NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

EXPLORING DETECTION OF UNMANNED AERIAL SYSTEMS ON 5G NETWORKS VIA MACHINE LEARNING

by

Alexander D. Gore

June 2023

Co-Advisors: Britta Hale
Patrick McClure

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** June 2023	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE EXPLORING DETECTION OF UNMANNED AERIAL SYSTEMS ON 5G NETWORKS VIA MACHINE LEARNING	**5. FUNDING NUMBERS**
6. AUTHOR(S) Alexander D. Gore	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.	**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

The development and implementation of 5G network technologies, including beamforming and sidelinking, will significantly improve the capabilities of adversaries performing drone operations beyond line of sight. This research explores methods for addressing the challenges presented by 5G for drone detection, such as when data is encrypted. The approach generates datasets for drone network traffic by capturing packets between a ground control station and a simulated drone. Then the individual communication flows are separated, and statistical fingerprints are constructed from the extracted temporal features: mean, median, and standard deviation of inter-arrival times, and packet direction ratio. These fingerprints are used to train and test a random forest classifier, which distinguishes drone traffic flows simulated over WiFi or Ethernet from normal 5G traffic flows with 99% accuracy and an F1 score greater than 98% in less than one tenth of a second. The classifier also detects drone traffic from data sent across a different transmission system than it was trained on with an F1 score greater than 97%. While due to tool limitations the drone data was not tested over 5G, detection aspects between drone data and other normal 5G data such as the data directional rate show promise regardless of transmission method. The proposed method's high performance and exclusive use of temporal features make it a promising direction to explore for 5G drone detection.

14. SUBJECT TERMS UAV, surveillance, drone, 5G, detection, machine learning, decision trees, packet analysis	**15. NUMBER OF PAGES** 69
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

EXPLORING DETECTION OF UNMANNED AERIAL SYSTEMS ON 5G NETWORKS VIA MACHINE LEARNING

Alexander D. Gore
Ensign, United States Navy
BS, San Diego State University, 2022

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
June 2023

Approved by: Britta Hale
Co-Advisor

Patrick McClure
Co-Advisor

Gurminder Singh
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The development and implementation of 5G network technologies, including beamforming and sidelinking, will significantly improve the capabilities of adversaries performing drone operations beyond line of sight. This research explores methods for addressing the challenges presented by 5G for drone detection, such as when data is encrypted. The approach generates datasets for drone network traffic by capturing packets between a ground control station and a simulated drone. Then the individual communication flows are separated, and statistical fingerprints are constructed from the extracted temporal features: mean, median, and standard deviation of inter-arrival times, and packet direction ratio. These fingerprints are used to train and test a random forest classifier, which distinguishes drone traffic flows simulated over WiFi or Ethernet from normal 5G traffic flows with 99% accuracy and an F1 score greater than 98% in less than one tenth of a second. The classifier also detects drone traffic from data sent across a different transmission system than it was trained on with an F1 score greater than 97%. While due to tool limitations the drone data was not tested over 5G, detection aspects between drone data and other normal 5G data such as the data directional rate show promise regardless of transmission method. The proposed method's high performance and exclusive use of temporal features make it a promising direction to explore for 5G drone detection.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

1 Introduction **1**

1.1 Motivation 1

1.2 Thesis Organization 2

2 Background **3**

2.1 UAS Detection Methods 3

2.2 Network Traffic Flows 5

2.3 Related Work 7

2.4 Cellular Drones 9

2.5 5G 11

3 Methodology **15**

3.1 Approach 15

3.2 Equipment 16

3.3 Normal Traffic Dataset Selection 23

3.4 Drone Traffic Dataset Collection 23

3.5 Feature Extraction 30

3.6 Classifier Development 32

4 Results **35**

4.1 Metrics 35

4.2 Standard Evaluation 36

4.3 Cross-System Evaluation 41

4.4 Feature importance analysis 43

4.5 Limitations. 45

5 Conclusion and Future Work **47**

5.1 Conclusion. 47

5.2 Future Work 47

List of References **49**

Initial Distribution List **53**

List of Figures

Figure 2.1 Illustration of Flows for Different Systems 6

Figure 2.2 Illustration of WiFi vs. Cellular for Drone Communications. . . . 9

Figure 2.3 Illustration of the Gaps in Coverage for Drones with Current Cellular Infrastructure. 11

Figure 2.4 Illustration of 5G Sidelinking Enabling Direct and Indirect Signal Relaying. 13

Figure 3.1 Amarisoft AMARI Callbox Mini. 17

Figure 3.2 Raspberry Pi 4 with Waveshare SIM8200EA-M2 5G HAT. 18

Figure 3.3 Boeing Insitu ScanEagle. 19

Figure 3.4 3D Robotics IRIS Multicopter. 20

Figure 3.5 DJI Air 2S 21

Figure 3.6 PX4 Simulation Environment. 26

Figure 3.7 An Example Data Collection Flight Shown in QGroundControl. . 27

Figure 3.8 Gazebo Simulating an Example Data Collection Flight. 28

Figure 4.1 k-Fold Cross Validation Confusion Matrices for WiFi Drone and Normal 5G Traffic. 38

Figure 4.2 k-Fold Cross Validation Confusion Matrices for Ethernet Drone and Normal 5G Traffic. 39

Figure 4.3 k-Fold Cross Validation Confusion Matrices for Combined WiFi and Ethernet Drone, and Normal 5G Traffic. 40

Figure 4.4 Confusion Matrix Averaged across 20 Runs of Training on WiFi and Testing on Ethernet. 42

Figure 4.5 Confusion Matrix Averaged across 20 Runs of Training on Ethernet and Testing on WiFi. 43

Figure 4.6 Mean Decrease Impurity (MDI) Feature Importance Percentages. 44

List of Tables

Table 3.1 Drone Traffic Packet Capture Metadata. 29

Table 3.2 Dataset Classes. 32

Table 4.1 k-Folds Cross Validation Scores (Percentages). 37

Table 4.2 Cross-System Evaluation Scores from 20 Runs (Percentages). . . . 42

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

3GPP 3rd Generation Partnership

BLOS Beyond Line of Sight

CSV Comma Separated Values

DS Differentiated Service

GCS Ground Control Station

gNB gNodeB

GPIO General Purpose Input/Output

GPS Global Positioning System

GSM Global System for Mobile Communications

HAT Hardware Attached on Top

HITL Hardware in the Loop

IP Internet Protocol

ISR Intelligence Surveillance and Reconnaissance

LTE Long Term Evolution

MAC Media Access Control

MDI Mean Decrease Impurity

NIDS Network Intrusion Detection System

OSI Open Systems Interconnection

pcap Packet Capture

PX	Pixhawk
QGC	QGroundControl
RAM	Random Access Memory
RF	Radio Frequency
RMS	Root Mean Square
SBC	Single Board Computer
SDR	Software Defined Radio
SIM	Subscriber Identity Module
SITL	Software in the Loop
TCP	Transmission Control Protocol
UAS	Unmanned Aerial System
UAV	Unmanned Aerial Vehicle
UDP	User Datagram Protocol
UE	User Equipment
UMTS	Universal Mobile Telecommunications System
USB	Universal Serial Bus

CHAPTER 1:
Introduction

Both military and commercial grade Unmanned Aerial Vehicles (UAVs), commonly called drones, have proved to be an integral part of the war in Ukraine, with roles extending beyond just surveillance and reconnaissance. They have also demonstrated powerful capabilities for launching kinetic strikes [1]. Owing to their low cost to obtain and operate, drones are not only weapons of war, operated by militaries, but widely available to the general public. In this arena, they have become a powerful tool for taking photos and videos of restricted areas, eavesdropping on private signals and networks, stalking individuals, and disrupting government and commercial activities. Because of their small size, maneuverability, and ability to quickly enter and exit a location, it is extremely difficult to prevent unauthorized drones from performing malicious activities. Interdiction systems have been developed, however any interdiction system first requires a robust and reliable detection system.

The numerous benefits offered by drones suggest that they will be instrumental in the smart cities of the future [2], a vision that the engineers behind 5G describe as "the communications system that can finally achieve what has long been promised—anyone anywhere can get in touch with whoever or whatever" [3]. This vision of the future is not without dangers, from malicious operators, or simply from a malfunctioning drone striking people, property, or infrastructure. Several research groups have noted the need for technology that can detect, track, and interdict unauthorized or malicious drones in the smart cities and smart battlefields of the future [2], [4]. This thesis examines the use of machine learning classifiers based on packet flow analysis as a method for detecting drones, using methods applicable to detecting drones operating on 5G networks.

1.1 Motivation

The vast majority of commercially available drones use a direct radio link, often WiFi. These point-to-point connections require a relatively uninterrupted line of sight between the drone and its controller, thereby limiting the range at which the drone can be operated. Cellular infrastructure supports connections that span the globe, but these connections are

not yet suitable for use with drones. 5G, the next generation cellular network standard, implements several technologies that will allow future operators to fly drones anywhere in the world from anywhere else in the world. Specifically, cellular infrastructure and *sidelinking* capability allows for a drone's signal to be passed along by multiple devices, overcoming the limitations of a single transmitter's finite range, and eliminating dependence on controller-to-drone direct linking. The challenges involved in detecting and tracking 5G–enabled drones will therefore be far more complex than with the current generation of drones. This thesis explores the first steps towards addressing them.

Techniques for detecting drones based on identifying the communication traffic sent between a drone and its controller, collectively called a UAS have been demonstrated to be effective [2], [5], [6]. However, they have all been targeted at the current standard of WiFi controlled drones. There has not yet been research on the extent to which these techniques are suited for detection of future 5G controlled drones.

There has been an increased focus in the literature on the effectiveness of WiFi detection when UAS communications are encrypted. However, most detection methods still consider the length of captured packets, and attributes of their Internet Protocol (IP) headers. Both of these features are obscured by the encryption implemented by 5G [7]. The objective of this research is to determine the effectiveness of exclusively time-based packet flow analysis on non-5G UAS traffic alongside the normal traffic generated by 5G-enabled devices by generating drone traffic datasets, performing temporal feature extraction, and using machine learning classifiers for detection, so as to explore detection of encrypted 5G drone communications in future 5G smart environments.

1.2 Thesis Organization

The remainder of the thesis is organized as follows: Chapter 2 provides an overview of drone detection methods, UAS communications, 5G technologies, and discusses past work in this field; Chapter 3 describes the methodology for collecting and processing data, and developing machine learning models; Chapter 4 evaluates the machine learning models results; Chapter 5 analyzes lessons learned and outlines future work in the domain.

CHAPTER 2:
Background

There have been a number of methods explored for interdiction of unauthorized drones. These include kinetic strikes, as well as jamming of communication signals, to cause the operator to lose control of the drone [8]. For drones that are not remote controlled, but Global Positioning System (GPS) guided, a spoofed GPS signal can be fed to the drone to neutralize it [4], [9]. Experiments have been done using nets to catch drones or force them to crash. Some of these used a gun that fires the net at the drone, while others have used another drone to carry the net and ensnare the unauthorized drone [4]. Finally, police in the Netherlands made waves in 2016 by training eagles to attack unauthorized drones and bring them back to the authorities, Only a year later they retired the anti-drone eagles, stating the eagles were often distracted from their task, expensive to keep, and rarely used [10]. An automated system circumvents many of these disadvantages.

Some background is necessary to understand later parts of the thesis. This chapter will provide an overview of the categories of drone detection, the various wireless communications protocols used in UASs, which include one or more drones and one or more Ground Control Stations (GCSs), and the machine learning techniques that will be employed.

2.1 UAS Detection Methods

As stated in [2], "the first step towards the neutralization of a threat is a robust and reliable detection," after all, the ability to interdict a target is useless without the ability to detect and track the target. Current drone detection techniques largely fall into five categories: acoustic, visual, radar, Radio Frequency (RF), and traffic flow detection. Traffic flow detection is also known as WiFi detection in the literature, based on the specific subcase of the WiFi protocol vs the general category of network traffic protocol flow analysis.

2.1.1 Acoustic Sensors

Acoustic detection leverages the telltale sound that is produced by a drone's electric motors and propellers. An array of microphones can be set up around a perimeter to listen for

this sound and can even triangulate the location of a drone once it is heard [11]. Acoustic detection works in the dark and does not require a direct line of sight. However, it is ineffective in noisy environments, such as urban or industrial locations, which is what many of the targets of drone surveillance and attacks tend to be. An acoustic detection system is also subject to a high degree of interference from "friendly" drones, and cannot be mounted onboard a drone for the same reason.

2.1.2 Computer Vision

Computer Vision detection systems utilize cameras in the visual and thermal spectrum to monitor a particular area, and computer vision techniques to identify a UAV in frame. Once the UAV is identified, the system can track its location across subsequent frames [12]. Visual detection has several drawbacks, mainly that a camera can only see what is in its line of sight, meaning a video detection system will need a large number of cameras for a location that has any natural or man-made obstacles. The system's ability may also be severely impacted by darkness, rain, fog, or dust [4]. This is a vulnerability that could be intentionally exploited by an adversary with the use of smoke or other obstructions.

2.1.3 Radar

Radar detection systems can avoid a number of visual detection's drawbacks, as they are not as affected by visual obstructions and can even be configured to work without a direct line of sight. Also, they do not require any active RF signals from the target drone. The relatively low radar cross section of most commercial UAVs, particularly quadcopters, as well as the dielectric similarities between the plastics used for their construction and air, makes radar traditional anti-aircraft radar ineffective. UAVs can be detected by the micro-Doppler signatures produced by their propellers, with limited effectiveness [13].

2.1.4 Radio Frequency

RF detection listens for the control and video signals between the drone and its controller, it is typically performed by Software Defined Radios (SDRs) and does not require a line of sight. A method for gaining access to the frequency hopping sequence employed by a commercial drone is described in [14], which also notes that their system opens the door for jamming or hijacking of the signal. The authors of [4] note that with the proliferation of

SDRs, adversarial drone operators can configure their radio links in nearly any frequency band, making RF detection much more challenging.

2.1.5 Communication Flow Features

Detection can also take place by identifying protocol communication features that are fairly unique to UASs. WiFi is an example protocol. WiFi detection is the name commonly given to traffic flow analysis based drone detection systems, as the majority of current commercial UASs have historically used WiFi for communication [2], but others may also be used for UAS communication [15]. As this thesis will examine the use of traffic flow analysis for detection of drones, the process will be referred to as traffic-flow-based detection or flow-based detection. Predating the field of drone detection, flow based detection is used in cybersecurity in Network Intrusion Detection Systems (NIDSs) [16]. Because both traffic flow and RF detection involve capturing a drone's communication signals, neither can detect drones operating in an autonomous state. There are, however, several key distinctions between the two. RF detection analyzes signals at the physical layer, while flow-based detection analyzes flows of packets at the network, or IP, layer of the Open Systems Interconnection (OSI) model or higher [8].

2.2 Network Traffic Flows

In the field of computer networks, *traffic* refers to the set of signals sent over a network over a certain time domain. This thesis uses the terminology of the OSI model, which uses *packet* to refer to a protocol data unit at the network layer. There are many different definitions for the term "flow," RFC 3917 defines a flow as:

> "A set of IP packets passing an observation point in the network during a certain time interval. All packets belonging to a particular flow have a set of common properties. Each property is defined as the result of applying a function to the values of:
>
> 1. One or more packet header field (e.g., destination IP address), transport header field (e.g., destination port number), or application header field (e.g., RTP header fields)

2. One or more characteristics of the packet itself (e.g., number of MPLS labels, etc.)
3. One or more of fields derived from packet treatment (e.g., next hop IP address, the output interface, etc.)

A packet is defined to belong to a flow if it completely satisfies all the defined properties of the flow." [17]

In this thesis, the common set of properties shared by packets in a flow will be the set of source and destination IP addresses, making a flow the bidirectional set of packets between two IP addresses.

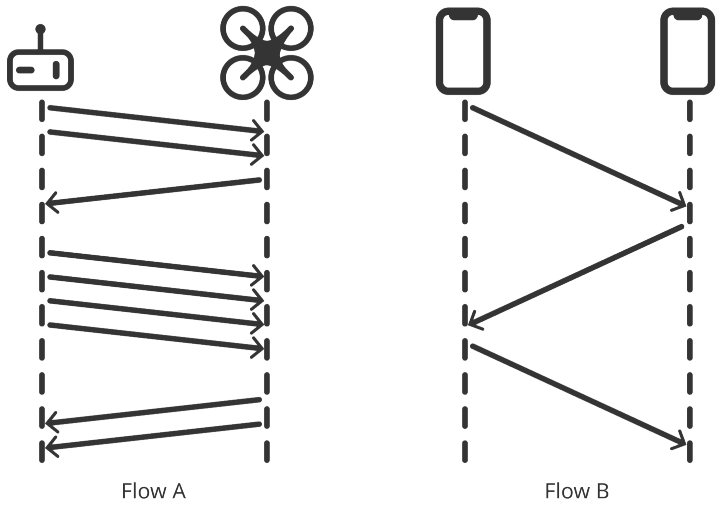

Figure 2.1. Illustration of Flows for Different Systems

As shown in Figure 2.1, different flows have different characteristics in the timing of their packets. One may be slower, while another may be faster, one may be steady, while another is bursty. One may be symmetric, while another is asymmetric. This information can be used to fingerprint the different types of flows. The unique fingerprints of drones can be used to identify them on the network.

2.3 Related Work

Several papers in the literature describe how to build a system that can perform flow based detection over WiFi. The first paper to implement one such system, [5], was published in 2018. The system consisted of a laptop using *Wireshark*, a network packet sniffer, to capture traffic on nearby WiFi channels. The captured packets were then exported to a Packet Capture (pcap) file, which was analyzed by partitioning the flows over a predefined time window, and extracting features to determine a statistical fingerprint for each flow. These fingerprints were then fed into a machine learning classifier, which would detect the unique fingerprints of drone traffic flows. The authors defined flows as containing packets with the same two Media Access Control (MAC) addresses as source and destination, regardless of the direction. This system extracted the following features for construction of each flow's statistical fingerprint:

1. Total number of packets
2. Total duration of the flow
3. Average packet length
4. Root Mean Square (RMS) of packet length
5. Average packets interarrival time
6. RMS of packets interarrival time
7. Number of embedded packets
8. Number of packets with specific Differentiated Service (DS) status flags

The authors updated their model to be able to detect drones that were disabling their video feeds in an attempt to avoid detection, which they called a *stealth attack*, in 2019 [2].

Also in 2019, another group published results of a WiFi flow based detection system that was able to not only detect drones, but determine if the drone was actively flying or not [18]. In 2020 they updated their model to be able to detect the specific control inputs being sent to the drone [6]. Instead of only extracting the average and RMS values of a flow, this model extracted a matrix, containing the following extracted features for every individual packet in the flow.

1. Packet size
2. Interarrival time between the n-th packet and the $(n+1)$-th packet.

3. Mean packet size over a predefined time window W
4. Mean interarrival time over W
5. Standard deviation of packet size over W
6. Standard deviation of interarrival time over W

The authors designed their model to detect encrypted drone traffic, hence their omission of IP header flags as a feature. However, they still use packet size as a feature, despite the ease with which it can be modified by an adversarial operator to avoid detection. Packet size can also be obscured by some forms of encryption. In [6], they address the increased difficulty an adversary would have with modifying timing. Chief among these is the the fact that artificial delays added to packets may help avoid detection. However, this will also degrade performance and control of the drone, and more packet delay leads to negative impact on performance. The authors also note that no commercial products support implementing packet delays.

2.3.1 Random Forest Classifiers

In both [2] and [6], random forest classifiers were demonstrated to be very effective for flow based detection. Random forests are desirable since they build on decision tree classifiers by considering many decision trees and utilizing the "wisdom in crowds." This thesis will employ random forest classifiers for these reasons, and because they are more interpretable than competing methods, such as neural networks. Decision trees, which make up random forests, are interpretable because they make classifications by asking a series of yes or no questions about a provided sample until they reach a decision, and these questions can be reviewed to determine which metrics the classifier is using.

2.3.2 Technological Advancements

Modern commercial amateur drones differ from the remote controlled planes and helicopters of the past for several reasons, technological improvements have enabled higher payloads and longer flight times, and onboard computers allow for semi-autonomous flight. These advancements have created drones that require far less knowledge and expense than their precursors, lowering the barrier to entry. Concurrent advances in video cameras and wireless technology have allowed for high-definition video to be transmitted in real time from drones, creating a commercial and hobbyist use-case for drones: aerial photography and

videography, drastically increasing the number of unmanned aircraft in the skies, and necessitating new technologies for regulating their use [19].

2.4 Cellular Drones

Stated explicitly in [5], "WiFi-based detection methods rely on the fact that a large majority of commercial low-cost amateur drones are controlled and/or transmit video streaming on WiFi bands." WiFi, like the hobbyist radio remote control systems that came before it and the new proprietary transmission systems created by drone manufacturers (e.g., DJI's *Ocusync*), still utilize point-to-point radio links and point-to-point communication security protocols. Ultimately, cellular connections offer a number of advantages, and though they have yet to be implemented in the commercial space, 5G is being designed to enable widespread cellular drone connections [20]. Similar to how cellular networks allow users to make phone and video calls Beyond Line of Sight (BLOS), and even between continents, they could be employed to control drones BLOS all over the world, providing virtually unlimited range and benefiting from the cellular infrastructure already in place.

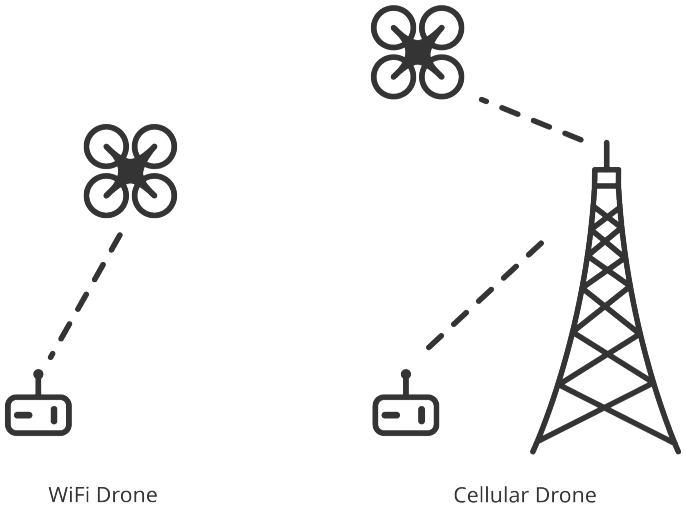

Figure 2.2. Illustration of WiFi vs. Cellular for Drone Communications.

In the literature, there are a number of experiments on the feasibility of cellular connected drones [19], [21], [22], although several factors have inhibited cellular connected commercial drones from entering the commercial space. Cellular technologies of the past such as Global System for Mobile Communications (GSM) and Universal Mobile Telecommunications System (UMTS) (i.e., 2G and 3G, respectively) lacked sufficient bit rate and had too high latency for effective drone control or video transmission [19]. More current cellular technologies, e.g. LTE and 5G, will be looked at next.

2.4.1 Limitations of LTE

4G Long Term Evolution (LTE) increased bit rates and lowered latency times, yet like the networks before it, LTE was developed and configured for terrestrial devices, leading to issues for flying devices. First, since most users are on the ground, cellular base station antennas are aimed downwards. Though up-tilted antennas have begun to be deployed for providing service to passengers in aircraft, there aren't enough to provide consistent coverage to drones, which fly far lower [23]. This creates both "dead zones" where there is no coverage, and a large amount of *handover* (i.e., switching a device from one base station to another) which can disrupt communications. As demonstrated by [23], these handovers are often unnecessary. However, they are executed because the current cellular implementations do not take into account the unique location of drones, which can have a direct line of sight to base stations that are too far away to provide sufficient service, something that does not happen with terrestrial devices [23].

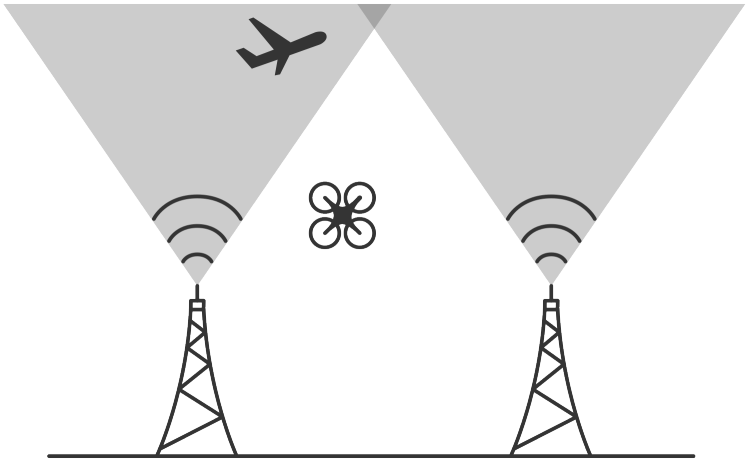

Figure 2.3. Illustration of the Gaps in Coverage for Drones with Current Cellular Infrastructure.

2.5 5G

Supplanting 4G LTE, 5G is the next generation standard for cellular networks. 5G offers upload and download speeds that are far superior to 4G networks [24]. The high data rate of 5G also provides for much lower latency in UAS communications, allowing for much more precise drone control. This benefit is compounded in systems with multiple drones operating in conjunction, and or control of drones from much longer distances, including control far beyond visual range. The low latency of 5G also allows for computationally intensive tasks, such as computer vision, to be offloaded to more powerful terrestrial computers, thereby decreasing drone weight and power draw and increasing flight time, while increasing drone "intelligence" [22]. One new technology that will enable 5G to better support drones is *beamforming*. Beamforming is the technique of creating constructive and destructive interference with multiple antenna elements to "aim" the signal in a particular direction, instead of having it radiate in all directions equally. With beamforming the base station antenna does not need to be tilted down for terrestrial users, or up for aerial users, as the signal can be precisely targeted in the specific direction of the particular device to which it is currently transmitting. 5G base stations will consist of a large array of antennas, which

can leverage advanced beam-management algorithms to specifically to provide service to drones [20]. In addition to beamforming, 5G networks are also being built with a higher concentration of base stations, and drone support has been specifically outlined as one motivation for these design choices [20], [21].

2.5.1 Sidelinking

Perhaps the most powerful technology that 5G can bring to bear for drone connections is *sidelinking*. Sidelinking was announced by the 3rd Generation Partnership (3GPP) in Release 16 of the 5G standard, and it outlines how a user device, commonly referred to as a User Equipment (UE), can function as a base station (gNodeB (gNB) in the 5G lexicon). According to the engineers cultivating the technology, "the 5G vision has always been this unifying platform," not just a cellular network to connect phones, but a network for connecting anything wireless [25]. Sidelinking was first envisioned as allowing vehicles to communicate directly with each other, but is not restricted in its application. It can theoretically be used for any two devices to communicate with each other without going through a base station, rendering itself a fundamental part of 5G alongside downlink and uplink [26]. As depicted in Figure 2.4, 5G sidelinking allows for not only a communication between two devices, but it can also be used to relay packets from base stations through UEs that are in-coverage to UEs that are out-of-coverage, with provisions for continuing this relay through multiple out-of-coverage UEs [26].

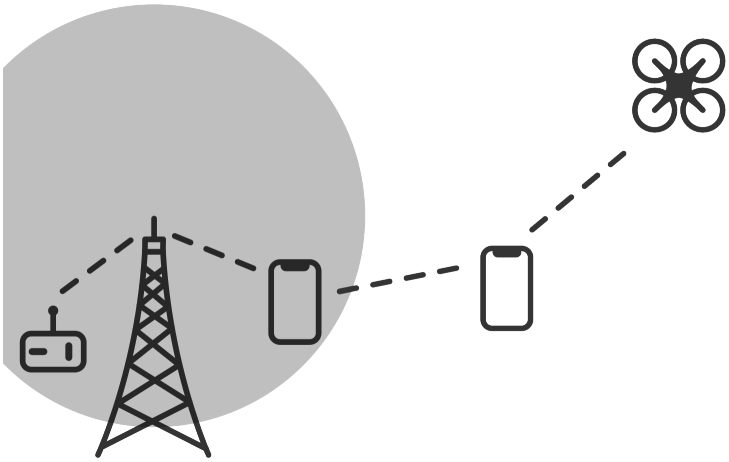

Figure 2.4. Illustration of 5G Sidelinking Enabling Direct and Indirect Signal Relaying.

This invites the capability to connect one 5G device with any other 5G device in the world as long as there is an unbroken chain of 5G devices between it and a base station. This, coupled with the far superior bit rate and latency compared to LTE, produces powerful new capabilities for commercial drones. The mesh-style networks made possible by 5G also creates new challenges for detection and interdiction of unauthorized drones, as a physical search for a pilot is no longer limited to the immediate area of the drone. Any signals between drone and pilot also must now be traced back through each hop of the relay to the original sender, instead of just being traced back a single hop. Sidelinking introduces challenges for capturing UAS signals, particularly those that are not routed through a base station, because the network becomes less centralized. Despite the challenges in collection, the analysis process, which is the focus of this research, is unaffected by sidelinking.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 3: Methodology

This chapter introduces the approach, equipment used, and the steps performed for generating a dataset of drone communications, extracting the important features, and training a random forest classifier to detect them.

3.1 Approach

Section 2.1 discussed several different approaches to drone detection, of these methods, this research will pursue traffic-flow-based detection. This method was chosen because it is not impacted by environmental factors (e.g., time of day, weather, ambient noise) as severely as other methods discussed. The approach taken differs from those described in [6] and [2], as only temporal features will be analyzed, to improve detection of encrypted drone traffic. The steps performed in this research are as follows: First drone traffic and normal traffic will be collected, then the individual traffic flows will be separated, and time-based features will be extracted from the flows to construct temporal statistical fingerprints. Finally, these statistical fingerprints will be used to train and test machine learning models to determine their effectiveness for distinguishing drone traffic from normal traffic. The random forest classifier that performs drone detection takes a feature vector that describes a particular flow as input, and outputs a prediction of if that flow is drone traffic or normal traffic. The feature vectors are constructed by separating the traffic captured in a particular time window, and extracting the following features: The number of packets sent during the window, the mean, median, and standard deviation of the inter-arrival times of those packets, and the packet flow direction ratio. These features are the only information about a given traffic flow that the classifier receives.

3.1.1 Constraints of the approach

Unfortunately, the available 5G equipment tested was not capable of providing the required data; this is discussed in more detail in Section 3.4. Thus, no actual 5G drone traffic was collected or analyzed, removing the straightforward approach of actually flying or simulating a drone on a 5G network with real or simulated 5G traffic on the same network. Instead,

several workarounds had to be implemented. First, simulated drone traffic was transmitted over WiFi and Ethernet, while the normal traffic it was detected against came from a 5G dataset, since the kind of traffic commonly seen on 5G networks is what a 5G drone detector will need to filter out.

It must be noted that because the drone traffic and the normal traffic were sent over different transmission systems, it is possible that the classifier is making classifications based on 5G vs. not-5G, instead of drone vs. not-drone. This concern is addressed by examining the features the model places the most importance in, and how they are affected by 5G. However, this research does not seek to provide a comprehensive analysis of methods for 5G drone detection, and it does still provide insights regarding possible directions for developing such methods, including an initial exploration into the hardware, software, and steps required to perform temporal flow-based detection, as well as first insights to the ability of machine learning classifiers to use temporal statistical fingerprints to differentiate different types of traffic, even if the use of different transmission systems prevents this differentiation from being fully drone vs. not-drone.

3.2 Equipment

This section will introduce the 5G equipment that was explored, the drones that are discussed and used for dataset generation, and the software to be used for drone control and simulation.

3.2.1 5G Equipment

The following 5G equipment was investigated for capturing 5G traffic, however due to limitations in the Amarisoft Callbox Mini's ability to capture encrypted 5G traffic, the Raspberry Pi's ability to run the drone control software, and the Waveshare HAT's ability to forward traffic to and from a second computer, none of this 5G equipment was used for the final data collection. However, this equipment may provide some direction for potential methodology in follow-on work. The particular 5G traffic capture attempts performed in this research are described in Section 3.4.

AMARI Callbox Mini

Built by Amarisoft, a French telecommunications software company founded in 2012, the Callbox Mini is marketed as "the 4G/5G network on your desk" [27] and functions as a 3GPP compliant gNB. The Callbox Mini supports up to 500 UEs, 200 Mbps downlink and 75 Mbps uplink, 2x2 MIMO, and supports 4G LTE and 5G connections [27]. The Callbox has an integrated computer running a Fedora Linux operating system. The Callbox Mini was successfully used to create a 5G network, unfortunately, the integration between the SDR and integrated computer does not allow for the collection of encrypted traffic, which is a limitation discussed in Section 3.4.

Figure 3.1. Amarisoft AMARI Callbox Mini. Source: [27].

Raspberry Pi 4 with Waveshare SIM8200EA-M2 5G HAT

The Raspberry Pi 4 is a Single Board Computer (SBC) computer with a 64-bit quad-core ARM processor, 4 GB of Random Access Memory (RAM), and four Universal Serial Bus (USB) 3.0 ports, supporting WiFi, Bluetooth, and Ethernet, connectivity [28]. The Raspberry Pi also has a 40 pin General Purpose Input/Output (GPIO) header, commonly used for attaching sensors, motors, or other external hardware. The SIM8200EA-M2 5G Hardware Attached on Top (HAT) gives the Raspberry Pi 5G connectivity, it supports 3G, 4G, and 5G, has a USB 3.1 port, 3GPP AT command support, and one Subscriber Identity

Module (SIM) card slot [29]. Figure 3.2 shows the the SIM8200EA-M2 HAT attached to the Raspberry Pi 4 in the provided enclosure, which securely holds the two together and has mount points for five antennae. The HAT provides 5G network connectivity to the Raspberry Pi, allowing it to connect to the Amarisoft Callbox Mini creating the private 5G network that was used to test forwarding and simulating drone traffic over 5G.

Figure 3.2. Raspberry Pi 4 with Waveshare SIM8200EA-M2 5G HAT and Enclosure. Source: [29].

3.2.2 Drones

Boeing Insitu ScanEagle

Built by Insitu, a Boeing Company, ScanEagle is an agile Intelligence Surveillance and Reconnaissance (ISR) UAV platform used by the U.S. military on land and at sea. ScanEagle has been deployed around the world since 2004, and is operated by military, government, and commercial organizations. It has a length of 5.6 feet, wingspan of 10.2 feet, endurance of up to 18 hours, ceiling of 19.500 feet, and a cruise speed of 50–60 knots. Notably, ScanEagle

can be integrated with open-architecture ground control stations [30]. ScanEagle traffic was originally analyzed and replayed over the Amarisoft Callbox 5G network by a Raspberry Pi with a Waveshare HAT, as the first dataset generation approach.

Figure 3.3. Boeing Insitu ScanEagle. Source: [30].

3D Robotics IRIS

Quadcopters are among the most common commercial and hobbyist drones, as advances in electric motors have made the four simple motors that power a quadcopter more cost effective than the single complex motor that lifts a conventional helicopter, and advances in computers have made the complex flight controllers required for stability to be cost and weight effective. Able to take off and land vertically with no extra equipment and little open space, as well as to hover, quadcopters are preferred to fixed wing drones by commercial and hobbyist users [31]. With the first models shipping in 2013, the IRIS quadcopter was developed and built by 3D Robotics, a Silicon Valley startup that went out of business in 2016 [32]. IRIS measures 1.8 feet from motor to motor, endurance of 10-15 minutes, and can fly autonomously by following pre-planned or real time waypoints. IRIS was developed around an open source Pixhawk flight controller and autopilot system [33], it is now the default airframe in the Gazebo simulation environment included with the PX4 autopilot software [34], further described in subsection 3.2.4. When purchased with a camera and

gimbal, IRIS costed $1269. No physical 3DR IRIS drone was operated in this research, however all drone simulations performed simulated an IRIS drone, and all drone traffic used for classifier training and testing was sent to to or from this simulated IRIS.

Figure 3.4. 3D Robotics IRIS Multicopter. Source: [33].

DJI Air 2S

In the decade since 2013, drones have become far more widespread. Chinese manufacturer DJI has been at the forefront of lowering the barrier for entry in terms of technical knowledge required and cost [32]. The latest entry level drone from DJI, the Mini 3, costs only $469, folds down to 6x4 inches, has an endurance of 51 minutes, and transmits video up to 10 kilometers [35]. This enormous leap in affordability and capability demonstrates the rapidly growing threat that drones pose. The Air 2S was not used in any part of this research, but is discussed to provide a frame of reference for the current state of affordable consumer drones.

Figure 3.5. DJI Air 2S. Source: [35].

3.2.3 Drone Control Software

There are several discrete software elements that make up a drone's control system. This thesis examines drones controlled by open-source software in the Pixhawk ecosystem, namely QGroundControl (QGC), MAVLink, and PX4.

MAVLink

Released in 2009, MAVLink is a messaging protocol for drone communications developed by Lorenz Meier for use on a custom flight controller developed by his Pixhawk team. Since being open sourced, MAVLink has been adopted by a number of autopilot projects, including AutoQuad and Ardupilot [36], [37]. Additionally, a project called RosettaDrone provides a wrapper to allow DJI drones, like the Air 2S described previously, to use MAVLink communications, for use with custom ground control stations like QGroundControl [38]. MAVLink uses both Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) connections, depending on the type of traffic being carried (e.g., control, video, etc.) and how it is configured. MAVLink protocol traffic is the only form of drone traffic that is explored in this research.

QGroundControl

QGroundControl is the open source ground control software developed by the Pixhawk team, providing flight control and mission planning for drones running flight control software that

communicates with the MAVLink protocol, and runs on Windows, OS X, Linux, iOS, and Android [39]. QGC provides a simple point and click interface for setting points for a drone to fly to or orbit around, shown in Figure 3.7.

PX4

The fourth iteration of the Pixhawk (PX) flight control software, PX4 is an open source autopilot for UAVs as well as ground, surface, and subsurface drones [34]. PX4 has a large open source community, including academics, hobbyists, and industry [36].

3.2.4 Drone Simulation

Simulation is important for development of new drone control software, as it allows developers to find bugs and see how their code works in a simulated world, where a malfunction will not endanger anyone or damage the drone it's running on. There are two types of drone simulation, Software in the Loop (SITL) simulation, where the software runs on a computer, and Hardware in the Loop (HITL) simulation, where the software is actually running on the flight controller [34]. For the purposes of this thesis, SITL simulation will be used. SITL is useful for experimentation as it requires no specialized hardware.

PX4 SITL and Gazebo

PX4's SITL simulation can be used to control simulated drones in Gazebo, a 3D simulation environment for autonomous vehicles. The PX4 simulation connects via MAVLink to a ground controller and a simulation environment, both of which can be on the same computer as the PX4 SITL instance, or another computer on the network.

MSI Prestige 15 A10SC Laptop

In this research, the PX4 and Gazebo simulations and QGroundControl were run on MSI Prestige 15 A10SC Laptops running Ubuntu 20.04.5. Both laptops used have an Intel Core i7 U-Processor, an NVIDIA GeForce GTX 1650 Mobile GPU, 32 GB of RAM. As they do not have Ethernet ports, MSI Ethernet to USB-C adapters were used for wired connections. Two of these MSI Prestige 15 laptops are used to run the drone simulation used for drone traffic dataset generation.

3.3 Normal Traffic Dataset Selection

The first step towards building a machine learning model is constructing a dataset. In order for a system to identify drone traffic, there must also be training with non-drone traffic. 5G–enabled drones operate on the same networks as normal users so a 5G drone detector should be trained with drone traffic, and "normal traffic." A publicly available dataset was used because of the limitations of the available 5G equipment. Due to how new 5G is, there are not a large number of 5G datasets, and the majority of them measure signal strength and other metadata. Fortunately, Idaho National Laboratory recently published a paper exploring machine learning assisted network security for 5G, and publicly released their 5G dataset, containing packet captures of normal traffic and attack traffic [40]. For the purposes of this thesis, only the normal traffic is considered. Idaho National Laboratory's dataset is well suited to this research, as it contains packet capture files containing normal internet traffic on a 5G network. The recorded traffic includes "streaming YouTube videos, accessing 500 popular websites, downloading files via FTP, mounting a SAMBA share and downloading files from it, and having a conference call via Microsoft Teams" [40]. This sort of activity is what one would expect to see on a public 5G network, and is thus referred to as normal traffic.

Two captures were selected from the dataset, totalling eight hours of traffic and 5.4 million packets in total. Only IP, (i.e., network-layer packets and above were exported for use, since packets at lower layers are unable to be routed across large interconnected networks, such as commercial cellular networks. Wireshark was used to export the packet data to the Comma Separated Values (CSV) format. This dataset was combined with the captured drone data to form the standard training and testing datasets, functioning as the "background" traffic that the traffic-flow-based detection system would have to pick drone traffic out of. Thus, the "normal 5G traffic" part of the dataset used was not simulated, but consists of real traffic captured on a real 5G network by researchers at Idaho National Laboratory [40].

3.4 Drone Traffic Dataset Collection

As of yet, there are no publicly available datasets of 5G drone traffic, likely stemming from the limited availability of 5G enabled drones. In addition, there are no publicly available datasets containing packet captures of drone traffic over any transmission system. Therefore

one research effort of this thesis is to investigate generating a suitable dataset. Several dataset generation methods are considered in this section.

3.4.1 Attempts to replay prerecorded drone traffic over 5G

The first method explored for generating a dataset was to take packet captures recorded by the ground control station controlling a ScanEagle drone over a WiFi radio link, and use the command-line program *tcpreplay* to forward the traffic over an Amarisoft Callbox Mini network-in-a-box. This method has several drawbacks. First, the timing of the packets is already determined by the UAS it was recorded on, which was not using 5G as a transmission system, forwarding over a 5G network would introduce some timing variances. However, the overall structure of the flows would likely not change considerably. The packets captured en route by the Callbox Mini were also found to be unencrypted. As a result, their lengths were not masked by the encryption used by 5G, but rather by the current transport security protocol. This resulted in them being exactly the same, making detection of packets belonging to drone traffic flows trivial. Because of these limitations, this method was not used to generate any of the datasets used in this research.

3.4.2 Attempts to forward simulated drone traffic over 5G

The second method that was attempted was to simulate drone traffic using the two Raspberry Pis with 5G HATs to run the PX4 drone controller and QGC ground control software, and using the Amarisoft Callbox's 5G network as the data link. This approach was unsuccessful because the ARM-based Raspberry Pis were unable to run the flight control and ground control software. This limitation was addressed by attempting to run the software on two laptops, and then connecting each laptop to a 5G-connected Raspberry Pi via Ethernet. In this setup, the Raspberry Pis were configured to forward the drone traffic from the laptops over the Amarisoft 5G network, where the resulting 5G drone traffic could be captured and used to train the detection system. Unfortunately, this approach could not be implemented because of the networking limitations imposed by the Waveshare HAT.

3.4.3 Successful drone traffic simulation over WiFi and Ethernet

The approach that was successful, and ultimately used, was to forego the use of 5G as a transmission system for the drone traffic. The equipment used in this setup was two laptops,

both running Ubuntu 20.04, with one simulating a PX4 powered drone, and the other acting as its ground control station running QGC.

For the simulation, laptop one was running PX4 SITL and the Gazebo environment. PX4 simulations must be very realistic to provide engineers, who are designing and building drones, with as accurate a picture of how their designs will function in the real world as possible. This provides them with some surety that they will not have any bugs that could lead to malfunctions that could cause damage to the drone or environment in the real world. Performing simulation with the PX4 SITL involves running the same software as would be run onboard a real drone, again providing a high degree of realism. The other laptop, laptop two was running a full, native instance of QGC, which was not simulated and was not aware that it was only communicating with a simulated drone.

Figure 3.6 shows the different pieces of software that make up a PX4 simulation. In the setup used for data collection in this research, the "PX4 on SITL" software, as well as the "Simulator" (Gazebo), were running on laptop one, while "QGroundControl/Other GCS" (QGC) was running on laptop two. "API/Offboard" was not performed, as the simulated drone was not offboarding any processing or information. Therefore, the uppermost MAVLink connection, between port 14580 and ports 14540–14549, did not occur. Since the PX4 SITL and Gazebo were both running on laptop one, the MAVLink connection between "random port" and TCP 4560 was all within laptop one, and, therefore, not captured. This connection also would not exist in a real world situation, so it is not of interest for this research. Finally, the MAVLink connection between port 18570 and port 14550 is the data link between the simulated drone and the ground control station, or, in this setup, laptop one and laptop two. This connection carries the traffic that will be used to detect the drone. All of the packets sent over this data link, by PX4 and QGC are MAVLink carried by UDP, carried by IP.

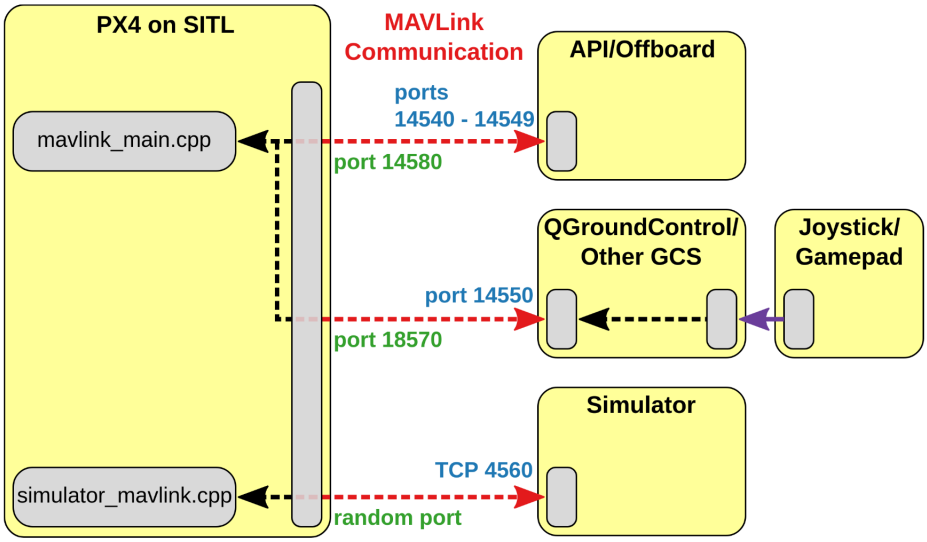

Figure 3.6. PX4 SITL Environment. Source: [34].

In this setup, all of the software being run is the same as would be run onboard a physical UAS, and the communication signals between the drone and the ground station are actually being sent over the network between the two laptops, just as they would be sent between the two physical devices. However, the physical environment the drone is flying in is being simulated, as well as the drone's hardware. This has the benefit of being easily reproducible with specialized hardware requirements.

Traffic was captured with the two laptops running the UAS simulation in two different configurations. In the first configuration, the laptops were both connected to a typical campus WiFi network, with environmental interference from other users, networks, and buildings. in the second configuration, the laptops were connected directly to each other via Ethernet. Without the proper equipment to capture traffic from a 5G-enabled drone, it is not possible to verify the effectiveness of flow-based detection for detecting 5G-enabled drones. However, this research seeks to preliminarily assess its effectiveness for distinguishing drones from normal 5G traffic, even if many of the features that are commonly used for drone detection, such as packet length and IP header information, are obscured, as they would be when sent over a 5G connection.

In order to collect representative MAVLink data, the simulated IRIS drone was flown around the environment in a variety of different ways, commands were issued to go to waypoints both close to and far from the drone's current location, and to orbit in large and small circles, clockwise and counter-clockwise. Figure 3.7 shows an example of a path flown by the drone in a data collection flight, the drone can be seen orbiting a point counter-clockwise at the moment the screenshot was taken. Figure 3.8 shows the display of the Gazebo simulation at a different point in the same flight, with the drone flying around in a simulated environment. Each flight followed a different path in a further effort to capture a variety of drone traffic.

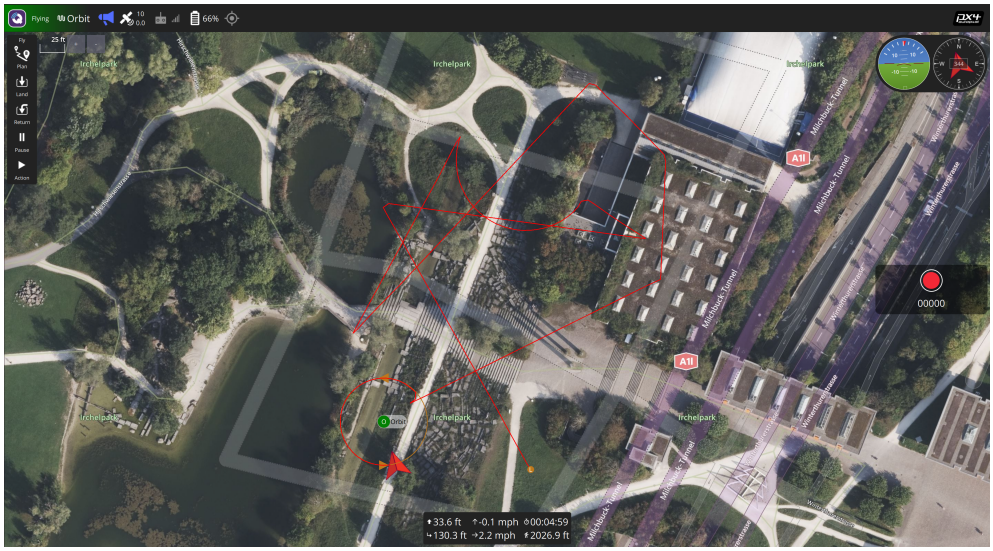

Figure 3.7. An Example Data Collection Flight Shown in QGroundControl.

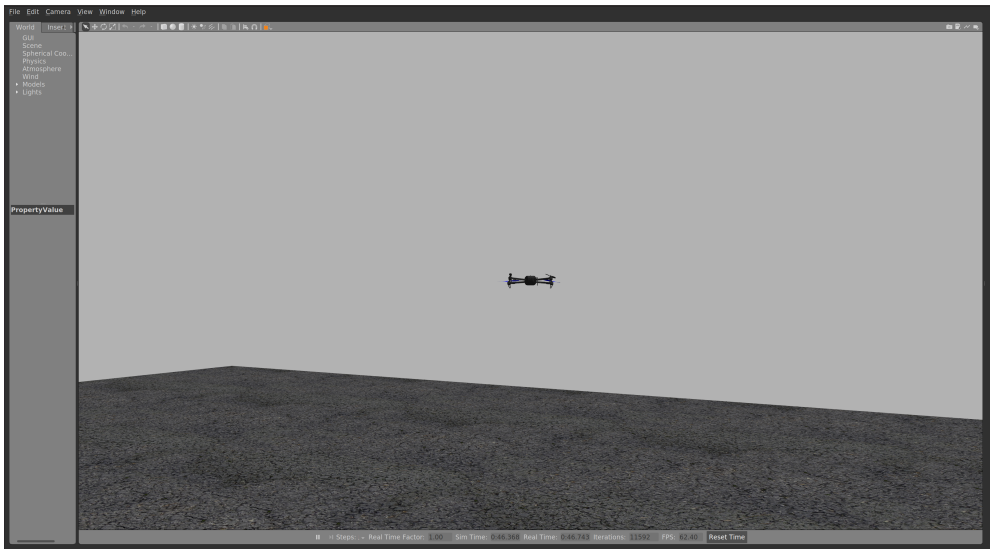

Figure 3.8. Gazebo Simulating an Example Data Collection Flight.

Packets were captured by running Wireshark on both laptops. Expectedly, both captures showed the same number of packets, as expected. As a result, the ground control station laptop's captures were chosen arbitrarily for feature extraction. The drone communication traffic was captured by running Wireshark on the ground control station laptop and exporting only MAVLink packets, which are UDP IP packets. Table 3.1 gives more information on captures used to train and test the machine learning detection model.

Table 3.1. Drone Traffic Packet Capture Metadata.

Duration (seconds)	Packet count	Transmission system
221	71120	WiFi
332	96929	WiFi
259	65481	WiFi
555	209619	WiFi
390	135989	WiFi
412	152464	Ethernet
595	221163	Ethernet
459	179039	Ethernet

The captured drone traffic in Table 3.1 makes up the positive class of the training and testing data, i.e., the traffic that the detector is trying to detect, while the normal 5G traffic dataset sourced in Section 3.3 makes up the negative class, i.e., the background traffic that the detector should ignore. In this research the only 5G traffic used is the normal 5G traffic, and the only WiFi and Ethernet traffic used is drone traffic. Chapter 4 describes several evaluations of how the model's ability to detect drone traffic varies based on the type of drone traffic used for training or testing (e.g., WiFi or Ethernet). However, no attempt is made to perform identification between WiFi and Ethernet drone traffic, they are both simply considered drone traffic.

Despite being different transmission systems, the packets sent over Ethernet, WiFi, and 5G are all IP packets when analyzed at the network layer, which is what flow-based detection does. Because of this, feature vectors extracted from flows sent over different transmission systems can be combined to create datasets for training and testing machine learning classifiers.

3.5 Feature Extraction

Following the data collection stage, the packet captures were exported from Wireshark as CSV files, and these files were used for feature extraction. Before the traffic can be used to train and test a machine learning classifier, feature vectors must be constructed from the data. The first step of feature extraction for traffic flows is separating and grouping the captured packets into their respective flows. The following notation is largely taken from [2] and [6], with some modification, as this thesis uses only time-based features. The exclusive use of temporal features enhances the model's ability to detect encrypted drone traffic, because commonly used features such as packet length and packet header fields are obscured by the encryption present in 5G traffic.

As discussed in Section 2.2, a given flow f^k is the stream of packets with the unique unordered set of source and destination IP addresses $k = (A_s, A_d)$ where: $(A_s, A_d) = (A_d, A_s)$. A particular flow lasts from when the first packet is sent to the last, which can be any (and a possibly infinite) length of time. In order to perform detection in a timely manner a time window W is defined, and only packets captured within W are considered to be part of a given flow.

The only data collected from each individual packet p_n is its source and destination set k_n, and arrival time t_n. Arrival time is defined as the number of seconds elapsed since the beginning of W when the current packet is captured by Wireshark on the ground control station laptop, regardless of if it originated on that laptop or not. Wireshark records time with a fidelity of ten nanoseconds. Thus, a packet is described fully by the list: $p_n = [k_n, t_n]$. As a flow is described by its source and destination addresses, each flow can be viewed as a list of N^k arrival times:

$$f^k = [t_1^k, ..., t_i^k, ..., t_N^k], \quad t_N^k - t_1^k < W \tag{3.1}$$

where N^k is the total number of packets in the flow. This list is then analyzed to construct a statistical fingerprint of the flow, a feature vector consisting of:

1. Feature x_1, the total number of packets in the flow f^k, previously defined as:

$$N^k = ||f^k|| \tag{3.2}$$

2. Feature x_2, the mean interarrival time of packets in the flow f^k is computed as:

$$\mu^k = \frac{1}{N^k - 1} \sum_{i=1}^{N^k - 1} (t_{i+1}^k - t_i^k) \tag{3.3}$$

3. Feature x_3, the median interarrival time of packets in the flow is computed as:

$$m^k = t_c^k, \quad c = \frac{(N^k + 1)}{2} \tag{3.4}$$

4. Feature x_4, the standard deviation of packet interarrival times is computed as:

$$\sigma^k = \sqrt{\frac{1}{N^k - 1} \sum_{i=1}^{N^k} (t_{i+1}^k - t_i^k)^2} \tag{3.5}$$

5. Feature x_6, the packet direction ratio r^k, is the ratio of packets with the more "talkative" address a as their source to those with that address as their destination and the less "talkative" address b in their source. Put more simply, it is the ratio of packets flowing in one direction to packets flowing in the other direction, ordered so that it is always between 0.5 and 1.0.

$$r^k = \frac{a \Rightarrow b}{b \Rightarrow a} \tag{3.6}$$

All elements are given a label y^k to indicate the type of traffic it is for training, validation, and testing of the model.

$$y^k = \begin{cases} 0 & \text{if normal 5G traffic} \\ 1 & \text{if drone traffic} \end{cases} \tag{3.7}$$

Each sample in the dataset is a flow fingerprint consisting of the feature vector $F(k) = N^k, \mu^k, m^k, \sigma^k, r^k$, and a corresponding label y^k.

A time window W of 0.1 seconds was chosen for the final model after preliminary tests. Packets were processed sequentially and before adding a given packet p to its corresponding flow f^k. If the elapsed time between t_1^k and p was greater than W, p, and all subsequent packets belonging to f^k, were added to a new flow $f^{k\prime}$, with this process repeating until all packets were processed.

The described features were extracted from the packet dissection CSVs via a Python script and saved to separate CSVs, to be entered into the machine learning classifier. The three datasets generated by the feature extraction process are described in Table 3.2

Table 3.2. Dataset Classes.

Transmission system	Samples
5G normal traffic	65,933
WiFi drone traffic	14,985
Ethernet drone traffic	14,273

3.6 Classifier Development

Once feature vectors have been extracted from the traffic flows, a machine learning classifier can be trained to distinguish those belonging to drone traffic from those belonging to normal traffic. As discussed in Section 2.3, random forest classifiers have several benefits for the task of automatic drone detection, and past research has demonstrated their effectiveness. This thesis used libraries from scikit-learn to implement random forests.

3.6.1 Ensemble Methods

Random forests are a type of classifier called ensemble methods. Scikit-learn describes ensemble methods as combining the predictions of several base estimators in order to improve the robustness of the predictions [41]. Random forests are a specific type of ensemble method called an averaging method, this means the base estimator are independent, and their predictions are averaged. According to scikit-learn "on average, the combined estimator is usually better than any of the single base estimators because its variance is reduced," this is what is referred to by the "wisdom of crowds." In random forests the base estimator is a decision tree. Decision trees make predictions by "learning simple decision rules inferred from the data features" [41]. Decision trees have several key advantages, they are simple to understand, can be visualized, and do not require data normalization. One

disadvantage of decision trees is their tendency toward overfitting, this tendency is however mitigated by the injected randomness in random forests. Overfitting occurs when a model conforms too closely to the training data, resulting in lower prediction performance on future data.

Random forests were implemented with scikit-learn using the following hyperparameters:

- n_estimators = 100, the number of trees in the forest.
- criterion = "gini", the function to measure the quality of a split.
- max_depth = None, the maximum depth of trees in the forest. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.
- min_samples_split = 2, the minimum number of samples required to split an internal node
- min_samples_leaf = 1, the minimum number of samples required to be at a leaf node.
- min_weight_fraction_leaf = 0.0, the minimum weighted fraction of the sum total of weights required to be at a leaf node.
- max_features = "sqrt", the number of features to consider when looking for the best split. If "sqrt" then max_features is the square root of the number of features.
- max_leaf_nodes = None, the maximum number of leaf nodes allowed in a tree.
- min_impurity_decrease = 0.0, a node will be split if this split induces a decrease of the impurity greater than or equal to 0.
- random_state = 1, controls the sampling of features considered when looking for the best split at each node. Using a random state provides reproducibility during development.
- class_weight = None, the weights associated with classes in the dataset. If not given, all classes will have weight one.
- ccp_alpha = 0.0, Complexity parameter used for minimal cost-complexity pruning. If 0, no pruning is performed.
- max_samples = None, the maximum number of samples to draw from X to train each base estimator.

3.6.2 Class Imbalance

In a real-world application, the amount of normal traffic would far outnumber the amount of drone traffic on a 5G network. Therefore, if the dataset used to train the random forest model was representative, it would have a strong *class imbalance*. Class imbalance can cause problems for machine learning models, particularly in cases when the minority class is the positive class. An example of this is a "lazy" classifier that classified every flow as normal traffic, it would be correct the vast majority of the time, but it would also be useless. In order to avoid the complications of class imbalance a large amount of drone traffic was captured relative to the normal 5G dataset. The resulting datasets have a class imbalance of 4.6:1. Experimentation with random undersampling and oversampling showed a decrease in performance, so the classes were left unbalanced.

Initially, the model suffered from poor recall, stemming from a high degree of false negatives. This was remedied by adjusting the decision threshold from 0.5 to 0.3. The decision threshold is the percentage of decision trees in the random forest that must classify a sample as a positive (i.e., drone) for the random forest to make a positive classification. Lowering the threshold caused the model to make a positive classification more often, resulting in a decrease in false negatives at the cost of a slight increase in false positives.

CHAPTER 4:
Results

The modeling approach was evaluated in two ways. First it was trained and tested on the different types drone traffic collected in Section 3.4.3 by using datasets constructed from the classes in Table 3.2. The second evaluation was performed by training the model on one type of drone traffic and testing it on the other. The purpose of the first evaluation is to determine the ability of the model to detect the captured drone traffic, which is not 5G traffic, from the selected normal 5G traffic. Because this is not a direct comparison, the second evaluation was performed in an attempt to generalize the detection from just one transmission system towards MAVLink drone traffic in general.

4.1 Metrics

In the classifications performed in this research, a sample flow is either drone traffic, or not. Thus, each sample is labelled with a binary 1 or 0, indicating drone traffic or normal traffic respectively. The model is tested by providing it with the feature vector of a sample and recording the label it predicts. Before evaluating the model's performance the metrics used for binary classification will be defined and analyzed.

- True Positive (TP): label = 1, prediction = 1
- False Positive (FP): label = 0, prediction = 1
- True Negative (TN): label = 0, prediction = 0
- False Negative (FN): label = 1, prediction = 0
- Correct Prediction: label = prediction, includes TP, TN
- Incorrect Prediction: label $\neq$ prediction, includes FP, FN
- Accuracy: The percentage of predictions that are correct, defined as:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{4.1}$$

- Precision: The percentage of positive predictions that are correct, defined as:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4.2}$$

- Recall: The percentage of positive samples that were correctly predicted, defined as:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4.3}$$

- F1: The harmonic mean of recall and precision, defined as:

$$\text{F1} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = 2 \cdot \frac{TP}{TP + FP + FN} \tag{4.4}$$

Each metric provides different insights about the performance of the model. Accuracy is the most commonly used metric. It shows how often the model was correct. However, accuracy can be misleading in cases where there is class imbalance, such as the case discussed in Section 3.6. In that dataset, the positive case is less than 1% of the total dataset. A "lazy" classifier that always predicted 0 would have an accuracy greater than 99%, but would be useless. Precision and recall are useful metrics for unbalanced datasets as they each correspond to the rate of false positives and false negatives. However, like accuracy, using only precision or only recall can lead to the model developing biases that diminish it's usefulness. This is why F1 is a useful metric. F1 balances the competing metrics of precision and recall, therefore a model with a high F1 score will have a high, yet balanced, rate of correct predictions.

Confusion matrices can be useful tools for visualizing these metrics. A confusion matrix displays all of the predictions made by the model broken down into true positives, true negatives, false positives, and false negatives.

4.2 Standard Evaluation

The feature vectors extracted from the collected traffic and described in Table 3.2 were combined into several different datasets, to evaluate the model's effectiveness in different scenarios. Each dataset contains the same normal 5G background traffic, and some selection of drone traffic to be detected. The following datasets were used for the first evaluation, in subsequent tables and figures the parenthesized shorthand names will be used:

1. (WiFi): WiFi drone traffic + normal 5G traffic: 80,519 samples
2. (Ethernet): Ethernet drone traffic + normal 5G traffic: 80,206 samples

3. (Combined): WiFi drone traffic + Ethernet drone traffic + normal 5G traffic: 94,791 samples

These three datasets were each evaluated using k-fold cross-validation with $k = 5$. In k-fold cross-validation with $k = 5$, the dataset is split into five random sets of equal size, four are used to train the model and the fifth is used to test it. This process is repeated with each of the sets being used as the test set, for a total of five cross-validations, after which the average scores are evaluated. The scores generated from the cross-validation are presented as percentages in Table 4.1.

Table 4.1. k-Folds Cross Validation Scores (Percentages).

Dataset	WiFi	Ethernet	Combined
Accuracy	99.38	99.98	99.48
Precision	98.50	99.94	99.25
Recall	98.08	99.99	99.02
F1	98.29	99.94	99.14

Figure 4.1 shows the confusion matrices from the k-fold cross-validation of the model trained and tested on the WiFi dataset. A confusion matrix shows the prediction that the model made for every sample in the test set, along with the true label of the sample. The cells in the top left and bottom right show the correct predictions, respectively the true negatives and true positives. The cells in the top right and bottom left show the incorrect predictions, respectively the false positives and false negatives. In this and all subsequent matrices, normal 5G traffic is labelled 0 and drone traffic is labelled 1. This dataset produced a similar amount of false positives as false negatives, which is also shown in the similar precision and recall scores in the "WiFi" column of Table 4.1.

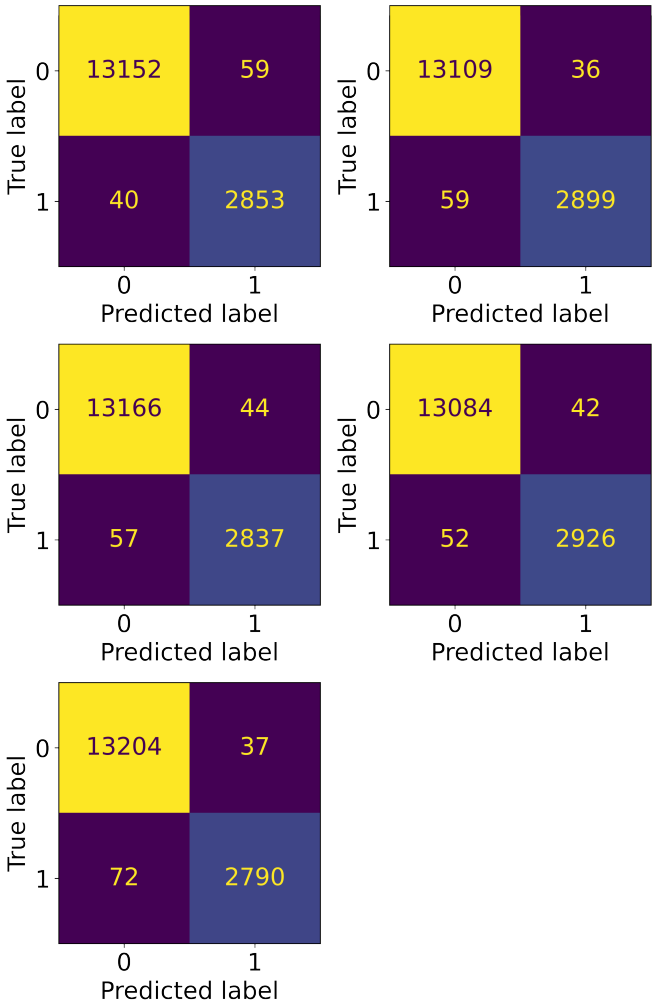

Figure 4.1. k-Fold Cross Validation Confusion Matrices for WiFi Drone and Normal 5G Traffic.

Figure 4.2 shows the confusion matrices from the k-fold cross-validation of the model trained and tested on the Ethernet dataset. The most notable aspect of this confusion matrix is that out of the 16,041 samples the model always predicted five or less incorrectly, resulting in the 99.9% accuracy, precision, recall, and F1 scores in Table 4.1. The model's far superior performance in detecting Ethernet drone traffic over WiFi drone traffic may stem from the lower amount of dropped packets and overall greater simplicity of the medium making the patterns in the drone traffic stand out more. This interpretation would suggest that in a real world environment, where there is environmental interference, the model would perform

more similarly to the WiFi results, since the WiFi traffic was captured on a campus network with multiple users and environmental interference from other buildings and networks. 5G signals in a future smart environment will also be subjected to a high degree of interference, so this approach would need to be modified to be more resilient to such interference to be optimally effective in those environments.

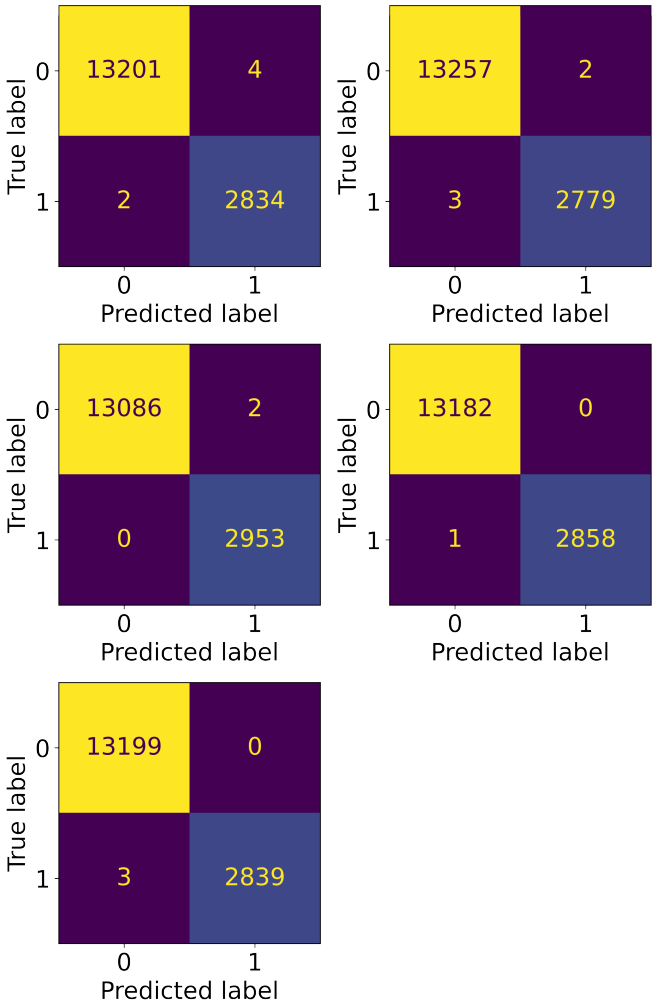

Figure 4.2. k-Fold Cross Validation Confusion Matrices for Ethernet Drone and Normal 5G Traffic.

Figure 4.3 shows the confusion matrices from the k-fold cross validation of the model trained and tested on the dataset with both WiFi and Ethernet traffic. This confusion matrix shows performance is slightly better than just using the WiFi dataset, but not as good as the Ethernet results, further indicating that the Ethernet traffic is easier for the model to detect.

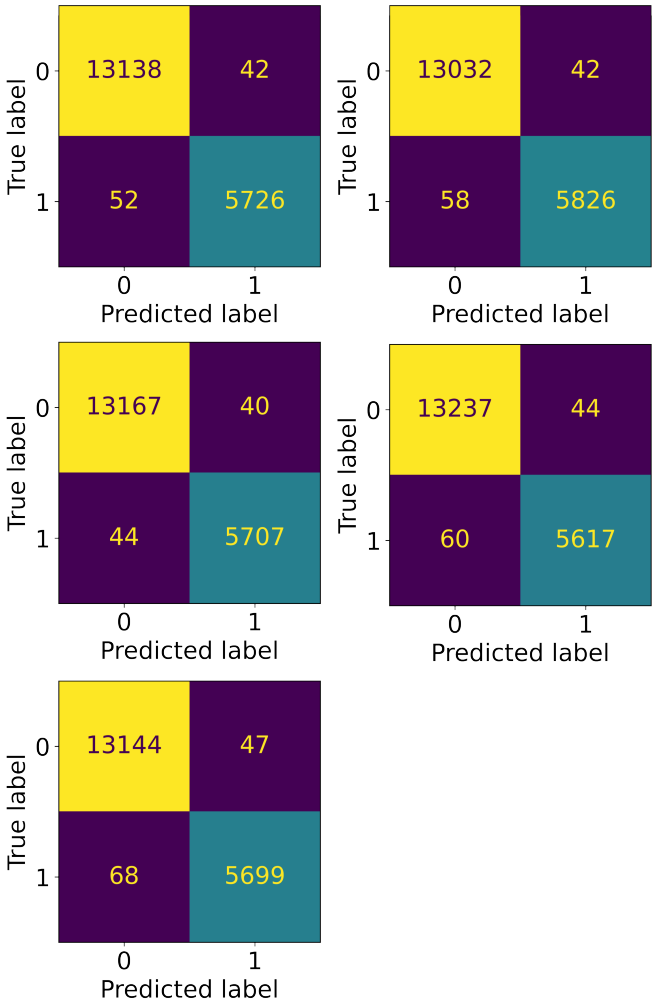

Figure 4.3. k-Fold Cross Validation Confusion Matrices for Combined WiFi and Ethernet Drone, and Normal 5G Traffic.

These results are promising as they indicate that the model is not just picking up the quirks of the transmission system (e.g., WiFi vs. 5G, Ethernet vs. 5G), as it is able to generalize its predictions to drone traffic regardless of transmission system. This suggests that these

techniques may be effective for detecting drone traffic regardless of the transmission system. However, this suggestion can be further strengthened by performing a cross-system evaluation.

4.3 Cross-System Evaluation

The cross-system evaluation involved training the random forest model on a dataset of normal traffic and drone traffic from only one transmission system (e.g., Ethernet), and then testing it on a second dataset of different normal traffic, and drone traffic from only the other transmission system (e.g., WiFi). The purpose of this evaluation is to further examine how much the model is making classifications based on the transmission system, and how much it is actually detecting the fingerprints of drone communications. The following datasets were used for this evaluation, in subsequent tables and figures the parenthesized shorthand names will be used:

1. WiFi drone traffic and normal 5G traffic (WiFi): 47,207 samples
2. Ethernet drone traffic and normal 5G traffic (Ethernet): 47,586 samples

The first dataset contains the WiFi drone traffic and half of the normal 5G traffic, and the other contains the Ethernet drone traffic and the other half of the normal 5G traffic. This was done so that there would be no overlap in the training and testing datasets in terms of normal 5G traffic, since each dataset was used in its entirety for training or testing. The averaged scores, and their standard deviations, from 20 runs of the model are presented as percentages in Table 4.2.

Table 4.2. Cross-System Evaluation Scores from 20 Runs (Percentages).

Training set	WiFi		Ethernet	
Testing set	Ethernet		WiFi	
	μ	σ	μ	σ
Accuracy	99.27	0.01	98.02	0.16
Precision	97.65	0.02	99.96	0.01
Recall	99.98	0.01	93.58	0.52
F1	98.80	0.01	96.67	0.28

Surprisingly, the model performed extremely well in this evaluation. The high performance of the model in detecting drone traffic flows despite them being sent over a different transmission system than those it was trained on further demonstrates that the model is agnostic to the transmission system, be that Ethernet, WiFi, or potentially 5G. In terms of accuracy and F1, the model performed better when trained on WiFi and tested on Ethernet than the opposite. This fits with the higher scores achieved on the Ethernet dataset in Section 4.2, further indicating that the pattern the model is looking for is clearer in the Ethernet traffic.

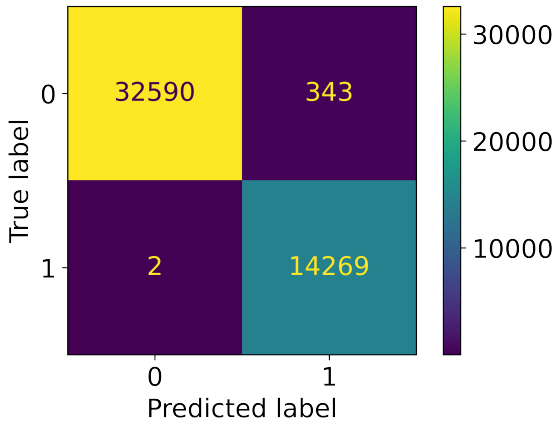

Figure 4.4. Confusion Matrix Averaged across 20 Runs of Training on WiFi and Testing on Ethernet.

The scores in Table 4.2 show that training on WiFi and testing on Ethernet resulted in a recall of 99.98, but a precision of only 97.65. Training on Ethernet and testing on WiFi resulted in a precision of 99.96, but a recall of only 93.58. This discrepancy shows that training on WiFi leads to more false positives, while training on Ethernet leads to more false negatives, an effect that can can be seen clearly in the 334 false positives to 2 false negatives in Figure 4.4, and the 1004 false negatives to 6 false positives in Figure 4.5. These results suggest that training on Ethernet and testing on WiFi causes the model to over-detect drones, and training on WiFi and testing on Ethernet causes the model to under-detect drones.

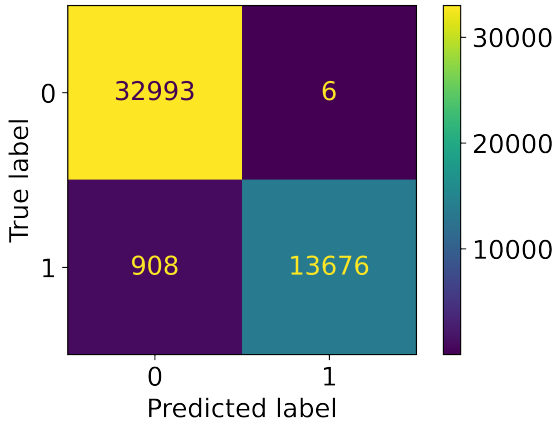

Figure 4.5. Confusion Matrix Averaged across 20 Runs of Training on Ethernet and Testing on WiFi.

4.4 Feature importance analysis

The random forest classifier used for drone detection is an ensemble of 100 decision trees, and each decision tree is made up of an arbitrary number of decisions, with each decision being based on a single feature. MDI importance can be used to measure the amount that a random forest considers each feature in the feature vector when making classifications [42]. The average feature importances from the random forest classifier trained on normal 5G traffic and the combined WiFi and Ethernet drone traffic are displayed as percentages in Figure 4.6.

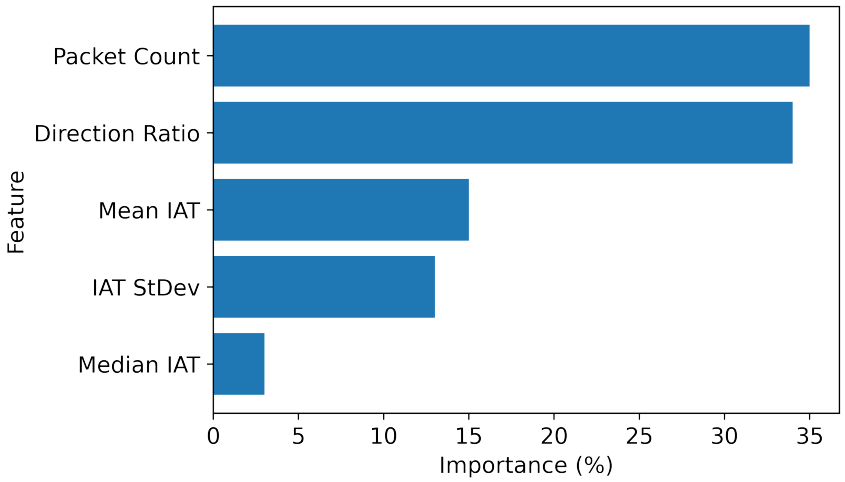

Figure 4.6. MDI Feature Importance Percentages.

It is clear that the model learned that the features that best differentiate the two classes are packet count during the time window and direction ratio. This behavior is further indicated by training and testing the model with only these two features as inputs, and finding that performance as given by F1 score from k-fold cross-validation (described in more detail in Section 4.2) was reduced by less than one percent. Of these two features, direction ratio is largely unaffected by the transmission system, since it is a result of the application's purposes. Put simply, if two devices are communicating, the ratio of how many messages one of them sends vs how many it receives is reliant on what they are communicating. For instance, nearly all of the drone traffic has a very high ratio, this is because the drone is constantly sending updates on its position and status to the control station, while the control station is only sending commands to the drone when they are entered by the operator, which is not the many times per second that updates are sent back by the drone. This is different from other applications such as a video call, in which both devices are sending roughly the same amount of traffic to each other, but similar to a video stream, where the server device is sending a large amount of traffic and the client device is sending very little, much like the drone traffic. While these ratios may be affected to some extent by 5G or WiFi or Ethernet, the fundamental differences in the reasons that different applications generate network traffic will not change. All of this supports the concept that the high reliance the model has on flow direction ratio indicates that it is picking up on the differences in traffic

from different applications (e.g., drone, not-drone).

4.5 Limitations

The results achieved by the model are very promising. However, the model does have a few limitations and, while these results provide insights about possible directions for 5G drone detection, they are by no means comprehensive. Because the model was not evaluated on 5G drone traffic, its aptitude for 5G drone detection is unknown, although the cross-system analysis indicates that it is adaptable to different transmission systems. Another notable limitation is that only MAVLink protocol drone traffic was explored, it is unknown how well this approach would be suited to detection of drones using other communication protocols.

The greatest limitation to this methodology is that the drone traffic and the normal traffic were captured on different transmission systems, so it cannot be stated with certainty that the model was detecting drones, and not just detecting non-5G traffic. The packet count could be impacted by the higher transfer speeds of 5G, the largest packet counts in the dataset belong to 5G flows, however the majority of 5G flows in this dataset have much lower packet counts than the non-5G flows in the dataset, the 5G traffic also has a far higher spread in terms of packet count. None of these values can make a strong claim that the feature of packet count is or is not affected by the transmission system, meaning the results of this model are not comprehensive, but are an exploration of direction for future systems.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 5:
Conclusion and Future Work

5.1 Conclusion

This thesis explored the use of machine learning and traffic flow analysis with the ultimate goal of detecting drones on 5G networks. Time-based feature extraction techniques were developed, and random forests were shown to be able to distinguish encrypted MAVLink drone traffic from everyday user 5G traffic, based on less than one tenth of a second of traffic. The random forest classifier model achieved F1 scores greater than 96% detecting drone traffic captured on both WiFi or Ethernet, no matter which type of drone traffic it was trained on. These results suggest that these techniques would be effective for detecting 5G drone traffic as well.

Chapter 2 gave an overview of the various detection methods in the literature and described traffic flows and traffic flow-based detection, particularly the related work in WiFi detection was described. Additionally, the current limitations and future possibilities of 5G drones were explored.

Chapter 3 first described the equipment used in this research, and the currently available 5G and drone technology, as well as summarizing the drone control and simulation software used. The standard 5G traffic dataset was then introduced and the drone data collection methods were explained. Finally, the techniques for feature extraction were demonstrated and the random forest model was described.

Chapter 4 defined metrics for model evaluation and then evaluated the model, finding F1 scores greater than 98% in standard cross validation and greater than 96% when testing across transmission systems. Finally, the model's limitations were discussed.

5.2 Future Work

5G drones are still in their infancy, so there is much future work to be done on detecting and tracing them. The results found in this research could be strengthened by addressing the

limitations described in Section 4.5. Particularly, research could be furthered by testing the model on real 5G drone traffic, and non-MAVLink drone traffic. The techniques developed in this research were a proof of concept for a real time 5G drone detection system. Future work could pursue using these techniques to build such a system.

In this research, all traffic was prerecorded and separated into sections of time window W, one tenth of a second. A real-time detection system would listen in on a live network and perform the capturing and processing automatically in intervals of W. The capture of 5G traffic could be performed either at the base station or out of the air via SDR. Tools, such as Wireshark or tcpdump, could be configured to automatically export all captured packets at intervals of W, then the extraction and detection algorithms could be run on the exported traffic. The only new capabilities needed to create such a real-time detection system from the techniques produced by this research would be 5G traffic capture capabilities and 5G-connected drones to detect. Such future 5G testing could use the PX4 SITL simulation with QGC discussed previously.

List of References

[1] K. Chávez, "Learning on the fly: Drones in the russian-ukrainian war," *Arms Control Today*, vol. 53, no. 1, pp. 6–11, 2023.

[2] I. Bisio, C. Garibotto, F. Lavagetto, A. Sciarrone, and S. Zappatore, "Blind detection: Advanced techniques for wifi-based drone surveillance," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 938–946, 2019 [Online]. Available: https://doi.org/10.1109/TVT.2018.2884767

[3] K. E. Skouby and P. Lynggaard, "Smart home and smart city solutions enabled by 5G, IoT, AAI and CoT services," in *2014 international conference on contemporary computing and informatics (IC3I)*. IEEE, 2014, pp. 874–878.

[4] I. Guvenc, F. Koohifar, S. Singh, M. L. Sichitiu, and D. Matolak, "Detection, tracking, and interdiction for amateur drones," *IEEE Communications Magazine*, vol. 56, no. 4, pp. 75–81, 2018.

[5] I. Bisio, C. Garibotto, F. Lavagetto, A. Sciarrone, and S. Zappatore, "Unauthorized amateur uav detection based on wifi statistical fingerprint analysis," *IEEE Communications Magazine*, vol. 56, no. 4, pp. 106–111, 2018.

[6] S. Sciancalepore, O. A. Ibrahim, G. Oligeri, and R. Di Pietro, "Pinch: An effective, efficient, and robust solution to drone detection via network traffic analysis," *Computer Networks*, vol. 168, p. 107044, 2020.

[7] K. Norrman, P. K. Nakarmi, and E. Fogelström. (2018). 5G Security — Enabling a Trustworthy 5G System. [Online]. Available: https://www.ericsson.com/en/reports-and-papers/white-papers/5g-security---enabling-a-trustworthy-5g-system

[8] C. Thiessen and B. Hale, "Blocking uass with bytes, not bullets," *USNI*, vol. 148, no. 9, 2022 [Online]. Available: https://www.usni.org/magazines/proceedings/2022/september/blocking-uass-bytes-not-bullets

[9] G. Ding, Q. Wu, L. Zhang, Y. Lin, T. A. Tsiftsis, and Y.-D. Yao, "An amateur drone surveillance system based on the cognitive internet of things," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 29–35, 2018.

[10] N. Times. (2017). Dutch police drops drone-hunting eagles project. [Online]. Available: https://nltimes.nl/2017/12/07/dutch-police-drops-drone-hunting-eagles-project

[11] M. Benyamin and G. H. Goldman, "Acoustic detection and tracking of a class i uas with a small tetrahedral microphone array," Army Research Lab Adelphi MD, Rep., 2014.

[12] S. R. Ganti and Y. Kim, "Implementation of detection and tracking mechanism for small uas," in *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2016, pp. 1254–1260 [Online]. Available: https://doi.org/10.1109/ICUAS.2016.7502513

[13] F. Fioranelli, M. Ritchie, H. Griffiths, and H. Borrion, "Classification of loaded/unloaded micro-drones using multistatic radar," *Electronics Letters*, vol. 51, no. 22, pp. 1813–1815, 2015.

[14] H. Shin, K. Choi, Y. Park, J. Choi, and Y. Kim, "Security analysis of fhss-type drone controller," in *Information Security Applications: 16th International Workshop, WISA 2015, Jeju Island, Korea, August 20–22, 2015, Revised Selected Papers 16*. Springer, 2016, pp. 240–253.

[15] C. H. Lee, C. Thiessen, D. L. Van Bossuyt, and B. Hale, "A systems analysis of energy usage and effectiveness of a counter-unmanned aerial system using a cyber-attack approach," *Drones*, vol. 6, no. 8, 2022 [Online]. Available: https://doi.org/10.3390/drones6080198

[16] M. Crotti, M. Dusi, F. Gringoli, and L. Salgarelli, "Traffic classification through simple statistical fingerprinting," *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 1, pp. 5–16, 2007.

[17] J. Quittek, T. Zseby, B. Claise, and S. Zander, "Rfc 3917: Requirements for ip flow information export (ipfix)," The Internet Society, USA, Rep., 2004.

[18] S. Sciancalepore, O. A. Ibrahim, G. Oligeri, and R. Di Pietro, "Detecting drones status via encrypted traffic analysis," in *Proceedings of the ACM Workshop on Wireless Security and Machine Learning*, 2019, pp. 67–72.

[19] L. Sundqvist, "Cellular Controlled Drone Experiment: Evaluation of Network Requirements," Master's thesis, Aalto University. School of Electrical Engineering, 2015 [Online]. Available: http://urn.fi/URN:NBN:fi:aalto-201512165670

[20] V. Marojevic, I. Guvenc, R. Dutta, M. L. Sichitiu, and B. A. Floyd, "Advanced wireless for unmanned aerial systems: 5g standardization, research challenges, and aerpaw architecture," *IEEE Vehicular Technology Magazine*, vol. 15, no. 2, pp. 22–30, 2020.

[21] R. Muzaffar, C. Raffelsberger, A. Fakhreddine, J. L. Luque, D. Emini, and C. Bettstetter, "First experiments with a 5g-connected drone," in *Proceedings of the 6th ACM Workshop on Micro Aerial Vehicle Networks, Systems, and Applications*, 2020, pp. 1–5.

[22] M. Jurčević and I. Vitas, "5g-connected drone for public road safety-research challenges and future research roadmap," in *2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO)*. IEEE, 2021, pp. 417–422.

[23] A. Fakhreddine, C. Bettstetter, S. Hayat, R. Muzaffar, and D. Emini, "Handover challenges for cellular-connected drones." New York, NY, USA: Association for Computing Machinery, 2019 [Online]. Available: https://doi.org/10.1145/3325421.3329770

[24] T. Sharma, K. Ritesh, N. Chauhan, and S. Agarwal, "Analogous study of 4g and 5g," in *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, 2016, pp. 2137–2140.

[25] M. Koziol. (2020). 5G Just Got Weird. [Online]. Available: https://spectrum.ieee.org/5g-release-16

[26] S.-Y. Lien *et al.*, "3gpp nr sidelink transmissions toward 5g v2x," *IEEE Access*, vol. 8, pp. 35 368–35 382, 2020 [Online]. Available: https://doi.org/10.1109/ACCESS.2020.2973706

[27] Amarisoft. Amari Callbox series. [Online]. Available: https://www.amarisoft.com/products/test-measurements/amari-lte-callbox/

[28] R. Pi. Raspberry Pi 4 Tech Specs. [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/

[29] Waveshare. SIM8200EA-M2 5G HAT for Raspberry Pi, 5G/4G/3G Support, Snapdragon X55, Multi Mode Multi Band. [Online]. Available: https://www.waveshare.com/sim8200ea-m2-5g-hat.htm

[30] Insitu. ScanEagle. [Online]. Available: https://www.insitu.com/products/scaneagle

[31] P. Parihar, P. Bhawsar, and P. Hargod, "Design & development analysis of quadcopter," *Compusoft*, vol. 5, no. 6, p. 2128, 2016.

[32] R. Mac. (2017). Behind the Crash of 3D Robotics, North America's Most Promising Drone Company. [Online]. Available: https://www.forbes.com/sites/ryanmac/2016/10/05/3d-robotics-solo-crash-chris-anderson/?sh=365969fe3ff5

[33] D. Robotics. (2014). IRIS Multicopter All-in-one-aerial performance. [Online]. Available: https://web.archive.org/web/20140131183317/http://3drobotics.com/iris/

[34] P. Autopilot. PX4 Autopilot User Guide. [Online]. Available: https://docs.px4.io/main/en/

[35] DJI. DJI Air 2S. [Online]. Available: https://store.dji.com/product/dji-mini-3?vid=128001

[36] Auterion. The story of PX4 and Pixhawk. [Online]. Available: https://auterion.com/company/the-history-of-pixhawk/

[37] MAVLink. MAVLink Developer Guide. [Online]. Available: https://mavlink.io/en/

[38] MAVLink. MAVLink Developer Guide. [Online]. Available: https://mavlink.io/en/

[39] QGroundControl. QGroundControl: Intuitive and Powerful Ground Control Station for the MAVLinkProtocol. [Online]. Available: http://qgroundcontrol.com/

[40] C. Coldwell *et al.*, "Machine learning 5g attack detection in programmable logic," in *2022 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2022, pp. 1365–1370.

[41] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[42] G. Louppe, L. Wehenkel, A. Sutera, and P. Geurts, "Understanding variable importances in forests of randomized trees," in *Advances in Neural Information Processing Systems*, C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, Eds. Curran Associates, Inc., 2013, vol. 26 [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2013/file/e3796ae838835da0b6f6ea37bcf8bcb7-Paper.pdf

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

DUDLEY KNOX LIBRARY

NAVAL POSTGRADUATE SCHOOL

WWW.NPS.EDU

WHERE SCIENCE MEETS THE ART OF WARFARE