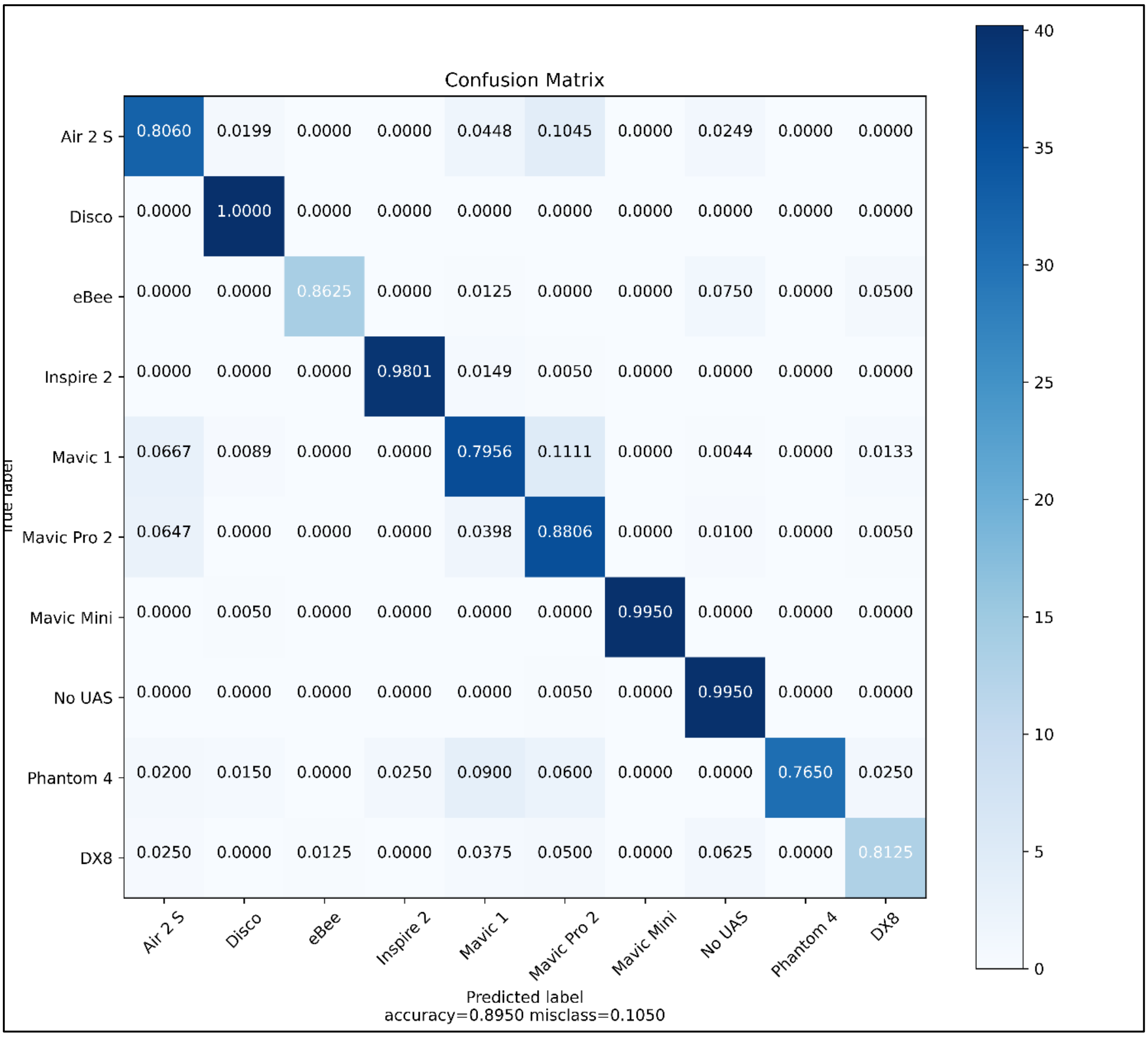
Figure 166 shows the cross validation confusion matrix for graphical signal representation spectrogram with machine learning classifier LR. Very minor misclassifications are seen for the eBee, Air 2 S, Mavic Pro, Mavic Pro 2, no UAS and the Phantom 4. The overall misclassifications are only 1.6% and over accuracy is high at 98.4%, only 0.3% lower than the PSD implementation with LR in Figure 165.



	Air 2 S	Disco	eBee	Inspire 2	Mavic 1	Mavic Pro 2	Mavic Mini	No UAS	Phantom 4	DX8
Air 2 S	0.8408	0.0149	0.0050	0.0000	0.0846	0.0498	0.0000	0.0000	0.0050	0.0000
Disco	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
eBee	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Inspire 2	0.0348	0.0249	0.0000	0.8905	0.0448	0.0000	0.0000	0.0000	0.0050	0.0000
Mavic 1	0.0444	0.0000	0.0000	0.0000	0.8444	0.1111	0.0000	0.0000	0.0000	0.0000
Mavic Pro 2	0.0348	0.0000	0.0000	0.0000	0.0348	0.9204	0.0000	0.0000	0.0000	0.0100
Mavic Mini	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
No UAS	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000
Phantom 4	0.0300	0.0150	0.0050	0.0050	0.0100	0.0100	0.0000	0.0000	0.9200	0.0050
DX8	0.0000	0.0000	0.0125	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.9875

Figure 167 - Cross Validation Confusion Matrix PSD Graphical Signal Representation kNN

Figure 168 shows the confusion matrix for the kNN machine learning classifier with graphical signal representation spectrogram. While in Figure 167 with the PSD misclassifications were observed between the OcuSync and Lightbridge datalinks, the spectrogram observes misclassifications for all the classes apart from the Parrot Disco. Figure 168 shows that even the no UAS class has misclassification and both of the new signals introduced including the eBee X and the DX8 DSMX. This suggests that representation of the signal in the frequency domain using the PSD will produce higher accuracy results for classification of 10 UAS signal types over the spectrogram and time domain representation.



	Air 2 S	Disco	eBee	Inspire 2	Mavic 1	Mavic Pro 2	Mavic Mini	No UAS	Phantom 4	DX8
Air 2 S	0.8060	0.0199	0.0000	0.0000	0.0448	0.1045	0.0000	0.0249	0.0000	0.0000
Disco	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
eBee	0.0000	0.0000	0.8625	0.0000	0.0125	0.0000	0.0000	0.0750	0.0000	0.0500
Inspire 2	0.0000	0.0000	0.0000	0.9801	0.0149	0.0050	0.0000	0.0000	0.0000	0.0000
Mavic 1	0.0667	0.0089	0.0000	0.0000	0.7956	0.1111	0.0000	0.0044	0.0000	0.0133
Mavic Pro 2	0.0647	0.0000	0.0000	0.0000	0.0398	0.8806	0.0000	0.0100	0.0000	0.0050
Mavic Mini	0.0000	0.0050	0.0000	0.0000	0.0000	0.0000	0.9950	0.0000	0.0000	0.0000
No UAS	0.0000	0.0000	0.0000	0.0000	0.0000	0.0050	0.0000	0.9950	0.0000	0.0000
Phantom 4	0.0200	0.0150	0.0000	0.0250	0.0900	0.0600	0.0000	0.0000	0.7650	0.0250
DX8	0.0250	0.0000	0.0125	0.0000	0.0375	0.0500	0.0000	0.0625	0.0000	0.8125

Figure 168 - Cross Validation Confusion Matrix Spectrogram Graphical Signal Representation kNN

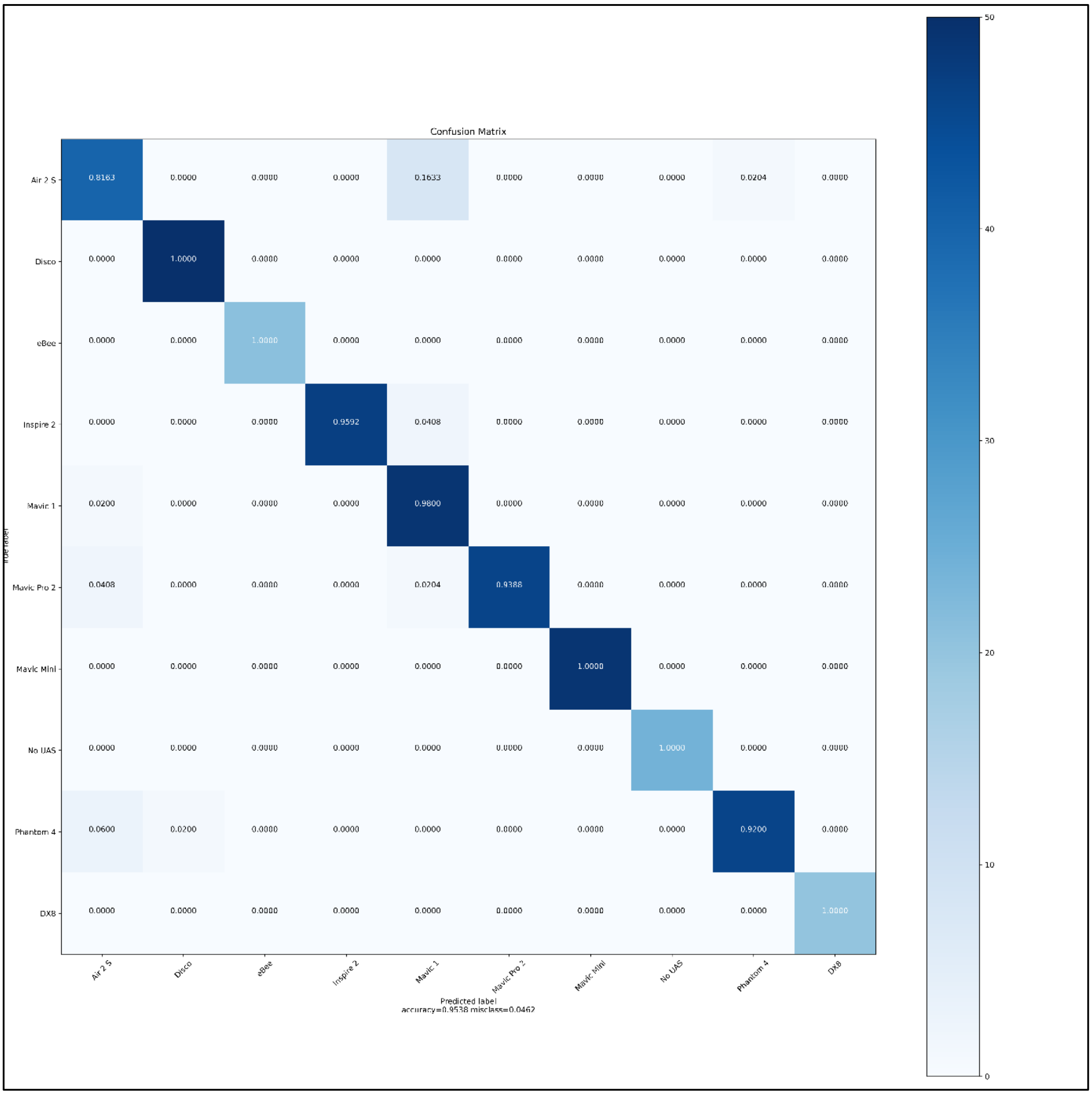
Holdout Dataset Results

Table 49 shows the hold-out dataset evaluation results for the re-trained supervised model for the 10 classes of UAS type using VGG-16 CNN feature extraction followed by combinations of machine learning classifiers LR and kNN and graphical signal representations PSD and spectrogram. When compared with the results from Table 48 a slight decrease is observed for the holdout dataset accuracy and f1-score in general. As the decrease is only marginal it is suggested that the model does not appear to be overfitting. Table 49 shows that the LR classifier with the spectrogram graphical signal representation is the highest performing combination which is in contrast to the PSD in the cross validation results in Table 48. The spectrogram and LR combination gives a result of over 98% accuracy for the holdout dataset. This result is the same as the cross validation result of 98%, indicating that the model is not overfitting.

Table 49 - Re-trained Supervised Model Hold-out Results

Classifier	Image	Metric	Hold-out Evaluation
LR	PSD	Accuracy (%)	95.4
LR	PSD	F1-Score (%)	95.4
LR	SPEC	Accuracy (%)	98.0
LR	SPEC	F1-Score (%)	98.0
kNN	PSD	Accuracy (%)	88.3
kNN	PSD	F1-Score (%)	88.1
kNN	SPEC	Accuracy (%)	82.9
kNN	SPEC	F1-Score (%)	82.4

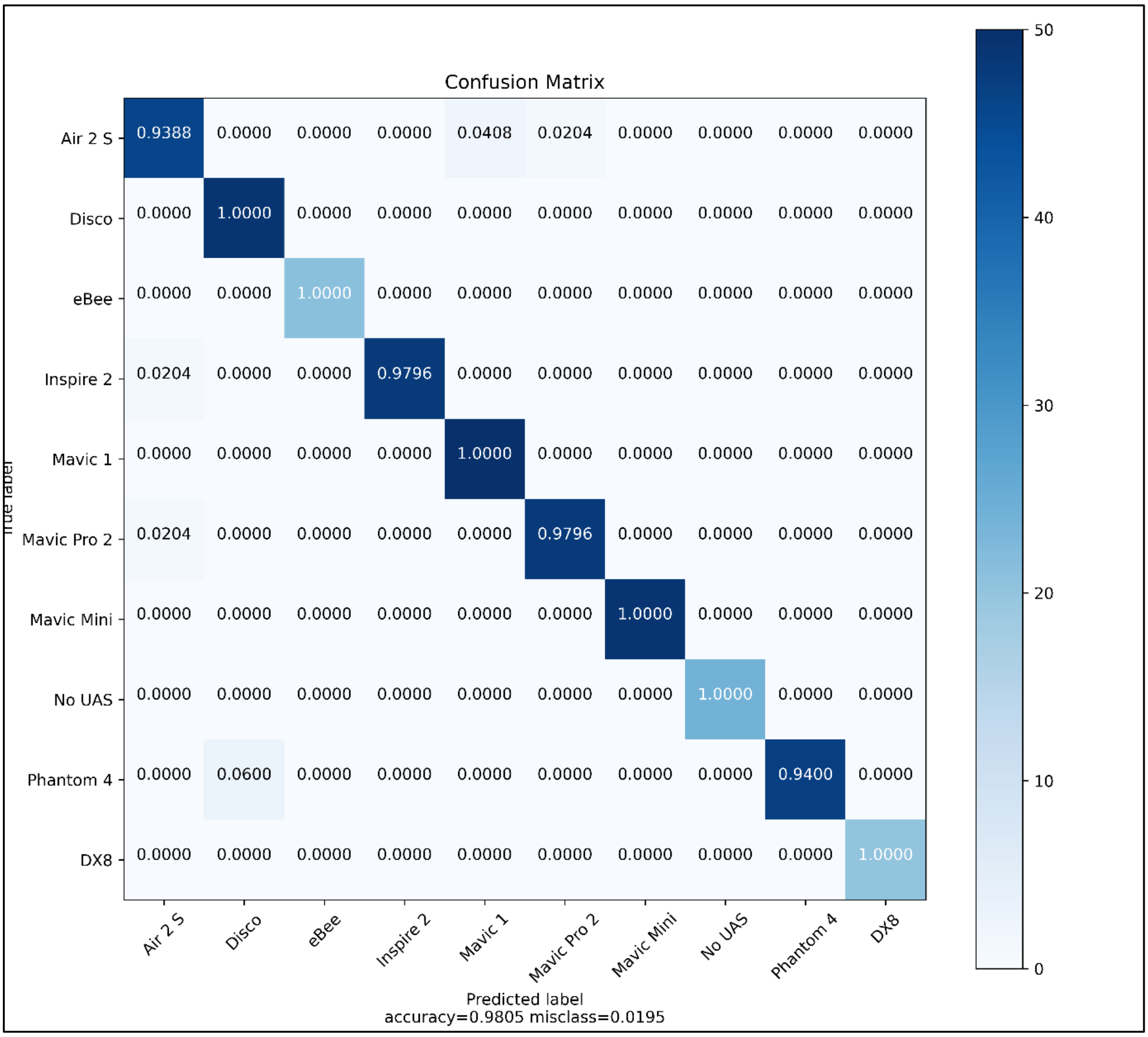
Figure 171 shows the holdout dataset confusion matrix for the kNN machine learning classifier with PSD graphical signal representation. It confirms the results from the cross validation confusion matrix in Figure 167 that misclassification only occurs between the variants of the OcuSync and Lightbridge DJI datalinks. The DJI Wi-Fi, Parrot Wi-Fi, eBee, DX8 DSMX and the class no UAS all perform without any misclassifications. In Figure 171 a slightly lower accuracy of 88.3% and an overall higher misclassification of 11.6% can be observed compared to the cross validation confusion matrix in Figure 167.



	Air 2 S	Disco	eBee	Inspire 2	Mavic 1	Mavic Pro 2	Mavic Mini	No UAS	Phantom 4	DX8
Air 2 S	0.8163	0.0000	0.0000	0.0000	0.1633	0.0000	0.0000	0.0000	0.0204	0.0000
Disco	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
eBee	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Inspire 2	0.0000	0.0000	0.0000	0.9592	0.0408	0.0000	0.0000	0.0000	0.0000	0.0000
Mavic 1	0.0200	0.0000	0.0000	0.0000	0.9800	0.0000	0.0000	0.0000	0.0000	0.0000
Mavic Pro 2	0.0408	0.0000	0.0000	0.0000	0.0204	0.9388	0.0000	0.0000	0.0000	0.0000
Mavic Mini	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
No UAS	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000
Phantom 4	0.0600	0.0200	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.9200	0.0000
DX8	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000

Figure 169 – Holdout Confusion Matrix PSD Graphical Signal Representation LR

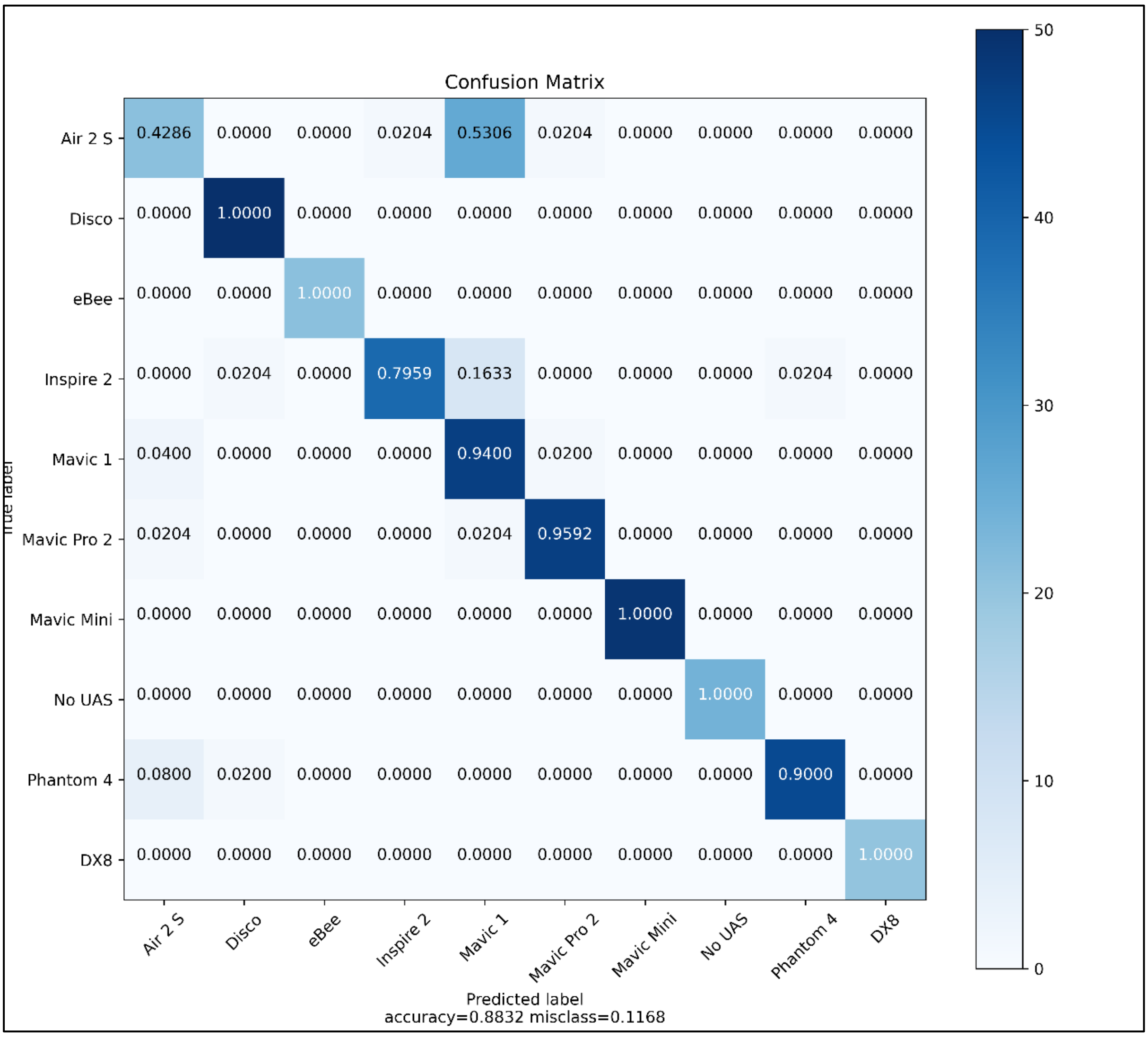
Figure 169 shows the holdout confusion matrix for the PSD with classifier LR. It can be observed, as with the PSD and kNN classifier in Figure 171 that the misclassifications occur between the OcuSync and Lightbridge datalinks. Distinguishing between different Wi-Fi datalinks, the eBee X, the DX8 DSMX transmitter and the class no UAS present no misclassifications. However overall accuracy remains high at 95.4% with misclassifications of 4.6%.



	Air 2 S	Disco	eBee	Inspire 2	Mavic 1	Mavic Pro 2	Mavic Mini	No UAS	Phantom 4	DX8
Air 2 S	0.9388	0.0000	0.0000	0.0000	0.0408	0.0204	0.0000	0.0000	0.0000	0.0000
Disco	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
eBee	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Inspire 2	0.0204	0.0000	0.0000	0.9796	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Mavic 1	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Mavic Pro 2	0.0204	0.0000	0.0000	0.0000	0.0000	0.9796	0.0000	0.0000	0.0000	0.0000
Mavic Mini	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
No UAS	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000
Phantom 4	0.0000	0.0600	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.9400	0.0000
DX8	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000

Figure 170 - Holdout Confusion Matrix Spectrogram Graphical Signal Representation LR

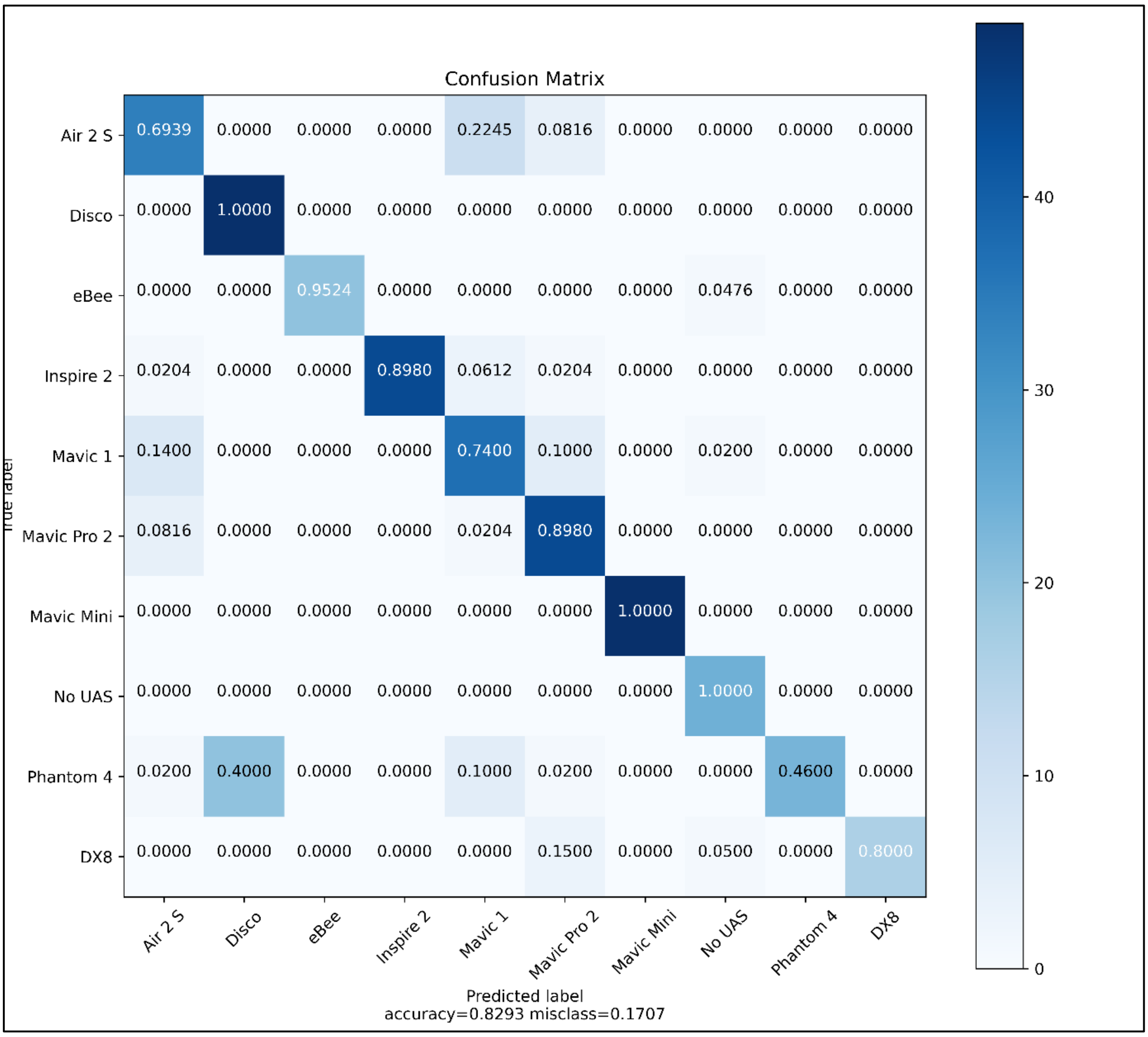
Figure 170 shows the confusion matrix for the holdout evaluation dataset for spectrogram graphical signal representations and machine learning classifier LR. This is the highest performing combination for the holdout evaluation results. Overall accuracy is 98.05% and misclassifications 1.95%. It can be seen in Figure 170 that all the misclassifications occur between the DJI OcuSync variants and the DJI Lightbridge variants. There are no misclassification between the two Wi-Fi datalinks or the eBee X, the DSMX on the DX8 transmitter and the class no UAS. The results for the holdout evaluation dataset in the confusion matrix in Figure 170 are exactly the same as the results in Figure 166 for the cross validation. This indicates that the model is not overfitting and the results can be trusted for predicting new UAS signals of these types.



	Air 2 S	Disco	eBee	Inspire 2	Mavic 1	Mavic Pro 2	Mavic Mini	No UAS	Phantom 4	DX8
Air 2 S	0.4286	0.0000	0.0000	0.0204	0.5306	0.0204	0.0000	0.0000	0.0000	0.0000
Disco	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
eBee	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Inspire 2	0.0000	0.0204	0.0000	0.7959	0.1633	0.0000	0.0000	0.0000	0.0204	0.0000
Mavic 1	0.0400	0.0000	0.0000	0.0000	0.9400	0.0200	0.0000	0.0000	0.0000	0.0000
Mavic Pro 2	0.0204	0.0000	0.0000	0.0000	0.0204	0.9592	0.0000	0.0000	0.0000	0.0000
Mavic Mini	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
No UAS	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000
Phantom 4	0.0800	0.0200	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.9000	0.0000
DX8	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000

Figure 171 - Holdout Evaluation Confusion Matrix PSD Graphical Signal Representation kNN

Figure 172 shows the holdout dataset evaluation results in the form of a confusion matrix for the spectrogram images and kNN machine learning classifier. Compared to Figure 168 an increase in misclassification with the DX8 DSMX transmitter is seen. The classes of Wi-Fi operated UAS and the class for no UAS present with no misclassifications. The overall accuracy is less than in Figure 168 at 82.9% and showing 17% misclassifications.



	Air 2 S	Disco	eBee	Inspire 2	Mavic 1	Mavic Pro 2	Mavic Mini	No UAS	Phantom 4	DX8
Air 2 S	0.6939	0.0000	0.0000	0.0000	0.2245	0.0816	0.0000	0.0000	0.0000	0.0000
Disco	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
eBee	0.0000	0.0000	0.9524	0.0000	0.0000	0.0000	0.0000	0.0476	0.0000	0.0000
Inspire 2	0.0204	0.0000	0.0000	0.8980	0.0612	0.0204	0.0000	0.0000	0.0000	0.0000
Mavic 1	0.1400	0.0000	0.0000	0.0000	0.7400	0.1000	0.0000	0.0200	0.0000	0.0000
Mavic Pro 2	0.0816	0.0000	0.0000	0.0000	0.0204	0.8980	0.0000	0.0000	0.0000	0.0000
Mavic Mini	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
No UAS	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000
Phantom 4	0.0200	0.4000	0.0000	0.0000	0.1000	0.0200	0.0000	0.0000	0.4600	0.0000
DX8	0.0000	0.0000	0.0000	0.0000	0.0000	0.1500	0.0000	0.0500	0.0000	0.8000

Figure 172 - Holdout Evaluation Confusion Matrix Spectrogram Graphical Signal Representation kNN

6.3 Discussion

6.3.1.1 Early Warning & Machine Learning Validation

Overall the results from the early warning testing showed that the Mavic 3 and the OcuSync 3.0+ was able to be detected on a system which was trained on previous versions of the OcuSync datalink, including the OcuSync 3.0 on the Air 2 S platform. This is a positive result for detecting UAS evolutionary datalinks on an early warning system. For example if a new version of a particular brank of UAS was released but had not been seen before, the early warning system would likely be able to classify the signal as a UAS if it had seen a previous version of the signal. It suggests that the CNN features are able to pick out patterns which are inherent between evolutions of the datalink. This could save resources for training data and time for retraining platforms. It could also have very positive implications for detecting new signals which have not been seen before as long if they were evolutionary signals. The machine learning validation in section 6.2.3.2 backed up these results showing that the Mavic 3 could be detected with over 96% accuracy and over 96% f1-score in every machine learning classifier and graphical signal representation combination.

With respect to the DX8 DSMZ transmitter and the eBee X, the machine learning validation results showed that the highest performing was machine learning classifier LR with PSD graphical signal representations. The more comprehensive validation showed it was possible to classify the DX8 DSMZ transmitter and eBee X with 100% accuracy and 100% f1-score even when the model was not trained on the signals. The LR classifier with PSD outperformed the other classifier and image combinations. The results overall indicate that although a supervised system should aim to be comprised of all datalink signal types, the classifier was able to distinguish between UAS detected and no UAS detected on signals which were not present in the original training set.

The system could also benefit from covering the whole 2.4GHz band. This would allow the CNN feature extractor and machine learning classifier access to the full range of features from the operating UAS. This could improve accuracy and provide a more comprehensive dataset. The dataset should consider types of signals, potentially by manufacturer first and then by signal type. For example DJI has three different datalinks, the Lightbridge, the OcuSync and Wi-Fi. If the supervised learning model contained at least one of each datalink type then this research indicates that it may be sufficient for training.

If a comprehensive dataset was to be made in the future all variants of the set up (for example each variant of the Mavic 3 compatible controllers) could be included within the training dataset so it is trained against all combinations that could occur. However, the difference in the controllers is with respect to the bit rate and EIRP variations. This research suggests that the bit rate does not affect the classification as the OcuSync 3.0 and 3.0+ both have different bit rates and EIRP but this should be tested across the different Mavic 3 controllers to confirm.

6.3.1.2 Re-trained Supervised Models

Overall PSD outperforms spectrograms for the 10 class UAS type supervised model cross validation results. On the holdout evaluation dataset, spectrogram with LR was the top performing combination of image and classifier. The new signals including the eBee X and the DX8 DSMX transmitter are classified without error using PSD graphical signal representations and kNN. The misclassifications with the kNN classifier occur between the OcuSync and Lightbridge variants. However, when considering the PSD signal representations the misclassification of those datalinks is low. The kNN and LR classifiers using PSD had no issue distinguishing between different Wi-Fi UAS.

As with the kNN classifier, the misclassifications that occur with LR and PSD occur between the DJI OcuSync and Lightbridge datalinks. This is the same for LR with spectrogram images. This suggests that the features the CNN is deriving or the pattern it is associating with those links are similar in some way. However, the misclassification is low. Overall the highest performing classifier and image combination was the LR machine learning classifier with spectrogram graphical signal representation. This combination could achieve over 98% accuracy for 10 class UAS type classification which includes the new signals the DX8 DSMX transmitter and the eBee X UAS.

Overall the section experiments have shown that a supervised model classifier trained for a set of UAS can detect unknown signals with high accuracy. This can be achieved on a low cost system consisting of a raspberry Pi and a BladeRF SDR. The last set of experiments in section 6.2.3.3 showed that a 10 class UAS type classifier could perform with 98% accuracy for confirming the type of UAS which has been detected. Most of the misclassifications occur between the OcuSync and Lightbridge variants of the datalinks but overall these misclassifications are small.

Chapter 7 - Conclusions

Many of today's systems and networks have RF access points. Cyber security does not always consider securing these links against cyber-attacks. The security of RF connectivity is paramount in protecting Critical National Infrastructure. The early warning of an anomalous signal in frequency bands of concern could indicate a cyber-attack or a malicious UAS entering a protected airspace. When addressing the £6.1 million funding that protecting wireless networks received in 2019 through the Secure Wireless Agile Networks (SWAN) partnership, a GCHQ spokesperson explained: "Though modern wireless standards incorporate encryption and authentication; the physical layers currently cannot be patched to the same degree as the higher layers in order to respond to new and emerging threats not foreseen at design time. It is important to develop new and novel protective measures, which should ideally include the ability to field-update every aspect of the physical layer radio operation to help mitigate such threats" [383].

As presented in section 1.2 the contributions of the research are are as follows:

(1) Classifying the flight mode of a UAS signal with high accuracy is an important step forward in the field and could provide vital information on the scene of a major incident for risk assessment. We improve accuracy by over 45% from previous research of flight mode classification confirmed using the same open source dataset. Further confirmation is proved with a larger dataset containing 11 more classes and tested in the presence of interference.

(2) Pre-trained CNNs for image classification can be employed for feature extraction using transfer learning on image based graphical representations of RF UAS and GPS jamming

signals producing high accuracy results and reducing the need for large datasets and computational resources.

(3) Supervised machine learning algorithms utilising transfer learning are capable of detecting UAS signals not captured in the original dataset, including evolutionary and non-evolutionary datalinks. This has significant implications for detecting unknown threats.

(4) Frequency domain graphical signal representations and deeper CNN architectures provide features which are robust to interference from other signals operating in the same frequency band such as Wi-Fi and Bluetooth.

(5) CNN feature extraction and transfer learning produces higher accuracy clustering for unsupervised learning compared to clustering raw data from the SDR and raw images but at a cost of time. Raw images are a good overall solution for timely clustering which could form part of an early warning system to confirm and/or cue other sensors.

(6) A low cost raspberry Pi and SDR based machine learning classification system can predict signals it was trained against and signals it was not trained against in a live environment.

This thesis will expand on the contributions by drawing out the conclusions from the research which have been split into the sections covered by the thesis outline.

7.1 Supervised Learning for RF Signal Classification

Classifying the flight mode of a UAS signal with high accuracy is an important step forward in the field and could provide vital information on the scene of a major incident for risk assessment. Up until this thesis the highest accuracy models for classifying the UAS flight mode was conducted by Al-Sa’d et al. [204] who achieved 46.8% accuracy for classifying 10 classes of flight mode. This thesis improves that to 91% accuracy using the same dataset of 10 flight mode classes. We then extend this work to include a larger dataset of more recent UAS platforms containing 21 classes and achieve over 95% accuracy with classifying the flight mode even in environments with Wi-Fi and Bluetooth interference in the band.

Transfer learning offers benefits for RF signal data as it reduces the need for large datasets while still providing high accuracy results. It offers two real advantages for classifying RF signal data in the form of image based graphical representations of the signal. Neural networks inherently require a lot of data to train from scratch, transfer learning allows us to employ smaller datasets and still achieve high accuracy classification results. This is highly significant for RF signal data as it is computationally intensive and it can be sensitive to collect legally. Secondly the training time for the entire model is reduced significantly compared with training a CNN from scratch therefore consuming less resources and providing a model in a much more efficient manner. It also allows for the re-training of models which are employed in different environments to be completed in a quicker time frame.

Providing the CNN feature extractor with different graphical signal representations can improve accuracy by allowing the CNN to benefit from the strengths of each individual representation. For GPS jammer classification it was shown that the novel concatenation of signal representations (PSD, spectrogram, raw constellation and histogram) was more effective that single representation images. It effectively allows the CNN to benefit from the strengths of each individual representation. The image concatenation dataset produced 98% (+/- 0.5%) classification accuracy with LR and SVM models and 96.3% (+/- 0.6%) with RF. The results, validated through 10-fold cross validation, showed that transfer learning using CNN VGG16 in conjunction with ML models LR, SVM, and RF and the concatenation of signal representations, produced high accuracy for the classification of GNSS jamming signals and outperformed previous work in the field.

Time domain features were shown to be less robust than frequency domain features when interference signals were introduced from Bluetooth and Wi-Fi. The DroneDetect dataset was used to test the effect of interference by deliberately capturing the UAS signals in the presence of Bluetooth and Wi-Fi interference. The introduction of interference resulted in a drop of less than 2% in accuracy for the classifier. Representing the signal in the frequency domain was shown to be more robust to noise. High accuracy can be maintained using LR as a classifier with CNN derived features. Bluetooth signals were shown to be more likely to interfere with detection and classification accuracy than Wi-Fi signals. GPS jamming signals were also evaluated with varying levels of SNR. Deep features were shown to be robust when SNR was degraded, with the concatenation of graphical signal representations producing 75% accuracy at -20dB SNR for 6 classes of GNSS jamming signal types.

The highest performing classifier and graphical representation combination is LR with PSD frequency domain representation especially in environments with interference. For the DroneRF experiments in section 4.4.1 there is little difference between machine learning classifiers LR and SVM for 10 class flight mode classification. However, LR does outperform the other classifiers when noise in introduced for the Drone Detect dataset in section 1.13.2, the gap gets greater as the interference is introduced and the number of classes is increased. The thesis concluded that the machine learning classifier to take forward for use with CNN feature extraction would be LR as it is the most robust to the introduction of interference with increased classes, especially when used in conjunction with frequency domain graphical signal representation PSD. This was shown to be true for both UAS signal classification and for the GPS jammer classification experiments whereby considering the PSD representation alone for very low levels of SNR (-10dB and lower) provided higher in accuracy that the concatenated signal representations.

Deeper CNN architectures improve accuracy and may be more resilient to noise when determining UAS features. Extending the evaluation of the DroneRF flight mode classification to consider a deeper CNN architecture of the ResNet-50, we found that accuracy could be further increased for 10 class flight mode classification from 87% with the VGG-16 feature extraction and LR classifier to 91% accuracy with the ResNet-50 feature extraction and LR classifier. An increase of 4% and both producing the highest results with the PSD graphical signal representations.

For the unsupervised learning experiments in section 5.5.2 the interference from Bluetooth and Wi-Fi signals introduced in the frequency band decreased v-measure scores significantly. Comparing the highest performing clean combination of PSD, VGG-16 and any initialisation saw a 0.7 drop in the same combination with interference. This shows that interference does affect the ability of k-means to perform clustering. However, the ResNet-50 feature extractor was able to achieve a v-measure score of 0.556 even in the presence of the interference with

PSD images and k++ initialisation. This suggests that the deeper CNN architecture is more resilient to noise when determining features for UAS signals through CNN feature extraction.

CNN feature extraction explainability is low. The explainability of the CNN feature extraction compared to traditional machine learning algorithms is low. Feature maps can be visually inspected but get harder to interpret the further into the process you go. However, the high accuracy results produced using the FE may offset the lack of explainability, especially if the system it is not being used for any ‘risk to life’ applications. In the application of UAS and GPS jammer detection and classification, the indication that something is present could trigger another sensor or even a human to visually verify the signal. Its use as an indicator is extremely valuable regardless of the lack of explainability, unless the system was producing a lot of false positives.

Misclassifications could be decreased by observing the whole of the RF spectrum. This could be achieved either by using a higher quality SDR which covers the 80MHz of the Wi-Fi band or using multiple SDRs so the 5GHz range can also be covered. The data could be combined at the raw level, as seen with the DroneRF dataset [337] or the images could be stitched together such as we have done with the concatenation of signals. As long as the CNN feature extractor was receiving all the information then it is believed that either approach would work well. For the DroneRF dataset the highest misclassifications occur with the Phantom 3. The Phantom 3 uses the 5GHz spectrum as well as the 2.4GHz frequency band so increasing the input to the CNN feature extractor to also cover the 5GHz range would provide the CNN will all the information to correctly classify the signal.

7.2 Unsupervised Learning for RF Signal Classification

CNN feature extraction and transfer learning improve accuracy as a precursor to unsupervised learning but at a cost of time. Raw PSD images with PCA initialisation of K-Means provides the best overall solution for timely clustering. In general PSD images – whether raw or used with VGG-16 or ResNet-50 features extraction, significantly outperform spectrogram images, indicating that frequency domain representation is better for clustering than time domain. Initialising the k-means clustering with PCA not only produces the highest v-measure score but it has the quickest inference time for raw images, VGG-16 and ResNet-50 feature extraction. Raw PSD images with PCA initialisation of K-Means provides the best overall solution for timely clustering. For example considering both inference time and v-measure score for the DroneRF dataset, PSD raw images without feature extraction produced a v-measure of 0.982 in 0.104 of a second, significantly quicker than using the ResNet-50 feature extractor which did have a marginally higher v-measure score.

Unsupervised learning could be used for a timely indication that a UAS is in the vicinity to confirm another sensor and/or cue another system, for example a supervised algorithm with much higher accuracy. The desired application of the system and interference environment must be considered when choosing the best combination of graphical signal representation, potential CNN feature extractor and K-means initialisation. For example if the system needs to display as high an accuracy as possible but time is not urgent then the correct choice might be a spectrogram with VGG-16 FE and PCA initialisation of K-Means clustering, which takes 6 seconds and produces a 0.814 v-measure score in a clean environment. In the presence of interference the PSD outperforms the spectrogram, still with the VGG-16 and k++ initialisation which takes 17 seconds and produces a 0.760 v-measure score. However, if the system needs to be highly time sensitive but can cope with a slightly lower accuracy then the best option would be image PSD with PCA which can complete in 0.280 seconds with a v-measure of 0.620 in a clean environment and 0.193 seconds with a v-measure of 0.60 in the presence of interference. In this instance the unsupervised learning could be used for a very timely indication that a UAS might be in the vicinity to confirm another sensor and/or cue another system, for example a supervised algorithm with much higher accuracy.

Representing a signal as an image and utilising CNN feature extraction is preferable to processing the raw data straight from the SDR for unsupervised clustering. Clustering graphical representations of the signal and utilising CNN feature extraction with transfer learning compared to raw RF data produces a higher V-measure score and if used in conjunction with PCA dimensionality reduction can compete in terms of clustering time with raw data. Using the raw data straight from the SDR without any pre-processing provided lower v-measure scores in section 5.5.3 for GPS jammer clustering and was more timely than using the graphical signal representations. The use of images allowed for a quicker clustering result to be calculated compared to raw data. When using CNN FE with transfer learning it was also observed that v-measure, ARI and AMI scores are generally in the region of 25% higher.

7.3 Low Cost Early Warning System

Low cost SDRs and hardware such as the Raspberry Pi are capable of running machine learning classifiers in real time for the purpose of early warning. The main advantage of using an SDR for this purpose is flexibility, the ability for system administrators to upgrade functionalities through a software download rather than a full hardware replacement. In terms of this application, it means that administrators could update and train models periodically without changing hardware. SDRs have the potential to alert a system administrator of an anomaly in the RF spectrum in a cost effective and timely manner and perform other functions such as triangulation of the signal which can track a UAS or mobile jammer using multiple SDR edge devices. SDRs provide a potential future solution which could lower the cost of RF signal classification significantly, especially when used in conjunction with hardware such as the Raspberry Pi.

A low cost Raspberry Pi 4 B connected to a BladeRF SDR and low cost antenna, is capable of running a CNN feature extractor and machine learning classifier as part of an early warning system for RF signal classification. However, the inference times ranged from 15-28 seconds for 2-class UAS detection and 18-28 seconds for classification. This suggests that for systems which require timely results the Raspberry Pi would be better suited to act as a repeater of the raw SDR data. This would enable the production of the graphical signal representations and the machine learning model prediction to be completed on a higher powered central control unit. If time was not of a concern then the Raspberry Pi is capable of making predictions as an edge device and could easily form part of a larger system made up of other sensors capable of faster indications to trigger higher accuracy results such as these.

7.4 Unknown Signal Classification

CNN feature extraction is able to pick out patterns which are inherent between evolutions of a datalink. The Mavic 3 was presented to the supervised machine learning model as an unknown signal that the model was not originally trained against and it was corrected classified a UAS for each model type. It can therefore be assumed that the CNN feature extraction for the Air 2 S which operates the OcuSync 3.0 datalink is close enough in terms of CNN extracted features to produce a positive result for the OcuSync 3.0+. The OcuSync 3.0+ is an evolution of the OcuSync 3.0. The OcuSync 3.0 and the OcuSync 3.0+ have differing EIRP and bit rate transmissions so CNN feature selection must not be based on transmission power or bit rate parameters but rather a pattern that the link is displaying is how it transmits. This is a positive result for detecting UAS evolutionary datalinks on an early warning system. For example if a new version of a particular brank of UAS was released but had not been seen before, the early warning system would likely be able to classify the signal as a UAS if it had seen a previous version of the signal. It suggests that the CNN features are able to pick out patterns which are inherent between evolutions of the datalink. This could save resources for training data and time for retraining platforms. It could also have very positive implications for detecting new signals which have not been seen before when new variants of a platform are first released or firmware/software is upgraded.

Unknown signal types which are not evolutionary can be detected using supervised models not originally trained against them. With respect to the DX8 DSMZ transmitter and the eBee X, the machine learning validation results showed that the highest performing combination was machine learning classifier LR with PSD graphical signal representations. This combination showed it was possible to classify the DX8 DSMZ transmitter and eBee X with 100% accuracy and 100% f1-score even when the model had not been trained on the original signals (or on any similar signals by the same manufacturer). The LR classifier with PSD outperformed the other classifier and image combinations. The results overall indicate that supervised system do not need to contain all platform types as the classifier was able to distinguish between UAS detected and no UAS detected on signals which were not present in the original training set. This could have large implications for reducing databases for supervised models and therefore subsequent training time and resource.

7.5 Future Work

Increased Spectrum Coverage. It would be beneficial to include the whole of the 80MHz Wi-Fi band for the feature extraction. This could be done in a couple of different ways. Increasing the quality of the SDR to cover the whole band or using multiple SDRs and combining the graphical signal representations as was done for the concatenation of signal types as described in section 0. Incorporating the 5GHz spectrum could also help the classifier increase accuracy in distinguishing between flight modes and UAS types.

Congested City Environments. It would be beneficial to evaluate the performance of these algorithms in a congested spectrum. For example, testing in a built up city environment where there will be a lot of activity in the Wi-Fi spectrum. The OcuSync 3.0 for example will hop to different frequency bands when it experiences interference. It is possible that these occurrences would need to be accounted for within the dataset but it really depends on the pattern that the feature extractor is using. For example it might not matter which particular band the signal is operating in if the features the CNN is using is not dependent on the part of the spectrum the UAS is operating in. This information will not be known until thorough testing is done within a constantly changing congested environment. While the experiments here introduce interference, it is introduced in a controlled manner and does not replicate a city environment where the activity in the band will be constantly changing. It may be found that certain locations such as airports have a relatively stable RF background but again this is not known until it is tested and the concept may work well with a changeable environment.

Simulated Signals versus Real Signals. The GPS jamming work has been tested with signals generated either using python or MATLAB. These simulated signals will be representative of real GPS jamming signals however the representation of real world noise is done using simulated AWGN. Testing with real world signals and therefore incorporating real world noise would be very difficult in the UK due to legal constraints. However, this work is essential to know whether the model would work in a real world environment.

Bibliography

[1] Merriam-Webster, “Drone Definition & Meaning - Merriam-Webster,” *Merriam-Webster Dictionary*. https://www.merriam-webster.com/dictionary/drone (accessed Nov. 16, 2022).

[2] “Radio frequency Definition & Meaning - Merriam-Webster.” https://www.merriam-webster.com/dictionary/radio frequency (accessed Nov. 16, 2022).

[3] *IEEE Std 1900.1-2019 (Revision of IEEE Std 1900.1-2008) : IEEE Standard for Definitions and Concepts for Dynamic Spectrum Access: Terminology Relating to Emerging Wireless Networks, System Functionality, and Spectrum Management.* IEEE, 2019.

[4] EU Agency for the Space Programme, “What is GNSS?,” *EU Agency Sp. Program.*, Dec. 2021, Accessed: Dec. 30, 2021. [Online]. Available: https://www.euspa.europa.eu/european-space/eu-space-programme/what-gnss.

[5] GPS.GOV, “GPS.gov: GPS Overview,” *GPS.GOV*. https://www.gps.gov/systems/gps/ (accessed Nov. 16, 2022).

[6] IBM, “What is Machine Learning?,” *IBM*. https://www.ibm.com/uk-en/cloud/learn/machine-learning (accessed Nov. 16, 2022).

[7] IBM, “What is Deep Learning?,” *IBM*. https://www.ibm.com/cloud/learn/deep-learning (accessed Nov. 16, 2022).

[8] A. Patil and M. Rane, “Convolutional Neural Networks: An Overview and Its Applications in Pattern Recognition,” *Smart Innov. Syst. Technol.*, vol. 195, pp. 21–30, 2021, doi: 10.1007/978-981-15-7078-0_3.

[9] S. Shackle, “The mystery of the Gatwick drone,” *The Gaurdian*, Dec. 2020, Accessed:

May 21, 2021. [Online]. Available: https://www.theguardian.com/uk-news/2020/dec/01/the-mystery-of-the-gatwick-drone.

[10] E. Gilmer, "D.C. Airport Incident Exposes Gaps in Counter-Drone Authorities," *Bloom. Gov.*, Jul. 2022, Accessed: Jul. 27, 2022. [Online]. Available: https://about.bgov.com/news/d-c-airport-incident-exposes-gaps-in-counter-drone-authorities/.

[11] Dedrone, "Airport Airspace Activity Study 2018," 2018.

[12] Civil Aviation Authority; and M. A. Military Aviation Authority;, "Analysis of Airprox in UK Airspace," *UK AirProx Board*, 2019.

[13] London Economics, "Economic impact to the UK of a disruption to GNSS Showcase Report," Apr. 2017.

[14] E. Colard, "Distributing high-precision time over optical networks in the 5G world," *GPS World*, Apr. 2020, Accessed: Apr. 14, 2021. [Online]. Available: https://www.gpsworld.com/distributing-high-precision-time-over-optical-networks-in-the-5g-world/.

[15] Eurocontrol, "Does Radio Frequency Interference to Satellite Navigation pose an increasing threat to Network efficiency, cost-effectiveness and ultimately safety?," *Eurocontrol*, Mar. 2019, Accessed: Dec. 30, 2021. [Online]. Available: https://www.eurocontrol.int/sites/default/files/2021-03/eurocontrol-think-paper-9-radio-frequency-intereference-satellite-navigation.pdf.

[16] E. Lo and J. Kohl, "Internet of things (IoT) discovery using deep neural networks," *Proc. - 2020 IEEE Winter Conf. Appl. Comput. Vision, WACV 2020*, pp. 795–803, 2020, doi: 10.1109/WACV45572.2020.9093371.

[17] ALADDIN, “ALADDIN – Advanced hoListic Adverse Drone Detection, Identification and Neutralization,” *ALADDIN Proj.*, 2020, Accessed: Jun. 22, 2021. [Online]. Available: https://aladdin2020.eu/.

[18] G. De Cubber *et al.*, “The SafeShore system for the detection of threat agents in a maritime border environment,” *IARP Work. Risky Interv. Environ. Surveill.*, no. May, pp. 6–9, 2017, doi: 10.5281/zenodo.1115552.

[19] C. J. Swinney and J. C. Woods, “K-Means Clustering Approach to UAS Classification via Graphical Signal Representation of Radio Frequency Signals for Air Traffic Early Warning,” *IEEE Trans. Intell. Transp. Syst.*, pp. 1–9, 2022, doi: 10.1109/TITS.2022.3202011.

[20] C. J. Swinney and J. C. Woods, “Low-Cost Raspberry-Pi-Based UAS Detection and Classification System Using Machine Learning,” *Aerosp. 2022, Vol. 9, Page 738*, vol. 9, no. 12, p. 738, Nov. 2022, doi: 10.3390/AEROSPACE9120738.

[21] C. J. Swinney and J. C. Woods, “A Review of Security Incidents and Defence Techniques Relating to the Malicious Use of Small Unmanned Aerial Systems,” *IEEE Aerosp. Electron. Syst. Mag.*, vol. 37, no. 5, pp. 14–28, 2022, doi: 10.1109/MAES.2022.3151308.

[22] C. J. Swinney, “Faithful Wingmen or Killer Robots: AI Applications for Air Power,” *Air Sp. Power 2022 Maint. Our Lead. Edge*, pp. 98–99, Jul. 2022, Accessed: Jul. 29, 2022. [Online]. Available: https://issuu.com/globalmediapartners/docs/raf_air_and_space_power_2022.

[23] C. J. Swinney and J. C. Woods, “GPS Jamming Signal Classification with CNN Feature Extraction in low Signal-to-Noise Environments,” *Int. J. Cyber Situational Aware.*, vol. 6, no. 1, pp. 1–21, Feb. 2022, doi: 10.22619/IJCSA.2021.100135.

[24] C. J. Swinney and J. C. Woods, “The Effect of Real-World Interference on CNN Feature Extraction and Machine Learning Classification of Unmanned Aerial Systems,” *Aerospace*, vol. 8(7), no. 179, pp. 1–18, 2021, doi: https://doi.org/10.3390/aerospace8070179.

[25] C. J. Swinney and J. C. Woods, “Unmanned Aerial Vehicle Operating Mode Classification Using Deep Residual Learning Feature Extraction,” *Aerospace*, vol. 8, no. 3, p. 79, Mar. 2021, doi: 10.3390/aerospace8030079.

[26] C. J. Swinney, “RF Detection and Classification of Unmanned Aerial Vehicles in Environments with Wireless Interference,” *2021 Int. Conf. Unmanned Aircr. Syst. Athens, Greece. June 15-18, 2021*, pp. 1494–1498, 2021, doi: 10.1109/ICUAS51884.2021.9476867.

[27] C. J. Swinney and J. C. Woods, “GNSS Jamming Classification via CNN , Transfer Learning and the Novel Concatenation of Signal Representations,” *2021 Int. Conf. Cyber Situational Awareness, Data Anal. Assess.*, pp. 1–9, Jun. 2019, doi: 10.1109/CYBERSA52016.2021.9478250.

[28] C. J. Swinney and J. C. Woods, “Unmanned Aerial Vehicle Flight Mode Classification using Convolutional Neural Network and Transfer Learning,” *2020 16th Int. Comput. Eng. Conf.*, pp. 83–87, 2020, doi: 10.1109/ICENCO49778.2020.9357368.

[29] Manchester University, “Digital Trust & Security Seminar Series: Carolyn Swinney,” *Digital Futures: Eventbrite*, May 04, 2022. https://www.eventbrite.co.uk/e/digital-trust-security-seminar-series-carolyn-swinney-tickets-251423353017 (accessed Aug. 10, 2022).

[30] C. J. Swinney and J. C. Woods, “DroneDetect Dataset: A Radio Frequency dataset of Unmanned Aerial System (UAS) Signals for Machine Learning Detection and

Classification," *IEEE DataPort*, Jun. 2021, doi: 10.21227/5jjj-1m32.

[31] C. J. Swinney and J. C. Woods, "Raw IQ dataset for GNSS GPS jamming signal classification," *Zenodo*, Mar. 2021, doi: 10.5281/ZENODO.4629685.

[32] P. Mackenzie and F. Kanellos, "Cyberspace Operations," *Joint Air Power Competence Centre*, Jan. 2021. https://www.japcc.org/chapters/c-uas-cyberspace-operations/ (accessed Aug. 10, 2022).

[33] Research and Markets, "Global Commercial Drone Market Report 2021-2028: Increasing," *Intrado Glode Newsire*, May 2021, Accessed: Jun. 25, 2021. [Online]. Available: https://www.globenewswire.com/en/news-release/2021/05/26/2236133/28124/en/Global-Commercial-Drone-Market-Report-2021-2028-Increasing-Economic-Opportunities-Provided-by-Drones-Favourable-Regulatory-Landscape.html.

[34] K. Townsend, "What Security Threats Are Posed By Drones?," *Avast*, Sep. 2019, Accessed: Jun. 25, 2021. [Online]. Available: https://blog.avast.com/what-security-threats-are-posed-by-drones.

[35] Department for Transport, "Small Remotely Piloted Aircraft Systems (drones) Mid-Air Collision Study," *Dep. Transp.*, no. July, p. 18, 2017, [Online]. Available: https://www.gov.uk/government/publications/drones-and-manned-aircraft-collisions-test-results.

[36] M. Gajewski, "Drone strikes commercial aircraft in Quebec: Garneau," *CTV News*, Oct. 15, 2017. https://www.ctvnews.ca/canada/drone-strikes-commercial-aircraft-in-quebec-garneau-1.3633035 (accessed Oct. 20, 2021).

[37] P. Gregg, "Risk in the Sky?," *University of Dayton Research Institute*, Sep. 13, 2018.

https://udayton.edu/udri/news/18-09-13-risk-in-the-sky.php (accessed Jun. 25, 2021).

[38] BBC News, “Venezuela President Maduro survives ‘drone assassination attempt,’” *BBC News*, Aug. 05, 2018. https://www.bbc.co.uk/news/world-latin-america-45073385 (accessed Jun. 25, 2021).

[39] BBC News, “Drone crash causes Hollywood electricity blackout,” *BBC News*, Oct. 28, 2015. https://www.bbc.co.uk/news/technology-34656820 (accessed Oct. 20, 2021).

[40] S. French, “This drone crashing into a bike race is every cyclist’s nightmare,” *Market Watch*, May 09, 2017. https://www.marketwatch.com/story/this-drone-crashing-into-a-bike-race-is-every-cyclists-nightmare-2017-05-09 (accessed Oct. 20, 2021).

[41] J. Lee, “Drone crashes into Space Needle during New Year’s Eve fireworks setup,” *The Seattle Times*, Jan. 11, 2017. https://www.seattletimes.com/photo-video/video/watch-drone-crashes-into-space-needle-during-new-years-eve-fireworks-setup/ (accessed Oct. 20, 2021).

[42] M. Gajewski, “Drone strikes commercial aircraft in Quebec: Garneau,” *CTV News*, Oct. 15, 2017. https://www.ctvnews.ca/canada/drone-strikes-commercial-aircraft-in-quebec-garneau-1.3633035 (accessed Jun. 25, 2021).

[43] BBC News, “Photos show ‘weaponised commercial drones’ in Iraq ,” *BBC News*, Jan. 18, 2017. https://www.bbc.co.uk/news/technology-38663394 (accessed Oct. 19, 2021).

[44] B. Heubl, “Conflict groups arm consumer drones for terror attacks,” *E&T Mag.*, Apr. 2021, Accessed: Oct. 19, 2021. [Online]. Available: https://eandt.theiet.org/content/articles/2021/04/conflict-groups-arm-consumer-drones-to-deliver-death-and-terror/.

[45] K. Hartmann and K. Giles, “UAV exploitation: A new domain for cyber power,” *Int.*

Conf. Cyber Conflict, CYCON, vol. 2016-Augus, pp. 205–221, 2016, doi: 10.1109/CYCON.2016.7529436.

[46] B. Riley-Smith, “Isil plotting to use drones for nuclear attack on West,” *Telegraph*, Apr. 01, 2016. https://www.telegraph.co.uk/news/2016/04/01/isil-plotting-to-use-drones-for-nuclear-attack-on-west/ (accessed Aug. 16, 2021).

[47] J. DeFranco, “Dark Side of Delivery: The Growing Threat of Bioweapon Dissemination by Drones,” *Def. IQ*, Jan. 2020, doi: 10.1142/P1081.

[48] R. Majeed, N. A. Abdullah, M. F. Mushtaq, and R. Kazmi, “Drone Security: Issues and Challenges,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 5, pp. 720–729, 2021, doi: 10.14569/IJACSA.2021.0120584.

[49] L. Zhen, “Chinese criminal gangs spreading African swine fever to force farmers to sell pigs cheaply so they can profit,” *South China Morning Post*, Dec. 14, 2019. https://www.scmp.com/news/china/politics/article/3042122/chinese-criminal-gangs-spreading-african-swine-fever-force (accessed Oct. 19, 2021).

[50] J. Taylor, “Public Statement from the Plenary Meeting of the Missile Technology Control Regime, Auckland,” *MTCR*, Oct. 19, 2019. https://mtcr.info/public-statement-from-the-plenary-meeting-of-the-missile-technology-control-regime-auckland-11-october-2019/ (accessed Oct. 19, 2021).

[51] Daily Mail, “China unveils its terrifying ‘suicide drones’ that can be launched in swarms ,” *Daily Mail*, Oct. 15, 2020. https://www.dailymail.co.uk/news/article-8843745/China-unveils-terrifying-weaponised-drone-troop-launched-truck.html (accessed Oct. 19, 2021).

[52] D. Hambling, “Israel used world’s first AI-guided combat drone swarm in Gaza

attacks," *New Scientist*, 2021. https://www.newscientist.com/article/2282656-israel-used-worlds-first-ai-guided-combat-drone-swarm-in-gaza-attacks/ (accessed Oct. 19, 2021).

[53] G. Lykou, D. Moustakas, and D. Gritzalis, "Defending airports from uas: A survey on cyber- attacks and counter-drone sensing technologies," *Sensors (Switzerland)*, vol. 20, no. 12, pp. 1–35, 2020, doi: 10.3390/s20123537.

[54] R. Weinmann and B. Schmotzle, "TBONE – A zero-click exploit for Tesla MCUs," 2020.

[55] M. McNabb, "Drones and Cybersecurity: An Expert Opinion on Protecting Industry Against Drone and Data Attacks," *DRONELIFE*, Oct. 17, 2019. https://dronelife.com/2019/10/17/drones-and-cybersecurity-an-expert-opinion-on-protecting-industry-against-drone-and-data-attacks/ (accessed Oct. 19, 2021).

[56] B. Edwards, "Cybersecurity And Drones: A Threat From Above," *Forbes*, Feb. 25, 2021. https://www.forbes.com/sites/forbestechcouncil/2021/02/25/cybersecurity-and-dronesa-threat-from-above/?sh=76d173477b0d (accessed Oct. 19, 2021).

[57] E. Fink, "This drone can steal what's on your phone," *CNN Business*, Mar. 20, 2014. https://money.cnn.com/2014/03/20/technology/security/drone-phone/ (accessed Oct. 19, 2021).

[58] N. Barker, "Development of a Drone-Mounted Wireless Attack Platform," *Theses Diss.*, 2020, [Online]. Available: https://scholar.afit.edu/etd/3224.

[59] F. Le Roy *et al.*, "Risk assessment of SDR-based attacks with UAVs To cite this version : HAL Id : hal-02283926 Risk assessment of SDR-based attacks with UAVs," 2019.

[60] A. Staniforth, “Countering Drones - An Evolving Cybersecurity Requirement,” *Defence IQ*, Feb. 13, 2020. https://www.defenceiq.com/cyber-defence-and-security/articles/countering-drones-an-evolving-cybersecurity-requirement (accessed Oct. 19, 2021).

[61] S. Taylor, “The Scariest Cybersecurity Statistics ,” *Restore Privacy*, Jul. 15, 2019. https://restoreprivacy.com/2019-cyber-security-statistics/ (accessed Oct. 19, 2021).

[62] G. Markarian and A. Staniforth, “Countermeasures for Aerial Drones,” *Artech House*, 2021. https://books.google.co.uk/books?id=In0qEAAAQBAJ&pg=PA36&lpg=PA36&dq=pineapple+wi-fi+mounted+on+drone&source=bl&ots=G7lTH1SIyr&sig=ACfU3U0vayj3uddxXIO4ay7Ay0SX9FwfCg&hl=en&sa=X&ved=2ahUKEwiboZ35k9bzAhUIilwKHZojAzoQ6AF6BAgNEAM#v=onepage&q=pineapple wi-fi m (accessed Oct. 19, 2021).

[63] J. Venable and L. Ries, “DJI Placed on the Entity List for Human Rights Abuses, but Concerns About Data Security Should Not Be Overlooked | The Heritage Foundation,” *The Heritage Foundation*, Jan. 07, 2021. https://www.heritage.org/cybersecurity/commentary/dji-placed-the-entity-list-human-rights-abuses-concerns-about-data (accessed Oct. 19, 2021).

[64] S. Walters, “How Can Drones Be Hacked? The updated list of vulnerable drones & attack tools ,” *Medium*, Oct. 29, 2016. https://medium.com/@swalters/how-can-drones-be-hacked-the-updated-list-of-vulnerable-drones-attack-tools-dd2e006d6809 (accessed Oct. 20, 2021).

[65] C. G. Leela Krishna and R. Murphy, “A review on cybersecurity vulnerabilities for unmanned aerial vehicles,” *Auvsi Xponential 2018*, pp. 0–5, 2018.

[66] BBC News, “Drone stalker jailed for spying on ex-girlfriend ,” *BBC News*, Nov. 20, 2020. https://www.bbc.co.uk/news/uk-wales-55018682 (accessed Oct. 20, 2021).

[67] A. Mitchelson, “‘Peeping Tom drones’ prompt calls for a close look at privacy laws,” *Writing Portfolio Alana Mitchelson*, Apr. 27, 2017. https://alanamitchelson.wordpress.com/2017/04/27/peeping-tom-drones-prompt-calls-for-a-close-look-at-privacy-laws/ (accessed Oct. 20, 2021).

[68] R. Patterson, “Sydney couple catch drone spying on them from fifth floor balcony,” *Daily Telegraph*, May 31, 2017. https://www.dailytelegraph.com.au/newslocal/manly-daily/sydney-couple-catch-drone-spying-on-them-from-fifth-floor-balcony/news-story/5f67377e6bb14f0d4e566340ae462ba7 (accessed Oct. 20, 2021).

[69] B. Crumley, “Foul! National soccer teams accuse rivals of spying with drones,” *DroneDJ*, Jun. 03, 2021. https://dronedj.com/2021/06/03/foul-national-soccer-teams-accuse-rivals-of-spying-with-drones/ (accessed Oct. 20, 2021).

[70] J. Crump, “‘Unidentified’ drones swarming US warships raise alarm,” *The Independant*, Apr. 06, 2021. https://www.independent.co.uk/news/world/americas/drones-us-navy-warships-unidentified-b1827342.html (accessed Jun. 25, 2021).

[71] M. F. Al-Sa’d, A. Al-Ali, A. Mohamed, T. Khattab, and A. Erbad, “RF-based drone detection and identification using deep learning approaches: An initiative towards a large open source drone database,” *Futur. Gener. Comput. Syst.*, vol. 100, pp. 86–97, Nov. 2019, doi: 10.1016/j.future.2019.05.007.

[72] M. M. Alaboudi, M. Abu Talib, and Q. Nasir, “Radio frequency-based techniques of drone detection and classification using machine learning,” *ACM Int. Conf. Proceeding Ser.*, pp. 278–282, 2020, doi: 10.1145/3449301.3449348.

[73] S. Calder, “Gatwick drone disruption cost over £50m ,” *The Independent*, Jan. 22, 2019. https://www.independent.co.uk/travel/news-and-advice/gatwick-drone-airport-cost-easyjet-runway-security-passenger-cancellation-a8739841.html (accessed Jun. 25, 2021).

[74] D. Lee, “Drone sighting disrupts major US airport,” *BBC News*, Jan. 2019, Accessed: May 21, 2021. [Online]. Available: https://www.bbc.co.uk/news/technology-46968419.

[75] Japan Today, “Drone disrupts operations at Kansai airport,” *Japan Today*, Nov. 2019, Accessed: May 21, 2021. [Online]. Available: https://japantoday.com/category/national/Drone-disrupts-operations-at-Kansai-airport.

[76] BBC News, “Heathrow airport drone investigated by police and military,” *BBC News*, Jan. 2019, Accessed: May 21, 2021. [Online]. Available: https://www.bbc.co.uk/news/uk-46804425.

[77] DW News, “Frankfurt Airport halts flights after drone sighted ,” *DW News*, Mar. 2019, Accessed: May 21, 2021. [Online]. Available: https://www.dw.com/en/frankfurt-airport-halts-flights-after-drone-sighted/a-48030789.

[78] E. Mee, “Flights resume at Dublin Airport after drone sighting ,” *Sky News*, Feb. 2019, Accessed: May 21, 2021. [Online]. Available: https://news.sky.com/story/dublin-airport-suspends-flights-after-drone-sighting-11643644.

[79] International Airport Review, “Drone sighting at Dubai International Airport temporarily suspends flights,” *Int. Airpt. Rev.*, Feb. 2019, Accessed: May 21, 2021. [Online]. Available: https://www.internationalairportreview.com/news/81308/drone-dubai-suspend-flights/.

[80] BBC News, “Changi Airport: Drones disrupt flights in Singapore,” *BBC News*, Jun. 2019, Accessed: May 21, 2021. [Online]. Available: https://www.bbc.co.uk/news/business-48754432.

[81] S. Corr, “Apache attack helicopters assist Essex Police in hunt for Stansted Airport drone,” *Bishops Stortford Indep.*, Nov. 2020, Accessed: May 21, 2021. [Online]. Available: https://www.bishopsstortfordindependent.co.uk/news/army-helicopters-help-police-hunt-drone-above-stansted-airport-9142882/.

[82] ITV News, “Flights grounded for two hours at Frankfurt airport after drone sighting,” *ITV News*, Mar. 2020, Accessed: May 21, 2021. [Online]. Available: https://www.itv.com/news/2020-03-02/flights-grounded-for-two-hours-at-frankfurt-airport-after-drone-sighting.

[83] T. Snuggs, “Madrid airport forced to close for two hours after drone sightings ,” *Sky News*, Feb. 2020, Accessed: May 21, 2021. [Online]. Available: https://news.sky.com/story/madrid-airport-forced-to-close-for-two-hours-after-drone-sightings-11925578.

[84] US News, “Flights Halted at North Carolina Airport After Drone Sighted | North Carolina News,” *US News*, Mar. 2021, Accessed: May 21, 2021. [Online]. Available: https://www.usnews.com/news/best-states/north-carolina/articles/2021-03-10/flights-halted-at-north-carolina-airport-after-drone-sighted.

[85] NZ Herald, “Drone spotted 30 metres from plane at Auckland Airport,” *NZ Her.*, Apr. 2021, Accessed: May 21, 2021. [Online]. Available: https://www.nzherald.co.nz/nz/drone-spotted-30-metres-from-plane-at-auckland-airport/JLFJA4D6OLRHIAHHRZO3W3LVXY/.

[86] B. Forrest and B. McGill, “Drones Flying Near Airports, Infrastructure Prompt U.S.

Action," *Wall Street Journal*, May 20, 2021. https://www.wsj.com/articles/drones-flying-near-airports-infrastructure-prompt-u-s-action-11621533604 (accessed Jun. 25, 2021).

[87] C. Phillips and C. Gaffey, "Most French Nuclear Plants 'Should Be Shut Down' Over Drone Threat," *News Weej*, Feb. 24, 2015. https://www.newsweek.com/2015/03/06/most-french-nuclear-plants-should-be-shut-down-over-drone-threat-309019.html (accessed Jun. 25, 2021).

[88] C. Arthur, "UK's first drone conviction will bankrupt me, says Cumbrian man," *The Guardian*, Apr. 02, 2014. https://www.theguardian.com/world/2014/apr/02/uk-first-drone-conviction (accessed Jun. 25, 2021).

[89] Reuters, "Greenpeace crashes Superman-shaped drone into French nuclear plant," *Reuters*, Jul. 03, 2018. https://www.reuters.com/article/uk-france-nuclear-greenpeace-idUKKBN1JT17G (accessed Jun. 25, 2021).

[90] D. Hambling, "Dozens More Mystery Drone Incursions Over U.S. Nuclear Power Plants Revealed," *Forbes*, Sep. 07, 2020. https://www.forbes.com/sites/davidhambling/2020/09/07/dozens-more-drone-incursions-over-us-nuclear-power-plants-revealed/?sh=387b92996296 (accessed Jun. 25, 2021).

[91] A. Tait, "Attack of the drones: the mystery of disappearing swarms in the US midwest," *The Guardian*, Apr. 18, 2021. https://www.theguardian.com/world/2021/apr/18/attack-of-the-drones-the-mystery-of-disappearing-swarms-in-the-us-midwest (accessed Jun. 25, 2021).

[92] D. Hambling, "'Drone Swarm' Invaded Palo Verde Nuclear Power Plant Last September," *Forbes*, Jul. 30, 2020.

https://www.forbes.com/sites/davidhambling/2020/07/30/drone-swarm-invaded-palo-verde-nuclear-power-plant/?sh=2e66545d43de (accessed Jun. 25, 2021).

[93] D. Rose, “Drone attack on suspected Iranian nuclear production plant,” *The Times*, Jun. 25, 2021. https://www.thetimes.co.uk/article/drone-attack-on-suspected-iranian-nuclear-production-plant-3vvg52hbf (accessed Jun. 25, 2021).

[94] R. T. Ioannides, T. Pany, and G. Gibbons, “Known Vulnerabilities of Global Navigation Satellite Systems, Status, and Potential Mitigation Techniques,” *Proc. IEEE*, vol. 104, no. 6, pp. 1174–1194, Jun. 2016, doi: 10.1109/JPROC.2016.2535898.

[95] D. Last, “GNSS: The Present Imperfect,” *Insid. GNSS*, 2010, [Online]. Available: http://admin.insidegnss.com/auto/may10-Last.pdf.

[96] Centre for the Protection of National Infrastructure, “Critical National Infrastructure,” *CPNI*, Mar. 2021, Accessed: Apr. 14, 2021. [Online]. Available: https://www.cpni.gov.uk/critical-national-infrastructure-0.

[97] G. Pescaroli *et al.*, “Cascading Effects of Global Positioning and Navigation Satellite Service Failures,” *Inst. RISK DISASTER Reduct. MULLARD Sp. Sci. Lab. CASCADING Eff. Glob. Position. Navig. Satell. Serv. Fail.*, 2019, doi: 10.14324/000.rp.10076568.

[98] Ofcom, “Radio frequency jammers,” 2021, Accessed: Apr. 14, 2021. [Online]. Available: https://www.ofcom.org.uk/spectrum/interference-enforcement/spectrum-offences/jammers.

[99] Amazon, “Generic GPS signal interference blocker,” *Amazon.co.uk*, Accessed: Sep. 16, 2020. [Online]. Available: https://www.amazon.co.uk/Grneric-interference-anti-tracking-tracking-anti-positioning/.

[100] T. Morong, P. Puričer, and P. Kovář, “Study of the GNSS jamming in real environment,” *Int. J. Electron. Telecommun.*, vol. 65, no. 1, pp. 65–70, 2019, doi: 10.24425/ijet.2019.126284.

[101] B. Heubl, “How illegal drone jammers are sold to Europe,” *IET E&T*, Mar. 2021, Accessed: Apr. 14, 2021. [Online]. Available: https://eandt.theiet.org/content/articles/2021/03/how-drone-jammers-are-sold-to-europe-and-the-uk/.

[102] T. Merz, “GPS jammer costs driver $32,000 after interfering with plane signals,” *Telegr.*, Aug. 2013, Accessed: Sep. 17, 2020. [Online]. Available: https://www.telegraph.co.uk/technology/news/10228874/GPS-jammer-costs-driver-32000-after-interfering-with-plane-signals.html.

[103] The Economist, “Satellite-navigation systems such as GPS are at risk of jamming,” *The Economist*, Jun. 08, 2021. https://www.economist.com/science-and-technology/2021/05/06/satellite-navigation-systems-such-as-gps-are-at-risk-of-jamming (accessed Dec. 30, 2021).

[104] A. Douglas, “GNSS jamming and how to mitigate it,” *IEEE 1588*, Apr. 2020, Accessed: Sep. 16, 2020. [Online]. Available: https://blog.meinbergglobal.com/2020/04/12/gnss-jamming-and-how-to-mitigate-it/.

[105] P. L. Lineswala and S. N. Shah, “Designing of SDR based malicious act: IRNSS jammer,” *Lect. Notes Inst. Comput. Sci. Soc. Telecommun. Eng. LNICST*, vol. 220 LNICST, pp. 237–246, 2018, doi: 10.1007/978-3-319-73712-6_25.

[106] R. Ferreira, J. Gaspar, P. Sebastião, and N. Souto, “Effective GPS Jamming Techniques for UAVs Using Low-Cost SDR Platforms,” *Wirel. Pers. Commun.*, vol. 115, no. 4, pp. 2705–2727, 2020, doi: 10.1007/s11277-020-07212-6.

[107] O. Glomsvoll and L. K. Bonenberg, “GNSS Jamming Resilience for Close to Shore Navigation in the Northern Sea,” *J. Navig.*, vol. 70, no. 1, pp. 33–48, 2017, doi: 10.1017/S0373463316000473.

[108] M. Pham and K. Xiong, “A survey on security attacks and defense techniques for Connected and Autonomous Vehicles,” *arXiv*, pp. 1–24, 2020.

[109] T. Kraus, R. Bauemfeind, and B. Eissfeller, “Survey of In-Car Jammers - Analysis and modeling of the RF signals and IF samples (suitable for active signal cancellation),” *24th Int. Tech. Meet. Satell. Div. Inst. Navig. 2011, ION GNSS 2011*, vol. 1, no. September, pp. 430–435, 2011.

[110] R. M. Ferre, A. D. La Fuente, and E. S. Lohan, “Jammer classification in GNSS bands via machine learning algorithms,” *Sensors (Switzerland)*, vol. 19, no. 22, Nov. 2019, doi: 10.3390/s19224841.

[111] A. C. Tribble, “The software defined radio: Fact and fiction,” *2008 IEEE Radio Wirel. Symp. RWS*, pp. 5–8, 2008, doi: 10.1109/RWS.2008.4463414.

[112] J. Mitola, “Software radios-survey, critical evaluation and future directions,” *[Proceedings] NTC-92 Natl. Telesystems Conf.*, pp. 13/15-13/23, doi: 10.1109/NTC.1992.267870.

[113] J. Raúl Machado-Fernández, “Software Defined Radio: Basic Principles and Applications Software Defined Radio: Principios y aplicaciones básicas Software Defined Radio: Princípios e Aplicações básicas,” *Rev. Fac. Ing. (Fac. Ing.), Enero-Abril*, vol. 24, no. 38, pp. 79–96, 2015.

[114] E. Blossom, “GNU Radio: Tools for Exploring the Radio Frequency Spectrum,” *Linux J.*, 2004, Accessed: Nov. 06, 2019. [Online]. Available:

https://www.linuxjournal.com/article/7319.

[115] D. Sinha, A. K. Verma, and S. Kumar, “Digital Front End in Software Defined Radio,” no. August 2019, pp. 0–6, 2016.

[116] T. A. Sturman, “An Evaluation of Software Defined Radio-An Overview,” 2006.

[117] A. Haghighat, “A review on essentials and technical challenges of software defined radio,” in *Proceedings - IEEE Military Communications Conference MILCOM*, 2002, vol. 1, pp. 377–382, doi: 10.1109/milcom.2002.1180471.

[118] E. T. A. Sturman, J. Fitzpatrick, M. Rupp, and I. Cox, “An Evaluation of Software Defined Radio – Main Document Administration page Date of issue,” no. 410000262, 2006, doi: 10.13140/RG.2.1.4454.2800.

[119] R. Krishnan, R. G. Babu, S. Kaviya, N. P. Kumar, C. Rahul, and S. S. Raman, “Software defined radio (SDR) foundations, technology tradeoffs: A survey,” in *IEEE International Conference on Power, Control, Signals and Instrumentation Engineering, ICPCSI 2017*, Jun. 2018, pp. 2677–2682, doi: 10.1109/ICPCSI.2017.8392204.

[120] R. Akeela and B. Dezfouli, “Software-defined Radios: Architecture, state-of-the-art, and challenges,” *Comput. Commun.*, vol. 128, no. March, pp. 106–125, Sep. 2018, doi: 10.1016/j.comcom.2018.07.012.

[121] G. Ulbricht, “Analog-to-digital Conversion – the Bottleneck for Software Defined Radio Frontends ANALOG-TO-DIGITAL CONVERSION – THE BOTTLENECK FOR SOFTWARE,” no. June 2012, 2015.

[122] J. Wang *et al.*, “A software defined radio evaluation platform for WBAN systems,” *Sensors (Switzerland)*, vol. 18, no. 12, pp. 1–15, 2018, doi: 10.3390/s18124494.

[123] N. Marappa, “ScholarWorks at WMU Design of Digital Down Converter Chain for Software Defined Radio Systems on FPGA.” Accessed: Nov. 11, 2019. [Online]. Available: https://scholarworks.wmich.edu/masters_theses/664.

[124] T. Kazaz, C. Van Praet, M. Kulin, P. Willemen, and I. Moerman, “Hardware Accelerated SDR Platform for Adaptive Air Interfaces,” 2017.

[125] H. M. Ahmed, “Directions in DSP Processors,” *IEEE J. Sel. Areas Commun.*, vol. 8, no. 8, pp. 1420–1427, 1990, doi: 10.1109/49.62820.

[126] A. G. ; T. S. ; M. M. ; E. Auslander, “DSP-based architectures for mobile communications: past, present and future - IEEE Journals & Magazine,” *IEEE Communications Magazine*, 2000. https://ieeexplore.ieee.org/document/815456 (accessed Nov. 11, 2019).

[127] I. Kuon, R. Tessier, and J. Rose, “FPGA Architecture: Survey and Challenges,” *Found. Trends R Electron. Des. Autom.*, vol. 2, no. 2, pp. 135–253, 2008, doi: 10.1561/1000000005.

[128] N. Grover and M. K. Soni, “Reduction of Power Consumption in FPGAs - An Overview,” *Inf. Eng. Electron. Bus.*, vol. 5, pp. 50–69, 2012, doi: 10.5815/ijieeb.2012.05.07.

[129] W. Hussain, J. Isoaho, and J. Nurmi, “The evolution of software defined radio,” *Comput. Platforms Software-Defined Radio*, pp. 1–4, Jan. 2016, doi: 10.1007/978-3-319-49679-5_1.

[130] Sdr, “Software Defined Radio Implementation,” 2010.

[131] N. Arya, K. Sharma, and G. Singh, “Analysis of channels for Software Defined Radio using LabVIEW,” *Int. J. Recent Res. Asp.*, vol. 1, pp. 1–6, 2014.

[132] A. K. Fritz E. Froehlich, “The Froehlich/Kent Encyclopedia of Telecommunications:,” *Froehlich/Kent Encycl. Telecommun. Vol. 18 - Wirel.*, p. 143, Accessed: Nov. 06, 2019. [Online]. Available: https://books.google.co.uk/books?id=mj1o_v17jQAC&pg=PA143&lpg=PA143&dq=The+Modular+Multifunction+Information+Transfer+System+(MMITS)+Forum&source=bl&ots=D107sVp6Fs&sig=ACfU3U2K5d7fY9KNwXGETuw7tpI41xHYaQ&hl=en&sa=X&ved=2ahUKEwjyqbPXrNXlAhVgThUIHYAsAG0Q6AEw.

[133] P. L. Francis, “Challenges and Risks Associated with the Joint Tactical Radio System Program,” 2003, Accessed: Nov. 06, 2019. [Online]. Available: https://books.google.co.uk/books?id=UX6ovAEACAAJ&dq=Joint+Tactical+Radio+System&hl=en&sa=X&ved=0ahUKEwjfuZfPsNXlAhVxVBUIHfhhCRkQ6AEILjAB.

[134] T. H. A. M. Wyglinski, M. Nekovee, “Cognitive Radio Communications and Networks: Principles and Practice,” 2009.

[135] C. Moy and M. Raulet, “High-Level Design Methodology for Ultra-Fast Software Defined Radio Prototyping on Heterogeneous Platforms,” *Adv. Electron. Telecommun.*, vol. 1, no. 1, 2010.

[136] Nutaq, “A short history of software-defined radio (SDR) technology,” *Nutaq*, Accessed: Nov. 07, 2019. [Online]. Available: https://www.nutaq.com/blog/short-history-software-defined-radio-sdr-technology.

[137] GNU Radio, “About GNU Radio,” *GNU Radio*, Accessed: Nov. 12, 2019. [Online]. Available: https://www.gnuradio.org/about/.

[138] B. Wire, “Vanu Inc Announces Commercial Availability of Anywave Base Station First FCC-Certified Software Radio Device Deployed Live in Mid Tex Cellular Network,” *Bus. Wire*, Accessed: Nov. 07, 2019. [Online]. Available:

https://www.businesswire.com/news/home/20050314005270/en/Vanu-Announces-Commercial-Availability-Anywave-Base-Station.

[139] M. A. A. Osman, M. Y. Mustafa, and G. M. Taha, “Software defined radio on FPGA,” *Proc. 2016 Conf. Basic Sci. Eng. Stud. SGCAC 2016*, pp. 171–176, 2016, doi: 10.1109/SGCAC.2016.7458024.

[140] N. Instruments, “Software Defined Radio Past Present and Future,” *National Instruments*. https://www.ni.com/en-gb/innovations/white-papers/17/software-defined-radio--past--present--and-future.html (accessed Nov. 01, 2019).

[141] R. G. Machado and A. M. Wyglinski, “Softwaredefined radio Bridging the analog digital divide,” *Proc. IEEE*, vol. 103, no. 3, pp. 409–423, Mar. 2015, doi: 10.1109/JPROC.2015.2399173.

[142] RTL-SDR, “About RTL-SDR.” https://www.rtl-sdr.com/about-rtl-sdr/ (accessed Jan. 08, 2020).

[143] Great Scott Gadgets, “HackRF One.” https://greatscottgadgets.com/hackrf/one/ (accessed Jan. 08, 2020).

[144] Nuand, “bladeRF USB 3.0 Software Defined Radio,” Accessed: Jan. 08, 2020. [Online]. Available: https://www.nuand.com.

[145] Ettus Research, “USRP B200/B210.” Accessed: Jan. 08, 2020. [Online]. Available: www.ettus.com.

[146] GitHub Mossmann, “Hackrf Schematic,” *Github*, Accessed: Jan. 08, 2020. [Online]. Available: https://github.com/mossmann/hackrf/blob/master/doc/hardware/hackrf-one-schematic.pdf.

[147] Amazon.co.uk Computers and Accessories, “Zinniaya RTL SDR Blog V3 RTL2832U

1PPM TCXO HF BiasT SMA Software Aluminium shielded case Defined Radio R820T2 tuner,” *Amazon*, Accessed: Jan. 08, 2020. [Online]. Available: https://www.amazon.co.uk/Zinniaya-RTL2832U-Software-Aluminium-shielded/.

[148] Amazon.co.uk Computers and Accessories, “HackRF One,” *Amazon*, Accessed: Jan. 08, 2020. [Online]. Available: https://www.amazon.co.uk/Great-Scott-Gadgets-HackRFOne-HackRF.

[149] RoboSavvy, “bladeRF x40,” *RoboSavvy*, Accessed: Jan. 08, 2020. [Online]. Available: https://robosavvy.com/store/bladerf-x40.html.

[150] Taylor Killian, “SDR Showdown HackRF vs bladeRF vs USRP,” *Taylor Kill.*, 2013, Accessed: Jan. 07, 2020. [Online]. Available: http://www.taylorkillian.com/2013/08/sdr-showdown-hackrf-vs-bladerf-vs-usrp.html.

[151] RTLSDR, “RTL-SDR Tutorial Receiving NOAA Weather Satellite Images,” *RTLSDR*, 2013. https://www.rtl-sdr.com/rtl-sdr-tutorial-receiving-noaa-weather-satellite-images/ (accessed Jan. 07, 2020).

[152] F. Eichelberger, “Using Software Defined Radio to Attack ‘Smart Home’ Systems,” 2015, Accessed: Nov. 15, 2019. [Online]. Available: https://pen-testing.sans.org/events/.

[153] M. Pozniak, D. Sadhukhan, and P. Ranganathan, “RF Exploitation and Detection Techniques using Software Defined Radio: A Survey,” Sep. 2019, pp. 345–350, doi: 10.1109/eit.2019.8834203.

[154] Qinetic, “What does the UK need to do to pursue its spectrum resilience objectives,” *Qinetic*, no. January, pp. 1–12, 2018.

[155] RF Globalnet, “Bastille Issues Warning On Radio Based Hacking Risks To National

Infrastructure," *RF Glob.*, Mar. 2017, Accessed: Apr. 02, 2020. [Online]. Available: https://www.rfglobalnet.com/doc/bastille-issues-warning-on-radio-based-hacking-risks-to-national-infrastructure-0001.

[156] O. A. Ibrahim, A. M. Hussain, G. Oligeri, and R. Di Pietro, "Key is in the Air: Hacking Remote Keyless Entry Systems," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2019, vol. 11552 LNCS, pp. 125–132, doi: 10.1007/978-3-030-16874-2_9.

[157] H. Government, "The Key Principles of Cyber Security for Connected and Automated Vehicles," *HM Gov.*

[158] T. Trippel, O. Weisse, W. Xu, P. Honeyman, and K. Fu, "WALNUT Waging Doubt on the Integrity of MEMS Accelerometers with Acoustic Injection Attacks." Accessed: Nov. 15, 2019. [Online]. Available: https://spqr.eecs.umich.edu/walnut/.

[159] B. I. Woodrow, "How Space based ADS B Will Change ATC for NATS UK in the North Atlantic," 2019, Accessed: Nov. 15, 2019. [Online]. Available: https://www.aviationtoday.com/2019/03/26/space-based-ads-b-will-change-atc-nats-uk-north-atlantic/.

[160] D. Moser, P. Leu, A. Ranganathan, F. Ricciato, and S. Capkun, "Investigation of Multi-device Location Spoofing Attacks on Air Traffic Control and Possible Countermeasures," 2016, doi: 10.1145/2973750.2973763.

[161] K. Wang, S. Chen, and A. Pan, "Time and Position Spoofing with Open Source Projects," *Black Hat Conf.*, 2015, Accessed: Nov. 17, 2019. [Online]. Available: https://www.blackhat.com/docs/eu-15/materials/eu-15-Kang-Is-Your-Timespace-Safe-Time-And-Position-Spoofing-Opensourcely-wp.pdf.

[162] A. Solodov, A. Williams, S. Al Hanaei, and B. Goddard, "Analyzing the threat of unmanned aerial vehicles (UAV) to nuclear facilities," *Secur. J.*, vol. 31, no. 1, pp. 305–324, 2018, doi: 10.1057/s41284-017-0102-5.

[163] V.-P. Thai, W. Zhong, T. Pham, S. Alam, and V. Duong, "Detection, Tracking and Classification of Aircraft and Drones in Digital Towers Using Machine Learning on Motion Patterns," *2019 Integr. Commun. Navig. Surveill. Conf.*, pp. 1–8, Jul. 2019, doi: 10.1109/icnsurv.2019.8735240.

[164] A. Schumann, L. Sommer, J. Klatte, T. Schuchert, and J. Beyerer, "Deep cross-domain flying object classification for robust UAV detection," *2017 14th IEEE Int. Conf. Adv. Video Signal Based Surveillance, AVSS 2017*, Oct. 2017, doi: 10.1109/AVSS.2017.8078558.

[165] AVSS, "Drone-vs-Bird Detection Challenge," *WOSDETC*, 2019, Accessed: May 21, 2021. [Online]. Available: https://wosdetc2019.wordpress.com/challenge/.

[166] M. Saqib, S. Daud Khan, N. Sharma, and M. Blumenstein, "A study on detecting drones using deep convolutional neural networks," Oct. 2017, doi: 10.1109/AVSS.2017.8078541.

[167] A. Schumann, L. Sommer, J. Klatte, T. Schuchert, and J. Beyerer, "Deep cross-domain flying object classification for robust UAV detection," Oct. 2017, doi: 10.1109/AVSS.2017.8078558.

[168] B. Taha and A. Shoufan, "Machine Learning-Based Drone Detection and Classification: State-of-the-Art in Research," *IEEE Access*, vol. 7, pp. 138669–138682, 2019, doi: 10.1109/ACCESS.2019.2942944.

[169] C. Aker and S. Kalkan, "Using deep networks for drone detection," Oct. 2017, doi:

10.1109/AVSS.2017.8078539.

[170] R. Yoshihashi, T. T. Trinh, R. Kawakami, S. Y. Csiro-Data61, M. Iida, and T. Naemura, “Differentiating Objects by Motion: Joint Detection and Tracking of Small Flying Objects,” 2018.

[171] J. Peng, C. Zheng, P. Lv, T. Cui, Y. Cheng, and L. Si, “Using images rendered by PBRT to train faster R-CNn for UAV detection,” *Comput. Sci. Res. Notes*, vol. 2802, no. May, pp. 13–18, 2018, doi: 10.24132/CSRN.2018.2802.3.

[172] D. R. Lee, W. Gyu La, and H. Kim, “Drone Detection and Identification System using Artificial Intelligence,” in *9th International Conference on Information and Communication Technology Convergence: ICT Convergence Powered by Smart Intelligence, ICTC 2018*, Nov. 2018, pp. 1131–1133, doi: 10.1109/ICTC.2018.8539442.

[173] E. Unlu, E. Zenou, and N. Rivière, “Generic Fourier Descriptors for Autonomous UAV Detection,” pp. 1–5, 2018.

[174] M. Nalamati, A. Kapoor, M. Saqib, N. Sharma, and M. Blumenstein, “Drone detection in long-range surveillance videos,” Sep. 2019, doi: 10.1109/AVSS.2019.8909830.

[175] A. Coluccia *et al.*, “Drone vs. Bird detection: Deep learning algorithms and results from a grand challenge,” *Sensors*, vol. 21, no. 8, pp. 1–27, 2021, doi: 10.3390/s21082824.

[176] P. Andraši, T. Radišić, M. Muštra, and J. Ivošević, “Night-time Detection of UAVs using Thermal Infrared Camera,” *Transp. Res. Procedia*, vol. 28, pp. 183–190, 2017, doi: 10.1016/j.trpro.2017.12.184.

[177] E. Diamantidou, A. Lalas, K. Votis, and D. Tzovaras, “Multimodal Deep Learning

Framework for Enhanced Accuracy of UAV Detection," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Sep. 2019, vol. 11754 LNCS, pp. 768–777, doi: 10.1007/978-3-030-34995-0_70.

[178] F. Svanstrom, C. Englund, and F. Alonso-Fernandez, "Real-Time Drone Detection and Tracking With Visible, Thermal and Acoustic Sensors," *2020 25th Int. Conf. Pattern Recognit.*, pp. 7265–7272, 2021, doi: 10.1109/icpr48806.2021.9413241.

[179] Y. Wang, Y. Chen, J. Choi, and C. C. J. Kuo, "Towards Visible and Thermal Drone Monitoring with Convolutional Neural Networks," *APSIPA Trans. Signal Inf. Process.*, vol. 8, pp. 1–12, 2019, doi: 10.1017/ATSIP.2018.30.

[180] B. H. Kim, D. Khan, C. Bohak, W. Choi, H. J. Lee, and M. Y. Kim, "V-RBNN based small drone detection in augmented datasets for 3D LADAR system," *Sensors (Switzerland)*, vol. 18, no. 11, p. 3825, Nov. 2018, doi: 10.3390/s18113825.

[181] UK Airfields, "Runway Lengths." http://www.ukairfields.org.uk/runway-lengths.html (accessed Nov. 19, 2022).

[182] D. Khan *et al.*, "Ladar data generation fused with virtual targets and visualization for small drone detection system," Oct. 2018, vol. 10797, p. 17, doi: 10.1117/12.2500525.

[183] B. H. Kim, D. Khan, W. Choi, and M. Y. Kim, "Real-time counter-UAV system for long distance small drones using double pan-tilt scan laser radar," May 2019, vol. 11005, p. 8, doi: 10.1117/12.2520110.

[184] M. Salhi and N. Boudriga, "Multi-array spherical LIDAR system for drone detection," in *International Conference on Transparent Optical Networks*, Jul. 2020, vol. 2020-July, doi: 10.1109/ICTON51198.2020.9203381.

[185] J. Mezei, V. Fiaska, and A. Molnar, “Drone sound detection,” *CINTI 2015 - 16th IEEE Int. Symp. Comput. Intell. Informatics, Proc.*, no. November 2015, pp. 333–338, 2016, doi: 10.1109/CINTI.2015.7382945.

[186] J. Busset *et al.*, “Detection and tracking of drones using advanced acoustic cameras,” *Unmanned/Unattended Sensors Sens. Networks XI; Adv. Free. Opt. Commun. Tech. Appl.*, vol. 9647, p. 96470F, Oct. 2015, doi: 10.1117/12.2194309.

[187] M. Nijim and N. Mantrawadi, “Drone classification and identification system by phenome analysis using data mining techniques,” Sep. 2016, doi: 10.1109/THS.2016.7568949.

[188] S. Jeon, J. W. Shin, Y. J. Lee, W. H. Kim, Y. H. Kwon, and H. Y. Yang, “Empirical study of drone sound detection in real-life environment with deep neural networks,” in *25th European Signal Processing Conference, EUSIPCO 2017*, Oct. 2017, vol. 2017-Janua, pp. 1858–1862, doi: 10.23919/EUSIPCO.2017.8081531.

[189] A. Bernardini, F. Mangiatordi, E. Pallotti, and L. Capodiferro, “Drone detection by acoustic signature identification,” *IS T Int. Symp. Electron. Imaging Sci. Technol.*, no. January, pp. 60–64, 2017, doi: 10.2352/ISSN.2470-1173.2017.10.IMAWM-168.

[190] B. K. Kim, H. S. Kang, and S. O. Park, “Drone classification using convolutional neural networks with merged doppler images,” *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 1, pp. 38–42, Jan. 2017, doi: 10.1109/LGRS.2016.2624820.

[191] X. Yue, Y. Liu, J. Wang, H. Song, and H. Cao, “Software Defined Radio and Wireless Acoustic Networking for Amateur Drone Surveillance,” *IEEE Commun. Mag.*, vol. 56, no. 4, pp. 90–97, Apr. 2018, doi: 10.1109/MCOM.2018.1700423.

[192] Y. Seo, B. Jang, and S. Im, “Drone Detection Using Convolutional Neural Networks

with Acoustic STFT Features," Feb. 2019, doi: 10.1109/AVSS.2018.8639425.

[193] E. Matson, B. Yang, A. Smith, E. Dietz, and J. Gallagher, "UAV Detection System with Multiple Acoustic Nodes Using Machine Learning Models," in *Proceedings - 3rd IEEE International Conference on Robotic Computing, IRC 2019*, Mar. 2019, pp. 493–498, doi: 10.1109/IRC.2019.00103.

[194] Z. Shi, X. Chang, C. Yang, Z. Wu, and J. Wu, "An Acoustic-Based Surveillance System for Amateur Drones Detection and Localization," *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 2731–2739, Mar. 2020, doi: 10.1109/TVT.2020.2964110.

[195] M. Z. Anwar, Z. Kaleem, and A. Jamalipour, "Machine Learning Inspired Sound-Based Amateur Drone Detection for Public Safety Applications," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2526–2534, Mar. 2019, doi: 10.1109/TVT.2019.2893615.

[196] H. Shin, K. Choi, Y. Park, J. Choi, and Y. Kim, "Security analysis of FHSS-type drone controller," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016, vol. 9503, pp. 240–253, doi: 10.1007/978-3-319-31875-2_20.

[197] P. Nguyen, M. Ravindranathan, A. Nguyen, R. Han, and T. Vu, "Investigating cost-effective RF-based detection of drones," *DroNet 2016 - Proc. 2nd Work. Micro Aer. Veh. Networks, Syst. Appl. Civ. Use, co-located with MobiSys 2016*, no. June, pp. 17–22, 2016, doi: 10.1145/2935620.2935632.

[198] Z. Shi, M. Huang, C. Zhao, L. Huang, X. Du, and Y. Zhao, "Detection of LSSUAV using hash fingerprint based SVDD," Jul. 2017, doi: 10.1109/ICC.2017.7996844.

[199] P. Nguyen, H. Truong, M. Ravindranathan, A. Nguyen, R. Han, and T. Vu, "Matthan: Drone presence detection by identifying physical signatures in the drone's RF

communication," *MobiSys 2017 - Proc. 15th Annu. Int. Conf. Mob. Syst. Appl. Serv.*, pp. 211–224, 2017, doi: 10.1145/3081333.3081354.

[200] S. Abeywickrama, L. Jayasinghe, H. Fu, S. Nissanka, and C. Yuen, "RF-based Direction Finding of UAVs Using DNN," *2018 IEEE Int. Conf. Commun. Syst. ICCS 2018*, pp. 157–161, 2018, doi: 10.1109/ICCS.2018.8689177.

[201] C. Zhao, C. Chen, Z. Cai, M. Shi, X. Du, and M. Guizani, "Classification of Small UAVs Based on Auxiliary Classifier Wasserstein GANs," *2018 IEEE Glob. Commun. Conf. GLOBECOM 2018 - Proc.*, 2018, doi: 10.1109/GLOCOM.2018.8647973.

[202] M. Ezuma, F. Erden, C. K. Anjinappa, O. Ozdemir, and I. Guvenc, "Detection and classification of UAVs using RF fingerprints in the presence of interference," *arXiv*, vol. 1, no. January, pp. 60–76, 2019, doi: 10.1109/OJCOMS.2019.2955889.

[203] X. Huang, K. Yan, H. C. Wu, and Y. Wu, "Unmanned Aerial Vehicle Hub Detection Using Software-Defined Radio," *IEEE Int. Symp. Broadband Multimed. Syst. Broadcast. BMSB*, vol. 2019-June, 2019, doi: 10.1109/BMSB47279.2019.8971851.

[204] M. Al-Sa'd, M. S. Allahham, A. Mohamed, A. Al-Ali, T. Khattab, and A. Erbad, "DroneRF dataset: A dataset of drones for RF-based detection, classification, and identification," vol. 1, 2019, doi: 10.17632/F4C2B4N755.1.

[205] X. Liang, Y. Jiang, and T. A. Gulliver, "An improved sensing method using radio frequency detection," *Phys. Commun.*, vol. 36, p. 100763, 2019, doi: 10.1016/j.phycom.2019.100763.

[206] S. Al-Emadi and F. Al-Senaid, "Drone Detection Approach Based on Radio-Frequency Using Convolutional Neural Network," *2020 IEEE Int. Conf. Informatics, IoT, Enabling Technol. ICIoT 2020*, pp. 29–34, 2020, doi:

10.1109/ICIoT48696.2020.9089489.

[207] I. Nemer, T. Sheltami, I. Ahmad, A. U. H. Yasar, and M. A. R. Abdeen, “Rf-based UAV detection and identification using hierarchical learning approach,” *Sensors,* vol. 21, no. 6, pp. 1–23, 2021, doi: 10.3390/s21061947.

[208] H. Lv, F. Liu, and N. C. Yuan, “Drone Presence Detection by the Drone’s RF Communication,” *J. Phys. Conf. Ser.*, vol. 1738, no. 1, 2021, doi: 10.1088/1742-6596/1738/1/012044.

[209] S. Rahman and D. A. Robertson, “Radar micro-Doppler signatures of drones and birds at K-band and W-band,” *Sci. Reports 2018 81*, vol. 8, no. 1, pp. 1–11, Nov. 2018, doi: 10.1038/s41598-018-35880-9.

[210] J. J. M. De Wit, R. I. A. Harmanny, and G. Prémel-Cabic, “Micro-Doppler analysis of small UAVs,” *Eur. Microw. Week 2012 “sp. Microwaves”, EuMW 2012, Conf. Proc. - 9th Eur. Radar Conf. EuRAD 2012*, pp. 210–213, 2012.

[211] P. Molchanov, R. I. A. Harmanny, J. J. M. De Wit, K. Egiazarian, and J. Astola, “Classification of small UAVs and birds by micro-Doppler signatures,” *Int. J. Microw. Wirel. Technol.*, vol. 6, no. 3–4, pp. 435–444, 2014, doi: 10.1017/S1759078714000282.

[212] R. I. A. Harmanny, J. J. M. De Wit, and G. Prémel Cabic, “Radar micro-Doppler feature extraction using the spectrogram and the cepstrogram,” *Eur. Microw. Week 2014 "Connecting Futur. EuMW 2014 - Conf. Proceedings; EuRAD 2014 11th Eur. Radar Conf.*, pp. 165–168, 2014, doi: 10.1109/EuRAD.2014.6991233.

[213] J. J. M. De Wit, R. I. A. Harmanny, and P. Molchanov, “Radar micro-Doppler feature extraction using the Singular Value Decomposition,” *2014 Int. Radar Conf. Radar*

2014, pp. 1–6, 2014, doi: 10.1109/RADAR.2014.7060268.

[214] N. Mohajerin, J. Histon, R. Dizaji, and S. L. Waslander, “Feature extraction and radar track classification for detecting UAVs in civillian airspace,” in *IEEE National Radar Conference - Proceedings*, 2014, pp. 674–679, doi: 10.1109/RADAR.2014.6875676.

[215] F. Fioranelli, M. Ritchie, H. Borrion, and H. Griffiths, “Classification of Loaded/Unloaded Micro-Drones Using Multistatic Radar,” *Core*, 2015.

[216] S. Zulkifli and A. Balleri, “Design and Development of K-Band FMCW Radar for Nano-Drone Detection,” *IEEE Natl. Radar Conf. - Proc.*, vol. 2020-Septe, 2020, doi: 10.1109/RadarConf2043947.2020.9266538.

[217] V. Semkin, M. Yin, Y. Hu, M. Mezzavilla, and S. Rangan, “Drone detection and classification based on radar cross section signatures,” *2020 Int. Symp. Antennas Propagation, ISAP 2020*, pp. 223–224, 2021, doi: 10.23919/ISAP47053.2021.9391260.

[218] M. Ritchie, F. Fioranelli, H. Griffiths, and B. Torvik, “Micro-drone RCS analysis,” in *2015 IEEE Radar Conference - Proceedings*, Oct. 2015, pp. 452–456, doi: 10.1109/RadarConf.2015.7411926.

[219] M. Jahangir and C. Baker, “Robust Detection of Micro-UAS Drones with L-Band 3-D Holographic Radar,” Oct. 2016, doi: 10.1109/SSPD.2016.7590610.

[220] J. Lundén and V. Koivunen, “Deep learning for HRRP-based target recognition in multistatic radar systems,” Jun. 2016, doi: 10.1109/RADAR.2016.7485271.

[221] B. Torvik, K. E. Olsen, and H. Griffiths, “Classification of Birds and UAVs Based on Radar Polarimetry,” *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 9, pp. 1305–1309, Sep. 2016, doi: 10.1109/LGRS.2016.2582538.

[222] B. S. Oh, X. Guo, F. Wan, K. A. Toh, and Z. Lin, "Micro-Doppler Mini-UAV Classification Using Empirical-Mode Decomposition Features," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 2, pp. 227–231, 2018, doi: 10.1109/LGRS.2017.2781711.

[223] G. J. Mendis, J. Wei, and A. Madanayake, "Deep learning cognitive radar for micro UAS detection and classification," Aug. 2017, doi: 10.1109/CCAAW.2017.8001610.

[224] J. Ren and X. Jiang, "Regularized 2-D complex-log spectral analysis and subspace reliability analysis of micro-Doppler signature for UAV detection," *Pattern Recognit.*, vol. 69, pp. 225–237, Sep. 2017, doi: 10.1016/j.patcog.2017.04.024.

[225] P. Zhang, L. Yang, G. Chen, and G. Li, "Classification of drones based on micro-doppler signatures with dual-band radar sensors," in *Progress in Electromagnetics Research Symposium*, Nov. 2017, vol. 2017-Novem, pp. 638–643, doi: 10.1109/PIERS-FALL.2017.8293214.

[226] N. Regev, I. Yoffe, and D. Wulich, "Classification of single and multi propelled miniature drones using multilayer perceptron artificial neural network," in *IET Conference Publications*, 2017, vol. 2017, no. CP728, doi: 10.1049/cp.2017.0378.

[227] L. Fuhrmann, O. Biallawons, J. Klare, R. Panhuber, R. Klenke, and J. Ender, "Micro-Doppler analysis and classification of UAVs at Ka band," *Proc. Int. Radar Symp.*, pp. 1–9, 2017, doi: 10.23919/IRS.2017.8008142.

[228] X. Ma, B. S. Oh, L. Sun, K. A. Toh, and Z. Lin, "EMD-Based Entropy Features for micro-Doppler Mini-UAV Classification," *Proc. - Int. Conf. Pattern Recognit.*, vol. 2018-Augus, pp. 1295–1300, 2018, doi: 10.1109/ICPR.2018.8546180.

[229] Y. Sun, H. Fu, S. Abeywickrama, L. Jayasinghe, C. Yuen, and J. Chen, "Drone Classification and Localization Using Micro-Doppler Signature with Low-Frequency

Signal," in *2018 IEEE International Conference on Communication Systems, ICCS 2018*, Jul. 2018, pp. 413–417, doi: 10.1109/ICCS.2018.8689237.

[230] D. Habermann, E. Dranka, Y. Caceres, and J. B. R. Do Val, "Drones and helicopters classification using point clouds features from radar," in *2018 IEEE Radar Conference, RadarConf 2018*, Jun. 2018, pp. 246–251, doi: 10.1109/RADAR.2018.8378565.

[231] L. Wang, J. Tang, and Q. Liao, "A Study on Radar Target Detection Based on Deep Neural Networks," *IEEE Sensors Lett.*, vol. 3, no. 3, pp. 2019–2022, 2019, doi: 10.1109/LSENS.2019.2896072.

[232] S. Samaras, V. Magoulianitis, A. Dimou, D. Zarpalas, and P. Daras, "UAV Classification with Deep Learning Using Surveillance Radar Data," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Sep. 2019, vol. 11754 LNCS, pp. 744–753, doi: 10.1007/978-3-030-34995-0_68.

[233] B. Choi and D. Oh, "Classification of Drone Type Using Deep Convolutional Neural Networks Based on Micro- Doppler Simulation," *ISAP 2018 - 2018 Int. Symp. Antennas Propag.*, no. 1, pp. 17–18, 2019.

[234] C. Guo, Y. He, H. Wang, T. Jian, and S. Sun, "Radar HRRP Target Recognition Based on Deep One-Dimensional Residual-Inception Network," *IEEE Access*, vol. 7, pp. 9191–9204, 2019, doi: 10.1109/ACCESS.2019.2891594.

[235] W. S. Chen, J. Liu, and J. Li, "Classification of UAV and bird target in low-altitude airspace with surveillance radar data," *Aeronaut. J.*, vol. 123, no. 1260, pp. 191–211, Feb. 2019, doi: 10.1017/aer.2018.158.

[236] M. Messina and G. Pinelli, “Classification of Drones with a Surveillance Radar Signal,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11754 LNCS, no. September, pp. 723–733, 2019, doi: 10.1007/978-3-030-34995-0_66.

[237] A. Coluccia, G. Parisi, and A. Fascista, “Detection and Classification of Multirotor Drones in Radar Sensor Networks: A Review,” doi: 10.3390/s20154172.

[238] M. Passafiume, N. Rojhani, G. Collodi, and A. Cidronali, “Modeling Small UAV Micro-Doppler Signature UsingMillimeter-Wave FMCW Radar,” *MDPI Electron.*, 2021, doi: 10.3390/electronics10060747.

[239] S. R. Ganti and Y. Kim, “Implementation of detection and tracking mechanism for small UAS,” *2016 Int. Conf. Unmanned Aircr. Syst. ICUAS 2016*, pp. 1254–1260, Jun. 2016, doi: 10.1109/ICUAS.2016.7502513.

[240] M. F. Al-Sa’d, A. Al-Ali, A. Mohamed, T. Khattab, and A. Erbad, “RF-based drone detection and identification using deep learning approaches: An initiative towards a large open source drone database,” *Futur. Gener. Comput. Syst.*, vol. 100, pp. 86–97, Nov. 2019, doi: 10.1016/j.future.2019.05.007.

[241] P. SHERPA, “ALADDIN,” *Horiz. 2020 Eur. Union*, no. 740859, pp. 1–38, 2020.

[242] D. C. Measures, “Domestic Drone Countermeasures,” *We Measure Everything About Home Stuffs*, 2020.

[243] B.-P. Teh, “RF techniques for detection, classification and location of commercial drone controllers,” *Aerosp. Def. Symp.*, 2017.

[244] X. Shi, C. Yang, W. Xie, C. Liang, Z. Shi, and J. Chen, “Anti-Drone System with Multiple Surveillance Technologies: Architecture, Implementation, and Challenges,”

IEEE Commun. Mag., vol. 56, no. 4, pp. 68–74, 2018, doi: 10.1109/MCOM.2018.1700430.

[245] Z. Zhang, C. Zeng, M. Dhameliya, S. Chowdhury, and R. Rai, “Deep learning based multi-modal sensing for tracking and state extraction of small quadcopters,” *arXiv*, 2020.

[246] M. R. Brust, G. Danoy, P. Bouvry, D. Gashi, H. Pathak, and M. P. Goncalves, “Defending Against Intrusion of Malicious UAVs with Networked UAV Defense Swarms,” in *Proceedings - 2017 IEEE 42nd Conference on Local Computer Networks Workshops, LCN Workshops 2017*, Nov. 2017, pp. 103–111, doi: 10.1109/LCN.Workshops.2017.71.

[247] J. Rothe, M. Strohmeier, and S. Montenegro, “A concept for catching drones with a net carried by cooperative UAVs,” *2019 IEEE Int. Symp. Safety, Secur. Rescue Robot. SSRR 2019*, pp. 126–132, 2019, doi: 10.1109/SSRR.2019.8848973.

[248] M. Goodrich, “Drone Catcher: ‘Robotic Falcon’ can Capture, Retrieve Renegade Drones,” *Michigan Tech News*, Jan. 07, 2016. https://www.mtu.edu/news/2016/01/drone-catcher-robotic-falcon-can-capture-retrieve-renegade-drones.html (accessed Nov. 04, 2021).

[249] C. Ford, “OpenWorks Engineering wins large European orders for its anti-drone SkyWall system,” *Business Live*, Feb. 02, 2021. https://www.business-live.co.uk/manufacturing/openworks-engineering-wins-large-european-19753723 (accessed Nov. 04, 2021).

[250] T. Ong, “Dutch police will stop using drone-hunting eagles since they weren’t doing what they’re told,” *The Verge*, Dec. 12, 2017. https://www.theverge.com/2017/12/12/16767000/police-netherlands-eagles-rogue-

drones (accessed Nov. 04, 2021).

[251] J.-P. Yaacoub, H. Noura, O. Salman, and A. Chehab, “Security analysis of drones systems: Attacks, limitations, and recommendations,” *Internet of Things*, vol. 11, p. 100218, Sep. 2020, doi: 10.1016/j.iot.2020.100218.

[252] J. Saballa, “France Test Fires Anti-Drone Laser,” *Defense Post*, Jul. 09, 2021. https://www.thedefensepost.com/2021/07/09/france-anti-drone-laser/ (accessed Nov. 04, 2021).

[253] M. Dempsey, “Shooting drones out of the sky with Phasers,” *BBC News*, Oct. 15, 2019. https://www.bbc.co.uk/news/business-49984415 (accessed Nov. 04, 2021).

[254] India News, “Security forces use pump action guns to kill low-flying drones,” *India News*, Sep. 20, 2021. https://www.indiatoday.in/india/story/security-forces-use-pump-action-guns-to-kill-low-flying-drones-1854674-2021-09-20 (accessed Nov. 04, 2021).

[255] T. Multerer *et al.*, “Low-cost jamming system against small drones using a 3D MIMO radar based tracking,” *Eur. Microw. Week 2017 “A Prime Year a Prime Event”, EuMW 2017 - Conf. Proceedings; 14th Eur. Microw. Conf. EURAD 2017*, vol. 2018-Janua, pp. 299–302, 2017, doi: 10.23919/EURAD.2017.8249206.

[256] K. Parlin, M. M. Alam, and Y. Le Moullec, “Jamming of UAV remote control systems using software defined radio,” *2018 Int. Conf. Mil. Commun. Inf. Syst. ICMCIS 2018*, no. May, pp. 1–6, 2018, doi: 10.1109/ICMCIS.2018.8398711.

[257] Z. Li, Z. Xiao, B. Wang, B. Y. Zhao, and H. Zheng, “Scaling Deep Learning Models for Spectrum Anomaly Detection,” p. 10, 2019, doi: 10.1145/3323679.3326527.

[258] M. Sliti, W. Abdallah, and N. Boudriga, “Jamming Attack Detection in Optical UAV Networks,” in *International Conference on Transparent Optical Networks*, Sep. 2018,

vol. 2018-July, doi: 10.1109/ICTON.2018.8473921.

[259] W. Chen, Y. Dong, and Z. Duan, "Manipulating drone dynamic state estimation to compromise navigation," *2018 IEEE Conf. Commun. Netw. Secur. CNS 2018*, 2018, doi: 10.1109/CNS.2018.8433205.

[260] Y. M. Kwon, J. Yu, B. M. Cho, Y. Eun, and K. J. Park, "Empirical Analysis of MAVLink Protocol Vulnerability for Attacking Unmanned Aerial Vehicles," *IEEE Access*, vol. 6, pp. 43203–43212, Aug. 2018, doi: 10.1109/ACCESS.2018.2863237.

[261] Y. Lei, L. Zeng, Y. X. Li, M. X. Wang, and H. Qin, "A Lightweight Authentication Protocol for UAV Networks Based on Security and Computational Resource Optimization," *IEEE Access*, vol. 9, pp. 53769–53785, 2021, doi: 10.1109/ACCESS.2021.3070683.

[262] O. Westerlund and R. Asif, "Drone Hacking with Raspberry-Pi 3 and WiFi Pineapple: Security and Privacy Threats for the Internet-of-Things," *2019 1st Int. Conf. Unmanned Veh. Syst. UVS 2019*, pp. 5–7, 2019, doi: 10.1109/UVS.2019.8658279.

[263] B. E. B. Bekdil, "Turkey eyes directed-energy weapons as key priority," *Defense News*, 2021. https://www.defensenews.com/industry/techwatch/2021/03/15/turkey-eyes-directed-energy-weapons-as-key-priority/ (accessed Jul. 02, 2021).

[264] Raytheon Missiles and Defense, "Phaser High-Power Microwave System," *Raytheon Missiles and Defense*, 2021. https://www.raytheonmissilesanddefense.com/capabilities/products/phaser-high-power-microwave (accessed Jul. 02, 2021).

[265] CISION, "SAVAGE - Smart Anti-Drone Weapon," *CISION PR Newswire*, Oct. 21, 2019. https://www.prnewswire.com/news-releases/savage---smart-anti-drone-weapon-

300941541.html (accessed Jul. 02, 2021).

[266] AerialX, “DroneBullet,” *AerialX*, 2021. https://aerialx.com/dronebullet/ (accessed Jul. 02, 2021).

[267] K. D. Atherton, “Trained Police Eagles Attack Drones On Command,” *Popular Science*, Feb. 02, 2016. https://www.popsci.com/eagles-attack-drones-at-police-command/ (accessed Jul. 02, 2021).

[268] I. Guvenc, F. Koohifar, S. Singh, M. L. Sichitiu, and D. Matolak, “Detection, Tracking, and Interdiction for Amateur Drones,” *IEEE Commun. Mag.*, vol. 56, no. 4, pp. 75–81, Apr. 2018, doi: 10.1109/MCOM.2018.1700455.

[269] P. L. Lineswala and S. N. Shah, “Performance analysis of different interference detection techniques for navigation with Indian constellation,” *IET Radar, Sonar Navig.*, vol. 13, no. 8, pp. 1207–1213, Aug. 2019, doi: 10.1049/iet-rsn.2019.0091.

[270] J.-H. Lee, H.-P. Kim, and J.-H. Won, “GNSS Cloud-Data Processing Technique For Jamming Detection And Localization,” *Int. Tech. Symp. Navig. Timing, Toulouse, Fr.*, Oct. 2018, doi: 10.31701/itsnt2018.23ï.

[271] H. Kim, G. Jin, and J. Won, “GNSS cloud-data processing technique for jamming detection, identification, and localisation,” *IET Radar, Sonar Navig.*, vol. 14, no. 8, pp. 1143–1149, Aug. 2020, doi: 10.1049/iet-rsn.2019.0518.

[272] H. Xu, Y. Cheng, J. Liang, and P. Wang, “A Jamming Recognition Algorithm Based on Deep Neural Network in Satellite Navigation System,” *China Satell. Navig. Conf. 2020*, vol. 652 LNEE, pp. 701–711, May 2020, doi: 10.1007/978-981-15-3715-8_63.

[273] S. Rajendran, W. Meert, V. Lenders, and S. Pollin, “Unsupervised Wireless Spectrum Anomaly Detection with Interpretable Features,” *IEEE Trans. Cogn. Commun. Netw.*,

2019, doi: 10.1109/TCCN.2019.2911524.

[274] N. Tandiya, A. Jauhar, V. Marojevic, and J. H. Reed, “Deep Predictive Coding Neural Network for RF Anomaly Detection in Wireless Networks,” *2018 IEEE Int. Conf. Commun. Work. (ICC Work.*, 2018.

[275] Coxlab, “PredNet,” Accessed: Apr. 02, 2020. [Online]. Available: https://coxlab.github.io/prednet/.

[276] T. J. O’Shea, T. Roy, and T. Erpek, “Spectral detection and localization of radio events with learned convolutional neural features,” *25th Eur. Signal Process. Conf. EUSIPCO 2017*, vol. 2017-Janua, pp. 331–335, 2017, doi: 10.23919/EUSIPCO.2017.8081223.

[277] Y. Arjoune, F. Salahdine, S. Islam, E. Ghribi, and N. Kaabouch, “A Novel Jamming Attacks Detection Approach Based on Machine Learning for Wireless Communication,” 2020. Accessed: Jul. 16, 2020. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02509430.

[278] Z. Wu, Y. Zhao, Z. Yin, and H. Luo, “Jamming signals classification using convolutional neural network,” *2017 IEEE Int. Symp. Signal Process. Inf. Technol. ISSPIT 2017*, pp. 62–67, Jun. 2018, doi: 10.1109/ISSPIT.2017.8388320.

[279] Y. Yang and L. Zhu, “An efficient way for satellite interference signal recognition via incremental learning,” *2019 Int. Symp. Networks, Comput. Commun. ISNCC 2019*, 2019, doi: 10.1109/ISNCC.2019.8909134.

[280] J. Price, Y. Li, K. Al Shamaileh, Q. Niyaz, N. Kaabouch, and V. Devabhaktuni, “Real-time Classification of Jamming Attacks against UAVs via on-board Software-defined Radio and Machine Learning-based Receiver Module,” *IEEE Int. Conf. Electro Inf. Technol.*, vol. 2022-May, pp. 252–256, 2022, doi: 10.1109/eIT53891.2022.9813923.

[281] Z. Hao, W. Yu, and W. Chen, "Recognition method of dense false targets jamming based on time-frequency atomic decomposition," *J. Eng.*, vol. 2019, no. 20, pp. 6354–6358, 2019, doi: 10.1049/joe.2019.0147.

[282] R. R. Fu, "Compound jamming signal recognition based on neural networks," *Proc. - 2016 6th Int. Conf. Instrum. Meas. Comput. Commun. Control. IMCCC 2016*, pp. 737–740, 2016, doi: 10.1109/IMCCC.2016.163.

[283] R. Zhang and S. Cao, "Support vector machines for classification of automotive radar interference," *2018 IEEE Radar Conf. RadarConf 2018*, pp. 366–371, 2018, doi: 10.1109/RADAR.2018.8378586.

[284] M. Kong, J. Liu, Z. Zhang, and Y. Qiao, "Radio ground-to-air interference signals recognition based on support vector machine," *Int. Conf. Digit. Signal Process. DSP*, vol. 2015-Septe, pp. 987–990, 2015, doi: 10.1109/ICDSP.2015.7252025.

[285] Y. Junfei, L. Jingwen, S. Bing, and J. Yuming, "Barrage jamming detection and classification based on convolutional neural network for synthetic aperture radar," in *International Geoscience and Remote Sensing Symposium (IGARSS)*, Oct. 2018, vol. 2018-July, pp. 4583–4586, doi: 10.1109/IGARSS.2018.8519373.

[286] G. Shao, Y. Chen, and Y. Wei, "Convolutional Neural Network-Based Radar Jamming Signal Classification with Sufficient and Limited Samples," *IEEE Access*, vol. 8, pp. 80588–80598, 2020, doi: 10.1109/ACCESS.2020.2990629.

[287] K. Simonyan and A. Zisserman, "VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION," *ICLR 2015*, 2015, Accessed: Mar. 26, 2020. [Online]. Available: http://www.robots.ox.ac.uk/.

[288] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition,"

Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2016-Decem, pp. 770–778, Dec. 2016, doi: 10.1109/CVPR.2016.90.

[289] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” *Commun. ACM*, 2017, Accessed: May 10, 2020. [Online]. Available: http://code.google.com/p/cuda-convnet/.

[290] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.

[291] G. Richard, T. Learning, and M. Learning, “Transfer Learning methods for temporal data To cite this version : ethodes d ’ apprentissage statistique de type temporelles multivari ´ Thèse de doctorat,” 2022.

[292] E. Tsalera, A. Papadakis, and M. Samarakou, “Comparison of pre-trained cnns for audio classification using transfer learning,” *J. Sens. Actuator Networks*, vol. 10, no. 4, 2021, doi: 10.3390/jsan10040072.

[293] K. Palanisamy, D. Singhania, and A. Yao, “Rethinking CNN Models for Audio Classification,” 2020, [Online]. Available: http://arxiv.org/abs/2007.11154.

[294] T. Koike, K. Qian, Q. Kong, M. D. Plumbley, B. W. Schuller, and Y. Yamamoto, “Audio for Audio is Better? An Investigation on Transfer Learning Models for Heart Sound Classification,” *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, vol. 2020-July, pp. 74–77, 2020, doi: 10.1109/EMBC44109.2020.9175450.

[295] Z. Ren, N. Cummins, V. Pandit, J. Han, K. Qian, and B. Schuller, “Learning image-based representations for heart sound classification,” *ACM Int. Conf. Proceeding Ser.*, vol. 2018-April, no. 143, pp. 143–147, 2018, doi: 10.1145/3194658.3194671.

[296] H. Zhou, X. Bai, and J. Du, “An investigation of transfer learning mechanism for acoustic scene classification,” *2018 11th Int. Symp. Chinese Spok. Lang. Process. ISCSLP 2018 - Proc.*, pp. 404–408, Jul. 2018, doi: 10.1109/ISCSLP.2018.8706712.

[297] B. Nagarajan and V. Ramana Murthy Oruganti, “Cross-Domain Transfer Learning for Complex Emotion Recognition,” *Proc. 2019 IEEE Reg. 10 Symp. TENSYMP 2019*, pp. 649–653, Jun. 2019, doi: 10.1109/TENSYMP46218.2019.8971023.

[298] R. Müller, F. Ritz, S. Illium, and C. Linnhoff-Popien, “Acoustic anomaly detection for machine sounds based on image transfer learning,” *ICAART 2021 - Proc. 13th Int. Conf. Agents Artif. Intell.*, vol. 2, pp. 49–56, 2021, doi: 10.5220/0010185800490056.

[299] E. Dufourq, C. Batist, R. Foquet, and I. Durbach, “Passive acoustic monitoring of animal populations with transfer learning,” *Ecol. Inform.*, vol. 70, p. 101688, Sep. 2022, doi: 10.1016/J.ECOINF.2022.101688.

[300] N. B. Thota and D. Umma Reddy, “Improving the Accuracy of Diabetic Retinopathy Severity Classification with Transfer Learning,” *Midwest Symp. Circuits Syst.*, vol. 2020-Augus, pp. 1003–1006, Aug. 2020, doi: 10.1109/MWSCAS48704.2020.9184473.

[301] C. Jayakumari, V. Lavanya, and E. P. Sumesh, “Automated Diabetic Retinopathy Detection and classification using ImageNet Convolution Neural Network using Fundus Images,” *2020 Int. Conf. Smart Electron. Commun.*, pp. 577–582, Sep. 2020, doi: 10.1109/ICOSEC49089.2020.9215270.

[302] K. Weimann and T. O. F. Conrad, “Transfer learning for ECG classification,” *Sci. Rep.*, vol. 11, no. 1, pp. 1–12, 2021, doi: 10.1038/s41598-021-84374-8.

[303] M. Salem, S. Taheri, and J. S. Yuan, “ECG Arrhythmia Classification Using Transfer

Learning from 2- Dimensional Deep CNN Features," *2018 IEEE Biomed. Circuits Syst. Conf. BioCAS 2018 - Proc.*, Dec. 2018, doi: 10.1109/BIOCAS.2018.8584808.

[304] A. Pal, R. Srivastva, and Y. N. Singh, "CardioNet: An Efficient ECG Arrhythmia Classification System Using Transfer Learning," *Big Data Res.*, vol. 26, p. 100271, Nov. 2021, doi: 10.1016/J.BDR.2021.100271.

[305] J. Venton, P. J. Aston, N. A. S. Smith, and P. M. Harris, "Signal to Image to Classification: Transfer Learning for ECG," *2020 11th Conf. Eur. Study Gr. Cardiovasc. Oscil. Comput. Model. Physiol. New Challenges Oppor. ESGCO 2020*, Jul. 2020, doi: 10.1109/ESGCO49734.2020.9158037.

[306] C. Brown, A. Grammenos, T. Xia, P. Cicuta, and C. Mascolo, "Exploring Automatic Diagnosis of COVID-19 from Crowdsourced Respiratory Sound Data."

[307] N. Sharma *et al.*, "Coswara - A database of breathing, cough, and voice sounds for COVID-19 diagnosis," *Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH*, vol. 2020-Octob, pp. 4811–4815, 2020, doi: 10.21437/INTERSPEECH.2020-2768.

[308] J. Han *et al.*, "An early study on intelligent analysis of speech under COVID-19: Severity, sleep quality, fatigue, and anxiety," *Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH*, vol. 2020-Octob, pp. 4946–4950, 2020, doi: 10.21437/Interspeech.2020-2223.

[309] A. Imran *et al.*, "AI4COVID-19: AI enabled preliminary diagnosis for COVID-19 from cough samples via an app," *Informatics Med. unlocked*, vol. 20, Jan. 2020, doi: 10.1016/J.IMU.2020.100378.

[310] I. LAHSAINI, M. EL HABIB DAHO, and M. A. CHIKH, "Deep transfer learning

based classification model for covid-19 using chest CT-scans," *Pattern Recognit. Lett.*, vol. 152, pp. 122–128, Dec. 2021, doi: 10.1016/J.PATREC.2021.08.035.

[311] T. Garg, M. Garg, O. P. Mahela, and A. R. Garg, "Convolutional Neural Networks with Transfer Learning for Recognition of COVID-19: A Comparative Study of Different Approaches," *AI 2020, Vol. 1, Pages 586-606*, vol. 1, no. 4, pp. 586–606, Dec. 2020, doi: 10.3390/AI1040034.

[312] A. Alotaibi, "Transfer Learning for Detecting Covid-19 Cases Using Chest X-Ray Images," *Int. J. Mach. Learn. Networked Collab. Eng.*, vol. 4, no. 1, pp. 21–29, 2020, doi: 10.30991/ijmlnce.2020v04i01.003.

[313] R. Mormont, P. Geurts, and R. Maree, "Comparison of deep transfer learning strategies for digital pathology," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, Dec. 2018, vol. 2018-June, pp. 2343–2352, doi: 10.1109/CVPRW.2018.00303.

[314] T. Kaur and T. K. Gandhi, "Automated brain image classification based on VGG-16 and transfer learning," *Proc. - 2019 Int. Conf. Inf. Technol. ICIT 2019*, pp. 94–98, Dec. 2019, doi: 10.1109/ICIT48102.2019.00023.

[315] M. Wildan Oktavian, N. Yudistira, A. Ridok, M. Wildan Oktavian, N. Yudistira, and A. Ridok, "Classification of Alzheimer's Disease Using the Convolutional Neural Network (CNN) with Transfer Learning and Weighted Loss," *arXiv*, p. arXiv:2207.01584, Jul. 2022, Accessed: Aug. 24, 2022. [Online]. Available: https://ui.adsabs.harvard.edu/abs/2022arXiv220701584W/abstract.

[316] M. Maqsood *et al.*, "Transfer Learning Assisted Classification and Detection of Alzheimer's Disease Stages Using 3D MRI Scans," *Sensors 2019, Vol. 19, Page 2645*, vol. 19, no. 11, p. 2645, Jun. 2019, doi: 10.3390/S19112645.

[317] U. A. H. Khan *et al.*, “Improving Prostate Cancer Detection with Breast Histopathology Images,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11435 LNCS, pp. 91–99, 2019, doi: 10.1007/978-3-030-23937-4_11.

[318] C. Chen, H. Seo, and Y. Zhao, “A novel pavement transverse cracks detection model using WT-CNN and STFT-CNN for smartphone data analysis,” *Int. J. Pavement Eng.*, vol. 0, no. 0, pp. 1–13, 2021, doi: 10.1080/10298436.2021.1945056.

[319] D. George and E. A. Huerta, “Deep Neural Networks to Enable Real-time Multimessenger Astrophysics,” *Phys. Rev. D*, vol. 97, no. 4, Dec. 2016, doi: 10.1103/PhysRevD.97.044039.

[320] G. D. O’Mahony, K. G. McCarthy, P. J. Harris, and C. C. Murphy, “Developing novel low complexity models using received in-phase and quadrature-phase samples for interference detection and classification in Wireless Sensor Network and GPS edge devices,” *Ad Hoc Networks*, vol. 120, p. 102562, 2021, doi: 10.1016/j.adhoc.2021.102562.

[321] S. Ackermann, K. Schawinski, C. Zhang, A. K. Weigel, and M. Dennis Turp, “Using transfer learning to detect galaxy mergers,” *Mon. Not. R. Astron. Soc.*, vol. 479, no. 1, pp. 415–425, Sep. 2018, doi: 10.1093/MNRAS/STY1398.

[322] D. N. F. Awang Iskandar *et al.*, “Classification of Planetary Nebulae through Deep Transfer Learning,” *Galaxies 2020, Vol. 8, Page 88*, vol. 8, no. 4, p. 88, Dec. 2020, doi: 10.3390/GALAXIES8040088.

[323] W. Wei *et al.*, “Deep transfer learning for star cluster classification: I. application to the PHANGS–HST survey,” *Mon. Not. R. Astron. Soc.*, vol. 493, no. 3, pp. 3178–3193, Apr. 2020, doi: 10.1093/MNRAS/STAA325.

[324] J. Guérin, S. Thiery, E. Nyiri, O. Gibaru, and B. Boots, “Combining pretrained CNN feature extractors to enhance clustering of complex natural images,” *Neurocomputing*, vol. 423, pp. 551–571, Jan. 2021, doi: 10.1016/J.NEUCOM.2020.10.068.

[325] R. Morales Ferre, E. S. Lohan, and A. De la Fuente, “Image datasets for jammer classification in GNSS,” Aug. 2019, doi: 10.5281/ZENODO.3370934.

[326] LabSat, “GNSS Frequency Guide,” *LabSat Freq. Guid.*, 2021, Accessed: May 12, 2021. [Online]. Available: https://www.labsat.co.uk/index.php/en/gnss-frequency-guide.

[327] T. Ebinuma, “Software-Defined GPS Signal Simulator,” *GitHub - osqzss/gps-sdr-sim*, 2018, Accessed: May 12, 2021. [Online]. Available: https://github.com/osqzss/gps-sdr-sim.

[328] NASA, “Earthdata,” *NASA Earth Data*, 2021, Accessed: May 12, 2021. [Online]. Available: https://urs.earthdata.nasa.gov/.

[329] M. Markowski, “Gnuradio Mini Projects,” *Udel*, 2021, Accessed: May 12, 2021. [Online]. Available: http://udel.edu/~mm/gr/.

[330] P. Steve and J.-B. Patel, “Image Transfer and Software Defined Radio using USRP and GNU Radio,” *Proj. Rep.*, 2016, [Online]. Available: http://academic.csuohio.edu/yuc/mobile/mcproj/3p-PatelJordan.pdf.

[331] GNU Radio, “Simulation example: BPSK Demodulation,” *GNU Radio*, 2021. https://wiki.gnuradio.org/index.php/Simulation_example:_BPSK_Demodulation (accessed May 12, 2021).

[332] Mathuranathan, “Simulate additive white Gaussian noise (AWGN) channel,” *Gaussian Waves*, Jun. 2015, Accessed: Feb. 26, 2021. [Online]. Available:

https://www.gaussianwaves.com/2015/06/how-to-generate-awgn-noise-in-matlaboctave-without-using-in-built-awgn-function/.

[333] M. Viswanathan, “Digital Modulations using Python,” 2019, Accessed: Feb. 24, 2021. [Online]. Available: https://www.amazon.co.uk/Digital-Modulations-using-Python-Color/dp/1712321633/ref=sr_1_3?dchild=1&keywords=digital+modulation+python&qid=1614176971&sr=8-3.

[334] V. Godet and M. Durham, “User Guide,” *Bebop Drone*, 2009.

[335] K. H. Kindervater, “The emergence of lethal surveillance,” *Secur. Dialogue*, vol. 47, no. 3, Jun. 2016, Accessed: Dec. 07, 2020. [Online]. Available: https://www.jstor.org/stable/26294130?seq=1.

[336] N. Instruments, “USRP-2943 - NI,” *National Instruments*, 2020. https://www.ni.com/en-gb/support/model.usrp-2943.html (accessed Dec. 07, 2020).

[337] M. S. Allahham, M. F. Al-Sa’d, A. Al-Ali, A. Mohamed, T. Khattab, and A. Erbad, “DroneRF dataset: A dataset of drones for RF-based detection, classification and identification,” *Data Br.*, vol. 26, 2019, doi: 10.1016/j.dib.2019.104313.

[338] C. J. Swinney and J. C. Woods, “DroneDetect Dataset: A Radio Frequency dataset of Unmanned Aerial System (UAS) Signals for Machine Learning Detection and Classification,” *IEEE DataPort*, Jun. 2021, Accessed: Jun. 14, 2021. [Online]. Available: https://ieee-dataport.org/open-access/dronedetect-dataset-radio-frequency-dataset-unmanned-aerial-system-uas-signals-machine.

[339] Nuand, “bladeRF 2.0,” *Nuand*, 2021. https://www.nuand.com/bladerf-2-0-micro/ (accessed Apr. 26, 2021).

[340] I. T. Nassar and T. M. Weller, “A Novel Method for Improving Antipodal Vivaldi

Antenna Performance," *IEEE Trans. Antennas Propag.*, vol. 63, no. 7, pp. 3321–3324, 2015, doi: 10.1109/TAP.2015.2429749.

[341] A. M. De Oliveira, M. B. Perotoni, S. T. Kofuji, and J. F. Justo, "A palm tree Antipodal Vivaldi Antenna with exponential slot edge for improved radiation pattern," *IEEE Antennas Wirel. Propag. Lett.*, vol. 14, pp. 1334–1337, 2015, doi: 10.1109/LAWP.2015.2404875.

[342] Tindie, "Ultra-WideBand Vivaldi Antenna 800MHz to 6GHz+ from Hex and Flex ," *Tindie,* Accessed: May 26, 2021. [Online]. Available: https://www.tindie.com/products/hexandflex/ultra-wideband-vivaldi-antenna-800mhz-to-6ghz/.

[343] "Mavic Mini - Specifications - DJI," *DJI,* Accessed: May 27, 2021. [Online]. Available: https://www.dji.com/uk/mavic-mini/specs.

[344] "DJI Lightbridge 2 - Product Information - DJI," *DJI,* Accessed: May 27, 2021. [Online]. Available: https://www.dji.com/uk/lightbridge-2/info.

[345] DJI, "Mavic Pro - Product Information," *DJI,* Accessed: May 27, 2021. [Online]. Available: https://www.dji.com/uk/mavic/info.

[346] DJI Store Sofia, "What Is DJI OcuSync And How Does It Work?," *DJI Store Sofia,* Dec. 2019, Accessed: May 27, 2021. [Online]. Available: https://store.dji.bg/en/blog/what-is-dji-ocusync-and-how-does-it-work.

[347] DroneLabs CA, "Mavic Air 2s," *DroneLabs CA,* Accessed: May 27, 2021. [Online]. Available: https://www.dronelabs.ca/products/mavic-air-2s-1.

[348] J. Brown, "Parrot Disco: Features, Reviews, Specifications, Competitors," *My Drone Lab,* Accessed: May 27, 2021. [Online]. Available:

https://www.mydronelab.com/reviews/parrot-disco.html.

[349] DJI Best Drones, “DJI OcuSync 2.0: What is This FPV Transmission System?,” *DJI Best Drones*. http://djibestdrones.com/dji-ocusync-2-0/ (accessed Aug. 15, 2022).

[350] OpenStax CNX, “Short Time Fourier Transform,” *Digital Signal Processing: A User’s Guide*. https://cnx.org/contents/qAa9OhlP@2.44:PmFjFoIu@5/Short-Time-Fourier-Transform (accessed Dec. 07, 2020).

[351] A. Kumar and M. Chari, “Efficient Audio Noise Reduction System Using Butterworth Chebyshev and Elliptical filter,” *Int. J. Multimed. Ubiquitous Eng.*, vol. 12, no. 1, pp. 225–238, 2017, doi: 10.14257/ijmue.2017.12.1.19.

[352] Stanford University, “Welch’s Method,” *Stanford University*, 2020. https://ccrma.stanford.edu/~jos/sasp/Welch_s_Method.html (accessed Dec. 07, 2020).

[353] P. Lutus, “Software-Defined Radios,” *Arachnoid*, 2018.

[354] J. Daniel and J. H. Martin, “Logistic Regression,” *Speech Lang. Process.*, pp. 1–19, 2019.

[355] B. R. Mete and B. R. Ensari, “Flower Classification with Deep CNN and Machine Learning Algorithms,” *2019 3rd Int. Symp. Multidiscip. Stud. Innov. Technol.*, Oct. 2019, Accessed: Oct. 26, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/8932908.

[356] A. Pandey, “The Math Behind KNN.,” *Artificial Intelligence in Plain English*, Jan. 07, 2021. https://ai.plainenglish.io/the-math-behind-knn-7883aa8e314c (accessed Jun. 23, 2021).

[357] A. Al-Masri, “What Are Overfitting and Underfitting in Machine Learning,” *Towar. Data Sci.*, Jun. 2019, Accessed: Jun. 23, 2020. [Online]. Available:

https://towardsdatascience.com/what-are-overfitting-and-underfitting-in-machine-learning-a96b30864690.

[358] M. Sanjay, "Why and how to Cross Validate a Model?," *Towar. Data Sci.*, 2018, Accessed: Jun. 23, 2020. [Online]. Available: https://towardsdatascience.com/why-and-how-to-cross-validate-a-model-d6424b45261f.

[359] J. Brownlee, "A Gentle Introduction to k-fold Cross-Validation," *Mach. Learn. Mastery*, May 2019, Accessed: Jun. 23, 2020. [Online]. Available: https://machinelearningmastery.com/k-fold-cross-validation/.

[360] R. James, Gareth Witten, Daniela Hastie, Trevor Tibshirani, "An Introduction to Statistical Learning," p. 184, 2017.

[361] N. Jouppi, "Google supercharges machine learning tasks with TPU custom chip," *Google*, May 18, 2016.

[362] D. Sonagara and S. Badheka, "Comparison of Basic Clustering Algorithms," *Int. J. Comput. Sci. Mob. Comput.*, vol. 3, no. 10, pp. 58–61, 2014, [Online]. Available: www.ijcsmc.com.

[363] E. Yesilyurt, "What is the Difference Between Hierarchical and Partitional Clustering," *Machine Learning Turkiye Medium*, Mar. 26, 2022. https://medium.com/machine-learning-türkiye/what-is-the-difference-between-hierarchical-and-partitional-clustering-84d5a4ceb4c0 (accessed Aug. 21, 2022).

[364] D. Pfitzner, R. Leibbrandt, and D. Powers, "Characterization and evaluation of similarity measures for pairs of clusterings," *Knowl. Inf. Syst. 2008 193*, vol. 19, no. 3, pp. 361–394, Jul. 2008, doi: 10.1007/S10115-008-0150-6.

[365] R. Suwanda, Z. Syahputra, and E. M. Zamzami, "Analysis of Euclidean Distance and

Manhattan Distance in the K-Means Algorithm for Variations Number of Centroid K," *J. Phys. Conf. Ser.*, vol. 1566, no. 1, 2020, doi: 10.1088/1742-6596/1566/1/012058.

[366] T. m K. Atmiya, "Survey on Exiting Method for Selecting Initial Centroids in K-means Clustering Survey on Exiting Method for Selecting Initial Centroids in K-means Clustering," *Int. J. Eng. Dev. Res.*, vol. 2, no. 2, 2014.

[367] D. Arthur and S. Vassilvitskii, "k-means++: The Advantages of Careful Seeding."

[368] G. Hamerly and C. Elkan, "Alternatives to the k-means algorithm that find better clusterings," *Int. Conf. Inf. Knowl. Manag. Proc.*, pp. 600–607, 2002, doi: 10.1145/584792.584890.

[369] I. T. Jollife and J. Cadima, "Principal component analysis: a review and recent developments," *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.*, vol. 374, no. 2065, Apr. 2016, doi: 10.1098/RSTA.2015.0202.

[370] C. Ding, "K -means Clustering via Principal Component Analysis," 2004.

[371] C. Ding and X. He, "K-means clustering via principal component analysis," *Proceedings, Twenty-First Int. Conf. Mach. Learn. ICML 2004*, pp. 225–232, 2004, doi: 10.1145/1015330.1015408.

[372] B. Li and B. Li, "An Experiment of K-Means Initialization Strategies on Handwritten Digits Dataset," *Intell. Inf. Manag.*, vol. 10, no. 2, pp. 43–48, Feb. 2018, doi: 10.4236/IIM.2018.102003.

[373] D. and G. Dua, "UCI Machine Learning Repository," *University of California, Irvine, School of Information and Computer Sciences*, 2017. http://archive.ics.uci.edu/ml (accessed Jul. 10, 2022).

[374] K. Parra, "Travis AFB launches small unarmed aircraft initiative, first on Air Force

installation," *RAF Mildenhall News*, Dec. 17, 2020. https://www.mildenhall.af.mil/News/Article-Display/Article/2449840/travis-afb-launches-small-unarmed-aircraft-initiative-first-on-air-force-instal/ (accessed Jul. 10, 2022).

[375] D. Daly, "5 Major Ways Airports are Using Drones," *Consortiq*, 2022. https://consortiq.com/uas-resources/5-major-ways-airports-are-using-drones (accessed Jul. 10, 2022).

[376] Scikit Learn, "2.3. Clustering — scikit-learn 1.0.2 documentation," *Scikit Learn*, 2022. https://scikit-learn.org/stable/modules/clustering.html#clustering-evaluation (accessed Feb. 17, 2022).

[377] DJI, "Inspire 2 - Product Information - DJI," *DJI*, 2022. https://www.dji.com/uk/inspire-2/info#specs (accessed Jul. 27, 2022).

[378] Raspberry Pi, "Buy a Raspberry Pi 4 Model B – Raspberry Pi," *Raspberry Pi*, 2022. https://www.raspberrypi.com/products/raspberry-pi-4-model-b/ (accessed Jul. 27, 2022).

[379] DJI, "Mavic 3 - Specs - DJI," *DJI*. https://www.dji.com/uk/mavic-3/specs (accessed Aug. 22, 2022).

[380] DJI, "DJI Air 2S - Specs - DJI," *DJI*. https://www.dji.com/uk/air-2s/specs (accessed Aug. 22, 2022).

[381] SenseFly, "eBee X Technical brochure," [Online]. Available: www.sensefly.com/cameras.

[382] A. Rudd, "What is the difference between DSM2 and DSMX? ," *NBCCOMEDYPLAYGROUND*, Dec. 06, 2020.

https://www.nbccomedyplayground.com/what-is-the-difference-between-dsm2-and-dsmx/ (accessed Aug. 22, 2022).

[383] University of Bristol, “SWAN Protecting wireless networks from cyber-attacks receives 6.1 million funding boost,” Oct. 17, 2019.