The Pennsylvania State University

The Graduate School

SMALL UNMANNED AERIAL SYSTEMS DETECTION AND CLASSIFICATION USING IEEE 802.11 SIGNALS

A Thesis in

Cybersecurity Analytics and Operations

by

Rahul Emani

© 2022 Rahul Emani

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

May 2022

The thesis of Rahul Emani was reviewed and approved by the following:

Nicklaus A. Giacobe
Assistant Teaching Professor of Information Sciences and Technology
Thesis Advisor

Edward J. Glantz
Teaching Professor of Information Sciences and Technology

Anna C. Squicciarini
Frymoyer Chair in Information Sciences and Technology

Timothy E. Kohler
Research and Development Engineer at The Applied Research Laboratory at Penn State

Edward J. Glantz
Teaching Professor of Information Sciences and Technology
Chair of the Graduate Program

ABSTRACT

Small Unmanned Aerial Systems (sUAS) or drones are enjoying great popularity due to their ease of use, low cost, and wide availability. Today, these systems can be piloted using cellular devices as well as other networked systems. Given their size, sUAS are generally undetectable, can be launched from almost anywhere, and have little to no regulations regarding takeoff and flying. Depending on the airframe design and application, a given system may even have the ability to carry a sizeable payload, e.g., agricultural drones used to spray crops. Given the aforementioned statements, one can readily see why these same "hobby systems" are also very popular with those wishing to invade privacy or cause harm and damage. For example, nothing is preventing someone with bad intentions from filling the spray reservoir with a fluid containing viruses or bacteria to launch a bioterrorism attack. Along the same line, if the airframe is of sufficient size to carry a container of liquid, there is nothing preventing someone from replacing the container with an explosive device. Therefore, there is a strong need to know when any drone is operating within a given geographic dome.

This thesis will thus explore the feasibility of using IEEE 802.11 signals for the detection and classification of Wi-Fi enabled hobbyist drones to provide drone operation accountability. In the end, drone flight data was collected, and analyzed for differentiating features to aid in drone detection and classification. Then, seven different machine learning algorithms were trained and evaluated for their effectiveness of differentiating a drone versus a wireless access point. After that, the seven machine learning algorithms were retrained and reevaluated for their effectiveness at identifying the specific make and model of drone. Together, the research conducted explored a drone detection and classification framework desperately needed for invoking C-sUAS responses.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

- 3D – Three Dimensional
- A-CFAR – Averaging-Constant False Alarm Rate
- ADS-B – Automatic Dependent Surveillance-Broadcast
- AI – Artificial Intelligence
- AIB – Axially Integrated Bispectra
- AP - Access Point
- BSSID - Broadcast Service Set Identification
- CART – Classification and Regression Trees
- CFAR - Constant False Alarm Rate
- CISA - Cybersecurity and Infrastructure Security Agency
- C-sUAS – Counter-small Unmanned Aerial Systems
- CSV - Comma Separated Value
- CYVR – Vancouver International Airport
- dB – Decibel
- DBF – Digital Beam Forming
- DSSS - Direct-Sequence Spread Spectrum
- FAA – Federal Aviation Administration
- FASST - Futaba Advanced Spread Spectrum Technology
- FCC - Federal Communications Commission
- FD – Fractal Dimension
- FHSS - Frequency-Hopping Spread Spectrum
- FRIA – FAA-Recognized Identification Area

- GHz – Gigahertz
- GNB – Gaussian Naïve Bayes
- Hz - Hertz
- IEEE - Institute of Electrical and Electronics Engineers
- IP - Internet Protocol
- KHz – Kilohertz
- KNN – K-Nearest Neighbors
- LDA – Linear Discriminant Analysis
- LR – Logistic Regression
- MAC – Media Access Control
- MAVLink - Micro Air Vehicle Link
- MHz – Megahertz
- m - Meter
- OFDM - Orthogonal Frequency-Division Multiplexing
- OUI – Organizationally Unique Identifier
- RC - Radio Control
- RF – Radio Frequency
- RMSE – Root Mean Square Error
- s - seconds
- SDAE-LOF – Stacked Denoising Autoencoder-Local Outlier Factor
- SIB – Square Integrated Bispectra
- SNR – Signal to Noise Ratio
- SSID - Service Set Identification
- SSL – Secure Sockets Layer

- sUAS – small Unmanned Aerial Systems
- TCAS - Traffic Collision Avoidance System
- TCP – Transmission Control Protocol
- TLS – Transport Layer Security
- UAV – Unmanned Aerial Vehicle
- UDP – User Datagram Protocol
- UDT – UDP-based Data Transfer Protocol
- WAP - Wireless Access Point

ACKNOWLEDGEMENTS

I would like to begin by thanking Dr. Timothy Kohler for his mentorship, guidance, and support of my research interests and endeavors. Second, I would like to thank Dr. Nicklaus Giacobe and Dr. Edward Glantz for their mentorship, guidance, and support and guidance throughout my time in the Bachelor of Science and Master of Science Cybersecurity Analytics and Operations programs at Penn State. I would not have known the Integrated Undergraduate/Graduate program existed without them. Third, I'd like to thank Dr. Anna Squicciarini for being my honors advisor and supporting my Schreyer Honors College endeavors. Finally, I would like to thank Dr. Jack Langelaan for mentorship in the Aerospace Engineering area of my research and coursework.

I would like to thank the CyberCorps: Scholarship for Service program for supporting my education and research.

I would also like to thank the Applied Research Laboratory at Penn State, the College of Information Sciences and Technology and the College of Engineering for supporting me throughout my time at Penn State.

I would like to give a special thanks to my parents and family for their continued support along my academic, personal, and professional journey.

Chapter 1

Introduction

This thesis investigates the feasibility of analyzing Institute of Electrical and Electronics Engineers (IEEE) 802.11 emissions to detect and classify Wi-Fi enabled small Unmanned Aerial Systems (sUAS). As UASs become more prevalent in the world, the likelihood of them being used for mischievous things is increasing. As such, there is a need to be able to detect and classify sUAS to prevent.

sUAS are small, cheap, easy to fly and easy to conceal which make them feasible to use by terrorist groups. Terrorism attacks by known terrorist cells are a worst-case scenario, however any person can theoretically fly sUAS with a malicious intent or payload. Imagine school shooters mounting a gun to sUAS. Eavesdropping, taking pictures for blackmail or extortion are all considered malicious uses of sUAS. There must be a way to take down a drone being flown with malicious intentions. The first step to sUAS defense is to be able to detect and classify sUAS. Thus, IEEE 802.11 signal analysis is a theoretical way to detect and classify sUAS. This chapter will thus outline the problem statement, research questions, importance and objective.

Problem Statement

Having a proper Counter small Unmanned Aerial Systems (C-sUAS) framework in place can help prevent and deter drone related disruptions, terrorism and espionage. C-sUAS has applications in aviation, infrastructure and human safety.

Research Questions

This research will be split into two separate parts. The first part is the effective detection of sUAS. The first step for C-sUAS is detecting if there is actually a sUAS present.

1. How can 802.11 signals be used to effectively detect a Wi-Fi enabled sUAS?

The second part will be classification of sUAS. Ideally it would include identifying the specific vehicle that is present by make and model.

2. How can 802.11 signals be used to classify a sUAS by make and model?

Importance

The importance of this thesis is to build the basic framework for C-sUAS by covering the detection and classification of small unmanned aerial systems (sUAS) using IEEE 802.11 signals. By looking into detection and classification, a preemptive C-sUAS response can be invoked with sufficient time to minimize or eliminate disruptions and delays. The research conducted will contribute to invoking specific C-sUAS responses based on make and model of drone.

Hartmaan and Giles estimated 15,000 unmanned aerial vehicles (UAVs) in service by 2020 and cited UAV video streams being intercepted by Iran and Ukraine as support for C-sUAS. However, in 2015 alone, UAV sales were approximately 15,000 a month which prompted the Federal Aviation Administration (FAA) to pass several regulations on UAVs (Hartmaan & Giles, 2016). sUAS growth has greatly exceeded projections in 2015 alone which shows widespread availability and drone ownership among the public. Last updated on January 11, 2022, the FAA reports “856,158 total drones registered [which include] 325,372 commercial drones registered, 527,174 recreational drones registered, [and] 3,612 paper registrations. There are 259,207 Remote pilots certified (FAA, n.d.-a). One can only guess the true number, since the FAA website only

covers registered sUAS. There are several unregistered sUAS which are not accounted for in this metric either due to sUAS owners' refusal or ignorance. The potential millions of drones in the United States alone can be used to launch attacks against critical infrastructure.

Critical Infrastructure

There are many sectors of critical infrastructure which are vulnerable to drone attacks. The importance section is broken into various subsections organized by critical infrastructure. The Cybersecurity and Infrastructure Security Agency's (CISA's) 16 critical infrastructure sectors include chemical, commercial facilities, communications, critical manufacturing, dams, defense industrial base, emergency services, energy, financial services, food and agriculture, government facilities, healthcare and public health, information technology, nuclear reactors materials and waste, transportation systems, and water and wastewater systems. All sectors are vulnerable to attack by drones. Certain sectors will be expanded upon below with specific examples to highlight the importance of proper C-sUAS tactics.

Chemical Sector

The chemical facilities sector is extremely interesting to analyze. Terrorism can include biological, and chemical warfare which can stem from theft at such facilities. Thus, contents must be safeguarded from the hands of malicious actors. Drones can be used to scout and surveil facilities. Drones can also be used to steal company proprietary information. This must be thwarted.

Drones can also carry and drop payloads such as thumb drives for phishing. Siegfried, a drug ingredient manufacturer, was breached with malware on May 21, 2021 (Bomgardner, 2021).

With connections to big pharma companies like Pfizer, this should be concerning since access to lifesaving drugs can be hampered. Imagine if malicious actors destroy lifesaving COVID vaccines using a drone.

Commercial Facilities Sector

In the midst of the unexpected COVID-19 pandemic, CISA has released a bulletin on "UNAUTHORIZED DRONE ACTIVITY OVER SPORTING VENUES." Since the novel Coronavirus resulted in state mandates restricting public gatherings, spectators decided to take matters into their own hands to get "real-time drone footage" (CISA, n.d.). CISA's bulletin addresses prevention, protection and response. As part of the protect section, it states to "identify drone detection resources" (CISA, n.d.). Most tactics for prevention in the bulletin cover regulatory countermeasures such as temporary flight restrictions or posting signs to prohibit drone use, but there are no tactics or procedures discussed for combatting drone use.

On February 23, 2022, the Singapore branch of the China Railway First Group was fined $22,000 for flying a DJI Mavic 2 Zoom in controlled airspace without a permit. The group endangered the public and aircraft operating in the vicinity of the construction site at 62A Marymount Road. The incident was an unintentional breach of controlled airspace, since the company was unaware of the sUAS rules pertaining to the location (Shiying, 2022).

On July 5, 2019, two versions of political propaganda flyers reading "STOP THE PRESS," or containing a swastika were dropped over both an Ariana Grande concert in Sacramento, CA and Sacramento State University via drone (Papenfuss, 2019). The drone-based dissemination of propaganda and hate speech is a threat to the people attending commercial facilities sector locations.

Imagine a malicious payload such as explosives or a chemical agent being deployed remotely by drone. The amount of chaos, fear and damage caused would be fiscally efficient without the requirement of any martyrs. Commercial facilities sector infrastructure tends to draw crowds of people making it a target with a potential for high fatality counts.

Government Facilities Sector

Drone defense capabilities at government facilities are severely lacking. “A month after an explosives-laden drone targeted U.S. forces at an Iraq base, the top American commander for the Middle East says finding better ways to counter such attacks is a top priority, and the United States is still behind the curve on solutions. Marine Gen. Frank McKenzie told reporters traveling with him that the use of small drones by Iranian-backed militias is only going to grow in the next few years” (Baldor, 2021). Research must be conducted to enhance drone defense capabilities to prevent explosive carrying drones from penetrating military base airspace. Troops are extremely vulnerable on bases to sUAS based attacks even in the future according to General McKenzie.

Another attack on a government building can be exemplified. “On January 23, 2015, a UAV invaded the White House for espionage activities” (Yang, Huo, Jiang, Zhao & Qiu, 2016). Imagine if an armed drone invaded the White House. It would cause panic among the general public and the executive branch. John F. Kennedy was assassinated by Lee Harvey Oswald who was a spectator of the motorcade and was physically present at the event. Of possible concern is the fact that future assassinations can theoretically be conducted remotely if using an armed drone. Government facilities must be secure with C-sUAS technology to maintain secrecy and personnel safety.

A third attack is the unauthorized drone use over the Taj Mahal in India, since it is a government landmark and sacred place. On March 1, 2022, at approximately 2:50 PM, the drone

entered a high security no fly zone. According to the article, this occurrence is one of many in the last several years at the Taj Mahal (IANS, 2022).

A fourth example is the Russia-Ukraine War where both Russia and Ukraine are deploying exploding "kamikaze" drones which dive from the air and explode on impact (De Vynck, Verma & Baran, 2022). World War 2 had traditional kamikaze pilots who crashed into strategic targets in order to damage them. This resulted in the loss of the aircraft and the pilot. In this new era, remote kamikaze pilots can cause the same damage using a cheaper aircraft without sacrificing pilot lives.

Transportation Systems Sector

On January 31, 2017, a man was arrested for almost flying a drone in an unsafe manner causing a collision hazard to a police helicopter during a rescue mission (Vanian, 2017). The Coast Guard commonly launches rescue missions from the beach, and drone activity is just another variable that rescue mission planners have to consider in today's world.

The "UK Airprox recorded 117 'near-miss' drone plane incidents in 2018" (O'Malley, 2019). During the 2018 Christmas travel time, drones were sighted operating within the restricted airspace surrounding London's Gatwick Airport. The sightings disrupted operations for two days and ruined an estimated 110,000 passengers' travel plans.

While the UK Airprox incident only had near-miss incidents, it had the potential to be a lot worse. The incident also highlights weaknesses in the sUAS defense. Airplanes, people or the airport could have been targeted. Midair collisions are also extremely hazardous to aircraft taking off or landing causing structural damage or even failure. A drone could have been flown into a jet engine which could have disabled the aircraft. 110,000 passengers' travel plans getting ruined is relatively mild compared to the hypothetical scenarios discussed above.

In the end, it is concerning that London Gatwick Airport helplessly cancelled flights during the two-day period when drones were sighted rather than having proper C-sUAS technology in place. Proper C-sUAS technology could have ensured that the approximately 110,000 passengers saw their families for Christmas.

On February 21, 2022, a drone carrying explosives attacked the King Abdullah Airport in the Saudi Arabian city of Jizan leaving 16 people wounded. Three were left in critical condition and airport property was damaged. The incident took place only two weeks after a similar drone attack left 12 people injured at Abha International Airport (The Union, 2022).

On February 27, 2022, unauthorized drone operations near the Francisco Sá Carneiro airport caused an Easyjet aircraft to divert to Lisbon and finally arrive at its original destination more than two hours after its scheduled arrival (Lusa, 2022). The sUAS caused unnecessary fuel waste, delays and posed a hazard to the Easyjet aircraft.

On March 2, 2022, a Jazz de Havilland DHC-8-402 passenger aircraft on departure reported sighting a sUAS at 1000 feet within a one nautical mile radius of Vancouver International Airport (CYVR). The sUAS was trespassing into controlled airspace creating a safety hazard for passenger aircraft in the vicinity of CYVR, and the perpetrator was not found (Avrodex, n.d.).

Emergency Services Sector

The emergency services sector must remain vigilant against nefarious sUAS use. According to a report by the United States Department of Justice (2020):

> All law enforcement agencies, whether or not they wish to begin a program for using drones for their own purposes, must consider a related but far more difficult challenge:

> how to anticipate, prevent, detect, and respond to the criminal use of drones, including use by terrorists. Drones can be extremely effective at reconnaissance for criminal purposes because they can fly past bollards, checkpoints, and other security mechanisms. (p. xvii)

The idea of arming drones for security and defense is also taking off among law enforcement and other agencies. A possible Artificial Intelligence (AI) armed drone attack may have transpired in Libya. “Western military experts are assessing whether an autonomous drone operated by artificial intelligence, or AI, killed people — in Libya last year — for the first time without a human controller directing it remotely to do so” (Dettmer, 2021). The US Air Force is also conducting research on armed drones. “The U.S. Air Force has wrapped up the first phase of its Golden Horde demonstration effort, putting the service one step closer to developing swarming smart weapons” (Insinna, 2021).

If law enforcement can arm drones, then nothing is stopping normal citizens from arming drones either. Standoffs and gun fights could be predominantly by drone in the future. Either way, arming drones is of possible concern. In the right hands, it can prevent and stop crime without risking officers on the front lines. However, in the wrong hands this technology can result in massacres where the perpetrator can possibly sit at home on their couch. If this technology were in the wrong hands, effective C-sUAS is a must.

Smuggling

On March 12, 2022, five Khalistani terrorists smuggled arms and narcotics from Pakistan to India using drones (Nanjappa, 2022). From provided pictures, the drone seems like a custom drone running MavLink, but no details were provided on the news article.

On March 10, 2022, a modified DJI Inspire 2 drone was used to drop contraband over the walls of Georgia prisons. All lights were covered to make video and visual detection tougher. After a car chase into the woods, police recovered oxycodone, ecstasy and marijuana (Evans, 2022). Police were only able to recover the drone due to the car-based ground station being found resulting in the subsequent car chase. There were no C-sUAS actions taken by the police when the drone was flying.

Having C-sUAS systems deployed at borders can prevent smuggling events such as the ones listed above.

Drone Autonomy

The use of AI in flying drones also introduces the concern of autonomy with the use of preprogrammed flight plans not requiring a traditional command and control link. The US Air Force's Golden Horde program (mentioned in the paragraph above) is looking into developing weapons that "behave semi-autonomously and use algorithms to seek high-priority targets" (Insinna, 2021).

Theoretically, the same algorithm to seek high-priority targets could be optimized, refined, and applied by terrorists to seek targets which cause the most damage or death. This is even scarier than other hypothetical scenarios, since this means theoretically a terrorist can go to sleep while a drone searches for targets to wreak havoc based on group specific agendas and desires.

Swarms

The concept of drone swarms is also an emerging topic. Most incidents described above involve only one or a few drones. However, the concept of swarms introduces flying multiple drones in unison or formation which can be considered a scaled extension of autonomy. "DARPA's OFFensive Swarm-Enabled Tactics (OFFSET) program" (Pomerleau, 2021) is US government sponsored and has been researching drone swarms operated by a single user.

Current C-sUAS Strategies and Limitations

Having effective C-sUAS strategies for deterring drone use in sensitive areas can prevent events such as the ones detailed above from occurring. sUAS use is increasing and no effective protocols exist for C-sUAS.

Michalski and Michalska (2017) bring up the fact that the French Army trains eagles to intercept and capture drones. Physical C-sUAS responses such as using guns, missiles, and birds typically have range limitations. Using animals and a net with a parachute may be viable for low flying sUAS, but these options do not seem feasible for higher-flying sUAS.

A C-sUAS weapons system which can launch a net with a mounted parachute was developed in 2017. Nets are effective when shot at sUAS, since motors get locked up and the vehicle becomes unusable. However, accuracy and range are a huge tradeoff of this system, since sUAS must be flying low in order for high accuracy C-sUAS using nets.

Another strategy is to use military anti-air missiles for drone defense (Michalski & Michalska, 2017). However, the limitation is that drones do not have a particular heat signature which makes the utilization of heat seeking missiles difficult. It is also important to note that missiles would not typically be used to take out a sUAS. Using missiles can be dangerous, due to

the high risk of collateral damage, and also missile costs also generally greatly outweigh the cost of sUAS. This makes using missiles for C-sUAS highly unfeasible as a countermeasure.

Current C-sUAS strategies and limitations help support the research into the application of cybersecurity in C-sUAS. The first step before activating a countermeasure is to detect drone activity in the first place, which supports the research being conducting in this thesis.

Summary

With the exponential growth of drones with varying payload capabilities, there have to be risk controls in place to ensure safety and prevent disruptions. Even law, policy, and governance have not kept up with the growth of drones which is a large problem.

By looking at the use of drones and incidents involving drones, the detection and classification of sUAS is the first step towards eliminating unauthorized drone activity, which provides the foundation for conducting this thesis research.

CISA's critical infrastructure sectors are all closely intertwined and related. For some sectors, there are examples of drones maliciously being used for destruction or disruption of critical infrastructure. For other sectors, there are examples of attacks which could be modified to use drones. Either way, the point of using different examples is amplify the idea that drone-based attacks are likely going to be a problem in the near future. Proper countermeasures and playbooks are a must to fight the war on drones. System detection and classification are the first two steps of successful C-sUAS.

Objective

The objective of this thesis is to provide a practical framework for the detection and classification of sUAS that emit IEEE 802.11 signals. Much of the technology for this research is already in existence. Network traffic analysis is not a new concept in cybersecurity and is most commonly used to detect indicators of compromise. Taking a similar approach of using network traffic analysis to distinguish specific drone networks versus ordinary networks, a drone detection and classification framework can be created. Evaluating various machine learning algorithms with various feature combinations can also provide an autonomous approach to drone detection and classification. The other option is manual network traffic analysis, which is infeasible by hand, but hard codable into a programming language. This is great for detecting known IEEE 802.11 drones with known characteristics but doesn't take into account the unknown drone which is not in the drone network database. In a day and age where custom drones are easy to build, encountering an unknown IEEE 802.11 drone is certainly a possibility.

The goal of this thesis is to take the various considerations stated above and see if a universal IEEE 802.11 drone detection and classification framework can be created. The intent being that the framework described above can be incorporated into an autonomous drone detection and classification system used by law enforcement agencies, critical infrastructure sectors, airports, federal buildings or the military encountering nefarious use of these sUAS.

Chapter 2

Literature Review

This chapter will provide a comprehensive literature review on various sUAS control methods, C-sUAS, the MavLink protocol and sUAS detection and classification methods

sUAS Control Methods

Altaway and Youssef (2016) identify three different control modes of sUAS such as "remote pilot control, remote supervised control and full autonomous control" (Altawy & Youssef, 2016). Depending on the control mode, the sUAS will be piloted with human input, human supervision or no human interaction. Rather than focusing on just make and model of the sUAS, the method of control should be detected as well. Fully autonomous drones have the ability to fly flight plans without any human input, while remote pilot-controlled drones must be within range of a base station in order to control. Different flight methods have different equipment requirements. A company called DuckDrone created a duck hunting simulation product where the drones follow a return home function once shot by an RF beam (DuckDrone, 2017). This is an example of autonomous flight capabilities. Using this knowledge, equipment emission differences can be used to attempt to classify drones.

C-sUAS Examples

Aaronia (2017) is a German company which specializes in C-sUAS by using pattern triggering for sUAS detection. Drone Detector is a product created by the company. Two types of 3D antennas are used which can either have 8 sectors with 16 antennas or 16 sectors with 32 antennas. The system allows for recording and playback and can surveil a radius up to 5 kilometers (Aaronia AG, 2017). Field tests with the DJI Phantom 4 yielded a detection range of 5 kilometers. The average time for this product to detect an sUAS is between 10 microseconds and 500 milliseconds.

There is research into creating an anti-drone system with multiple surveillance technologies called ADS-ZJU. The three surveillance systems used are audio, video and RF. In a field study, the system is used to detect a DJI Phantom 4 (Shi, Yang, Xie, Liang, Shi & Chen, 2018). The anti-drone system has an automatic jamming unit which does not jam until an intruding drone is detected. The anti-drone system also has a processing unit which handles "drone, acoustic, image and RF feature extraction" for drone detection and classification (Shi et al., 2018).

MavLink Protocol

There is research into reverse engineering private flight control protocols of sUAS in order to understand and "analyze flight control commands" (Ji, Wang, Tang & Li, 2017). Based on their research, the MAVLink protocol is split up into eight fields. Each command is 17 bytes long as well. Knowing how to decipher each command is important to reverse engineering the flight control protocol. The eight fields consist of the message header, payload length, sequence number, system ID, component ID, message ID, payload and checksum respectively (Ji et. al.,

2017). It is inferred that it is cluster number 5 which is responsible for the takeoff of the sUAS. However, cluster number 2 could be a possibility as well. Cluster number 5 and 2 both occur four times. Perhaps one cluster is responsible for takeoff and the other is responsible for landing. While MAVLink is out of scope for this research, it has a packet like structure similar to IEEE 802.11. Hence, the paper provides a framework for packet-based protocol analysis.

sUAS Detection and Classification Methods

There is research which highlights several sUAS detection methods such as audio, video, thermal, radar and radio frequency (RF) detection. Many drone motors emit high frequency noise at 40 kilohertz (KHz) which can aid in audio detection. Video detection involves using cameras which can see up to 350 feet. However, 350 feet away leaves little to no time to invoke a C-sUAS response which is troubling. A challenge is accurately identifying drones without the misclassification as a bird. Thermal detection also seems extremely tricky, since most drones do not have turbofan or turbo-jet engines which emit hot exhaust gasses. Most sUAS are plastic aerospace vehicles with electric motors. Due to the poor conductivity of plastic, thermal detection may not be feasible either. RF detection is a good option which allows for long range detection of sUAS with a chance to invoke a preemptive C-sUAS response. It is also interesting to note that the DJI Phantom uses a 2.4/5.8 gigahertz (GHz) signal for control signals using Frequency Hopping Spread Spectrum or Direct-Sequence Spread Spectrum (FHSS or DSSS) modulation paired with Futaba Advanced Spread Spectrum Technology (FASST) or Lightbridge technology, while video transmission only uses a 2.4 GHz signal with Orthogonal Frequency-Division Multiplexing (OFDM) modulation paired with Lightbridge/Wi-Fi technology. The FCC allocates the 27 and 49 megahertz (MHz) bands for radio control (RC) drones. While newer drones are generally in the 2.4 GHz and 5 GHz range. Some other license exempt bands include 433 MHz,

868 MHz and 5.8 GHz. Generally, the 433/868 MHz bands are used for telemetry while the 5.8 GHz band is used for video and the 2.4 GHz band for control. It is important to note that custom radios can be installed thus configuring the sUAS to operate on other bands. While this is in direct violation of the Federal Communications Commission's (FCC's) rules and mandates, it should be considered as a possible obfuscation technique which can be employed to make detection even harder (Bello, 2019).

Video Detection

In previous research conducted by Unlu, Zenou and Rivere, the authors use a dataset filled with bird and drone profiles for drone detection and classification (2018). The authors use generic Fourier descriptors for shape description which is scalable and effective. Using the algorithm, birds are correctly identified 93.7% of the time, however drones are only correctly identified 64.6% of the time (Unlu et. al., 2018). However, when combined with a neural network, the accuracy jumped to 100%. While there is not much research involving machine learning, seeing Unlu et al.'s success with using machine learning for identification, gives hope to a possible avenue of exploration. All the test data was also in black and white (Unlu et. al., 2018).

Radar

There is research which highlights the importance of passive radar to detect sUAS, since traditional radar can be extremely noisy and give away attempts to utilize radar detection. Passive radar is extremely difficult to detect, since the system collect gains of all antennas and then amplifies the signals. Radial velocity may be extremely important to consider, since sUAS generally fall in the range of [70,120] meters per second. In comparison, a low-speed bird travels

[8, 10] meters per second, a high-speed bird travels [200, 400] meters per second which leaves the general sUAS speed interval untouched. It is unclear which sUAS or class of sUAS Yang, Hou, Jiang, Zhao and Qiu analyzed. There certainly must be outliers to this interval as well. Averaging-Constant False Alarm Rate (A-CFAR), Pulse Compressing, and Digital Beam Forming (DBF) paired with radial velocity analysis can detect sUAS (Yang et. al., 2016).

There is subsequent research by Zhao and Su which analyzes the use of micro-doppler signals for radar-based sUAS detection. The rotation of the rotor blades provides the opportunity for micro-doppler signature classification. The wing beat frequency of birds fall in the range of 2 and 20 hertz (Hz) while the rotation frequency of small sUAS is 100 to 200 Hz. It may be interesting to look into the rotation frequency of larger sUAS and any exceptions to these numbers which may exist. sUAS in hover have rotor blades which operate at the same frequency, but different initial phase angles. Since rotors position can be expressed in polar format, it is important to note that the rotor positions will all be at different phase angles. The field tests from this research only test Root Mean Square Error (RMSE) between the range of 10 and 50 meters. Doppler interference was neglected. Field testing in University Park, PA near an airport and busy university town can help realize the effects of Doppler interference. Doppler interference must be researched in order to ensure real world application (Zhao & Su, 2016).

RF Detection

Radio Frequency (RF) emissions are everywhere and a byproduct of most technology. The most common frequency bands of sUAS signals lie in the 2.4 GHz and 5.8 GHz spectrums. Yang, Qin, Liang, and Gulliver analyze the DJI Phantom 4 sUAS which communicates with its controller using a 2.4 GHz RF signal. The research analyzes emissions at distances of 100 m, 500 m, 1500 m and 2400 m from the receiver. A second experiment analyzes emissions at 2500 m and

2800 m with noise and interference. A third experiment analyzes two sUAS at 2400 m and 2500 m. Using an artificial neural network algorithm can help detect interference. It seems as though background filtering removing static and non-static clutter can help detection rates be near 100% up to 2400 meters (Yang, Qin, Liang & Gulliver, 2018).

Nemer, Sheltami, Ahmad, Yasar, and Abdeen fed sUAS RF signals to XGBoost and K-Nearest Neighbor algorithms to detect and identify sUAS. Their "dataset consists of 227 segments of the recorded RF signals during multiple experiments. Each experiment was conducted using three types of drones/sUAS (Bebop, AR, and Phantom 3). The collected RF [recordings] contain [approximately] 10.25 [seconds (s)] of RF background data when the sUAS are off and 5.25 s of sUAS communication RF data. Data was also recorded for different flight modes: on, hovering, only flying, and flying with video recording" (Nemer et. al., 2021). In the experiment using 10 classes, the classification accuracy was around 99.11%.

Ezuma, Erden, Anjinappa, Ozdemir, and Guvenc attempt to detect and classify sUAS in the presence of external noise such as Wi-Fi and Bluetooth interference (2019). They were able to "achieve an accuracy of 98.13% in classifying 15 different controllers" (Ezuma et. al., 2019). Their research paper shows that real world RF detection is feasible even with various interference sources. Medaiyese, Ezuma, Lauf, and Adeniran take a very similar approach. "Because the UAV communication link operates in the same frequency band (i.e., 2.4 GHz) as WiFi and Bluetooth devices, [a Stacked Denoising Autoencoder-Local Outlier Factor (SDAE-LOF)] model was used to detect the presence of a UAV or UAV controller signal with an accuracy of 89.52%" (Medaiyese et. al., 2021). While these metrics are not as great as Ezuma et. al.'s findings, the metrics are still high enough to support RF detection and classification feasibility.

Wi-Fi Detection

Wi-Fi detection is not a new concept, since Bisio, Garibotto, Lavagetto, Sciarrone, and Zappatore use a Wi-Fi fingerprint-based approach to detect and classify UASs (Bisio et. al., 2018). Bisio et. al. differentiate "control, navigation and video packets" (Bisio et. al., 2018) by extracting individual node conversations and creating traffic flow maps. Average packet length and latency is also analyzed. Bisio et. al. define three flight modes which are classic, stealth, and mixed which refer to whether video is enabled or not (2018). Bisio et. al. were able to detect classic flight mode drones, but had trouble detecting drones using stealth flight modes.

Another example of detection via network traffic analysis is Sciancalepore, Adel Ibrahim, Oligeri, and Di Pietro's approach of designing the Picking a Needle in a Haystack (PiNcH) methodology. The attributes extracted by Sciancalepore et. al. are source and destination MAC addresses, arrival time and packet size. Broadcast traffic is thrown out by Sciancalepore et. al. due to non-variability. PiNcH performed extraordinarily up to 200 meters with detection rates being 97% and greater.

Nie, Han, Zhou, Xie and Jiang discuss detection and identification using Wi-Fi Signals and RF fingerprints (2021). The approach used amplitude and phase characteristics to analyze both Wi-Fi and ordinary RF signals. Nie et. al.'s "experimental results show that the average identification accuracy of the [fractal dimension (FD)] feature can reach 100% when [signal-to-noise (SNR)] = 10dB and 3dB. For [square integrated bispectra (SIB)] and [axially integrated bispectra (AIB)] features, when SNR = 10dB, the average identification accuracy is 96.11% and 97.23%, respectively. When SNR = 3dB, the average accuracy is 93.50% and 95.00%, respectively" (Nie et. al., 2021).

Yaacoub, Noura, Salman, and Chehab provide a comprehensive literature review on sUAS detection methods as well as cybersecurity-based C-sUAS responses (2020). Within the

review, Yaacoub et. al. discuss a network forensics-based approach “which involves the analysis of network data traveling through firewalls or intrusion detection systems. This allows a network-based investigation to detect and identify anomalies in the traffic” (2020).

Transponder Mandate

In 2020, pilots were required to fit and install ADS-B Out on all airplanes. This requirement means that all civil aircraft in controlled airspace are required to continuously transmit aircraft identification, position, speed and direction. Hartmaan and Giles bring up the idea of using Automatic Dependent Surveillance-broadcast (ADS-B) out and other safety/regulatory measures for controlling sUAS (Hartmaan & Giles, 2016). Expanding the requirement to cover sUAS could aid in detection and C-sUAS. Altawy and Youssef also describe implementing collision avoidance risk controls such as ADS-B out and traffic collision avoidance systems (TCAS). While this is a great risk control to implement, it will certainly increase the cost of producing a drone. Hobbyist drones are known to be cheap and implementing expensive ADS-B technology can make hobbyist drones unaffordable for most, since it would increase the manufacturing costs by thousands of dollars.

In 2021, the FAA’s Remote ID mandate went into effect. With the mandate, in flight information such as identification and location have to be broadcast beginning in 2023. In order to meet the rule requirements, remote pilots can operate a drone with Remote ID built in, operate a drone with an external Remote ID broadcast module, or operate without Remote ID equipment in an FAA-recognized identification area (FRIA). This provides accountability for drones operating in an unsafe manner if Remote ID is equipped (FAA, n.d.-b). It is safe to say that a malicious threat actor will not conform to this mandate and thus fly a payload carrying drone without Remote ID equipment in controlled airspace.

Chapter 3

Experimental Design and Methodology

This chapter will cover the equipment and software needed for the experiment, drone specifications, experimental design, and procedure.

Drone Equipment

The drone depicted in **Figure 1** (DJI, n.d.-a) is a DJI Phantom 3 which originally retailed at $799. The DJI Phantom 3 has a gimbal mounted camera which provides stable in-flight images for aerial photography applications.

Figure 1: DJI Phantom 3 Drone (DJI, n.d.-a)

The drone depicted in **Figure 2** (DJI, n.d.-b) is a DJI Tello which retails at $99. The DJI Tello has a camera. The DJI Tello is extremely small and has little to no payload capabilities. It serves as an introductory drone for beginner drone hobbyists. The drone can be integrated with programming languages and fly in various intelligent flight modes eliminating the need for a traditional controller.

Two DJI Tello drones were purchased to verify that a different drone of the same make and model that was not in the training dataset could be correctly detected and classified by the algorithm.

Figure 2: DJI Tello Drone (DJI, n.d.-b)

The drone depicted in **Figure 3** (Amazon, n.d.-a) is a DJI Mavic Air which retailed at $799. The DJI Mavic Air has a camera. The DJI Mavic Air drone can be flown via phone using 802.11, or with a controller using OcuSync. For this research, the drone was flown via phone to enable IEEE 802.11 signal emissions.

Figure 3: DJI Mavic Air Drone (Amazon, n.d.-a)

The drone depicted in **Figure 4** (Amazon, n.d.-c) is a Sanrock U61W which retails at $69.99. The drone is pocket sized and has little to no payload carrying capabilities. The Sanrock U61W has a camera. This drone was used to simulate an unknown drone attacking the drone detection and classification system, and thus was not part of the training set.

Figure 4: Sanrock U61W Drone (Amazon, n.d.-c)

The drone depicted in **Figure 5** (Zuhafa, n.d.) is a Zuhafa JY02 which retails at $61.99. The Zuhafa JY02 has a camera. The Zuhafa JY02 is difficult to fly, since a neutral input does not

cause the drone to hover. The Zuhafa JY02 is not longitudinally statically stable. Remote pilots must be vigilant of this.

Figure 5: Zuhafa JY02 Drone (Zuhafa, n.d.)

The drone depicted in **Figure 6** (Amazon, n.d.-b) is a Hasakee Q10 which retails at $79.99. The Hasakee Q10 has a camera. The Hasakee Q10 is also extremely small and has little to no payload capabilities.

Figure 6: Hasakee Q10 Drone (Amazon, n.d.-b)

Drone Specifications

Table 1 contains aggregated specifications of the drones used for this research. Specifications that were not found by reading manuals or researching online were marked as "Unknown." Data in Table 1 is extracted from manufacturer websites. All drones had cameras. Most drones had access points located in the drone itself. The only exception was the DJI Phantom 3 which had its wireless access point in the controller. DJI drones all had options to secure wireless access points. The DJI Tello can operate with no encryption, but the DJI Phantom 3 and DJI Mavic Air drones cannot run without network encryption. All other drones had no encryption options. Five out of six of the drones solely operate on the 2.4 GHz band, while the DJI Mavic switches between the 2.4 and 5.8 GHz bands based on flight environment. The max transmit power is listed for future research which can calculate the aircraft and controller distance relative to the ground station as a function of the inverse of radius squared.

Drone Make and Model	**Frequency**	**Default Encryption? Option to secure/unsecure?**	**Max Transmit Power**	**Camera?**	**Access Point Location**
DJI Tello	2.4 GHz	Default: Open Option to secure?: Yes	Aircraft: < 20 dBm (FCC)	Yes	Drone
DJI Phantom 3	2.4 GHz	Default: Encrypted Option to unsecure?: No	Aircraft: 27 dBm (FCC) Controller: 19 dBm (FCC)	Yes	Controller
DJI Mavic Air	2.4/5.8 GHz	Default: Encrypted Option to unsecure?: No	Aircraft: 2.4 GHz: 28 dBm (FCC) 5.8 GHz: 31 dBm (FCC) Controller: 2.4 GHz: 26 dBm (FCC) 5.8 GHz: 30 dBm (FCC)	Yes	Drone
Sanrock U61W	2.4 GHz	Default: Open Option to secure?: No	Aircraft: Unknown Controller: 10.53 dBm	Yes	Drone
Zuhafa JY02	2.4 GHz	Default: Open Option to secure?: No	Unknown	Yes	Drone
Hasakee Q10	2.4 GHz	Default: Open Option to secure?: No	Unknown	Yes	Drone

Table 1: Drone Specifications

Detection Equipment and Software

A Dell Latitude 7480 workstation running Ubuntu 18.04.06 was used with Wireshark, Aircrack-ng, tshark and Python 3.

Refer to **Appendix A** and **Appendix B** for the Python code. The following Python libraries were primarily used:

- os
    - Access operating system commands
- pandas
    - Dataframes to hold parsed data
- numpy
    - Various array operations and calculations
- sklearn
    - Various machine learning algorithms
- re
    - Regular expressions for line parsing

Other libraries contained in the code were either unused for future implementation or implemented in commented out code.

Visual Code Studio was used for code development. An external Wi-Fi Alfa AWUS036ACH adapter depicted in **Figure 7** (ALFA, n.d.) was used with the workstation to capture 802.11 traffic. Device specific drivers were required to be installed.

Figure 7: ALFA AWUS036ACH Wi-Fi Adapter (ALFA, n.d.)

The network adapter was manually put into monitor mode using iwconfig. Packet captures were exported to CSV format for Machine Learning analysis using Python.

Experimental Design

The feasibility of 802.11 signal analysis for sUAS detection and classification was explored. The 802.11 signals of drones and regular access points were captured using Wireshark, exported to CSV format, and fed to the machine learning classifier. Known drones were labeled as drones. All other 802.11 signals were labelled as access points for the training set.

Packet Capture Duration

Packet captures with sample duration of 30 seconds were chosen for IEEE 802.11 signal captures in Wireshark. This was chosen using the specification that a typical DJI drone's maximum speed is around 45 miles per hour. Assuming that a highly experienced remote pilot can fly the drone in 10 mile per hour tailwinds, this means the maximum ground speed is about 55 miles per hour. IEEE 802.11 signals for drones typically have a range between 2.5 to 5 miles. Other wireless access points are shielded between numerous walls and thus have a typical signal range of 150-300 feet. Ranges vary based on antenna setup. At 55 miles per hour, a drone is travelling 0.92 miles per minute. At this speed, with the IEEE 802.11 signal range characteristics above, it means that the signal can be detected from 2.72-5.43 minutes away from its destination. Without accounting for computing latency and processing time, a sample duration of 30 seconds gives the algorithm 5.44-10.86 (5-10) chances to successfully identify and classify a drone before it reaches the station.

Calculations were made with the following formulas:

1. Groundspeed=airspeed+tailwind or Groundspeed=airspeed-headwind
2. Max airspeed, and signal range were taken from drone spec sheet
3. Max tailwind component is based on remote pilot flight experience
4. Miles per minute=(miles per hour)/60
5. Detection Minutes=(Signal range in miles)/(Miles per minute)

Algorithms

For the scope of this research, only certain classification algorithms which are part of Python's scikit-learn toolbox were explored for accuracy. These algorithms include Logistic

Regression (LR), Linear Discriminant Analysis (LDA), K-Nearest Neighbors (KNN), Classification and Regression Trees (CART), Gaussian Naïve Bayes (GNB), Support Vector Machines (SVM), and Multi-layer Perceptron Neural Network (MLPNN). A combination of linear and nonlinear algorithms was analyzed. LR and LDA are linear while KNN, CART, GNB, SVM, and MLPNN are nonlinear algorithms. All algorithms were implemented in a supervised machine learning-based approach, even though some algorithms have unsupervised machine learning options.

Drone detection of whether fields came from a wireless access point or drone was done by each algorithm. Drone classification of identifying make and model of drone was also done by each algorithm.

Due to the difficulty in collecting data which can represent the entire drone and access point population, the use of machine learning algorithms should be considered preliminary research.

Logistic Regression

The Logistic Regression supervised machine learning algorithm was written and optimized for binary categorical classification. Logistic regression uses an s curve which is bounded between 0 and 1 to model a binary output variable (Belyadi & Haghighat, 2021). In the case of this research, the binary output variable for detection would take on values of access point or drone. In terms of the classification use case, the categorical output variable is not a binary classification problem. This algorithm is a strong candidate for detection, but a weak candidate for classification. Also, due to the overlap of features in drone networks and ordinary wireless networks, data that falls in the middle of the sigmoid function has the potential to become a false positive or false negative when using a binary classification algorithm.

Linear Discriminant Analysis

The Linear Discriminant Analysis supervised machine learning algorithm was written and optimized for binary categorical classification. Linear Discriminant Analysis essentially evaluates residuals to condense a two-axis problem into a one axis problem by generating a new axis to decrease variance between each class and maximize the distance between each class (GeeksforGeeks, 2021). In the case of this research, the binary output variable for detection would take on values of access point or drone. In terms of the classification use case, the categorical output variable is not a binary classification problem. This algorithm is a strong candidate for detection, but a weak candidate for classification. Also, due to the overlap of features in drone networks and ordinary wireless networks, data that falls on the newly generated axis has the potential to become a false positive or false negative when using Linear Discriminant Analysis for binary classification algorithm.

K-Nearest Neighbors

The K-Nearest Neighbors supervised machine learning algorithm was written and optimized for both classification and regression problems. K-Nearest Neighbors can use either a small K or large K which controls how many nearest neighbors it takes into account. K-Nearest Neighbors uses the K smallest distance vectors based on the data provided. The algorithm essentially uses the training set for real time decision making and does not handle large features, large training data, and categorical variables well (Belyadi & Haghighat, 2021). For the case of this research, the binary output variable for detection would take on values of access point or drone with no categorical features. In terms of the classification use case, the categorical output variable is not a binary classification problem, so it is expected that this algorithm performs well

depending on the training data provided. This algorithm is a strong candidate for classification, but a weak candidate for detection. Also, due to the overlap of features in drone networks and ordinary wireless networks, evaluating the nearest neighbors can lead to large variability and bias when using this algorithm. This can result in misclassification and class bias.

Classification and Regression Trees

Another name for Classification and Regression Trees is Decision Trees. The Classification and Regression Trees supervised machine learning algorithm was written and optimized for both classification and regression problems similar to K-Nearest Neighbors. Classification and Regression Trees' algorithmic performance can be attributed to the quality of binary trees it creates based on numerical or categorical input features (Belyadi & Haghighat, 2021). For the case of this research, the binary output variable for detection would take on values of access point or drone with no categorical features. In terms of the classification use case, the categorical output variable is not a binary classification problem, so it is expected that this algorithm performs well depending on the training data provided. This algorithm is a strong candidate for detection, but a weak candidate for classification due to the fact that it uses binary trees. Also, due to the overlap of features in drone networks and ordinary wireless networks, using binary trees can lead to large variability and bias when using this algorithm as well.

Gaussian Naïve Bayes

The Gaussian Naïve Bayes supervised machine learning algorithm was written and optimized for binary categorical classification. Gaussian Naïve Bayes assumes that the data provided follow a normal distribution and that each feature is independent of the other

(Majumder, n.d.). In the case of this research, the binary output variable for detection would take on values of access point or drone, and the features are independent of one another. In terms of the classification use case, the categorical output variable is not a binary classification problem. This algorithm is a strong candidate for detection, but a weak candidate for classification. Also, due to the overlap of features in drone networks and ordinary wireless networks, data that falls near the mean has the potential to become a false positive or false negative when using this binary classification algorithm.

Support Vector Machines

The Support Vector Machines supervised machine learning algorithm was also written and optimized for both classification and regression problems similar to Classification and Regression Trees and K-Nearest Neighbors. Support Vector Machines attempt to maximize the distance between support vectors unlike K-Nearest Neighbors which minimizes the distance between neighbors. Increased margins between support vectors result in increased algorithmic performance (Belyadi & Haghighat, 2021), however due to the overlap of features in drone networks and ordinary wireless networks, this may be infeasible. Support Vector Machines can be supervised or unsupervised, but for this research only supervised Support Vector Machines will be implemented. The binary output variable for detection would take on values of access point or drone with no categorical features. In terms of the classification use case, the categorical output variable is not a binary classification problem, so it is expected that this algorithm performs well depending on the training data provided. Training data class disparity can again result in misclassification and class bias. This algorithm is also a strong candidate for classification due to linearity of vectors, but also a potentially strong candidate for detection as well, since different kernel functions can be chosen.

Multi-layer Perceptron Neural Network

The Multi-layer Perceptron Neural Network supervised machine learning algorithm was also written and optimized for both classification and regression problems similar to Classification and Regression Trees, K-Nearest Neighbors and Support Vector Machines. Linear or nonlinear activation function can be chosen to optimize the function for each use case. Layers are created, and entries with similar characteristics are grouped into a single layer (Grosse, n.d.). For the case of this research, the binary output variable for detection would take on values of access point or drone with no categorical features. In terms of the classification use case, the categorical output variable is not a binary classification problem, so it is expected that this algorithm performs well depending on the training data provided. Training data class disparity can again result in misclassification and class bias due to layer creation by the neural network. This algorithm is also a strong candidate for classification due to the linear activation function choice, but also a potentially strong candidate for detection as well, since nonlinear activation functions can also be chosen.

Neural networks require large datasets in order to function effectively. An example is Unlu et. al.'s video-based drone detection research which use a dataset consisting of 419 bird silhouettes and 121 drone silhouettes. The size of the drone and wireless access point signal training set is not as large as Unlu et. al.'s training set. Thus, neural networks were explored as an experimental method of detecting and classifying sUAS with no expectation of high performance.

Machine Learning Algorithm Limitations

Machine learning algorithms require an evenly distributed class ratio of approximately one to one. Since the Wi-Fi drones are only a few entries out of many access points, the class

ratio is nowhere close to one to one. Hence, depending on the distribution, machine learning algorithms tend to just classify all entries as access points which can contribute to artificially high accuracy scores. The drone dataset collected is large enough to trust the machine learning results.

Anomaly detection is a more robust approach. At an 802.11 layer however, there are not many anomalous descriptors separating a drone network from an ordinary network. Aggregating drone data is the only way to have a centralized repository that is more descriptive of the Wi-Fi drone population. At the Internet Protocol (IP) layer, there are multiple anomalous descriptors that can be used for anomaly detection.

Data Feature Fields

This section details the various IEEE 802.11 features which were considered for use in the machine learning algorithms. IEEE 802.11 are a set of network standards consisting of hardware protocols for deploying local area networks. **Figure 8** (MathWorks, n.d.) highlights the format of a sample IEEE 802.11 frame with mandatory and non-mandatory fields colored.

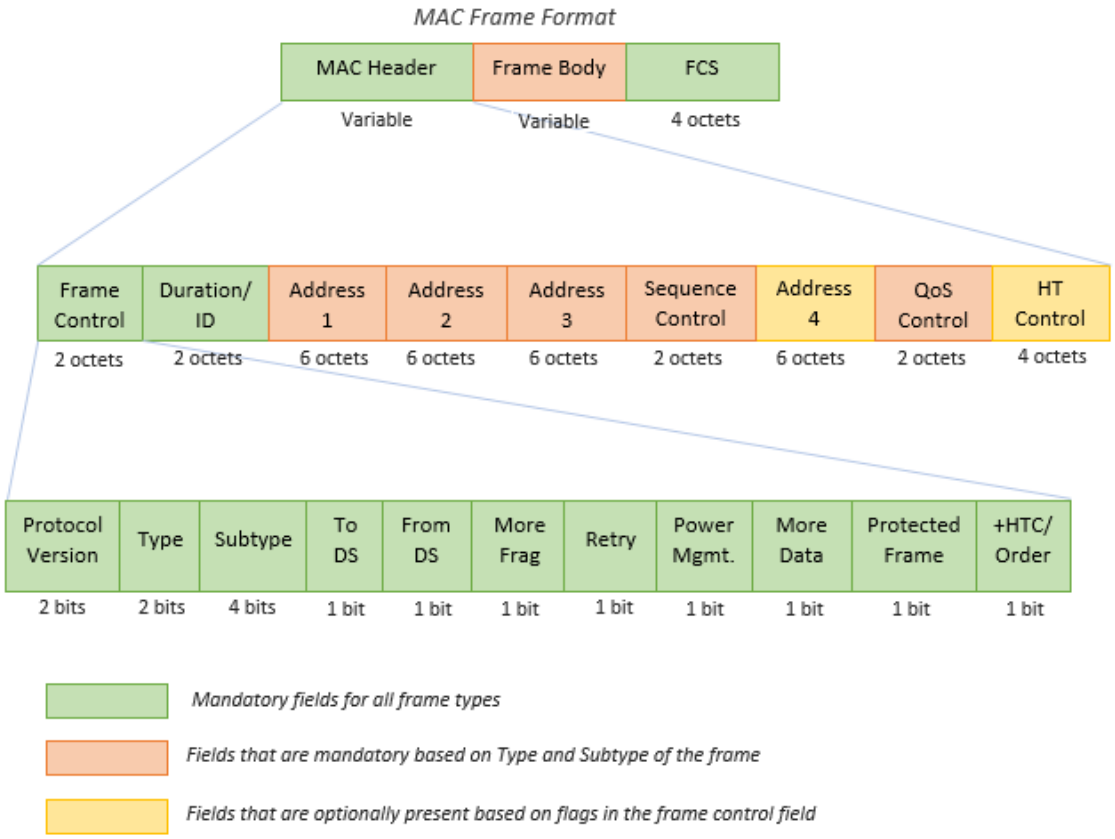

Figure 8: IEEE 802.11 Frame Format (MathWorks, n.d.)

BSSID

This field is the broadcast service set identification (BSSID) which contains Media Access Control (MAC) address information. MAC addresses have the first three octets as an Organizationally Unique Identifier (OUI). While OUI alone can be a key giveaway of drone hardware, MAC addresses can be spoofed. Thus, MAC address alone should not be used to detect the presence of a sUAS. OUIs also do not account for custom built Wi-Fi drones with wireless card manufacturer OUIs. This is an IEEE 802.11 field taken from the “Address 2” source MAC

address field in **Figure 8 (**MathWorks, n.d.) when an access point broadcasts a management frame. This field is not fed to the algorithm due to the next field which normalizes this metric.

Drone OUI?

This is a binary classification field of whether the OUIs of common drone manufacturers were present in the BSSIDs or not. If the OUIs were present, then the value of this field was 1. If not present, then the value of this field was 0. This field is a normalized field derived from BSSID and is not an IEEE 802.11 field.

SSID

SSID is the name of the access point which can be easily renamed in drone and wireless access point settings. **Figure 9** (Verges, 2017) shows the SSID parameter as part of an IEEE 802.11 frame in a sample Wireshark packet capture. This string is user customizable, however default DJI Phantom 3 SSIDs follow the format PHANTOM3_XXXXXX. Default DJI Tello SSIDs follow the format TELLO-XXXXXX. These strings are not unique and multiple access points can have the same SSID.

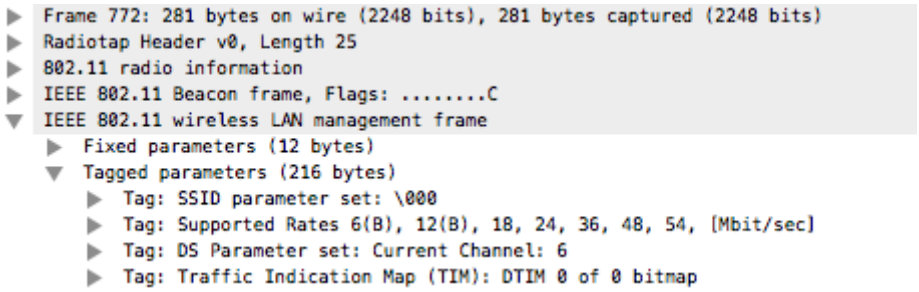

Figure 9: SSID Parameter (Verges, 2017)

This field is not fed to the algorithm due to the next field which normalizes this metric.

Drone SSID?

This field is a binary classification field of whether default drone SSID string formats are present in the SSID field or not. This is derived from the value of the SSID feature field above, and is not a component of an IEEE 802.11 frame. For example, an SSID of PHANTOM3_XXXXXX would have a value of 1 while an SSID of Linksys would have a value of 0.

Channel

There are 14 channels spaced 5 MHz apart for 802.11 wireless access points. Channels are another data point; however, access points optimize channel based on access points in the area to avoid interference. There is not much unique information distinguishing a drone channel vs a wireless access point's channel, so this field is not fed to the machine learning algorithm. Channel information is typically displayed under "Radiotap Header v0" in **Figure 9** (Verges, 2017), so it is part of IEEE 802.11 frames.

Beacons

Beacon frames are IEEE 802.11 management packets which are important for smooth operation of a wireless network. The frames indicate the presence of a wireless access point to potential endpoints. **Figure 10** (Nayarasi, 2014-a) shows the structure of an IEEE 802.11 beacon frame. Every time the frame depicted in **Figure 10** (Nayarasi, 2014-a) is sent by a specific BSSID, the number of beacons associated to a specific BSSID value increments by one.

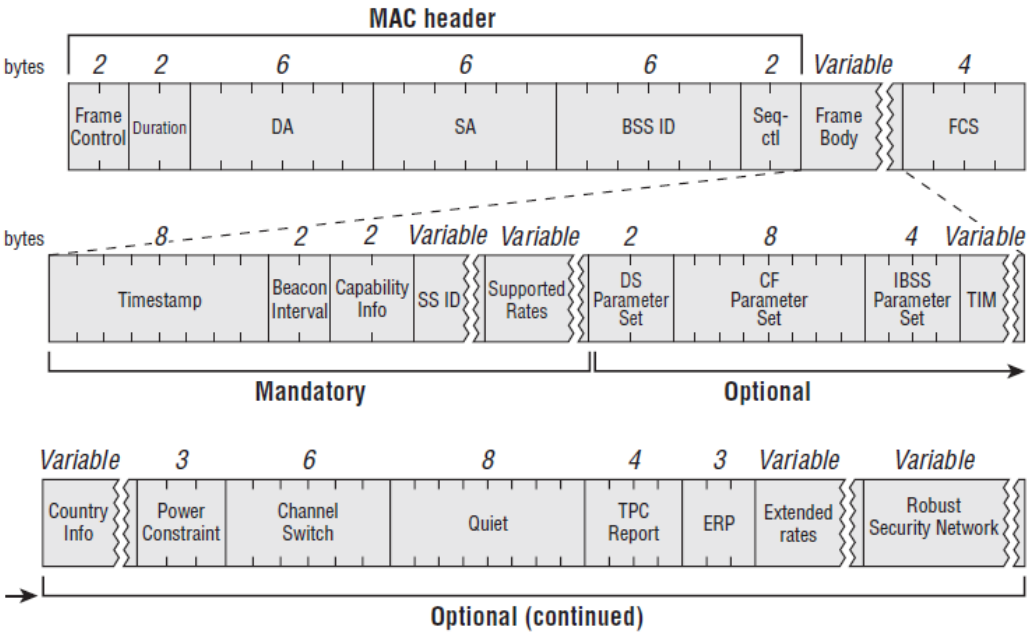

Figure 10: IEEE 802.11 Beacon Frame (Nayarasi, 2014-a)

Beacon Frame Length

As seen in **Figure 10** (Nayarasi, 2014-a), Beacon Frames have a total length in bytes which is the sum of the several subbytes depicted above. Beacon frame length refers to the total length of the beacon frame.

Beacon frames contain information such as timestamp, beacon interval, SSID and traffic indication map. The traffic indication map contains sleeping stations that the access point needs to send data to. Ideally, on a drone network all stations should be active at all times. The only information which can be changed by the user is theoretically the SSID. Thus, this justifies the next field.

This field is not fed to the algorithm due to the next field which normalizes this metric.

Beacon-length(SSID)

Drawing on the beacon frame length field derived from the beacon frame management packet depicted in **Figure 10** (Nayarasi, 2014-a), ideally, no matter what a drone's SSID is, the beacon frame length minus the length of the SSID metric should remain the same, since SSID length in bytes are subbytes of the total beacon frame length. Unless there are sleeping stations which are variable in a drone network, the beacon frame length should stay uniform throughout the drone access point's uptime. Normally in a drone network, there are no sleeping stations. The unique value calculated from subtracting the SSID length in bytes from the beacon frame length in bytes can be descriptive of a specific make and model of drone.

The field value is either the frame length minus the length of the SSID, 0 if there are no beacon frames present during the packet capture, or -1 if the beacon frame length is variable. The field value is also 0 if the unique BSSID is not mapped to a specific SSID.

Data Packets

This field is the total number of IEEE 802.11 data packets transmitted and received by a wireless access point. A sample data packet is shown in **Figure 11** (Nayarasi, 2014-b). If the

BSSID was contained in either the receiver address or transmitter address field, the data packet count for the specific BSSID was incremented by one. This field was not split up into transmitted and received packets, since the ratio of transmitted to received packets is generally one to one. Both transmitted and received packets went into the total count of data packets. In the info section of the 802.11 frame, it contains string QoS Data or other strings to identify a packet as a data packet. Packets with each unique BSSID as a receiver or sender will be counted as part of the data packet count.

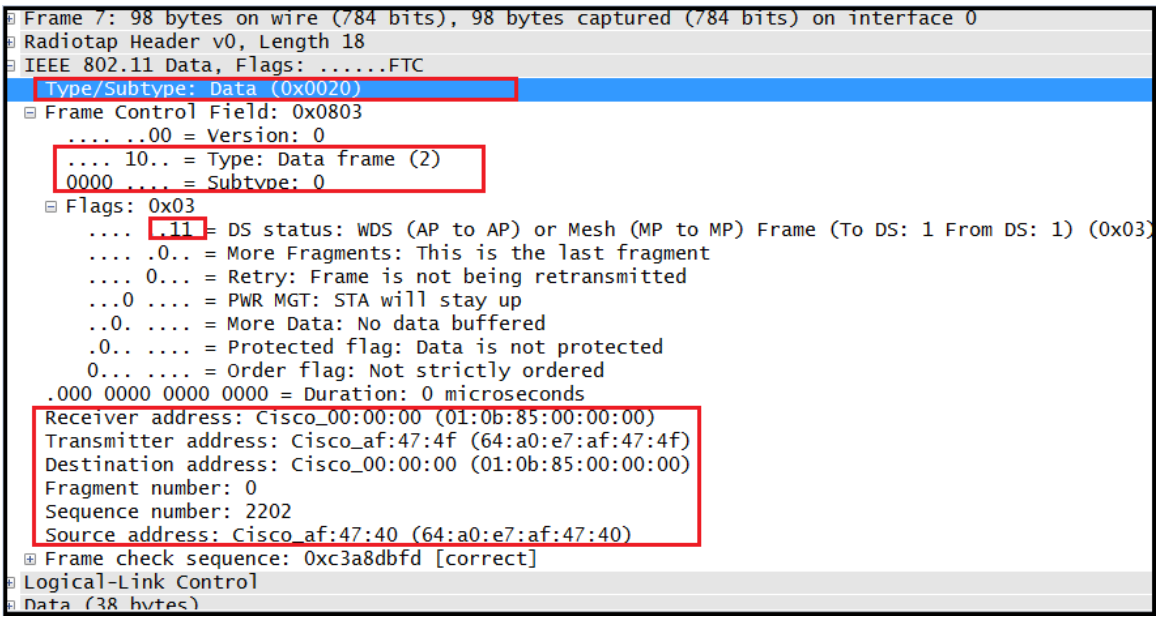

Figure 11: IEEE 802.11 Data Frame (Nayarasi, 2014-b)

Probe Reqs

A probe request is an IEEE 802.11 frame which is sent by clients to discover 802.11 networks in the immediate vicinity. A sample probe request is depicted below in **Figure 12** (Notter, 2017). Probe requests nowadays are extremely noisy, since wireless clients connect to

many networks in today's day and age. For example, an iPhone which connects to a home network, corporate network, guest network and friend's network sends probe requests to all the unique BSSIDs and SSIDs it has ever connected to. Probe requests are also sent to the broadcast address to gather information of all non-hidden 802.11 networks in the area.

This field was not fed to the algorithm, since there is not much differentiating a probe request of an access point versus a drone.

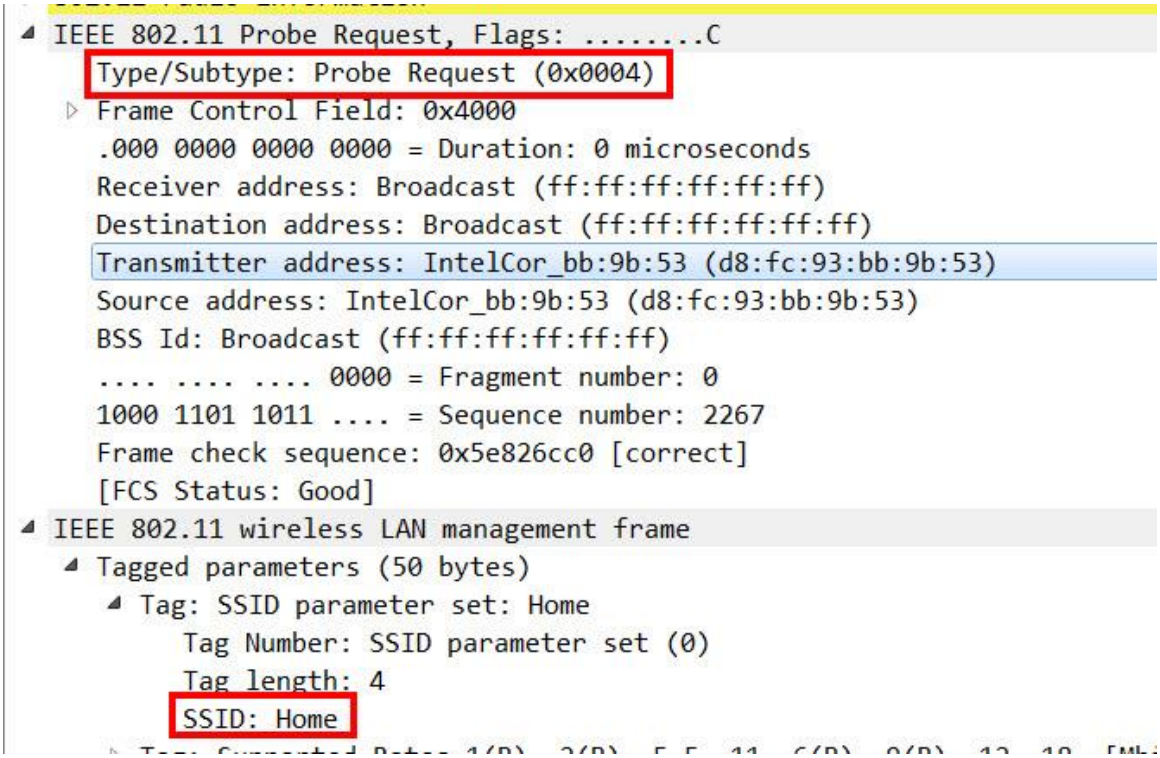

Figure 12: IEEE 802.11 Probe Request (Notter, 2017)

Probe Resp

The probe response is is what is sent back to clients which have sent a probe request either to its unique BSSID or the broadcast address depending on configuration settings. A

depiction of the previous sentence is shown in **Figure 13** (Meraki, 2020). The probe response is an IEEE 802.11 frame which contains data rates and encryption information to provide compatibility information to the client. If the client is compatible and wants to connect, then an authentication request is sent.

This field was not fed to the algorithm, since there is not much differentiating a probe response of an access point versus a drone.

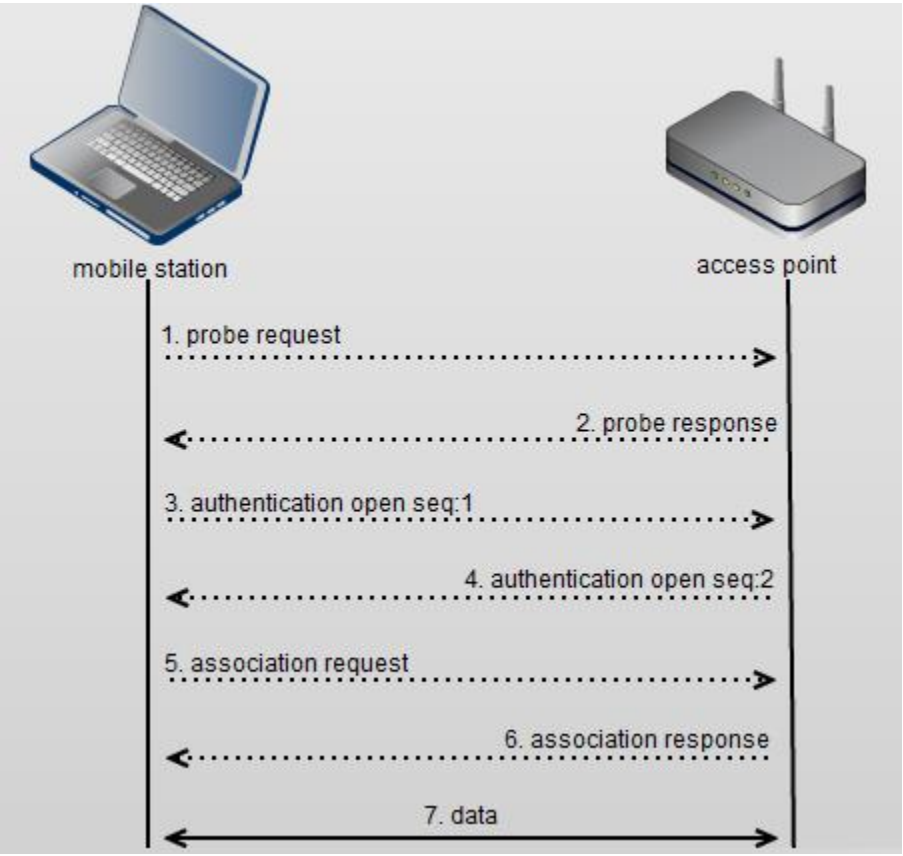

Figure 13: IEEE 802.11 Connection Process (Meraki, 2020)

Auths

This field covers the number of authentications to a network. Authentication facilitates the connection to the wireless access point. Theoretically, a drone launched close to the C-sUAS

ground station can capture authentication packets. In flight, there are normally no authentication packets, so this field value can potentially be zero. Referring back to **Figure 13** (Meraki, 2020), an authentication request is sent once a client wants to connect to the specific station. Authentication requests are an IEEE 802.11 frame.

This field was not fed to the algorithm, since authentications are a normal characteristic of any wireless network. Also in real world flight data, it was incredibly difficult to capture authentications from a takeoff zone located far away from the ground station. Incorporating more powerful signal capture equipment in future research can capture authentications more easily but was out of scope for this research.

Deauths

This field covers the number of deauthentications to a network. Similar to an authentication request shown in **Figure 13** (Meraki, 2020), a deauthentication request can be sent. Deauthentication requests are an IEEE 802.11 frame. Deauthentication can be access point initiated to renew a lease or can be a client side disconnect request. Normally, clients exceed the range of a wireless network, so this does not cause a deauthentication frame to be sent. Depending on make and model of a drone, a deauthentication may be sent upon landing and shut down to signal the end of the flight. Normally the value of this field is also zero in flight.

False positives of the auth/deauths field could be caused by a guest network being in the vicinity of the signal capture system. A place of business offering Wi-Fi can see multiple authentications and deauthentications over time. Normally, client based deauthentications do not happen, since many end users cache credentials after connecting for the first time.

This field was also not fed to the algorithm, since deauthentications are also a normal characteristic of any wireless network. Again, in real world flight data, it was incredibly difficult

to capture deauthentications from a takeoff zone located far away from the ground station. Incorporating more powerful signal capture equipment in future research can capture deauthentications more easily but was similarly out of scope for this research.

Signal Strength Variability

Signal strength variability refers to the variability of the signal strength in decibels. All signal strengths (depicted in **Figure 14** (Wireshark, n.d.-a)) associated with a specific BSSID were aggregated into an array and variability of the array values was calculated. Signal strength is a subset of radio information captured from IEEE 802.11 frames.

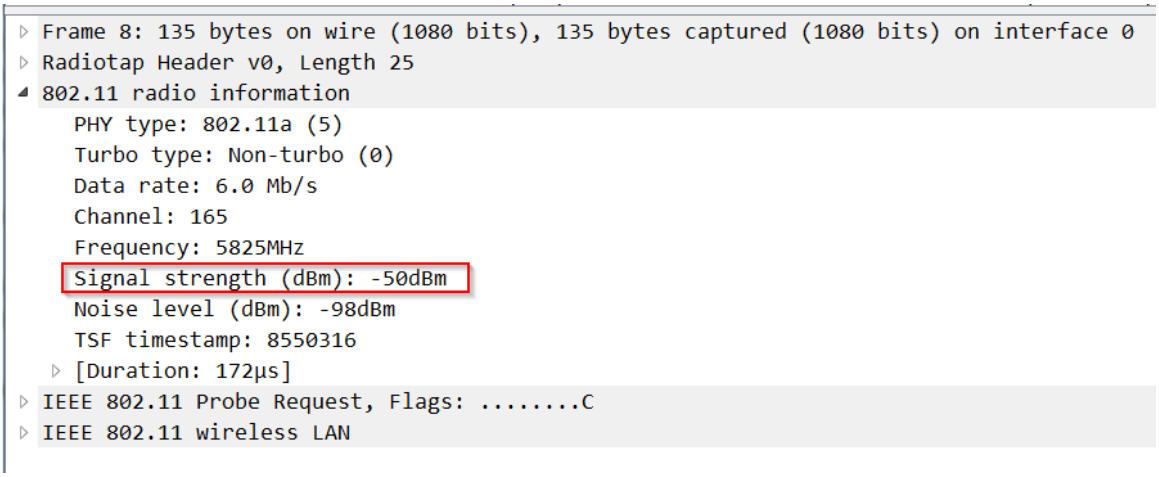

Figure 14: Signal Strength (Wireshark, n.d.-a)

A moving access point with respect to a fixed ground station theoretically has more signal strength variability than a fixed access point with respect to the same fixed ground station. False positives of this metric can include a cell phone's wireless hotspot, or a hotspot enabled car. Drones can travel faster or slower than cars depending on the specific location.

Nodes

Nodes refer to the number of unique MAC addresses each BSSID talks to. As shown in **Figure 15** (Geier, n.d.), there is source address and destination address information contained in IEEE 802.11 packets. The number of unique destination addresses a specific BSSID communicates with is used to calculate the number of nodes. Destination addresses of "FF:FF:FF:FF:FF:FF," and "00:00:00:00:00:00" are omitted from the number of nodes, since these are broadcast addresses. Only data frames are considered to aggregate nodes, since beacon and probe requests can occur between devices that are not necessarily connected to a given wireless network. This essentially provides a baseline of how many different client MAC addresses a given access point's MAC address is communicating with.

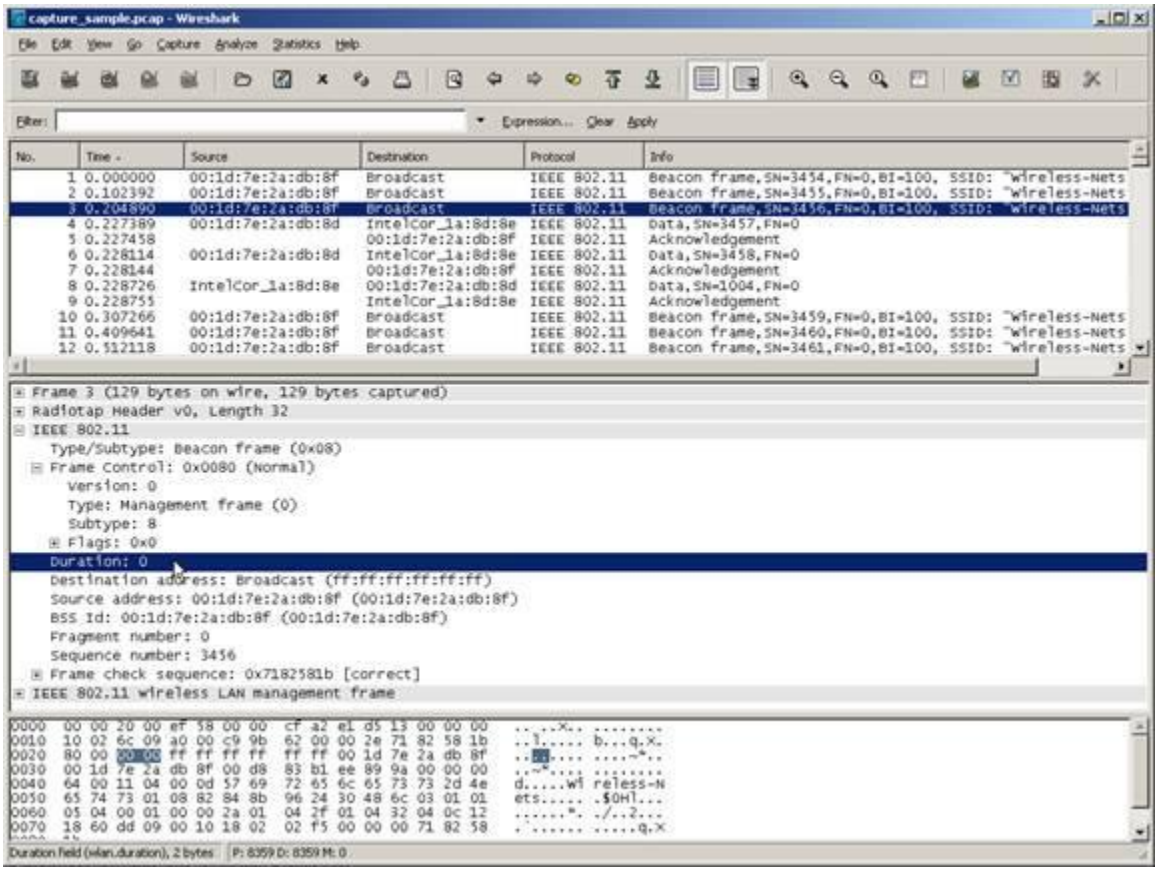

Figure 15: IEEE 802.11 Nodes (Geier, n.d.)

Additional Data Feature Fields after Breaking into Wi-Fi Network

None of the following fields were analyzed, since it requires breaking into a Wi-Fi network. This is generally illegal and thus was not implemented in the machine learning algorithm. Drone network traffic was analyzed, and a corresponding table was assembled.

Header

This value is derived from the Transmission Control Protocol or User Datagram Protocol (TCP or UDP) stream by looking at the first two bytes of each conversation. The headers of streams from drone networks should be different from the header streams from normal networks. A network with multiple unique headers has a field value -1. A network with a single header has a field value of the unique header. For example, a characteristic of the DJI Tello's UDP stream are headers which always start with CC. Using headers can provide make and model information and help differentiate between a normal wireless access point or drone. By following protocol streams as depicted in **Figure 16** (Wireshark, n.d.-b), analyzing the first few bytes of each conversation can help map stream related headers.

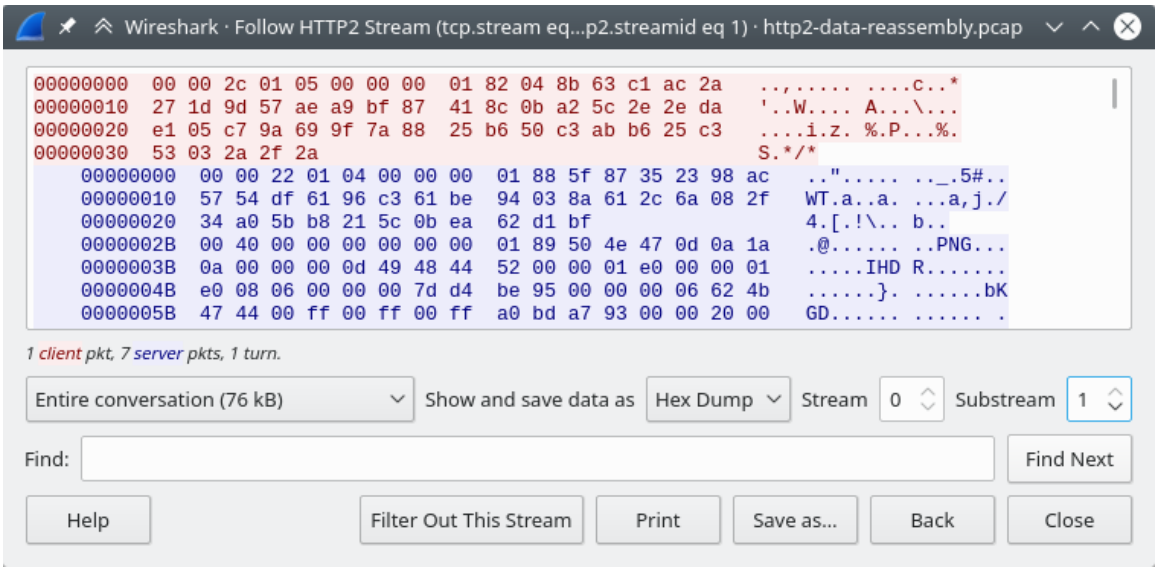

Figure 16: Stream Headers (Wireshark, n.d.-b)

TLS/SSL

This field has a value of 0 (Transport Layer Security or Secure Sockets Layer (TLS or SSL) not present) or 1 (TLS or SSL present) based on if there is any TLS or SSL traffic present. Drones in general should not have any TLS or SSL present due to the computing resources required for running protocols typically reserved for web servers. A sample packet capture containing TLS packets from Wireshark is shown in **Figure 17** (Ghosh, 2021).

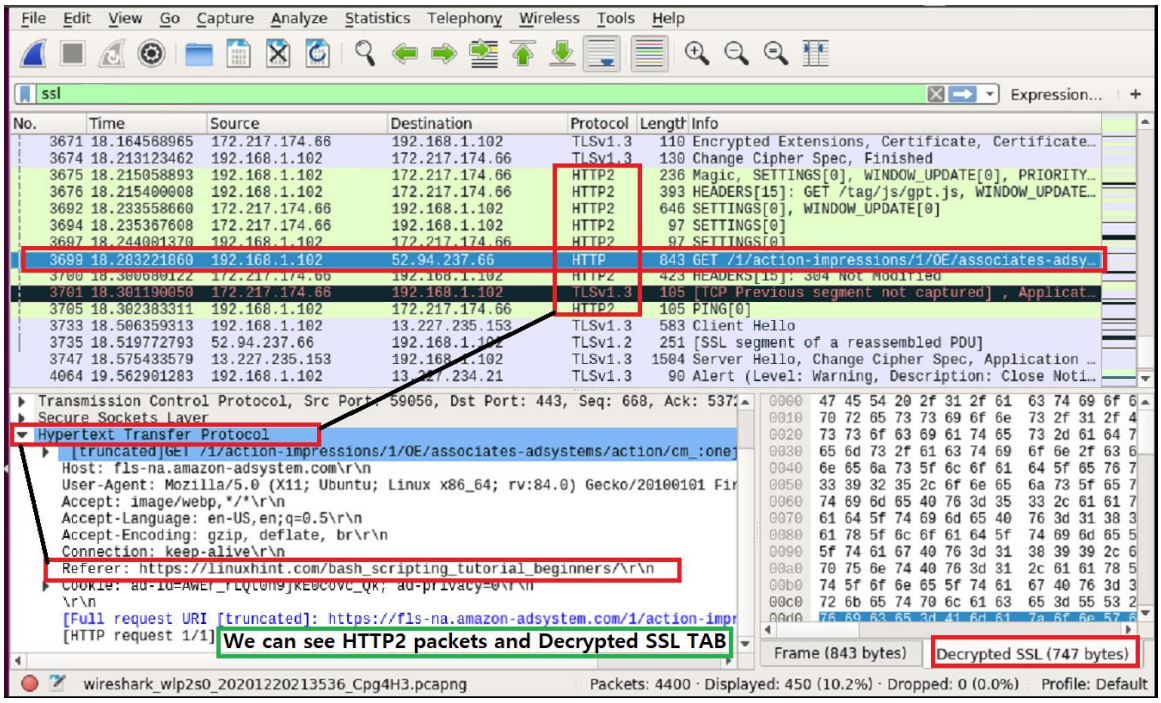

Figure 17: TLS Packet Capture (Ghosh, 2021)

IP Nodes

Similar to the nodes field discussed earlier, this field covers the number of unique IP addresses that an access point services. Instead of using source and destination MAC addresses, IP addresses are used. One IP address (typically .1) is that of the router. Thus, the number of IP addresses should be subtracted by one to identify the number of nodes that the access point services.

Drone networks generally have a low number of IP nodes compared to a normal enterprise or home network. A false positive of this metric is mobile hotspots which generally support a low number of hosts as well.

Classifier Fields

Drone/AP

This is a binary classifier. This field is either classified as drone or access point by the algorithm based on the data feature fields.

Make and Model

This potential drone classification field is either labeled AP signifying an access point or has make and model information about the drone. A sample value would be DJI Phantom 3. This was hard coded into the machine learning project and also fed through the algorithm, since the sample size of the same make and model of drone was extremely small, and small sample sizes are unreliable. The results determined classification effectiveness.

Manual Equipment Setup

Monitor Mode

The network card must be put into monitor mode in order to pick up IEEE 802.11 packets. The AWUS036ACH can be put into monitor mode using the following commands:

1. sudo ip link set [interface name] down
2. sudo iw [interface name] set monitor none
3. sudo ip link set [interface name] up

Procedure

The procedure is split up into 3 parts of manual network traffic analysis, automated network traffic analysis and flight data analysis. At the beginning, the only drone data collected was from the DJI Phantom 3 and DJI Tello. Later, more drones were acquired and added to the dataset.

1. Manual Network Traffic Analysis

    To begin, manual network traffic analysis of the drones was done. The network card was manually put into monitor mode and Wireshark was run to capture data. The propellers were taken off the drone and commands such as takeoff, land, pitch, roll and yaw were issued.

    Then, depending on the default Wi-Fi security employed on the drone, the security was either turned off or turned on to capture data at both a Layer 2 and Layer 3 level.

    Each packet capture was saved in an individual folder for drone data. The only metric that could not be used reliably was the signal strength variability, since the drone was not actually flying at this stage.

    At this stage, distinguishing characteristics included beacon frame lengths and beacon frame length minus the length of the SSID. This was verified by changing the SSID name in the individual drone settings and validating whether the beacon frame length minus the length of the SSID metric stays the same for all drones. Other distinguishing characteristics included the limited number of communication nodes, and large number of data packets on a drone network.

2. Automated Network Traffic Analysis

Python code (**Appendix A** and **Appendix B**) was used to parse 30 second long tshark capture output files to a csv file. The features extracted include BSSIDs, SSIDs, signal strength variability, number of beacon frames, beacon frame length, beacon frame length minus length of SSID, number of data frames, and number of hosts each BSSID communicates with. Additional fields such as Drone OUI, and Drone SSID are parsed from two dictionaries of known drone SSIDs and known drone OUIs. The file is then fed to the machine learning algorithms listed under the algorithms section which outputs a separate prediction csv file.

3. Flight Data

Python code was running while each drone was individually flown towards a static ground station. This simulates a payload carrying drone flying towards a sensitive location where the ground station is located. The DJI Phantom 3, one DJI Tello, the DJI Mavic Air, the Zuhafa JY02, and the Hasakee Q10 were incorporated in the drone class of the training dataset which had five entries. The aforementioned drones were flown in the manner described in the first sentence of the paragraph. The Sanrock U61W was excluded from the training dataset to simulate an unknown drone attacking the location of the ground station. The second DJI Tello was also excluded from the training dataset to simulate a known drone attacking the location of the ground station.

Chapter 4

This chapter will summarize findings made during drone network traffic analysis and experimental results of machine learning algorithm efficiency.

Beacon Frame Length

Network traffic analysis revealed that the beacon frame length of a DJI Phantom 3 with the default SSID following the format PHANTOM3_XXXXXX is 227 bytes. After renaming the DJI Phantom 3's SSID to SSID name "Rahul," the beacon frame length is 217 bytes.

Similarly, the beacon frame length of the DJI Tello with the default SSID following format Tello-XXXXXX is 162 bytes. After renaming the DJI Tello's SSID to SSID name "Rahul," the beacon frame length is 155 bytes. This affirms that a drone network typically has no sleeping stations, and that beacon frame length is variable due to SSID length, thus justifying the Beacon-length(SSID) field which normalizes the beacon frame length field.

To add, the Beacon-length(SSID) field for both DJI Tello drones is 150 bytes. Using the data from both DJI Tellos, it is assumed that beacon frame lengths stay the same across multiple drones of the same make and model.

Training Set

The training set used to train the machine learning models consisted of five drones (the DJI Phantom 3, one DJI Tello, the DJI Mavic Air, the Zuhafa JY02, and the Hasakee Q10 drones

discussed in **Chapter 3**) and 95 wireless access points. This sample is not representative of the population consisting of all 802.11 drones and wireless access points, so the results below are not generalizable. This is a limitation of the study due to budget constraints and ease of access for data collection. Accuracy scores are not reliable due to the variability with small data sets. However, the study is still interesting because it explores machine learning applications with an incomplete training set to prove a proof of concept. Successful detection of known and unknown drones not contained in the dataset should be considered research findings, since it supports the feasibility of a machine learning based IEEE 802.11 drone detection and classification system.

Training set data is shuffled using kfold and split using n_splits=5 for detection and n_splits=2 for classification. Since only five drones are in the dataset, shuffling can cause uneven train/test sets. Having a one-to-one ratio of wireless access points to drones can improve the class imbalance present.

Results

Machine learning algorithms are tabulated and listed below. The first section contains test and validation detection metrics, the second section contains model.predict() detection accuracy metrics, and the third section contains model.predict() classification accuracy metrics.

Test and Validation Results

The results below comprise of test and validation statistics of a training set with 95 and n (n changes with respect to the various data samples below) number of drones in the training set.

Two Drone Dataset

The training set used for test and validation only consisted of two drones and 95 wireless access points.

Two Drone Machine Learning Algorithm Detection Metrics

The baseline accuracy metrics in **Table 2** look excellent from a high level. From this accuracy score distribution, it seems that Support Vector Machines, Linear Discriminant Analysis and K-Nearest Neighbors were tied for the best performing algorithms with the same mean accuracy and standard deviation values of 0.950000 and 0.100000 respectively. The Multi-layer Perceptron Neural Network algorithm's performance was not comparable to the neural network algorithm's performance in Yang et. al.'s research.

Algorithm	**Mean of Accuracy**	**Standard Deviation of Accuracy**
Logistic Regression	0.883333	0.145297
Linear Discriminant Analysis	0.950000	0.100000
K-Nearest Neighbors	0.950000	0.100000
Classification and Regression Trees	0.941667	0.118145
Gaussian Naïve Bayes	0.850000	0.213437
Support Vector Machines	0.950000	0.100000
Multi-layer Perceptron Neural Network	0.883333	0.145297

Table 2: Mean and Standard Deviation of Detection Accuracy for Various Machine Learning Algorithms with Only DJI Tello and DJI Phantom 3

As discussed in Chapter 3, the only drone data collected in the training dataset was just from two drones. This uneven class ratio distribution can yield high false negative rates, since the algorithm likely evaluated all data as wireless access points. More drone data is required to build a more effective and robust machine learning algorithm. For this reason, anomaly detection algorithms were the best option for this scenario. Thus, confusion matrices and more in-depth data was not collected for machine learning algorithms in **Table 2**.

Four Drone Dataset

The training set used for test and validation only consisted of four drones and 95 wireless access points.

Four Drone Machine Learning Algorithm Detection Metrics

The baseline accuracy metrics in **Table 3** also look promising. From this accuracy score distribution, it seems Linear Discriminant Analysis and Logistic Regression were almost tied for the best performing algorithms with approximately the same mean accuracy and standard deviation values of around 0.97 and 0.03 respectively. Even for the four-drone dataset, the Multi-layer Perceptron Neural Network algorithm's performance was not comparable to the neural network algorithm's performance in Yang et. al.'s research.

Algorithm	**Mean of Accuracy**	**Standard Deviation of Accuracy**
Logistic Regression	0.971429	0.034993
Linear Discriminant Analysis	0.972381	0.033860
K-Nearest Neighbors	0.930476	0.002333
Classification and Regression Trees	0.958095	0.056697
Gaussian Naïve Bayes	0.958095	0.034259
Support Vector Machines	0.944762	0.027701
Multi-layer Perceptron Neural Network	0.943810	0.028156

Table 3: Mean and Standard Deviation of Detection Accuracy for Various Machine Learning Algorithms with Four Drone Training Set

Five Drone Dataset

The training set used for test and validation consisted of all five drones and 95 wireless access points.

Five Drone Machine Learning Algorithm Detection Metrics

The baseline accuracy metrics in **Table 4** also look excellent for all algorithms due to all algorithms having an accuracy score above 93%. From this accuracy score distribution, it seems that Gaussian Naïve Bayes performed best with the five-drone dataset. Gaussian Naïve Bayes had mean accuracy and standard deviation values of approximately 0.98 and 0.03 respectively. Linear Discriminant Analysis still was one of the better performing algorithms with mean accuracy and

standard deviation values of approximately 0.97 and 0.03 respectively. Classification and Regression Trees also had the same metrics and Linear Discriminant Analysis. Subsequently using the final five drone dataset, the Multi-layer Perceptron Neural Network algorithm's performance was not comparable to the neural network algorithm's performance in Yang et. al.'s research, but to reiterate, the training set composed is not representative of the population of IEEE 802.11 networks.

Algorithm	**Mean of Accuracy**	**Standard Deviation of Accuracy**
Logistic Regression	0.944762	0.027701
Linear Discriminant Analysis	0.972381	0.033860
K-Nearest Neighbors	0.944762	0.027701
Classification and Regression Trees	0.972381	0.033860
Gaussian Naïve Bayes	0.985714	0.028571
Support Vector Machines	0.930476	0.002333
Multi-layer Perceptron Neural Network	0.930476	0.045236

Table 4: Mean and Standard Deviation of Detection Accuracy for Various Machine Learning Algorithms with Five Drone Training Set

Five Drone Machine Learning Algorithm Classification Metrics

Using the full five drone dataset, make and model sUAS classification using test and validation data was tested for the first time with results shown in **Table 5**.

The baseline classification accuracy metrics in **Table 5** also look excellent. From this accuracy score distribution, it seems that Classification and Regression Trees performed best with the five-drone dataset. Classification and Regression Trees had mean accuracy and standard deviation values of approximately 0.97 and 0.00 respectively. No other algorithm performed nearly as well as Classification and Regression Trees.

Algorithm	**Mean of Accuracy**	**Standard Deviation of Accuracy**
Logistic Regression	0.930476	0.000000
Linear Discriminant Analysis	0.958333	0.013889
K-Nearest Neighbors	0.930556	0.013889
Classification and Regression Trees	0.972222	0.000000
Gaussian Naïve Bayes	0.944444	0.055556
Support Vector Machines	0.930556	0.013889
Multi-layer Perceptron Neural Network	0.944444	0.000000

Table 5: Mean and Standard Deviation of Classification Accuracy for Various Machine Learning Algorithms with Five Drone Training Set

Summary

Test and validation results are promising for getting baseline metrics on a static dataset and great for effectively training machine learning algorithms. Metrics can be skewed and bias

when using a training and validation dataset. Also, this training set is not representative of all 802.11 stations in existence. For effective drone detection and classification, models must effectively predict whether a given 802.11 station is a drone or ordinary access point.

Drone Detection

Using model.predict(), test data consisting of one drone and 59 access points was fed to the various machine learning algorithms for predictions. Either the full feature set consisting of drone OUI, drone SSID, Beacon-length(SSID), number of beacons, number of data packets, signal strength variability, and client nodes or the limited feature set which only consisted of number of beacons, number of data packets, signal strength variability, and client nodes was fed to the various machine learning algorithms to see whether each algorithm could distinguish a drone versus an access point.

Unknown Drone Detection

The Sanrock U61W drone not contained in the training dataset was flown with noise from other ordinary wireless access points to see if the algorithm can detect it. This was done to simulate an unknown drone attacking a wireless access point, since even the best training sets may not be inclusive of all IEEE 802.11 drones.

Full Feature Set

Table 6 consists of machine learning algorithm detection prediction accuracy metrics using the full feature set for the unknown drone flight data.

Linear Discriminant Analysis, K-Nearest Neighbors, Classification and Regression Trees, Support Vector Machines, and Multi-layer Perceptron Neural Network classified everything as a

normal access point and was unable to detect the unknown drone. Only Logistic Regression and Gaussian Naïve Bayes correctly detected the drone. Logistic Regression was the best performing algorithm

Algorithm	**Accuracy**	**Correctly Detected Drone?**
Logistic Regression	0.95 (57/60)	Yes
Linear Discriminant Analysis	0.98 (59/60)	No
K-Nearest Neighbors	0.98 (59/60)	No
Classification and Regression Trees	0.98 (59/60)	No
Gaussian Naïve Bayes	0.93 (56/60)	Yes
Support Vector Machines	0.98 (59/60)	No
Multi-layer Perceptron Neural Network	0.98 (59/60)	No

Table 6: Unknown Drone Full Feature Set Detection Metrics for Various Machine Learning Algorithms Using Model.predict()

Subpar detection performance was to be expected by the nonlinear classification algorithms, due to the class disparity leading to overfitting by K-Nearest Neighbors, Classification and Regression Trees, Support Vector Machines, and Multi-layer Perceptron.

Good performance by the binary classification optimized algorithms such as Logistic Regression, and Naïve Bayes was to be expected. An anomaly is Linear Discriminant Analysis' performance, but it seems that the normalized axis generated by the training set was not representative of the unknown drone. This is always a potential issue when non-training set data is fed to Linear Discriminant Analysis.

Limited Feature Set

Table 7 consists of machine learning algorithm detection prediction accuracy metrics using the limited feature set for the unknown drone flight data.

Algorithm	**Accuracy**	**Correctly Detected Drone?**
Logistic Regression	0.93 (56/60)	Yes
Linear Discriminant Analysis	0.93 (56/60)	Yes
K-Nearest Neighbors	0.98 (59/60)	No
Classification and Regression Trees	0.90 (54/60)	Yes
Gaussian Naïve Bayes	0.93 (56/60)	Yes
Support Vector Machines	0.98 (59/60)	No
Multi-layer Perceptron Neural Network	0.93 (56/60)	Yes

Table 7: Unknown Drone Limited Feature Set Detection Metrics for Various Machine Learning Algorithms Using Model.predict()

K-Nearest Neighbors, and Support Vector Machines classified everything as a normal access point and was unable to detect the unknown drone. Logistic Regression, Linear

Discriminant Analysis, Gaussian Naïve Bayes, and Multi-layer Perceptron correctly detected the drone. The Classification and Regression Trees algorithm also detected the unknown drone as a drone, but also classified 6 access points as drones as well. With the limited feature set, the Classification and Regression Trees algorithm was about 90% accurate (which is an improvement from the full feature set's algorithm performance) while correctly classifying the unknown drone as a drone. The best performing algorithm which correctly detected the drone was a tie between Logistic Regression, Linear Discriminant Analysis, Gaussian Naïve Bayes, and Multi-layer Perceptron.

Subpar detection performance was to be expected by the nonlinear classification algorithms, due to the class disparity leading to overfitting by Support Vector Machines and K-Nearest Neighbors.

Good performance by the binary classification optimized algorithms such as Logistic Regression, Linear Discriminant Analysis, and Naïve Bayes was to be expected. The decrease in features led to decreased algorithm performance in the case of binary classification algorithms. An anomaly is Classification and Regression Trees' performance, but it is biased based on its training data. Since the limited feature set training data only had numerical features, the algorithm's performance can be attributed to the binary decision tree's low error. Multi-layer Perceptron performs best when layers are independent and easily distinguishable. It is assumed that the limited feature set led to better layer grouping by drone vs access point by Multi-layer Perceptron.

Known Drone Detection with Second DJI Tello

The algorithm was tested with a different DJI Tello from the one contained in the training dataset. The first DJI Tello was in the training set while the second DJI Tello was flown to see if

the algorithms could detect it. Theoretically, the algorithms should detect any DJI Tello drone flown towards it.

Full Feature Set

Table 8 consists of machine learning algorithm detection prediction accuracy metrics using the full feature set for the known DJI Tello drone flight data.

Algorithm	**Accuracy**	**Correctly Detected Drone?**
Logistic Regression	0.95 (57/60)	Yes
Linear Discriminant Analysis	0.98 (59/60)	No
K-Nearest Neighbors	1.00	Yes
Classification and Regression Trees	1.00 (60/60)	Yes
Gaussian Naïve Bayes	0.95 (57/60)	Yes
Support Vector Machines	0.98 (59/60)	No
Multi-layer Perceptron Neural Network	0.98 (59/60)	No

Table 8: Known DJI Tello Drone Full Feature Set Detection Metrics for Various Machine Learning Algorithms Using Model.predict()

The Classification and Regression Trees and K-Nearest Neighbors algorithms correctly classified the access points and the DJI Tello drone with accuracy percentages of 100%. The Gaussian Naïve Bayes and Logistic Regression algorithms correctly classified the DJI Tello as a drone, but it also classified 3 access points as drones too with accuracy percentages of 95%. The Linear Discriminant Analysis, Support Vector Machines and Multi-layer Perceptron algorithms classified everything as an access point and could not identify the known DJI Tello drone.

Subpar detection performance was to be expected by the nonlinear classification algorithms, due to the class disparity leading to overfitting by Support Vector Machines and Multi-layer Perceptron.

Good performance by the binary classification optimized algorithms such as Logistic Regression, Linear Discriminant Analysis, and Naïve Bayes was to be expected. An anomaly is Classification and Regression Trees' performance, but it is biased based on its training data. Since the training data contained a DJI Tello, the algorithm's performance can be attributed to the DJI Tello already being in the training set. K-Nearest Neighbors performs best when residuals between various classes is minimized and known data is fed to the algorithm. It is assumed that the residual from the DJI Tello flight data to the drone group was minimal enough to prevent misclassification of the DJI Tello as an access point. It is also assumed that the second DJI Tello being a known drone entry contributed to the algorithm's performance.

Limited Feature Set

Table 9 consists of machine learning algorithm detection prediction accuracy metrics using the limited feature set for the known DJI Tello drone flight data.

The Logistic Regression, Linear Discriminant Analysis, Gaussian Naïve Bayes, and Multi-layer Perceptron algorithms all classified the drone as a drone and also incorrectly classified four access points as drones. All the algorithms' accuracy was approximately 93% tying all of them for best performance.

Algorithm	**Accuracy**	**Correctly Detected Drone?**
Logistic Regression	0.93 (56/60)	Yes
Linear Discriminant Analysis	0.93 (56/60)	Yes
K-Nearest Neighbors	0.98 (59/60)	No
Classification and Regression Trees	0.88 (53/60)	No
Gaussian Naïve Bayes	0.93 (56/60)	Yes
Support Vector Machines	0.98 (59/60)	No
Multi-layer Perceptron Neural Network	0.93 (56/60)	Yes

Table 9: Known DJI Tello Drone Limited Feature Set Detection Metrics for Various Machine Learning Algorithms Using Model.predict()

The K-Nearest Neighbors, Classification and Regression Trees, and Support Vector Machines algorithms could not correctly detect the drone. The Classification and Regression Trees algorithm performed even worse than the K-Nearest Neighbors and Support Vector Machines algorithms, since the Classification and Regression Trees algorithm incorrectly classified the drone as an access point and six access points as drones. The Classification and Regression Trees algorithm's accuracy was approximately 88% compared to the K-Nearest Neighbors and Support Vector Machines algorithms' accuracy of approximately 98%.

Subpar detection performance was to be expected by the nonlinear classification algorithms, since the class mismatch led to overfitting by K-Nearest Neighbors, Classification and Regression Trees, and Support Vector Machines. Classification and Regression Trees is

biased based on its training data, and since the known DJI Tello drone flown did not have identical characteristics to the training set DJI Tello drone, it could have led to misclassification.

Good performance by the binary classification optimized algorithms such as Logistic Regression, Linear Discriminant Analysis, and Naïve Bayes was to be expected. An anomaly is Multi-layer Perceptron's performance, since it is better at multi-layer classification rather than binary classification, however it seems that the drone layer defined from the training data was good enough to detect the known drone.

Aggregated Machine Learning Detection Results

Table 10 was compiled using all the accuracy results above from known and unknown drone machine learning algorithm data. The algorithms that had no results or extremely poor results were not listed at all. The best performing algorithms are highlighted in green. The limited feature set did not include the "Drone OUI," "Drone SSID" and "Beacon-len(SSID)" fields.

The algorithms generally performed better detecting a known drone type compared to an unknown drone type. This was to be expected, since algorithms are optimized and biased for known/training data.

Drone Type	Limited Feature Set with Accuracy Percentages	Full Feature Set with Accuracy Percentages
Known	Classification and Regression Trees: 90% (Incorrectly classified Drone as AP) Gaussian Naive Bayes: 93% Linear Discriminant Analysis: 93% Logistic Regression: 93% Multi-layer Perceptron: 93%	Classification and Regression Trees: 100% K-Nearest Neighbors: 100% Logistic Regression: 95%
Unknown	Classification and Regression Trees: 90% Gaussian Naive Bayes: 93% Linear Discriminant Analysis: 93% Logistic Regression: 93% Multi-layer Perceptron: 93%	Gaussian Naïve Bayes: 93% Logistic Regression: 95%

Table 10: Aggregated Machine Learning Detection Performance Table

Logistic Regression is the only algorithm which robustly performed at detecting both known and unknown drones regardless of feature set. It was only outperformed by Classification and Regression Trees, and K-Nearest Neighbors when attempting to detect the known drone with the full feature set. The 100% accuracy rate of Classification and Regression Trees, and K-Nearest Neighbors is an anomaly, since algorithm performance of 100% accuracy is highly unlikely. The top binary classification algorithms such as Gaussian Naïve Bayes, Linear Discriminant Analysis and Logistic Regression performed extremely well. Gaussian Naïve Bayes is the gold standard algorithm for spam vs ham classification. Drone versus access point classification is an extremely similar use case. Classification and Regression Trees, K-Nearest Neighbors and Multi-layer Perceptron all use layers/groups based on similar characteristics. With the full and limited dataset having unique distinguishing features to differentiate a drone versus

an access point, it makes sense that these algorithms also performed well at placing data in the drone and access point buckets.

Drone Classification

Using model.predict(), test data consisting of one known DJI Tello drone and 59 access points was fed to the various machine learning algorithms for predictions. Either the full feature set consisting of drone OUI, drone SSID, Beacon-length(SSID), number of beacons, number of data packets, signal strength variability, and client nodes or the limited feature set which only consisted of number of beacons, number of data packets, signal strength variability, and client nodes was fed to the various machine learning algorithms to see whether each algorithm could identify a known drone by make and model.

Full Feature Set

Table 11 consists of machine learning algorithm classification prediction accuracy metrics using the full feature set for the known DJI Tello drone flight data.

The Classification and Regression Trees algorithm correctly classified 59 access points and incorrectly classified the drone as the Zuhafa JY02. This was the best performing algorithm in terms of accuracy for an algorithm that actually classified the drone entry as some make and model of drone.

Algorithm	**Accuracy**	**Drone Classification**	**Correctly Classified Drone?**
Logistic Regression	0.92 (55/50)	Zuhafa JY02	No
Linear Discriminant Analysis	0.93 (56/60)	Hasakee Q10	No
K-Nearest Neighbors	0.98 (59/60)	AP	No
Classification and Regression Trees	0.98 (59/60)	Zuhafa JY02	No
Gaussian Naïve Bayes	0.98 (59/60)	AP	No
Support Vector Machines	0.98 (59/60)	AP	No
Multi-layer Perceptron Neural Network	0.98 (59/60)	AP	No

Table 11: Known DJI Tello Drone Full Feature Set Classification Metrics for Various Machine Learning Algorithms Using Model.predict()

The Linear Discriminant Analysis algorithm incorrectly classified 3 access points as DJI Phantom 3s and incorrectly classified the drone as a Hasakee Q10. The Logistic Regression algorithm incorrectly classified 4 access points as DJI Phantom 3s and incorrectly classified the drone as a Zuhafa JY02. The K-Nearest Neighbors, Gaussian Naïve Bayes, Support Vector Machines and Multi-layer Perceptron algorithms incorrectly classified everything as an access point.

Most binary classification and linear models are not good for predicting non binary values. Classification and Regression Trees shows promise with the little data provided even though it did not correctly classify the drone. K-Nearest Neighbors and Multi-layer Perceptron

will potentially yield better results with more training data with less of an access point to drone class mismatch and more entries of the same make and model. Classification and Regression Trees, K-Nearest Neighbors and Multi-layer Perceptron should be looked into for future classification research.

Limited Feature Set

Table 12 consists of machine learning algorithm classification prediction accuracy metrics using the limited feature set for the known DJI Tello drone flight data.

The Classification and Regression Trees algorithm incorrectly classified 3 access points as DJI Phantom 3s, 2 access points as DJI Tellos, and incorrectly classified the drone as a Hasakee Q10. The Linear Discriminant Analysis algorithm incorrectly classified 4 access points as DJI Phantom 3s, and incorrectly classified the drone as a Hasakee Q10. The Logistic Regression algorithm incorrectly classified 4 access points as DJI Phantom 3s, and incorrectly classified the drone as a Zuhafa JY02. The Multi-layer Perceptron algorithm incorrectly classified 4 access points as DJI Phantom 3s, and incorrectly classified the drone as a DJI Mavic Air. This is interesting, since the make of the drone was correctly identified by the Multi-layer Perceptron algorithm.

Algorithm	**Accuracy**	**Drone Classification**	**Correctly Classified Drone?**
Logistic Regression	0.92 (55/60)	Zuhafa JY02	No
Linear Discriminant Analysis	0.92 (55/60)	Hasakee Q10	No
K-Nearest Neighbors	0.98 (59/60)	AP	No
Classification and Regression Trees	0.90 (54/60)	Hasakee Q10	No
Gaussian Naïve Bayes	0.98 (59/60)	AP	No
Support Vector Machines	0.98 (59/60)	AP	No
Multi-layer Perceptron Neural Network	0.92 (55/60)	DJI Mavic Air	No

Table 12: Known DJI Tello Drone Full Feature Set Classification Metrics for Various Machine Learning Algorithms Using Model.predict()

The Gaussian Naïve Bayes, K-Nearest Neighbors, and Support Vector Machines algorithms incorrectly classified everything as an access point even with a reduced feature set.

To reiterate, binary classification and linear models are not good for predicting nonbinary values. As expected, algorithm performance decreased with a limited feature set, since there are less fields to distinguish various groups. More features provide more distinguishing characteristics between drone and ordinary wireless networks. Again, Classification and Regression Trees, K-Nearest Neighbors and Multi-layer Perceptron should be looked into for future classification research but fed the full feature set in order to optimize layers and distinction between various groups.

Classification Analysis

None of the algorithms were able to correctly classify the DJI Tello, however this is mostly due to only having one entry for each make and model. A better training data set with more multiple entries of the same make and model of drone can create a more effective classification machine learning algorithm. Classification at the moment can be hard coded based on OUI, SSID format, and beacon frame length minus length of the SSID. These characteristics are unique enough to distinguish make and model of drone.

Another thing to note is that the two DJI Tellos used multiple OUIs which could have contributed to the algorithms' weak performances. The first DJI Tello had an OUI of "60:60:1F," while the second DJI Tello not contained in the training set had an OUI of "34:D2:62." While both OUIs were registered to DJI, it was not in the known lists of drone OUIs in the Python code. Having a more complete OUI list can increase machine learning classification effectiveness for the full feature set but was out of scope for this research.

Moving on, even with subpar classification machine learning performance, it is not necessarily the end of the world for the feasibility of a drone detection system. The end goal is to invoke a C-sUAS response to take down a drone. The C-sUAS system can potentially launch all exploits in its database even if classification metrics stay extremely poor.

Drone Network Traffic Analysis

Manual drone network traffic analysis was conducted using Wireshark. Data was then collected and aggregated into a specifications table.

Table 13 contains drone characteristics of all the drones used in this research. Data was collected using Wireshark and extraneous packet capture data was excluded. Thus, there are no

Wireshark screenshots, but rather an aggregated table of pertinent information. The Beacon Frame-length(SSID) field was used for manual make and model classification of sUAS.

Drone Make and Model	**OUI**	**Default SSID Format**	**Default Network Security**	**Beacon Frame-length(SSID)**	**Data Transmission Protocols**	**Stream Data Header**
DJI Tello	60:60:1F & 34:D2:62	TELLO-XXXXXX	Open	150	Video: Skype Control: UDP	CC
DJI Phantom 3	60:60:1F	PHANTOM3-XXXXXX	WPA2-PSK	212	Video: UDT Control: TCP	55
DJI Mavic Air	60:60:1F	MAVIC_AIR-XXXXXX	WPA2-PSK	306	Unknown	Unknown
Sanrock U61W	AC:D8:29	Udirc-WiFi-XXXXXX	Open	102	Video: Skype Control: RC	63… (First 28 bytes the same)
Zuhafa JY02	30:7B:19	JY-FPV-XXXXXX	Open	180	Video: UDP Control: RC	40… (First 11 bytes the same)
Hasakee Q10	AC:D8:29	HASAKEE-Q10_XXXXXX	Open	162	Video: RTP/Skype Control: RC	93… (First 64 bytes the same for all packets)

Table 13: Network Traffic Analysis Characteristics by Drone Make and Model

Chapter 5

Conclusion

This chapter will discuss results to the research questions proposed, discuss main contributions, address limitations and suggest future research.

Research Question Results

The first research question was how 802.11 signals can be used to effectively detect a Wi-Fi enabled sUAS. The short answer is that beacon frame lengths contained in 802.11 signals can be used to detect Wi-Fi sUAS. Hardcoding various beacon frame lengths minus the length of the SSID into Python can yield manual detection as well as make/model classification of known drones.

Machine learning algorithms were evaluated to test the unknown drone use case when securing a facility using a ground station. However, results are not representative of the entire drone and wireless access point population, since the training set was small. Machine learning metrics in the results section show promise for a more automated approach, since they were able to classify known drones with high accuracy and low error. Unknown drones which were not contained in the dataset were correctly detected by some algorithms. There was high accuracy and some error involved when evaluating unknown drones with the limited training set. An inclusive and diverse dataset can strengthen these results and reduce unknown drone evaluation error.

The second research question was how 802.11 signals can be used to classify a sUAS by make and model. The answer to this question was also by using beacon frame lengths. A known drone can be correctly classified by make and model by using the alternate beacon frame minus length of SSID approach discussed above.

Machine learning algorithms were tested again to see whether a drone flying could be identified by make and model. The machine learning algorithms could not correctly classify the DJI Tello or other drones, which can be attributed to the small size of the training set or the suitability of the specific machine learning algorithm for classification. Only one entry for each drone make and model was contained in the training set, and there was a large class disparity of each individual make/model of drone versus ordinary wireless access points. However, the alternate manual classification method identified drones by make and model, but with more training data, machine learning algorithms could theoretically also be used for classification.

Findings and Main Contributions

Findings were made using network traffic analysis to extract descriptive fields such as nodes, signal strength variability, beacon frame lengths, and data packets. After conducting this research, the characteristics of drone networks can be described. Drone networks with a drone-based access point tend to have a fixed unique beacon frame length which SSID name is a part of, a generally low number of end nodes which are served, a medium to large signal strength variability with respect to a fixed ground station, a large number of data packets, and a possibility of displaying a drone manufacturer's OUI. Drone networks with a controller-based access point have all the previously described characteristics but tend not to have signal strength variability. Normal active wireless access points (WAPs) have their own unique beacon frame lengths which

SSID name is also a part of. WAPs are generally fixed in location, so signal strength variability is negligible. Data packets on a WAP can vary based on the type of browsing a user is doing.

Another finding of this research is that the Gaussian Naïve Bayes and Logistic Regression algorithms detected the unknown drone. The Classification and Regression Trees, K-Nearest Neighbors, and Logistic Regression algorithms detected the known drone. Both the unknown and known drones were detected even using the small training set with the full feature set assembled. Other algorithms such as Classification and Regression Trees, Linear Discriminant Analysis, and Multi-layer Perceptron were able to detect the unknown drone when using the limited feature set assembled. Other algorithms such as Gaussian Naïve Bayes, Linear Discriminant Analysis, and Multi-layer Perceptron were also able to detect the known drone when using the limited feature set assembled. However, in the cases of both the known drone and unknown drone, algorithm accuracy suffered when using the limited feature set compared to the full feature set. These results can vary depending on sample size of the training set.

A large contribution by aggregating these findings was the development of a drone detection and classification system using an 802.11 framework. A machine learning framework was built during the course of this research, but results were not significant due to a limited sample size and limited data. Various algorithms were evaluated to pave the road for future research in this field.

Limitations

The several limitations of this study will be addressed in this section.

Sample Size of Wireless Access Points

This study took 55 public wireless access points in the immediate vicinity into account, but that is not necessarily representative of all WAPs in all areas or even private WAPs. Beacon frame lengths, MAC addresses and SSID names are all variable values, and different locations can cause more false positives. IEEE 802.11 packet captures (PCAPs) of WAPs are not readily available online due to the sensitivity of such data. This proved to be a challenge for the research and can be a limitation for development of a universal drone detection and classification system. It can be noted that more in depth network traffic analysis at layer 3 rather than layer 2 is more universal and can better identify drone networks vs home networks.

Sample Size of Drones

Drones are expensive pieces of equipment and Wi-Fi enabled drones all vary in video transmission and control protocols. The first boundary to using machine learning for the detection of drones was getting five Wi-Fi enabled drones as part of the training set. These five drones are not necessarily representative of all Wi-Fi drones. The next step was looking at machine learning for classification, but that requires at least five samples of each make and model of drone which is not economical nor is it practical. Thus, classification was hard coded rather than feeding data through the algorithm for a second time. Together, the sample size of access points and sample size of drones were not large enough to represent the entire drone and access point population. Sharing drone data and making a centralized drone dataset repository is integral for future real-world implementation of a drone defense system.

Ethicality of Layer 3 Network Traffic Analysis

Layer 2 handles MAC addresses only, however by brute forcing a network's credentials and capturing the Extensible Authentication Protocol (EAPOL) handshake (for WPA2 clients), Layer 2 analysis at the IP level can be done. Hacking into Wi-Fi networks is unethical and illegal within the United States and browsing history can be used for extortion. Law enforcement needs warrants in order to break into personal networks. Thus, the ethicality and legality can be a huge barrier to analysis of layer 3 network traffic.

Beacon Frame Length Assumption

The assumption that the value of the beacon frame length minus the SSID length was based on data captured from the DJI Tello. There is no guarantee that this assumption holds true for all drones, but there was not enough funding to purchase multiple Wi-Fi drones. Also, Wi-Fi equipment on board can change at the discretion of the end user or manufacturer. There is also the chance that another wireless access point shares a common beacon frame length resulting in false positives. Third, hidden networks do not display SSIDs in 802.11 headers which can defeat the beacon frame length minus SSID length metric.

MAC Address Assumption

While not evaluated in the scope of this paper, MAC addresses can be spoofed. While there is no option to. The "Drone OUI" field depends on the drone keeping its manufacturer's default MAC address in order to evaluate the field as 1. Theoretically, patched firmware or swapping onboard equipment can defeat this feature's use as a classifier.

Future Research

Training Set Expansion

This research ran into budget and data collection constraints, so future research can collect more data to yield significant accuracy results for a machine learning based drone detection system. Having access to more access points and more drones of the same make and model can also yield significant accuracy results for a machine learning based drone classification system. Simulation can also be explored to artificially create data which mimics real world data. That can reduce class disparities and help combat machine learning algorithm overfitting. Classification field of make and model can also be split into two classification fields, so an unknown drone can be identified by make but not necessarily model.

Layer 3 Data Collection

This research only focused on Layer 2 features, but Layer 3 data collection was discussed. Future research can append Layer 3 features to the existing Layer 2 features presented in this research to see if detection and classification accuracy scores increase even more. Algorithms other than the seven implemented can also be evaluated to see if metrics change.

Non-IEEE 802.11 Drones

Drones that emit IEEE 802.11 signals are not representative of the entire drone population. Drones can use hobbyist remote control, MAVLink, OcuSync, or other custom protocols with radios using non-IEEE 802.11 protocols. A perfect example is OcuSync which is

DJI's 2.4/5.8 GHz custom protocol implementation. It was created to avoid range limitations of the IEEE 802.11 protocol. One can only assume that a transition away from IEEE 802.11 for drones is in the near future. Thus, a more universal drone RF detection and classification approach should be implemented for real world applications. To create a more robust drone detection system, the protocol scope should be expanded.

All protocols have a characteristic "layout." While fields such as BSSID, SSID, and Beacon frames may not be present, other characteristics will be present. Once these characteristics are identified, they could be incorporated into this framework.

C-sUAS

To continue, this research only covered detection and classification of IEEE 802.11 drones. But in order to defend against a malicious 802.11 drone, there has to be a C-sUAS response for it. Expanding on the previous paragraph, any drone running any protocol should also have a unique C-sUAS response to create a drone defense system. This way, a malicious drone can be successfully countered instead of just detected and classified. As of right now with drone detection and classification, the best one can do is evacuate a location since there is no way to deter the drone.

Anomaly Detection

Anomaly detection algorithms such as IsolationForest and Density-based spatial clustering of applications with noise (DBSCAN) can also be analyzed to overcome the limitations of machine learning. This way a centralized repository of drone and wireless network packet is not required, since the anomaly detection algorithm can evaluate a baseline for ordinary wireless

access points then trigger when a drone not matching the wireless access point baseline is detected. The limitations of machine learning were discussed in this paper, but anomaly detection was out of scope for this research.

References

Aaronia AG. (2017). Detection of UAV's based on their RF emissions.

ALFA. (n.d.). AWUS036ACH. ALFA Network Inc. Retrieved March 28, 2022, from https://www.alfa.com.tw/products/awus036ach

Altawy, R., & Youssef, A. M. (2017). Security, Privacy, and Safety Aspects of Civilian Drones: A Survey. *ACM Transactions on Cyber-Physical Systems*, *1*(2), 1–25. https://doi.org/10.1145/3001836

Amazon. (n.d.-a). *Amazon.com: DJI Mavic Air Quadcopter with Remote Controller—Arctic White: Toys & Games*. Retrieved February 25, 2022, from https://www.amazon.com/DJI-Mavic-Quadcopter-Remote-Controller/dp/B078WP48CH/ref=asc_df_B078WP48CH

Amazon. (n.d.-b). *Amazon.com: Q10 Mini Drones for Kids with Camera FPV Wifi 720P HD Remote Control Helicopter Toys Gifts for Boys Girls, Foldable RC Quadcopter with Wifi Live Video, Altitude Hold, Headless Mode, 3D Flips: Toys & Games*. Retrieved February 25, 2022, from https://www.amazon.com/Helicopter-Foldable-Quadcopter-Altitude-Headless/dp/B0964WHG8Q

Amazon. (n.d.-c). *Amazon.com: SANROCK U61W Drones with Camera for Kids Adult Beginner 720P HD & 2 Batteries, Mini Drone Toy Gift for Boy Girl WiFi FPV RC Quadcopter, Waypoints Fly, Headless Mode, Altitude Hold, Emergency Stop, RED : Toys & Games*. Retrieved February 25, 2022, from https://www.amazon.com/SANROCK-Intelligent-Operation-Altitude-Emergency/dp/B07R6L2QJZ

Avrodex. (n.d.). *Incident at VANCOUVER INTL BC (CYVR) | Avrodex*. Retrieved March 24, 2022, from https://avrodex.com/view/2022P0194

Baldor, L. (2021, May 23). *Top US general in the Mideast says more work needed to counter small drones*. Military Times. https://www.militarytimes.com/news/your-military/2021/05/23/top-us-general-in-the-mideast-says-more-work-needed-to-counter-small-drones/

Bello, A. (n.d.). *Radio Frequency Toolbox for Drone Detection and Classification*. https://doi.org/10.25777/9GKM-JD54

Belyadi, H., & Haghighat, A. (2021). Machine learning guide for oil and gas using Python a step-by-step breakdown with data, algorithms, codes, and applications. http://www.vlebooks.com/vleweb/product/openreader?id=none&isbn=9780128219300

Bisio, I., Garibotto, C., Lavagetto, F., Sciarrone, A., & Zappatore, S. (2019). Blind Detection: Advanced Techniques for WiFi-Based Drone Surveillance. *IEEE Transactions on Vehicular Technology*, *68*(1), 938–946. https://doi.org/10.1109/TVT.2018.2884767

Bomgardner, M. (2021, May 27). *Siegfried, Brenntag, and Symrise hit by cyberattacks*. Chemical & Engineering News. https://cen.acs.org/business/specialty-chemicals/Siegfried-Brenntag-Symrise-hit-cyberattacks/99/i20

CISA. (n.d.). *Unauthorized Drone Activity Over Sporting Venues*. 2.

Dettmer, J. (2021, June 7). *Possible First Use of AI-Armed Drones Triggers Alarm Bells*. VOA. https://www.voanews.com/a/africa_possible-first-use-ai-armed-drones-triggers-alarm-bells/6206728.html

De Vynck, G., Verma, P., & Baran, J. (2022). Exploding ‘kamikaze’ drones are ushering in a new era of warfare in Ukraine. *Washington Post*. https://www.washingtonpost.com/technology/2022/03/24/loitering-drone-ukraine/

DJI. (n.d.-a). *Buy DJI Mini SE - DJI Store*. Retrieved February 25, 2022, from https://store.dji.com/product/dji-mini-se-tm?site=brandsite&from=buy_now_bar&vid=105351

DJI. (n.d.-b). *Phantom 3 Standard—DJI*. DJI Official. Retrieved February 25, 2022, from https://www.dji.com/phantom-3-standard

DJI. (n.d.-c). *Tello | Ryze Tech—Feel the Fun*. Retrieved February 25, 2022, from https://store.dji.com/product/tello?vid=38421

DuckDrone, LLC. (2017). U.S. Patent No. 10,302,397 B1. U.S. Patent and Trademark Office.

Evans, D. (2022, March 10). *Possible contraband drop drone found following police chase* [Text.Article]. FOX 5 Atlanta; FOX 5 Atlanta. https://www.fox5atlanta.com/news/possible-contraband-drop-drone-found-following-police-chase

Ezuma, M., Erden, F., Anjinappa, C. K., Ozdemir, O., & Guvenc, I. (2019). Micro-UAV Detection and Classification from RF Fingerprints Using Machine Learning Techniques. *2019 IEEE Aerospace Conference*, 1–13. https://doi.org/10.1109/AERO.2019.8741970

FAA. (n.d.-a). *UAS by the Numbers* [Template]. https://www.faa.gov/uas/resources/by_the_numbers/

FAA. (n.d.-b). *UAS Remote Identification Overview* [Template]. https://www.faa.gov/uas/getting_started/remote_id/

GeeksforGeeks. (2021). *ML | Linear Discriminant Analysis—GeeksforGeeks*. https://www.geeksforgeeks.org/ml-linear-discriminant-analysis/

Geier, J. (n.d.). How to: Sniff Wireless Packets with Wireshark. Retrieved March 28, 2022, from http://www.wireless-nets.com/resources/tutorials/sniff_packets_wireshark.html

Ghosh, B. (2021). Decrypting SSL/TLS Traffic with Wireshark. https://linuxhint.com/decrypt-ssl-tls-wireshark/

Grosse, R. (n.d.). *Lecture 5: Multilayer Perceptrons*. 7.

Hartmann, K., & Giles, K. (2016). UAV exploitation: A new domain for cyber power. *2016 8th International Conference on Cyber Conflict (CyCon)*, 205–221. https://doi.org/10.1109/CYCON.2016.7529436

IANS. (2022). *Aircraft spotted flying across high-security no-flying zone of Taj Mahal; ASI seeks report from CISF*. Free Press Journal. https://www.freepressjournal.in/india/aircraft-spotted-flying-across-high-security-no-flying-zone-of-taj-mahal-asi-seeks-report-from-cisf

Insinna, V. (2021, June 7). *US Air Force completes tests of swarming munitions, but will they ever see battle?* Defense News. https://www.defensenews.com/air/2021/06/07/us-air-force-successfully-completes-tests-of-swarming-munitions-but-their-future-is-unclear/

Ji, R., Wang, J., Tang, C., & Li, R. (2017). Automatic Reverse Engineering of Private Flight Control Protocols of UAVs. *Security and Communication Networks*, *2017*, 1–9. https://doi.org/10.1155/2017/1308045

Lusa. (2022). *Drone forces flight diversion to Lisbon*. https://www.theportugalnews.com/news/2022-02-27/drone-forces-flight-diversion-to-lisbon/65514

Majumder, P. (n.d.). *Gaussian Naive Bayes*. Retrieved March 24, 2022, from https://iq.opengenus.org/gaussian-naive-bayes/

MathWorks. (n.d.). *802.11 MAC Frame Decoding—MATLAB & Simulink*. Retrieved March 28, 2022, from https://www.mathworks.com/help/wlan/ug/802-11-mac-frame-decoding.html

Medaiyese, O. O., Ezuma, M., Lauf, A. P., & Adeniran, A. A. (2021). Hierarchical Learning Framework for UAV Detection and Identification. *ArXiv:2107.04908 [Eess]*. http://arxiv.org/abs/2107.04908

Meraki. (2020, October 5). 802.11 Association Process Explained. Cisco Meraki. https://documentation.meraki.com/MR/WiFi_Basics_and_Best_Practices/802.11_Association_Process_Explained

Michalski, D., & Michalska, A. (2017). Protection against drone activity. *Security Forum*, *1*, 73–83. https://doi.org/10.26410/SF_1/17/8

Nanjappa, V. (2022, March 12). *5 Khalistani terrorists charged for smuggling arms, drugs using drone from Pak*. Https://Www.Oneindia.Com. https://www.oneindia.com/india/5-khalistani-terrorists-charged-for-smuggling-arms-drugs-using-drone-from-pak-3382282.html

Nayarasi. (2014-a). CWAP – 802.11 Data Frame Types. Mrn-Cciew. https://mrncciew.com/2014/10/13/cwap-802-11-data-frame-types/

Nayarasi. (2014-b). 802.11 Mgmt: Beacon Frame. Mrn-Cciew. https://mrncciew.com/2014/10/08/802-11-mgmt-beacon-frame/

Nemer, I., Sheltami, T., Ahmad, I., Yasar, A. U.-H., & Abdeen, M. A. R. (2021). RF-Based UAV Detection and Identification Using Hierarchical Learning Approach. *Sensors*, *21*(6), 1947. https://doi.org/10.3390/s21061947

Nie, W., Han, Z.-C., Zhou, M., Xie, L.-B., & Jiang, Q. (2021). UAV Detection and Identification Based on WiFi Signal and RF Fingerprint. *IEEE Sensors Journal*, *21*(12), 13540–13550. https://doi.org/10.1109/JSEN.2021.3068444

Notter, R. (2017, December 27). How A Hidden SSID Can Impact Your Roaming Experience. *Dot11 Exposed*. https://dot11.exposed/2017/12/27/how-a-hidden-ssid-can-impact-your-roaming-experience/

O'Malley, J. (2019, February 18). *Take me out: Creating 'No-Drone Zones' around airports*. https://eandt.theiet.org/content/articles/2019/02/take-me-out-creating-no-drone-zones-around-airports/

Papenfuss, M. (2019, May 7). Ariana Grande Concert Targeted By Drone Dropping Swastika Flyers. HuffPost UK. https://www.huffingtonpost.co.uk/entry/ariana-grande-concert-drone-swastika-flyers_uk_5cd170b7e4b0548b735fcc3b

Pomerleau, M. (2021, October 4). *Swarm grammar: DARPA to test whether single user can control 200 drones*. C4ISRNet. https://www.c4isrnet.com/unmanned/2021/10/04/swarm-grammar-darpa-to-test-single-user-controlling-200-drones-in-mock-city/

Sciancalepore, S., Ibrahim, O. A., Oligeri, G., & Di Pietro, R. (2020). PiNcH: An Effective, Efficient, and Robust Solution to Drone Detection via Network Traffic Analysis. Computer Networks, 168, 107044. https://doi.org/10.1016/j.comnet.2019.107044

Shi, X., Yang, C., Xie, W., Liang, C., Shi, Z., & Chen, J. (2018). Anti-Drone System with Multiple Surveillance Technologies: Architecture, Implementation, and Challenges. *IEEE Communications Magazine*, *56*(4), 68–74. https://doi.org/10.1109/MCOM.2018.1700430

The Union. (2022). *Saudi Arabia: 16 hurt in airport drone attack from Yemen*. Hosted. Retrieved March 24, 2022, from http://hosted.ap.org/theunion/article/78874e16a415137810be53f237cfdc0e/saudi-arabia-16-hurt-airport-drone-attack-yemen

United States Department of Justice. (2020). *Drones: A Report on the Use of Drones by Public Safety Agencies—And a Wake-Up Call about the Threat of Malicious Drone Attacks*. 128.

Unlu, E., Zenou, E., & Riviere, N. (2018). Using Shape Descriptors for UAV Detection. *Electronic Imaging*, *2018*(9), 128–128. https://doi.org/10.2352/ISSN.2470-1173.2018.09.SRV-128

Vanian, J. (2017). *Man Arrested For Flying Drone Near Police Helicopter*. The Drive. https://www.thedrive.com/news/7255/man-arrested-for-flying-drone-near-police-helicopter

Verges, F. (2017). Wireshark: How to check if a Wi-Fi network supports 802.11k – SemFio Networks. https://semfionetworks.com/blog/wireshark-how-to-check-if-a-wi-fi-network-supports-80211k/

Wireshark. (n.d.-a). *what is the capture/display filter to get RSSI information of WiFi users? - Ask Wireshark*. Retrieved March 28, 2022, from https://ask.wireshark.org/question/4598/what-is-the-capturedisplay-filter-to-get-rssi-information-of-wifi-users/

Wireshark. (n.d.-b). *7.2. Following Protocol Streams*. Retrieved March 28, 2022, from https://www.wireshark.org/docs/wsug_html_chunked/ChAdvFollowStreamSection.html

Xiaoqi Yang, Kai Huo, Weidong Jiang, Jingjing Zhao, & Zhaokun Qiu. (2016). A passive radar system for detecting UAV based on the OFDM communication signal. *2016 Progress in Electromagnetic Research Symposium (PIERS)*, 2757–2762. https://doi.org/10.1109/PIERS.2016.7735118

Yaacoub, J.-P., Noura, H., Salman, O., & Chehab, A. (2020). Security analysis of drones systems: Attacks, limitations, and recommendations. Internet of Things, 11, 100218. https://doi.org/10.1016/j.iot.2020.100218

Yang, S., Qin, H., Liang, X., & Gulliver, T. (2019). An Improved Unauthorized Unmanned Aerial Vehicle Detection Algorithm Using Radiofrequency-Based Statistical Fingerprint Analysis. *Sensors*, *19*(2), 274. https://doi.org/10.3390/s19020274

Zhao, Y., & Su, Y. (2020). The Extraction of Micro-Doppler Signal With EMD Algorithm for Radar-Based Small UAVs' Detection. *IEEE Transactions on Instrumentation and Measurement*, *69*(3), 929–940. https://doi.org/10.1109/TIM.2019.2905751

Zuhafa. (n.d.). *Www.zuhafa.com*. Retrieved February 25, 2022, from http://www.zuhafa.com

Appendix A

This section contains the Python code used for various machine learning algorithm evaluation.

Python Machine Learning Code

```
import sys
import os
import subprocess
import matplotlib
import pandas as pd
import scipy
import logging
import threading
import time
import csv
from datetime import datetime, timedelta
import numpy as np
from scapy.all import *
from time import sleep, perf_counter
from threading import Thread
from pandas import read_csv
from pandas.plotting import scatter_matrix
from matplotlib import pyplot
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import StratifiedKFold
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
```

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.neural_network import MLPClassifier
from sklearn.ensemble import IsolationForest
from sklearn.impute import SimpleImputer

dataset=read_csv('trainingset.csv')
df=read_csv(str(sys.argv[1]))# 'capture5.csv'
i=int(sys.argv[2])
print(dataset.shape)
print(dataset.head(20))
print(dataset.describe())
#print(dataset.groupby('Drone/AP').size())

array = dataset.values


X = array[:,3:9]
y = array[:,10]
X_train, X_validation, Y_train, Y_validation = train_test_split(X, y, random_state=1)

models = []
models.append(('LR', LogisticRegression(solver='liblinear', multi_class='ovr')))
models.append(('LDA', LinearDiscriminantAnalysis()))
models.append(('KNN', KNeighborsClassifier()))
models.append(('CART', DecisionTreeClassifier()))
models.append(('NB', GaussianNB()))
models.append(('SVM', SVC(gamma='auto')))
models.append(('MLP',MLPClassifier()))
# evaluate each model in turn
results = []
names = []
for name, model in models:
        kfold = StratifiedKFold(n_splits=5, random_state=1, shuffle=True)
        cv_results = cross_val_score(model, X_train, Y_train, cv=kfold,
scoring='accuracy')
        results.append(cv_results)
        model.fit(X,y)
        pred=model.predict(df.iloc[:,3:9])
        names.append(name)
        print('%s: %f (%f)' % (name, cv_results.mean(), cv_results.std()))
```

```
                pd.DataFrame(pred).to_csv(str(model)+str(i)+".csv",
header=['Drone/AP'],index=None)

        X = array[:,3:9]
        y = array[:,11]
        for name, model in models:
                kfold = StratifiedKFold(n_splits=2, random_state=1, shuffle=True)
                cv_results = cross_val_score(model, X_train, Y_train, cv=kfold,
scoring='accuracy')
                results.append(cv_results)
                model.fit(X,y)
                pred=model.predict(df.iloc[:,3:9])
                names.append(name)
                print('%s: %f (%f)' % (name, cv_results.mean(), cv_results.std()))
                pd.DataFrame(pred).to_csv(str(model)+"1_classification.csv",
header=['Drone/AP'],index=None)
```

Appendix B

This section contains the Python code used for 30-second-long packet capture parsing and conversion to CSV.

Python CSV Parsing Code

```
import sys
import os
import subprocess
import matplotlib
import pandas as pd
import logging
import threading
import time
import csv
from datetime import datetime, timedelta
import numpy as np
import re
from scapy.all import *
from time import sleep, perf_counter
from threading import Thread
from pandas import read_csv
from pandas.plotting import scatter_matrix
from matplotlib import pyplot
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import StratifiedKFold
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
```

```
        def wireshark(j):#j
            '''os.system("sudo tshark -a duration:30 -i [interface] -w output.pcap -o
nameres.mac_name:FALSE")
            os.system("sudo tshark -r output.pcap -o nameres.mac_name:FALSE -T fields -e
wlan.bssid | sort -u > BSSIDList.txt")
            os.system("sudo tshark -r output.pcap -o nameres.mac_name:FALSE -z conv,wlan -V
> conversations.txt")'''
            os.system("sudo touch capture" + str(j) + ".pcap")
            os.system("sudo tshark -a duration:30 -i [interface] -w capture" + str(j) + ".pcap -o
nameres.mac_name:FALSE")
            os.system("sudo tshark -r capture" + str(j) + ".pcap -o nameres.mac_name:FALSE -T
fields -e wlan.bssid | sort -u > BSSIDList" + str(j) + ".txt")
            os.system("sudo tshark -r capture" + str(j) + ".pcap -o nameres.mac_name:FALSE -z
conv,wlan -V > conversations" + str(j) + ".txt")

        def BSSID_Parser(j):#j
            BSSIDList=open("BSSIDList"+str(j)+".txt", "r")#
            BSSIDs = BSSIDList.readlines()
            for i in range(0,len(BSSIDs)):
                BSSIDs[i]=BSSIDs[i].strip('\n')
            try:
                BSSIDs.remove('ff:ff:ff:ff:ff:ff')
                BSSIDs.remove('00:00:00:00:00:00')
            except:
                pass
            BSSIDs=[BSSID for BSSID in BSSIDs if BSSID]
            df = pd.DataFrame(BSSIDs)
            #df.to_csv('realtimeCapture.csv',header=["BSSID"], index=False)
            return df

        def conversation_parser(j):#j
            Dict={}
            conversationList=open("conversations"+str(j)+".txt", "r")#
            content = conversationList.readlines()
            sigStrength=0
            BSSID=''
            SSID_Dict={}
            BSSID_Nodes_Dict={}
            Data_Frames_Dict={}
            Beacon_Frame_Length_Dict={}
            Beacon_Count_Dict={}
            power_variance_Dict={}
            beaconFrame=False
            dataFrame=False
            for line in content:
                if "bytes on wire" in line:
                    bytes = int(((re.findall(r":\s[0-9]*",line)[0])[1:]).strip())
                    #print(bytes)
                elif "Signal strength" in line:
```

```
    sigStrength=(re.findall(r"\bSignal strength\s+(.*)$",line)[0])
    sigStrength=int(sigStrength[7:].strip("dBm"))
elif "IEEE 802.11 Beacon frame" in line:
    beaconFrame=True
elif "IEEE 802.11 QoS Data" in line:
    dataFrame=True
elif "IEEE 802.11 Data" in line:
    dataFrame=True
elif "Receiver address:" in line:
    receiverAddress = str(re.findall(r'\:(.*)',line)[0]).strip()
    #print(receiverAddress)
elif "Transmitter address" in line:
    BSSID=str(re.findall(r'\:(.*)',line)[0]).strip()
    #print(BSSID)
    if BSSID in Dict:
        Dict[BSSID].append(sigStrength)
    elif not BSSID in Dict:
        Dict[BSSID]=[sigStrength]
    if BSSID in BSSID_Nodes_Dict:
        BSSID_Nodes_Dict[BSSID].append(receiverAddress)
    elif not BSSID in BSSID_Nodes_Dict:
        BSSID_Nodes_Dict[BSSID]=[receiverAddress]
    if beaconFrame:
        if BSSID in Beacon_Frame_Length_Dict:
            Beacon_Frame_Length_Dict[BSSID].append(bytes)
        else:
            Beacon_Frame_Length_Dict[BSSID]=[bytes]
        if BSSID in Beacon_Count_Dict:
            Beacon_Count_Dict[BSSID]=Beacon_Count_Dict[BSSID]+1
        else:
            Beacon_Count_Dict[BSSID]=1
        beaconFrame=False
    elif dataFrame:
        if BSSID in Data_Frames_Dict:
            Data_Frames_Dict[BSSID]=Data_Frames_Dict[BSSID]+1
        else:
            Data_Frames_Dict[BSSID]=1
        dataFrame=False
    #BSSID="
    #print(line)
elif "SSID:" in line:
    try:
        SSID=str(re.findall(r"\bSSID:\s+(.*)$",line)[0]).strip()
        if not SSID == 'Not supported':
            #print(BSSID+":"+SSID)
            SSID_Dict[BSSID]=SSID
        SSID="
    except:
        continue
```

```
            for BSSID in Dict:
                power_variance_Dict[BSSID]=np.var(Dict.get(BSSID))
            #print(power_variance_Dict)
            #print(SSID_Dict)
            #print(Beacon_Count_Dict)
            #print(BSSID_Nodes_Dict)
            #print(Beacon_Frame_Length_Dict)
            return SSID_Dict,power_variance_Dict, Beacon_Count_Dict, BSSID_Nodes_Dict,
Data_Frames_Dict, Beacon_Frame_Length_Dict

        def Decrypt(SSID_Dict):
            command = "sudo tshark -r output.pcap -w output_decrypt.pcap -o
nameres.mac_name:FALSE -o wlan.enable_decryption:TRUE "
            passwordList=open("drone_password_dictionary.txt", "r")
            for SSID in SSID_Dict:
                for password in passwordList.readlines():
                    command = command + "uat:80211_keys:\"wpa-pwd\",\"" + password.strip("\n")
+ ":" + SSID.strip() + "\""
            os.system(command)



        os.chdir("/home/[user]/DroneFlightData/ ")

        os.system("sudo ip link set [interface] down")
        os.system("sudo iw [interface] set monitor none")
        os.system("sudo ip link set [interface] up")
        #os.system("sudo rm signalStrength*")
        j=0
        Drone_OUIs=['60:60:1f'] #append more as necessary
        Drone_SSIDs=[["PHANTOM3_",15],
                     ["MAVIC_AIR-",16],
                     ["TELLO-",12],
                     ["JY-FPV-",13]] #append more as necessary
        while True:
            wireshark(j)#j
            df=BSSID_Parser(j)#j
            df.columns= ['BSSID']
            BSSIDs=[]
            SSIDs=[]
            power_variance=[]
            Beacon_count=[]
            data_count=[]
            BSSID_nodes=[]
            Drone_SSID=[]
            Drone_OUI=[]
            beacon_frame_lengths=[]
            beacon_minus_len_ssid=[]
            #print(df)
```

```
            SSID_Dict,power_variance_Dict, Beacon_Count_Dict, BSSID_Nodes_Dict,
Data_Frames_Dict, Beacon_Frame_Length_Dict=conversation_parser(j)#j
            for index,row in df.iterrows():
                BSSIDs.append(row['BSSID'].strip())
                for OUI in Drone_OUIs:
                    if OUI in row['BSSID'].strip():
                        Drone_OUI.append(1)
                else:
                    Drone_OUI.append(0)
                if row['BSSID'].strip() in SSID_Dict:
                    SSIDs.append(SSID_Dict[row['BSSID'].strip()])
                    for attribute in Drone_SSIDs:
                        if attribute[0] in SSID_Dict[row['BSSID'].strip()] and
len(SSID_Dict[row['BSSID'].strip()])==attribute[1]:
                            Drone_SSID.append(1)
                    else:
                        Drone_SSID.append(0)
                else:
                    SSIDs.append("")
                    Drone_SSID.append(0)
                if row['BSSID'].strip() in power_variance_Dict:
                    power_variance.append(power_variance_Dict[row['BSSID'].strip()])
                else:
                    power_variance.append(0)
                if row['BSSID'].strip() in Beacon_Count_Dict:
                    Beacon_count.append(Beacon_Count_Dict[row['BSSID'].strip()])
                else:
                    Beacon_count.append(0)
                if row['BSSID'].strip() in Data_Frames_Dict:
                    data_count.append(Data_Frames_Dict[row['BSSID'].strip()])
                else:
                    data_count.append(0)
                if row['BSSID'].strip() in Beacon_Frame_Length_Dict:
                    arr=np.array(Beacon_Frame_Length_Dict[row['BSSID'].strip()])
                    #print(arr)

#beacon_frame_lengths.append(Beacon_Frame_Length_Dict[row['BSSID'].strip()][0])
                    cond = np.all(arr == arr[0])
                    if cond:
                        #print(Beacon_Frame_Length_Dict[row['BSSID'].strip()][0])

beacon_frame_lengths.append(Beacon_Frame_Length_Dict[row['BSSID'].strip()][0])
                    else:
                        beacon_frame_lengths.append(-1)
                else:
                    beacon_frame_lengths.append(0)
                if row['BSSID'].strip() in BSSID_Nodes_Dict:
                    nodes=np.array(BSSID_Nodes_Dict[row['BSSID'].strip()])
                    nodes = list(set(nodes))
```

```
        try:
            nodes.remove('ff:ff:ff:ff:ff:ff')
            nodes.remove('00:00:00:00:00:00')
        except:
            pass
        BSSID_nodes.append(len(nodes))
    else:
        BSSID_nodes.append(0)
#print(len(SSIDs))
#print(len(BSSIDs))
iterator=0
for ssid in SSIDs:
    if not ssid=="":
        if beacon_frame_lengths[iterator] == 0 or beacon_frame_lengths[iterator] == -1:
            beacon_minus_len_ssid.append(0)
        else:
            beacon_minus_len_ssid.append(beacon_frame_lengths[iterator]-len(ssid))
    else:
        beacon_minus_len_ssid.append(0)
    iterator=iterator+1
lens_validation=[len(BSSIDs),
                 len(SSIDs),
                 len(power_variance),
                 len(Beacon_count),
                 len(data_count),
                 len(Drone_SSID),
                 len(Drone_OUI),
                 len(beacon_frame_lengths),
                 len(beacon_minus_len_ssid),
                 len(BSSID_nodes)]
small=min(lens_validation)
data = {'BSSIDs':BSSIDs[0:small],
        'SSIDs':SSIDs[0:small],
        'Beacon Frame Length':beacon_frame_lengths[0:small],
        'Power Variance':power_variance[0:small],
        'Beacon Count':Beacon_count[0:small],
        'Data Frame Count':data_count[0:small],
        'Drone SSID?':Drone_SSID[0:small],
        'Drone OUI?':Drone_OUI[0:small],
        'Beacon-len(SSID)':beacon_minus_len_ssid[0:small],
        'Nodes':BSSID_nodes[0:small]}
dataframe = pd.DataFrame.from_dict(data)
print(dataframe)
dataframe.to_csv('capture'+str(j)+'.csv',index=False)
j=j+1
file='capture'+str(j)+'.csv'
os.system("python ml.py "+file)
```