NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

COUNTERING SMALL UNMANNED AIRCRAFT SYSTEMS WITH ADVANCED DATA ANALYSIS AND MACHINE LEARNING

by

Robert Miske

March 2023

Thesis Advisor:	Ruriko Yoshida
Second Reader:	Ross J. Schuchard

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.			
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** March 2023	**3. REPORT TYPE AND DATES COVERED** Master's thesis	
4. TITLE AND SUBTITLE COUNTERING SMALL UNMANNED AIRCRAFT SYSTEMS WITH ADVANCED DATA ANALYSIS AND MACHINE LEARNING			**5. FUNDING NUMBERS**
6. AUTHOR(S) Robert Miske			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.			**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

In January 2021, the DOD released its first *Counter-Small Unmanned Aircraft Systems Strategy* to address the growing risk to military personnel, facilities, and assets posed by the rapid technological advancement and proliferation of sUAS. Existing counter-drone capabilities—heavily reliant on electronic warfare to disrupt the communication link between user and device—no longer address an evolving threat that includes autonomous drones, COTS technology, and an increasing number of drones in the airspace that can overwhelm a C-sUAS operator. To counter the increasingly complex small drone threat, the Army-led Joint Counter-sUAS Office is pursuing materiel and non-materiel solutions for its new system-of-systems approach. One vexing C-sUAS challenge involves radar detection systems discriminating some sUAS from other flying objects, like birds, due to their comparable size, slow movement, and low altitude. Inaccurate or inefficient sUAS classification using radar data can be a force protection threat due to the limited number of electro-optical sensors and human operators for classification at-scale. This thesis uses bird and drone radar track data from two different training environments to explore hidden structure in the data, develop independent unsupervised and supervised learning models using the two datasets, and experiment with data sampling and feature engineering to improve upon model robustness to different environments and dynamic environmental conditions.

14. SUBJECT TERMS small unmanned aircraft system, sUAS, UAS, unmanned aerial vehicle, UAV, counter unmanned aircraft system, C-UAS, C-sUAS, drone, counter-drone, radar, detection, classification, discrimination, machine learning, supervised learning, unsupervised learning, data analysis, air defense, strategy, feature engineering, model robustness			**15. NUMBER OF PAGES** 129
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

COUNTERING SMALL UNMANNED AIRCRAFT SYSTEMS WITH ADVANCED DATA ANALYSIS AND MACHINE LEARNING

Robert Miske
Lieutenant Colonel, United States Army
BS, Notre Dame University, 2001
MS, Defense Analysis (Terrorist Operations & Financing),
Naval Postgraduate School, 2012
MS, Information Operations, Naval Postgraduate School, 2013

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
March 2023

Approved by: Ruriko Yoshida
Advisor

Ross J. Schuchard
Second Reader

W. Matthew Carlyle
Chair, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

In January 2021, the DOD released its first *Counter-Small Unmanned Aircraft Systems Strategy* to address the growing risk to military personnel, facilities, and assets posed by the rapid technological advancement and proliferation of sUAS. Existing counter-drone capabilities—heavily reliant on electronic warfare to disrupt the communication link between user and device—no longer address an evolving threat that includes autonomous drones, COTS technology, and an increasing number of drones in the airspace that can overwhelm a C-sUAS operator. To counter the increasingly complex small drone threat, the Army-led Joint Counter-sUAS Office is pursuing materiel and non-materiel solutions for its new system-of-systems approach. One vexing C-sUAS challenge involves radar detection systems discriminating some sUAS from other flying objects, like birds, due to their comparable size, slow movement, and low altitude. Inaccurate or inefficient sUAS classification using radar data can be a force protection threat due to the limited number of electro-optical sensors and human operators for classification at-scale. This thesis uses bird and drone radar track data from two different training environments to explore hidden structure in the data, develop independent unsupervised and supervised learning models using the two datasets, and experiment with data sampling and feature engineering to improve upon model robustness to different environments and dynamic environmental conditions.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

List of Figures

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

AGL above ground level

AIC Akaike Information Criteria

AUC area under the curve

ATP Army Techniques Publication

C3 communications, command, and control

CART classification and regression tree

CBR chemical, biological, and radiological

COP common operational picture

COTS commercial-off-the-shelf

CTS Coastal Training Site

C-sUAS counter-small unmanned aircraft systems

C-UAS counter-unmanned aircraft systems

DARPA Defense Advanced Research Projects Agency

DHS Department of Homeland Security

DOD Department of Defense

EA executive agent

EDA exploratory data analysis

ELINT electronic intelligence

EO electro-optical

EW	electronic warfare
FOB	forward operating base
GPS	global positioning system
HSOAC	Homeland Security Operational Analysis Center
ICUAS	International Conference on Unmanned Aircraft Systems
IR	infrared
ISR	intelligence, surveillance, and reconnaissance
JCO	Joint C-sUAS Office
JSON	JavaScript Object Notation
KNN	k-nearest neighbors
LIDAR	light detection and ranging
m-D	micro-Doppler
MDV	minimum detection velocity
MFP	mobile force protection
MMV	maximum measurable velocity
MSS	mission support site
NDJSON	Newline Delimited JavaScript Object Notation
NNSS	Nevada National Security Site
PCA	principal component analysis
RCS	radar cross section
RF	radio frequency
RNN	recurrent neural network

ROC receiver operating characteristic

SDR software defined radio

SECARMY Secretary of the Army

SECDEF Secretary of Defense

SNR signal-to-noise ratio

sUAS small unmanned aircraft systems

TB2 Turkish Baykar Bayraktar

UAS unmanned aircraft systems

USBP U.S. Border Patrol

THIS PAGE INTENTIONALLY LEFT BLANK

Executive Summary

The rapid advancement of drone technology—including sensor miniaturization, battery longevity, flight efficiency, and improved control mechanisms—combined with the increasing affordability and commercial utility of drones, has given rise to their ubiquity in society. However, with the growing number of benevolent purposes for which drones can be used comes the responsibility of appropriately regulating drone use to minimize the potential for both high-risk accidental occurrences and nefarious activity by malevolent actors, including terrorists and hostile regimes alike. While unmanned aircraft systems (UAS) have been around for decades, the global proliferation of small unmanned aircraft systems (sUAS) presents a particularly vexing challenge for the U.S. Department of Defense (DOD) because of the requirement not only to protect U.S. airspace, installations, and critical infrastructure from this increasingly capable new threat, but to extend this force protection to a forward operating base (FOB) or transient mission support site (MSS), in addition to providing mobile force protection (MFP) in combat. Azerbaijan's series of attacks (using low-cost Turkish Baykar Bayraktar [TB2] drones) on its Armenian neighbor during a 44-day war in Nagorno-Karabakh in 2020 and Ukraine's stalwart defensive and counter-attacks (aided by [TB2] drones and thousands of other sUAS) to negate the overwhelming military advantage of the Russian advance early in the 2022 Russia-Ukraine War provide two striking examples of the vulnerability of legacy combat systems to the asymmetric threat of drones at-scale.

While there are multiple aspects of the counter-small unmanned aircraft systems (C-sUAS) problem set—from detection to kinetic or non-kinetic threat response—that defense industry is working to address, data scientists have been particularly drawn to the challenge of rapidly and efficiently discriminating sUAS from birds and other atmospheric clutter by radar systems. Radar systems generally have two primary problems with detecting and classifying sUAS. The first concerns the combination of their size (easily conflated with birds) and speed (either very fast or slow, including their hover capability). The second involves characterizing a diversity of sUAS types (between the two general rotary wing and fixed-wing categories) that have a variety of flight phenomenology, radar cross sections (RCS), optical emissions, reflectivity characteristics, and material structures. Although some research in this area has been dedicated to exploring a *system-of-systems* approach that includes other sensor types—such as electro-optical (EO)/infrared (IR), acoustic, and human surveillance—to reduce radar system vulnerabilities, this solution assumes a present-day luxury of having such a sensor suite working in tandem at a fixed location. The importance of pursuing this "gold standard" solution for efficiently moving from detection to classification of aerial objects, however, does not negate the continued importance of improving upon the discriminatory performance of radar systems, either *stand-alone* or within an overarching system of different sensor types.

In consultation with Anduril Industries and using radar track data of birds and drones from two distinctly different training environments, this thesis purports to achieve two objectives. First, we sought to validate (or improve upon) the performance of existing classification algorithms from defense industry using our independent unsupervised and supervised learning methodology with models trained in each of the two environments. Second, we attempted to enhance the *robustness* of our models to the two different environments and dynamic environmental conditions (i.e., precipitation and wind) that currently necessitate a lengthy and costly system calibration process in each new environment.

To achieve these two objectives, our research experimented with radar track data (provided by Anduril Industries) of hundreds of birds and drones from each of the two training environments by developing, testing, and validating the discriminatory performance of a variety of unsupervised and supervised learning models on birds and drones from both the environment in which we trained the models and the alternate environment. Through our independent methodology, our top-performing models for the two training environments successfully validated the performance of Anduril's classifier (as provided to us by our data sponsor) by achieving 97% and 98% accuracy (respectively for the two environments), with our models trained and validated in the same environment. However, our observation of a 20–25% decrease in accuracy (by the top-performing models) in an alternate environment validation and our intuition about the distinct differences in the datasets and models from the two environments prompted modifications to a second iteration of our methodology that achieved marginal improvements in model *robustness*. This thesis concludes with four recommendations to continue statistical and machine learning research using this methodology but exploring the collection of additional radar track data features with the goal of better capturing the flight phenomenology differences between birds and different types of drones.

Acknowledgments

I would like to thank my thesis advisory team, Dr. Ruriko "Rudy" Yoshida and LTC Ross Schuchard, for their unending patience, commitment, and encouragement through the highs and lows of the last couple of years in bringing this thesis to its culmination. I truly appreciated their willingness to allow me to take on this critically important topic for the U.S. Army and Department of Defense and explore multiple promising gambits with our data, while astutely keeping me grounded in scope.

I would also like to thank the tag team of Scott Sanders and Scott Zolendziewski, along with Sal Lombardo and all of the Anduril Industries engineering team, for providing access to their radar track data and sharing their valuable time and energy to allow me to get my research across the finish line. I hope that, in some way, our research and recommendations will be valuable to Anduril and others across defense industry working on this problem set.

Last in sequence—but certainly most deserving of my gratitude through what was a grueling and seemingly unending quest—I would like to thank my wife, Lacy, and our resilient children, Mary-Logan, Wilson, Jackson, Liadan, and Martha, for enduring the many hours I have spent on this thesis. If you are reading this, then we are finally at the end of this academic journey, and you are as much a part of this degree as I am. Thank you for your love, prayers, and support through it all.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1: Introduction

> The increased availability of inexpensive, capable sUAS [small unmanned aircraft systems] is allowing government, industry, and the public to employ what was once available only to the military and a small number of dedicated hobbyists. These systems are the fastest growing segment of the aviation industry, and this growth has dramatically increased the risk of sUAS hazards for the military. Improvements in sensor miniaturization, battery technology, flight performance, and control mechanisms, along with reductions in price and regulations, have led to increased interest in their commercial utility. As new technology continues to improve capabilities, commercial applications will also expand. Large commercial entities are pursuing UAS [unmanned aircraft systems] operations, resulting in a dramatic proliferation of highly capable drones occupying U.S. airspace and overflying DOD [Department of Defense] installations in the United States and around the world. To adapt to these changes in the air domain, the DOD must adopt a posture of anomaly detection by seeking ways to highlight abnormal behavior and focus attention on those sUAS identified as potential threats and hazards. (Miller 2021, p. 7)

1.1 Problem Statement

Towards the goal of rapidly and accurately detecting and classifying sUAS (specifically groups 1 and 2, Figure 2.1), current systems have achieved an accuracy rate of 85–88% using exclusively radar track data that includes the radar cross section (RCS) and trajectory (velocity components and altitude) of an approaching aerial object to discriminate sUAS from other objects like birds (Liu et al. 2021). Using a multi-modal detection system, the process usually takes 4–5 seconds from detection to classification. For enhanced security, the threshold of existing classification algorithms can be adjusted to ensure a fewer number of "false negatives" occur when classifying drones, but this adjustment can result in an unreasonably large number of "nuisance alerts" or "false positives" to be verified by a human-in-the-loop using additional optical imagery or some other sensor. The additional screening process can be cumbersome and fatigue human operators, or quickly overwhelm a limited quantity of electro-optical (EO) sensors, especially at a smaller forward operating base (FOB) or transient mission support site (MSS). Experimental testing using a recurrent neural network (RNN) has consistently achieved classification accuracy rates above 90% and as high as 98%, but this work must be verified in different environments and should be compared with more intuitive models employing supervised and unsupervised learning. The research in this thesis will use labeled (drone or bird) radar track data from two training sites with distinctly different environments to explore potentially hidden structure in the data, develop and test models using supervised and unsupervised learning with data from each training site, and evaluate the robustness of the respective models by validating their performance at both training sites. In doing so, this research gleans important insights for consideration in future model development for discrimination of drones and birds to enhance counter-small unmanned aircraft systems (C-sUAS) performance.

1.2 Scope of Research

Using labeled drone and bird radar track data from two unique training sites, this thesis applies a two-phase statistical and machine learning approach to develop, test, and validate preferred models for classifying sUAS across different environmental conditions. Beginning with mature radar track data, in 10 Hz increments, from a multi-modal detection system, our research sought to scrupulously explore the data from each of the two training sites for any potential hidden structure in the data using unsupervised learning methods in the first phase. After analyzing the initial results, we considered some of the more promising data clustering for integration with the development of supervised learning algorithms in the second phase of our model development. In the second phase, we developed and tested a range of different prominent supervised learning methods to compare with one another. We then repeated this two-phase process with an independent data set from an alternate training site having different environmental conditions. After validating our respective models on unseen data from the same training site on which we trained them, we then assessed the robustness of our top-performing models developed from each training site by performing a second validation on the unseen labeled bird and drone track data from the alternate site.

After performing this complete methodology once to establish a benchmark classification performance of our models on the alternate training site data, we attempted to reduce the apparent overfitting of our models and improve model robustness by altering our data sampling and adding three components of the acceleration (derived from the velocity components and timestep increments). In doing so, we observed improvements in the validation of our models both using data from the same training site on which we trained the models and on data from the alternate training site. Although we still observed a drop-off in prediction accuracy of all of our models during our validation on the alternate training site from the one on which we trained our models, we derived some insights through our methodology, feature engineering, and sampling adjustments that exploited the differences in the flight kinematics of birds and drones and can be further refined in future model development using statistical and machine learning with radar track data.

While some of our clustering analysis using unsupervised learning in the first phase of our methodology leads us to believe in the potential to also discriminate between the flight kinematics of different design types of sUAS (i.e., fixed-wing and rotary) for identifying particular threats or performing sponsor attribution, this thesis only designed and evaluated models based on their ability to distinguish birds from drones both in the same environment in which we trained them, and in a drastically different environment. Additionally, while counter-unmanned aircraft systems (C-UAS) systems often incorporate other sensors, including EO, infrared (IR), radio frequency (RF), acoustic, electronic intelligence (ELINT), and light detection and ranging (LIDAR), during the complete process from sUAS threat detection to mitigation, this research only considered features derived from radar sensors in the development of its models. Although exclusively using radar data for this research, we discuss the potential for generalizing this methodology to incorporate non-radar sensor data in Section 5.2.1.

1.3 Analytical Framework

This thesis is composed of five chapters. Chapter 2 provides a basis for understanding the problem set of detecting and classifying sUAS by providing the context of current UAS and detection system capabilities and a review of existing C-sUAS strategy and research to improve the discriminatory performance of radar detection systems. In Chapter 3, we describe our two-phase statistical learning approach using unsupervised and supervised learning with radar track data and the adjustments made in the second iteration of our methodology to improve upon our models' validation results against birds and drones in both the training environment and alternate environment. Chapter 4 explains the progression of our work in terms of the initial benchmark results and subsequent improvements in our methodology after analyzing the interim performance of our models from both training sites. In Chapter 5, we summarize the key findings of our research and provide some recommended areas of future research for improving C-sUAS performance using statistical and machine learning.

CHAPTER 2: Background

The prevalence, cost-effectiveness and increasing sophistication of UAS has opened the imaginations and ingenuity of many hobbyists, engineers, and entrepreneurs around the world to the unending possibilities of employing them for a myriad benevolent and beneficial purposes. The United States DOD, along with defense ministries across the globe, has successfully employed unmanned systems to support combat operations for decades, and this century has seen a proclivity of lethal kinetic strikes by unmanned systems as human trust in their functionality and precision has improved. In its Roadmap of UAS for 2010-2035, U.S. Army leadership describe "Army UAS supporting Army and Joint operations [as providing] the Warfighter a disproportionate advantage" over its adversaries (Dempsey and Rasmussen 2010, p. 7). At the time of the 2010 roadmap, only a handful of countries, with huge defense budgets and the operational mandate to conduct such warfare against legal combatants, retained the large-scale capability of unleashing lethal kinetic warfare with pinpoint accuracy through unmanned systems (Dempsey and Rasmussen 2010). However, in the last decade, the increasing demand for and prevalence of small unmanned systems in the commercial sector has paralleled a widespread technological advancement and distribution of unmanned systems with lethal payloads across the world (Miller 2021).

2.1 UAS Threat

Continuous drone testing and more frequent demonstration of successful large-scale and costly drone strikes by state and non-state actors has exacerbated tensions in recent years, evidenced by the strikes on Erbil Airport (Iddon 2021) and Saudi Arabia's Aramco Oil Refineries (Barrington 2021) in consecutive months in 2021. About 6 months earlier, Azerbaijan employed low-cost Turkish Baykar Bayraktar (TB2) drones in a series of attacks against their Armenian neighbor, destroying "185 T-72 tanks; 90 armored fighting vehicles; 182 artillery pieces; 73 multiple rocket launchers; 26 surface-to-air missile systems (including a Tor system and five S-300s); 14 radars or jammers; one SU-25 war plane; four drones and 451 military vehicles" during a 44-day war over the mountainous enclave of Nagorno-Karabakh (Dixon 2020, p. 4). In June 2020, the same TB2 drones avoided engagement and successfully employed laser-guided missiles to destroy dozens of Russian-made anti-aircraft vehicles in Syria. In addition to the financial and human toll that such strikes can impose on high-payoff targets, the attacks by Azerbaijan's drones against Armenian military assets and Iran-backed Houthi rebel's drones against Saudi Arabia's oil refineries also demonstrated the advent of a novel strategy in which unmanned aircraft exploited the vulnerabilities of air defense systems (Dixon 2020). By carrying out sophisticated reconnaissance and complex attacks with a succession of drone incursions to probe, saturate and find gaps in the existing air defense systems, follow-on armed drones were able to find and exploit those vulnerabilities to effectively have free reign on unsuspecting and unprotected high-value targets. Over the span of less than a year in 2020 and 2021, not only did these attacks highlight the vulnerabilities in sophisticated air defense systems using comparatively inexpensive drones, but they caused some to speculate that the attacks revealed the beginning of a new paradigm in the dimensions of warfare previously dominated by ground battles and traditional air power (Dixon 2020).

If there were still some naysayers skeptical about the immanency and pervasiveness of this shifting combat advantage and the vulnerability of expensive legacy combat power to comparatively lower cost combat drones, the loss of 400 out of Russia's 1200 tanks within the first month of its February 2022 invasion of Ukraine would cause anyone to pause and take a closer look. As one example of the asymmetric UAS threat, Ukraine's purchase of TB2 drones in the single-digit millions of dollars each pales in comparison to the Russian surface-to-air missile system shown destroyed by that same drone in a March 13, 2022 video and estimated to be worth up to $50 million (Wang 2022). After many military experts predicted Russia would over-match Ukrainian defenses and defeat its military within a few weeks due to its overwhelming combat power, two months later Russia was effectively stalemated and unable to advance on key population centers due, in large part, to a two-pronged Ukrainian drone strategy. In addition to the "U.S. provision of hundreds of kamikaze-like loitering drones that can hunt targets for hours before dropping down to detonate a deadly munition," Ukraine's complement of low-cost commercial-off-the-shelf (COTS) drones as spotters for artillery and other munitions proved effective in reducing the exposure of its own troops, stalling the Russian advance, and forcing the Russian military to divert a significant portion of its resources to conducting anti-drone warfare (Detsch 2022). Two months into the Russian invasion, experts noted an even greater impact of the COTS drone technology than in the 2020 Nagorno-Karabakh war, in which this era of drone prominence emerged. Also noteworthy in this 2022 instance of drone warfare in Ukraine, observers saw a shift from the heavier combat-capable drones to a ubiquity of Chinese-made DJI Mavic drones and the Polish and Turkish counterpart surveillance drones that provided both Russians and Ukrainians with a clearer common operational picture (COP) at times and improved each side's targeting to an effective stalemate (Detsch 2022). The war that originated with a large-scale air and ground invasion and massive aerial bombardment across all of Ukraine had been effectively reduced to both sides in the Donbas Region relying heavily on small expendable drones that could be bought at big-box stores like Walmart and operated by a child (Detsch 2022).

2.2 UAS Categories and Capabilities

The TB2 drone—featured in the 2020 Nagorno-Karabakh War, the 2021 Erbil and Saudi Arabia high profile attacks, and Russia's 2022 war in Ukraine—is an example of a group 4 UAS, considered a large UAS. Starting in 2009, the DOD began categorizing UAS in this new group system, from 1 to 5, increasing in capability. As shown in Figure 2.1, sUAS are those in groups 1, 2 and 3; groups 4 and 5 are larger and typically controlled or employed by state actors. In accordance with its standards, sUAS are lighter in weight (less than 1320 lbs), operate at a lower altitude (less than 18,000 ft MSL) and fly at a slower airspeed (less than 250 knots) than UAS in the larger groups (Miller 2021). As seen in the successful drone attacks in 2020, 2021, and 2022, larger UAS continue to present an existential threat and significant challenge for air defense systems across the world (including the U.S. Integrated Air and Missile Defense Battle Command System) while travelling farther, flying faster and carrying a larger explosive payload or more lethal weapon systems than sUAS. Due to their destructive capacity and ability to exploit vulnerabilities in air defense systems, novel strategies involving larger UAS will force the United States DOD and other defense ministries to continually adapt their air defense systems to try and stay ahead of the threat. In spite of the greater destructive capacity and higher payoff of the larger UAS threat, the rapid proliferation and lower cost of sUAS, combined with their increasing weaponization and other nefarious activities by criminals and non-state actors, can present an even greater challenge for C-UAS systems with the responsibility for protecting personnel, critical assets, or the operational maneuver of forces on the battlefield (Suits 2020).

UA Category	Maximum Gross Takeoff Weight (lbs.)	Normal Operating Altitude (feet)	Speed (KIAS)
Group 1	0-20	< 1200 AGL	100 knots
Group 2	21-55	< 3500 AGL	< 250 knots
Group 3	< 1320	< 18,000 MSL	< 250 knots
Group 4	> 1320		Any Airspeed
Group 5	> 1320	> 18,000 MSL	Any Airspeed
Legend: AGL – above ground level; MSL – mean sea level; KIAS – knots indicated airspeed			

Figure 2.1. The DOD categorizes UAS into Groups 1-5, with UAS in higher groups increasing in size, speed, and operating altitude. Groups 1-3 are considered sUAS. Source: Miller (2021, p. 29).

2.3 Adversarial sUAS

In focusing our research on countering the nefarious uses of sUAS, we relied heavily on the 2020 sUAS Adversary Capabilities Study by the RAND Corporation's Homeland Security Operational Analysis Center (HSOAC). The study assessed the nefarious sUAS uses by deriving use cases from multiple sources but primarily the Department of Homeland Security (DHS) C-UAS Capabilities Analysis Working Group. Since the use cases could not comprehensively address all types of targets and mission sets, the study focused on relevant technologies and particular high-impact scenarios (Wilson et al. 2020). After establishing a baseline set of nefarious sUAS uses, the study developed a framework of adversary sUAS use cases (Figure 2.2) by extrapolating the baseline nefarious uses into more generic categories of threats, or "threat vectors."

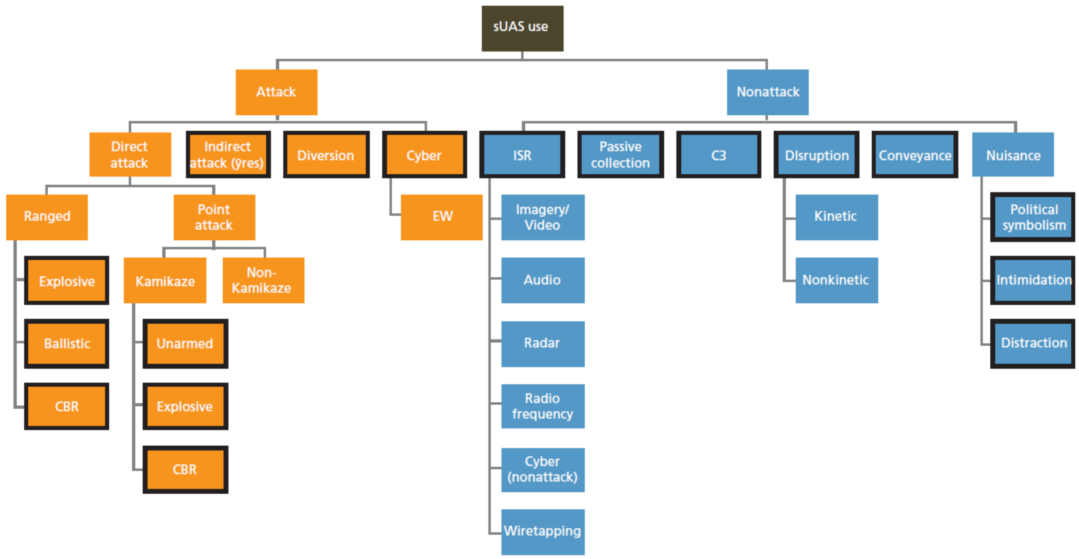


Figure 2.2. Framework of Adversary sUAS Use Cases Organized by Category (Threat Vector). Threat vectors highlighted in black are those that were used in a group risk analysis performed by RAND Corporation's Homeland Security Operational Analysis Center in its 2020 sUAS Adversary Capabilities Study. Source: Wilson et al. (2020, p. 68).

From this framework of threat vectors–broadly categorized into *attack* and *nonattack*, and further decomposed within each category–the study outlined 16 threat vectors (Table 2.1) on which to perform a risk assessment using relative likelihood and consequence. After applying a composite risk assessment of these threat vectors (Figure 2.3), the study identified a subset–intelligence, surveillance, and reconnaissance (ISR), conveyance, kamikaze explosive attack, and chemical, biological, and radiological (CBR) attack–for case studies as high-risk threat vectors scenarios (Wilson et al. 2020). Upon narrowing the focus of the case studies to these four high-risk threat vectors, the study proceeded to further analyze the likely and high-impact scenarios by determining the requisite performance capabilities necessary to effectively engage in each of the missions. Although sUAS optimally designed for the *nonattack* ISR or conveyance missions could conceivably be configured for *attack* missions, we focused on sUAS optimally designed for mission success in the high-risk *attack* missions: kamikaze explosive attack and CBR.

Table 2.1. sUAS Threat Vectors. Source: Wilson et al. (2020, p. 69–70).

Threat Vector	Description	Example
Kamikaze point attack	Adversary directs UAS chassis into a target, with only the UAS itself used as a weapon	UAS intentionally flies into aircraft engine
Kamikaze point attack (with explosives)	Adversary directs UAS chassis into a target, with the UAS containing explosives for greater damage	UAS with plastic explosive lands on facility and detonates
CBR attack	Adversary uses UAS to launch a CBR attack, which could be kamikaze, spraying a substance, or firing a projectile from stand-off	UAS deposits dirty bomb to facility roof at night, and bomb activates the next day; UAS sprays aerosolized anthrax above a facility
Stand-off attack (with firearm)	Adversary uses UAS equipped with a firearm to engage target	Uzi-mounted UAS shooting into crowd
Stand-off attack (with explosives)	Adversary uses UAS for delivery of explosives at range	UAS drops grenades into a crowd
Indirect ranged attack	Adversary uses UAS for target acquisition and range-finding for a human-operated long-range weapon	UAS surveils a facility for soft targets to cue mortars located hundreds of meters away
Diversion in support of attack	Adversary uses UAS to distract or divert friendly forces in support of a larger, manned attack	UAS swarm draws security forces to far side of facility while manned attack hits main entrance
Active cyberattack/ disruption	Adversary uses UAS as a platform for other devices to launch malicious cyberattack	UAS uses location to gain local network access and installs malware that provides remote users access/ privileges
Passive electronic collection	Adversary uses UAS as a platform for other devices to collect electronic information from target	UAS with Wi-Fi sniffer lands on facility roof and monitors traffic; UAS captures two-way radio transmission from law enforcement
Communications, command, and control (C3) attack	Adversary uses UAS to support command of and communication between physically distant adversary actors	UAS serves a mobile relay node for line-of-sight-limited communications (including for other UAS); UASs track multiple smuggler operations for cartel boss
ISR	Adversary uses UAS to detect, identify, and monitor friendly forces to hinder friendly operations	UAS finds and follows U.S. Border Patrol (USBP) agents to allow smugglers to evade them; UAS tracks security shift changes at facility
Disruption/ harassment	Adversary uses UAS flying in close proximity to friendly forces to hinder friendly operations	UAS swarm flies into law enforcement officers at outset of raid to buy adversary time; UAS buzzes aircraft during training exercise
Conveyance	Adversary uses UAS to convey illicit items across or into restricted areas	UAS transports drugs over border and into U.S. urban area; UAS transports barred weapons into prison
Political symbolism	Adversary uses UAS for act of political demonstration	UAS defaces hard-to-reach symbol at government facility; UAS spray paints slogan on government vehicles
Intimidation	Adversary uses UAS in publicized demonstration of capability to elicit concessions from friendly actors	Massive UAS swarm used in coordinated show of force; UAS delivers supposed explosive payload and operators demand ransom
Distraction	Adversary uses UAS to distract friendly actors not in support of an attack	UAS swarm pesters lone USBP agent on foot; UAS loiters above facility courtyard

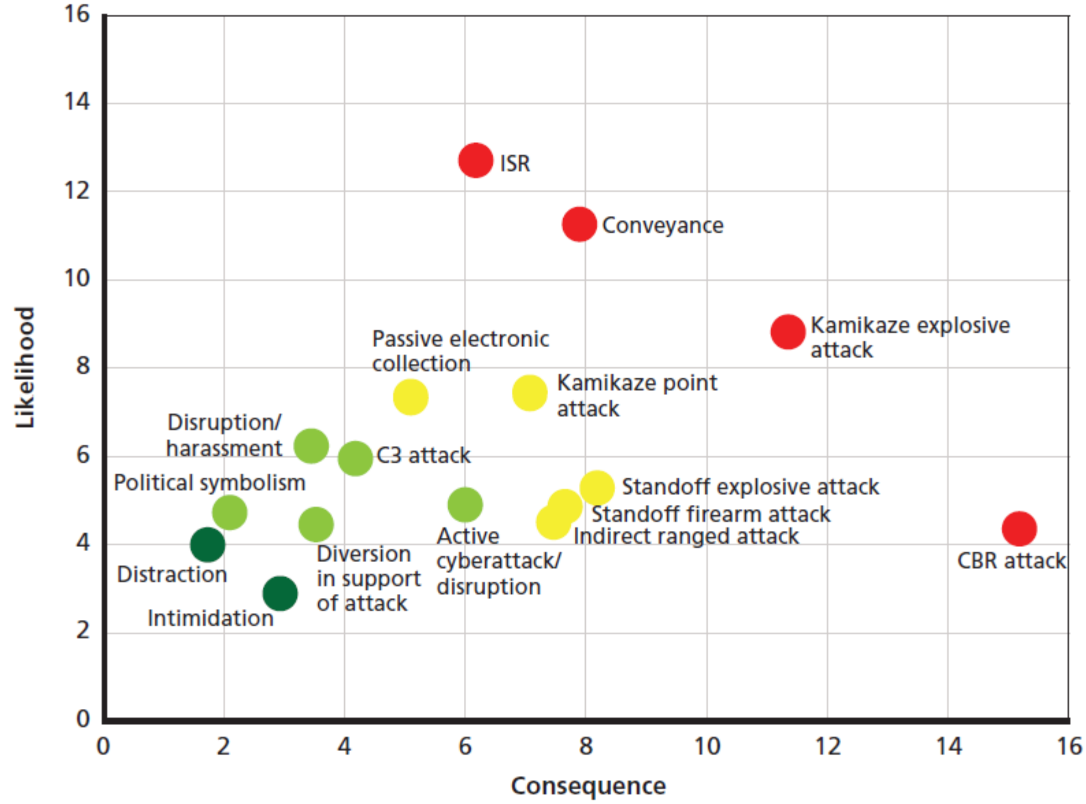


Figure 2.3. Risk Assessment of Adversarial sUAS Threat Vectors. The high-risk threat vectors, as determined by HSOAC team members and reviewers, are those colored red. Source: Wilson et al. (2020, p. 71).

To better identify the types of adversarial sUAS posing the highest risk while executing the likely and high-impact missions, the HSOAC study divided sUAS into three levels (low, moderate, and high) based on four performance characteristics: range (miles), endurance (minutes), payload (lbs), and speed (knots). The low, moderate, and high specifications for each of the four sUAS performance characteristics can be seen in Figure 2.4, and both the assessed sUAS performance level requirements and number of sUAS meeting those requirements, for each of the high-risk missions, can be seen in Figure 2.5.

Categories of sUAS Performance Capability

Performance Characteristic	Low	Moderate	High
Range (miles)	< 5	≤ 5, < 20	≤ 20
Endurance (minutes)	< 30	≤ 30, < 60	≤ 60
Payload (lbs)	< 2	≤ 2.2, < 10	≤ 10
Speed (knots)	< 25	≤ 25, < 75	≤ 75

Figure 2.4. Categories of sUAS Performance Capability. Source: Wilson et al. (2020, p. 72).

sUAS Performance Needs to Effectively Engage in Mission

					sUASs Capable of Completing Mission	
Scenario	Range	Endurance	Payload	Speed	%	Number
ISR	Low	High	Low	Low	23	332
Conveyance	High	High	High	Low	4	53
Kamikaze explosive attack	Low	Low	Moderate	High	5	72
CBR	Moderate	Moderate	High	Low	6	84

Figure 2.5. sUAS Performance Requirements by Mission. Source: Wilson et al. (2020, p. 73).

This triage of potentially adversarial sUAS and their respective capabilities is useful for understanding the C-UAS threat picture in terms of most likely or most dangerous courses of action and allows DHS, DOD or other agencies to better assess vulnerabilities in the protection of U.S. institutions or forces operating around the world. For the purposes of this research, understanding the commercially available sUAS platforms capable of effectively conducting high-risk missions also allows analysts to improve C-sUAS systems by studying the tracked flight motion patterns, known as white kinematics, to assist in the anomaly-detection necessary to discriminate drones from birds. As an example, at the time of this report, only 5% (72) of the commercially available sUAS were capable of carrying a moderate payload (between 2.2 and 10 lbs) and flying at a high speed (greater than 75 knots) in order to effectively engage in a kamikaze explosive attack mission. This information could be used to assess the range and endurance capabilities of those drones that can effectively engage in the mission to identify an appropriate sensor suite (in addition to radar) to improve timely classification at a shorter range. Although some research in this area, such as Liang et al. (2021), has explored methods for deriving malicious intent among multiple drones operating in the same air space, this research area continues to evolve as countries and localities establish and refine protocol and standards for authorized employment of sUAS.

2.4 The Challenge of sUAS

There are two primary problems involved in the detection and classification of sUAS. The first concerns the detection and classification of very small objects moving at either very fast or slow (including hover) speeds. Group 1 sUAS (i.e., DJI Phantom) are typically under 9 kg, portable and manually launched. Group 2 sUAS (i.e., Puma LE) typically weigh 9 to 25 kg and usually launch using more advanced mechanisms. In the visible, thermal, radar, and acoustic domains, the smaller signatures can be difficult to detect through the clutter echoes caused by various atmospheric conditions at greater ranges. The exceedingly slow or fast speeds pose a challenge for radar sensors because of their minimum detection velocity (MDV) or maximum measurable velocity (MMV) (Poitevin et al. 2017).

The second problem concerns the challenge of identifying the different types (rotary or fixed wing) sUAS because of their different flight phenomenology and diversity of sUAS characteristics in their phenomenology because each has a variety of "material structures, optical emissions, reflectivity characteristics, and radar cross sections" (Henderson 2020, p. 3). This variability in the sUAS platforms means that the entire problem, from detection to mitigation, cannot be addressed by a single system and will require a *system-of-systems* approach to be successful discriminating based on the "key detectable elements of an sUAS: shape, size, material structure, velocity, communication signals, and high-frequency propeller or blade movement or acoustics" (Henderson 2020, p. 3).

Assuming that a potentially adversarial UAS is approaching a military base or other high value target with a lethal kinetic capability, the C-UAS process can be decomposed into three steps (Figure 2.6) and becomes a problem of detection, identification, and tracking to be able to classify the UAS with high enough confidence to make a threat decision regarding an appropriate kinetic or non-kinetic response before it can effectively target friendly forces.

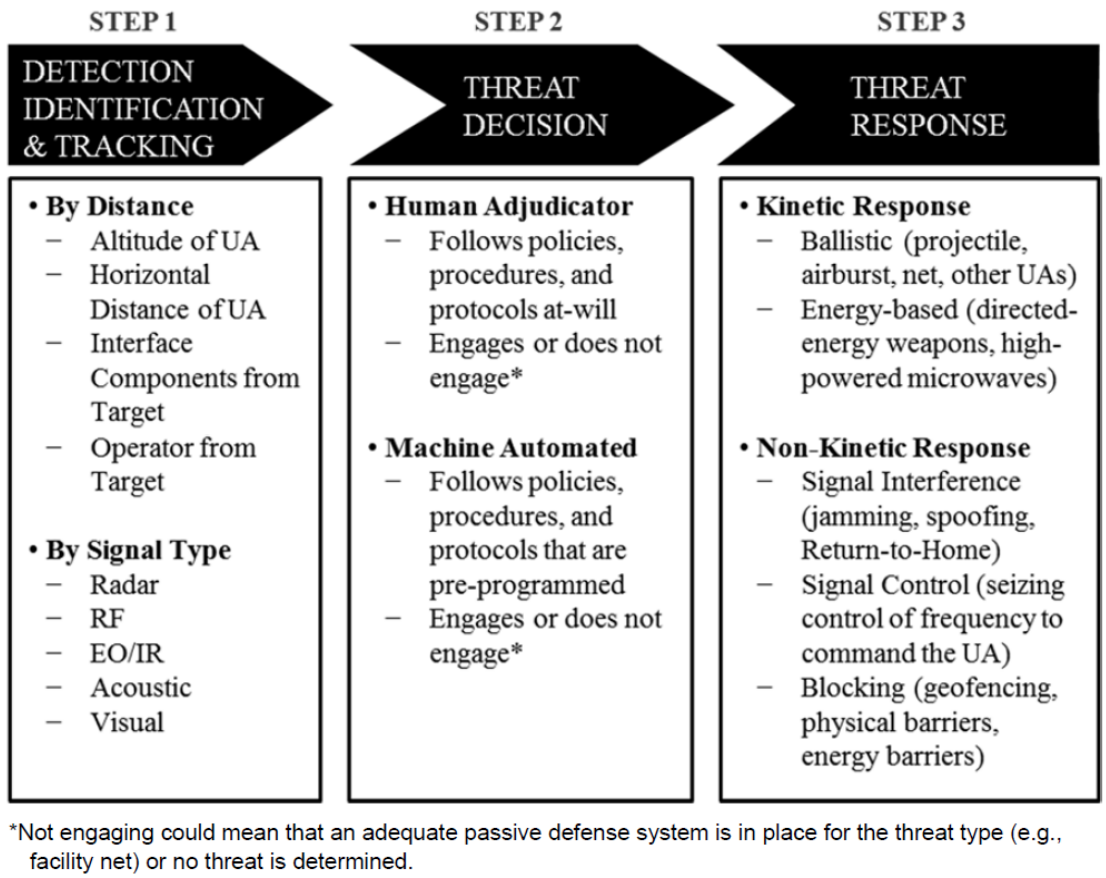


Figure 2.6. The C-UAS Process. Source: Herrera et al. (2017, p. 5)

Although the DOD has countermeasures designed to address this threat, Maj. Gen. Sean A. Gainey, director for the counter-drone technologies program, explained that "close to 90% of the military's counter-drone capabilities are electronic warfare-type systems [that] use lasers or microwave-signal propagation to disrupt the communications link between user and device" (Suits 2020, p. 2). The current counter-drone capabilities, which are heavily reliant on electronic warfare, will no longer be sufficient to address an evolving threat that includes autonomous drones and COTS technology, as well as an increasing number of drones in the airspace, including the potential for UAS swarm, that can overwhelm a limited number of sensors and a C-sUAS operator (Suits 2020).

In addition to the evolving UAS capabilities and their proliferation, sUAS employment options have also expanded. At the benign end of the spectrum, small drones have been a nuisance by entering the wrong air space at the wrong time. While occasionally innocent mishaps, adversaries can afford to test the airspace boundaries with minimal operational risk and a marginal financial loss if a drone is destroyed or intercepted and captured. However, as seen most recently in Russia's 2022 War in Ukraine, adversaries are using drones for a variety of purposes: lasing targets or spotting for indirect and air-to-ground fires, collecting intelligence, and becoming a weapon themselves (Detsch 2022). The ways in which adversaries can disguise malevolent activities and perform serious operations using sUAS is only growing. Additionally, with the incorporation of UAS technology developments in artificial intelligence and autonomy, drone swarms have become even easier to coordinate and integrate into operations and provide an offensive advantage against mismatched air defense systems (Judson 2021).

Concurrent with the challenge of discriminating and countering the adversarial sUAS threat in a rapidly growing population of drones across the globe, air defense systems (and their human operators) cannot afford to lose any time or be imprecise in the process from detection to classification of a drone. Since sUAS are both small and fast, if one or more pose a direct threat, the timeline for mitigation of the threat is critical and will require an effective defeat or neutralization of the sUAS within 40 s from 1 km out (Henderson 2020). Using the example of a DJI Phantom with a speed of 16 m/s and assuming a closest approach distance for a micro UAS of 300 m, a C-sUAS system requiring 10 s to detect and initiate an accurate track and 20 s for assessment and neutralization would need a minimum detection range of $300m + (20 + 10)s * 16m/s = 780m$ (Poitevin et al. 2017).

Depending on the environment and the air defense systems, this process of detection and classification can involve different complicating factors, including line-of-sight obstructions, atmospheric conditions, and obfuscation with other flying objects like birds. These unwanted reflections that can be detected by a radar system are called clutter and can be affected by "precipitation, terrain, urban landscape, sea surface, ground moving targets, and birds" (Poitevin et al. 2017, p. 2). While the radar data we used in our research was mature track data (after filtering out environment-related clutter), Anduril Industries acknowledged that its radar systems can lose precious seconds during the detection process to ensure a mature track as a result of the aforementioned environmental conditions. Additional improvements in reducing environmental clutter to establish a mature track are beyond the scope of this thesis but present another area for improving the efficiency of discriminatory performance when considering the entire sequence from detection to mitigation.

2.5 C-sUAS Strategy

In June 2016, the *International Conference on Unmanned Aircraft Systems* (ICUAS) brought together a variety of "groups of qualified military and civilian representatives worldwide, organization representatives, funding agencies, industry, and academia [to] discuss the current state of UAS advances, and the roadmap to their full utilization in civilian and public domains" (IEEE Robotics and Automation Society 2016, p. 1). The conference included presentations of current and future research opportunities and the essential technologies that need to be utilized for further advancement in UAS (IEEE Robotics and Automation Society 2016). While an overwhelming number of articles and presentations from ICUAS 2016 focused on a fascination with the new opportunities for the UAS technology to become increasingly integrated into the fabric of society through commercial industry, a few analysts had already recognized and begun work on the critical aspects of detecting and tracking sUAS, such as Ganti and Kim (2016).

Immediately following ICUAS 2016, analysts started contemplating the ways in which UAS, although still largely in their infancy, were a threat to defense and intelligence operations by providing advanced surveillance capabilities to enable adversaries to collect data and information to shape military tactics and swarming techniques as a means for adversaries to distract, to disorient, and disrupt. Earlier in the same year, several of the U.S. armed services were already exploring technologies to address the growing threat, though most of the initiatives were in the early stages (Yasin et al. 2016). Although the technological solutions for this growing threat were not yet in full development, the importance of a strategic framework could already be conceived based on the nature of the specific aspects of the C-UAS challenge (Figure 2.7) concerning shortfalls in sensor technology that the DOD, DHS, and other government agencies were beginning to address.

C-UAS Challenges

- Unmanned systems are readily available at low cost and highly capable
- Smaller unmanned vehicles are hard or impossible to detect.
- Radar has to find small UAVs amid birds and other clutter.
- Low-altitude UAVs can hide behind buildings and trees to escape detection.
- Determining intent isn't always easy—is the approaching quad copter a threat, or just a toy?
- Mitigation and defeat carries a cost and risks collateral damage.

Figure 2.7. C-UAS Challenges. Adapted from Yasin et al. (2016, p. 3).

If the C-UAS gaps in U.S. Army and joint doctrine and capabilities were not already self-evident, the 25th Infantry Division's Warfighter Exercise 20-03 revealed that "current Army C-UAS capabilities and doctrine, especially that found in Army Techniques Publication (ATP) 3-01.81, *Counter-Unmanned Aircraft Systems Techniques*, [were] insufficient to meet the demands of the present and future battlefields" due to materiel and organizational limitations at the echelons (brigade and below) mostly responsible for targeting sUAS groups 1 and 2 (Scott 2021, p. 69). Scott (2021) also noted that the insufficient C-sUAS resources experienced during the exercise relegated the division commander to a heavy reliance on electronic warfare (EW) to find and target UAS ground stations (Figure 2.8).

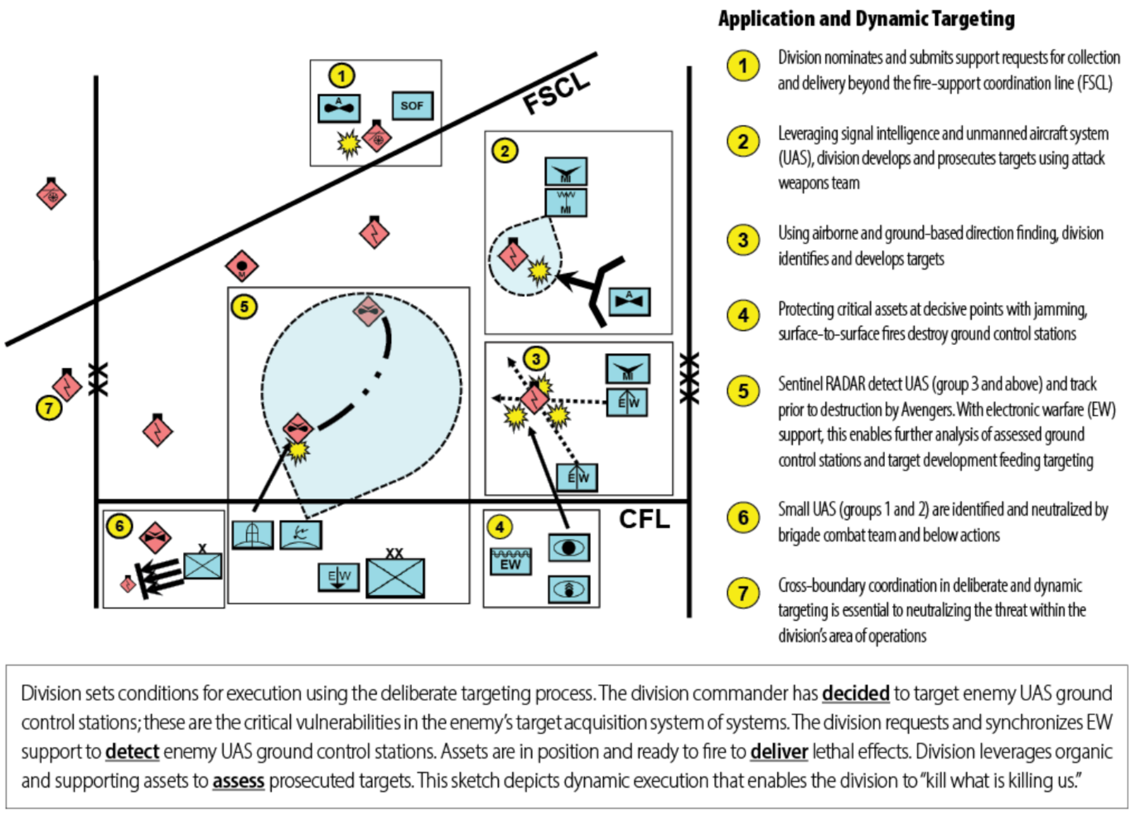


Figure 2.8. Visual Model of the 25th Infantry Division's C-UAS Targeting Efforts during Warfighter Exercise 20-03. Source: Scott (2021, p. 72)

Recognizing the sUAS threat posed by state and non-state actors alike, "in November 2019, the Secretary of Defense (SECDEF) designated the Secretary of the Army (SECARMY) as the DOD executive agent (EA) for C-sUAS" (Miller 2021, p. 3). Shortly thereafter, the SECARMY established the Joint C-sUAS Office (JCO) to lead, synchronize, and direct C-sUAS activities for unity of effort across the Department. About a year later, in December 2020, Acting SECDEF Christopher Miller signed the first U.S. DOD C-sUAS Strategy, consisting of three objectives: "enhanc[ing] the Joint Force through innovation and collaboration to protect DOD personnel, assets, and facilities at home and abroad, develop[ing] materiel and non-materiel solutions to facilitate the safe and secure execution of DOD missions while denying adversaries the ability to impede our objectives, and build[ing] and broaden[ing] our relationships with allies and partners to protect our interests at home and abroad" (Miller 2021, p. 3). To address some of the present C-sUAS gaps, the Army is already augmenting and integrating its existing Integrated Air and Missile Defense Battle Command System with its ongoing study of evolving C-sUAS capabilities to be institutionalized at the Joint C-sUAS academy within the Fires Center of Excellence at Fort Sill, OK by fiscal year 2024 (Suits 2020).

For UAS of a large enough size (typically group 3 and higher), existing air defense systems at the U.S. Army Division level have sufficient means to detect, classify and make a determination of whether the UAS is friendly or potentially foe with enough standoff to give a decision-maker a variety of options to degrade or diminish the threat (Scott 2021). However, in the case of sUAS, detection systems must be able to quickly identify, track and classify hundreds of small flying objects to distinguish birds from drones when they are first detected using passive radar sensors. To address this threat gap, the Defense Advanced Research Projects Agency (DARPA) mobile force protection (MFP) program is seeking to develop an "integrated system capable of defeating self-guided sUAS (i.e. those that do not rely on a radio or GPS receiver for their operation)" and has been pursuing promising sensing and neutralization technologies to complement existing MFP systems under development (GlobalSecurity 2021a, p. 1). Improvements in the DOD C-sUAS capability will clearly rely on technologies developed by commercial industries, evidenced by what GlobalSecurity (2021a, p. 2) describes as a C-UAS market estimated around "USD 1.8 Billion in 2020 and expected to grow to USD 5.47 Billion by 2028." In this market, ground based platforms, including fixed and vehicle mounted platforms, are expected to continue their dominance (GlobalSecurity 2021a).

In Section 2.6, we provide an overview of the key sUAS detection and mitigation technologies for executing steps 1 and 3 of the C-UAS process (Figure 2.6).

2.6 C-sUAS Detection and Mitigation Technologies

Since the 2016 ICUAS, a growing number of research efforts have been dedicated to detecting, tracking, identifying, and mitigating UAS. The predominant detection technologies can be broadly categorized into acoustic, vision (including EO and IR), passive RF or software defined radio (SDR), radar, and data fusion. Current mitigation technologies that are being developed include physical capture (i.e. containment netting), jamming (including RF communications, command, and control (C3) and global positioning system (GPS) jamming or spoofing), and destruction (including RF C3 intercept and control or attack using another UAS or projectile) (Wang et al. 2021). For the scope of this thesis, we assumed a high-threat sUAS, such as a kamikaze explosive or CBR attack threat, that would require positive classification as a drone for destruction (i.e., could not be physically captured or jammed). As such, we did not perform a wholesale review of current C-sUAS mitigation technologies and whether an alternative detection strategy might be preferred to the positive drone classification we aimed to achieve with our methodology. With this understood, we focused our review of detection technologies towards those systems that have achieved success in positively identifying sUAS with a high accuracy rate (minimal "false negatives") at the maximum range possible in all environmental conditions.

In Figure 2.9, we can clearly see that no single detection method is dominant across the key characteristics of range, position accuracy, and classification, without even considering the *robustness* of the detection method in all environmental conditions. Therefore, an optimal detection strategy must consider the trade-offs and relative importance of the key characteristics.

	Characteristics						
Detection Methods	Range	Position accuracy	Classification	Autonomous targets	Multiple targets	Low visibility conditions	Price
Human surveillance	**	***	*****	✓	✗	✗	****
Passive Electro-optical/infrared	***	****	****	✓	✗	✗	*
Acoustic	*	**	**	✓	✓	✓	***
Active Radar	****	****	***	✓	✓	✓	**

Figure 2.9. Comparison of key characteristics between C-UAS sensors. Source: Samaras et al. (2019, p. 2).

Human surveillance and EO/IR detection methods clearly outperform the top-performing radar and acoustic sensors in accurately classifying sUAS. However, human-in-the-loop surveillance systems, even in remote areas with minimal birds and drones, can easily become taxing on a handful of trained human operators. Other vision-discriminating systems, including EO/IR, also face limitations concerning weather conditions, line-of-sight, and their maximum effective ranges. In their review of real-time drone detection from EO sensors, Elsayed et al. (2021) explain that the use of IR thermal imaging and visual band imaging camera systems, from a cost-to-quality perspective, are extremely appealing, but each has unique limitations. They note that “IR performs poorly in complex backgrounds” or situations with no clear line-of-sight, such as in urban areas (Elsayed et al. 2021, p. 2). Low-cost commercial IR sensors also experience issues with moisture in some weather situations (Elsayed et al. 2021). Within the visible band, they conclude that no model or technique for EO sensors will work in all environments (Elsayed et al. 2021).

Meanwhile, radar sensors, despite sacrificing some classification accuracy, are uniquely capable of providing the long-range detection (from a few kilometers to tens of kilometers) in all light and weather conditions (Samaras et al. 2019). Their ability to detect and then track multiple aerial objects with sufficient stand-off is highly desirable for the high-threat environment in which our research is focused. Although the EO/IR detection ranges against sUAS groups 1 and 2 (Figure 2.1) continue to increase with the improvement of high-powered camera systems, the range advantage of radar systems (and additional decision space they offer) more than makes up for their comparatively less accurate classification performance.

Due to the trade-offs between range and classification accuracy, along with either self-imposed constraints or other environmental constraints specific to different types of sensors, an increasing amount of research has focused on data fusion from more than one sensor to take advantage of the unique benefits of the various types of sensors, in combination, to further optimize classification accuracy and reduce the “blind spots” that could be vulnerabilities with any one sensor type. Although research in sensor fusion for drone detection is relatively scarce, the available research points to this being a fertile area for achieving more accuracy and *robustness* in different environments, by contrast with any single sensor. In their review of sensor fusion results to date, Svanström et al. (2022) stress that while some results clearly demonstrate improvements in accuracy by compensating for weaknesses in individual sensors, other results indicate that more research is necessary to discover optimal combinations of heterogeneous data sources such as audio, visual, and RF surveillance to reduce information loss. Although our cursory review of the literature regarding sensor fusion for drone detection leads us to believe that an optimal detection strategy for achieving a sufficient classification accuracy at the greatest range in all or specific environmental conditions lies in some combination of different sensors, research in this area is beyond the scope of this thesis and discussed briefly in Section 5.2.1.

While acknowledging that an optimal sUAS detection system will likely involve the fusion of data from multiple different sensors, our research approached this problem set from the standpoint of protecting a remote and transient FOB or MSS that would not likely have the luxury of both a fixed passive EO/IR system and an active radar system acting in concert. In this scenario, a small number of operators would be reliant upon one or more mobile or vehicle-mounted radar systems that may be the last line of defense for detecting an adversarial sUAS. Although this austere and transient force posture would be limited to one or more radar systems, we assumed that the system could rely upon the necessary computing capacity *at the edge* to be able to take advantage of statistical and machine learning for sUAS discrimination.

In order to be effective classifying sUAS with RCS comparable to birds, radar-based drone detection research has generally gone in one of two directions: utilizing the micro-Doppler (m-D) signature or utilizing the kinematic data or other derived features from the Range Doppler (Samaras et al. 2019). In Section 2.7, we conclude this chapter by reviewing other prominent research that have looked into the improvement of classifying sUAS using the features of radar track data.

2.7 sUAS Radar Detection Research

In the growing field of sUAS detection and classification using radar data, the “radar m-D signature is the most commonly employed radar signal characteristic for automatic target classification” and has been used for a variety of sensing activities, “including ground moving targets, ship detection, human gait recognition, and other human activity” (Samaras et al. 2019, p. 7). The m-D signature has proven successful not only discriminating birds from drones but distinguishing different types of drones due to the ability to statistically describe and differentiate the intrinsic movements of the rotation of rotor blades on a rotary wing UAS or the flapping wings of a bird.

In 2016, the year of the ICUAS mentioned in Section 2.5, some promising results in this research area emerged. Mendis et al. (2016) claimed to be the first to apply deep learning techniques for radar signature extraction and recognition using the Doppler signatures and spectral correlation functions of three different micro UAS in a laboratory environment. Their work improved upon existing non-radar strategies using the distinct sound of UAS propellers and the radio signals from remotely controlled UAS, both of which were constrained by either the presence of wind and environmental noise or a fully autonomous UAS. From their laboratory research, they demonstrated the promise of machine learning by effectively detecting and classifying micro UAS with an accuracy above 90% when the signal-to-noise ratio (SNR) is >= 0 dB. In other analysis of time velocity diagrams of small helicopters and multicopters, Björklund (2018) performed feature extraction from the base velocity or body radial velocity, total bandwidth of the Doppler signal, offset of the total Doppler, etc. to attain greater than 90% classification accuracy, as well as model *robustness* across different target behaviors, ranges, and backgrounds. While the m-D signature for distinguishing drones from birds has been promising, much of the research has been at close range and done in artificial simulations due to the scarcity of radar sensors specialized for small target detection (Samaras et al. 2019). As a result, other research has turned its attention to extracting sources of information from the motion and RCS related features derived from surveillance radars.

There have been a variety of research exploring the most useful features for discriminating between the aerial tracks of drones and birds. While it is certainly possible for an adversarial controller of sUAS to mimic the flight kinematics of birds so as to reduce their likelihood of being classified as drones, doing so would also come at the cost of a more efficient or direct flight path towards an objective and increase the defensive response time for secondary (or combined) sensors (i.e., EO/IR) or employment of mitigation measures. Mimicking the flight kinematics of birds would also require additional investment by an adversary in collecting large amounts of pertinent bird track data to understand and replicate the flight patterns of its own sUAS. Given these considerations, it is likely that this field of detection research will continue to be fruitful in drawing insight from the anomalous flight kinematics of drones to improve discrimination performance.

Drone classification research using flight kinematics radar data begins with first optimizing the speed at which a multi-modal radar system can detect and begin processing the radar track data of an aerial object. Shin et al. (2016) used agent-based modeling to find the most efficient configuration of short and long-range radar sensors to provide reasonable detection rates at low cost. The arrangement and type of radar sensors can not only impact the range and speed at which the system can begin processing the aerial object's track data, but it can also affect the quality of some of the features of track data that we describe in Section 3.3. For this thesis however, we limited the scope of our problem set to improving the classification performance once the aerial object radar track has been validated or considered a mature track of an aerial object.

Some research in this area has focused specifically on the RCS while performing statistical analysis upon the flight telemetry of different types of multicopter and fixed-wing drones to improve detection (Sedivy and Nemec 2021). Others have analyzed the flight mechanics and behavior mode differences between drones and birds to derive a small number of features for a supervised learning random forest classification model, achieving a greater than 85% classification rate among three target types: drones, birds, and precipitation (Liu et al. 2021). In addressing the limitations of their research, Liu et al. (2021) make three specific recommendations for future work. These research recommendations include making improvements in the tracking accuracy of target motion characteristics, adding RCS to the target motion characteristics, and correlating aerial object track information with different surveillance environments to "distinguish between tracks with smooth and consistent motion patterns" (Liu et al. 2021, p. 9). Due to the importance of collecting a large quantity of labeled data for improving the machine learning methods at the heart of discriminating drones from birds, other research has focused specifically on the problem set of collecting accurate ground truth classification data (Sim et al. 2019).

This thesis carries forward some of the future work recommendations of Liu et al. (2021), includes RCS as a feature, as in Sedivy and Nemec (2021), and uses labeled radar track data from two uniquely different environments in order to explore potential improvements in classification accuracy while also introducing flight kinematic features with the intent of improving model *robustness* to the conditions affecting the flight of birds and drones in different environments. In Chapter 3, we review our methodology and model development to pursue these objectives.

CHAPTER 3: Methodology and Models

> COTS drones continue to evolve in complexity, threaten military and civilian safety, and disrupt operations. Anduril's end-to-end C-UAS system supports the entire kill chain in one, easy to use interface with precision, accuracy and reliability. Rogue drones are identified, tracked, and disabled in any environment, day or night. Anduril's goal is to provide human operators with a comprehensive picture enabling them to make critical decisions quickly. Sentry Towers connect to Lattice to detect and track rogue drones threatening perimeters of military bases, large public event venues or privately managed critical infrastructure. Lattice cuts through the noise and creates a shared real-time understanding of the battlespace. Lattice autonomously parses data from thousands of sensors and data sources into an intelligent common operating picture in a single pane of glass. Lattice uses technologies like sensor fusion, computer vision, edge computing, and machine learning and artificial intelligence to detect, track, and classify every object of interest in an operator's vicinity. (Anduril Industries 2022)

3.1 Methodology Strategy and Overview

The ultimate goal of our methodology is to improve upon a system's discriminatory performance using only radar data from the mature track of an unknown aerial object. Although our consultation with Anduril Industries for our data resulted in some exposure to its existing models and methodological approach that also informed a baseline understanding of its current best practices, we embarked upon an independent methodology that made no assumptions about what had already been tried and tested. Despite Anduril's relative satisfaction with the performance of its existing algorithms, the best performing algorithms relied heavily on a trial-and-error process and "black box" algorithms from extensive machine learning that seemed to lack a degree of mathematical or flight kinematics intuition. Additionally, while Anduril has incurred the costs associated with training its system in each new environment before achieving a satisfactory level of classification accuracy, it is unclear how much training is sufficient, how often the system might require retraining due to changes in the environment, and whether a system's models can be developed through a methodology that sustains their performance in different environments rather than requiring retraining in each new environment. We considered the degree to which a model performed as well in a different environment (the alternate training site) as in the environment (training site) in which we developed it, the model's *robustness*.

With this understanding of Anduril's limitations in the context of its own methodology, our approach sought to first corroborate the performance of its existing models using our independently developed models with its data. After validating the performance of our models on unseen bird and drone data from the same environment (training site) in which we developed our models, we then established a baseline performance of our top-performing models by validating them on unseen bird and drone data from a drastically different environment (the alternate training site). In the second iteration of our methodology, we then adjusted our sampling and added three additional features (acceleration components derived from the aerial track data) to assess whether our modifications could sustain or improve upon our baseline performance in the corresponding training environment while also increasing the model's robustness to a new environment (the alternate training site).

To do this, we employed a two-phase methodology (Figure 3.4) for training and testing a variety of models. In the first phase, we performed exploratory data analysis and used unsupervised learning methods to discover any hidden structure in the data. In the second phase, we built upon any insights about the structure of the data gained in the first phase to augment the development, training, and testing of our supervised learning models. Aside from the goal of improving the prediction accuracy of our models trained in a given environment, our desire for models that could sustain their prediction accuracy in a different environment led us think about and test other ways of distinguishing the flight kinematics of birds vs. drones regardless of the environment in which they are flying. By using a methodology that validated model performance in two distinctly different environments, we aimed to provide a more comprehensive and intuitive analysis that could lead to some useful insights for future model development rather than focusing exclusively on improving model performance in a given environment that may not translate well elsewhere.

In Section 3.2, we describe the overarching workflow, beginning with our data acquisition, in the context of a commonly used computational information design process. Section 3.3 provides an overview of the track data structure and a summary of the sample data we received from two distinct training environments. In Section 3.5, we conclude this chapter by describing the statistical learning approach and progression of steps we took to develop our models in the two-phase training and testing methodology.

3.2 Workflow

As a guide for ensuring a sound and relevant workflow, we derived our methodology from Dr. Benjamin Fry's "Computational Information Design" process (Fry 2004) that data scientists commonly reference in forming the building blocks of their research. In our adaptation of Fry's seven-step process (Figure 3.1), the first two steps, acquiring and parsing the data, involved several interactions with Anduril to provide requisite understanding of both the track data itself and some contextual understanding of the pre-processing and filtering of the radar sensor output that resulted in the mature (system validated) track data we received. For our analysis of radar track data from each testing site, Coastal Training Site (CTS) and Nevada National Security Site (NNSS), we parsed JavaScript Object Notation (JSON) and Newline Delimited JavaScript Object Notation (NDJSON) data from hundreds of files for sUAS (groups 1 and 2) and birds that the system's sensors detected and visually validated. For CTS, this included 233 files containing exclusively bird track data and 144 files containing exclusively drone (sUAS groups 1 and 2) track data. For NNSS, this included 79 files of bird track data and 112 files of drone track data. Figure 3.2 provides a summary of the track data received for the two training sites.

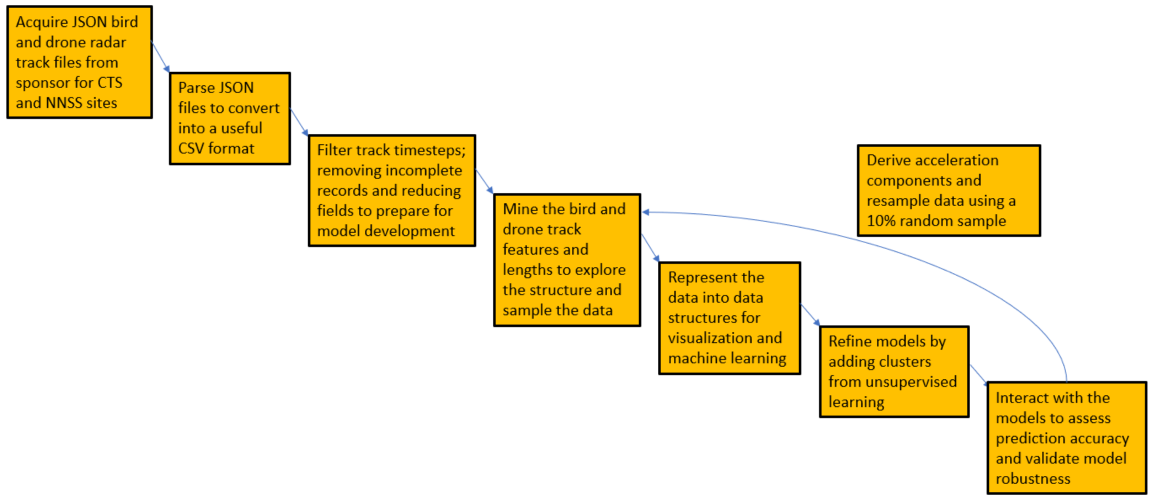


Figure 3.1. Our thesis workflow, as a computational information design process, consisted of two separate data acquisitions from our sponsor and a reiteration of model development after validating model performance and robustness. Adapted from Fry (2004).

In step three, we initially filtered the mature track data to extract eighteen numerical features, one categorical (binary) response variable, a sub-category (in the case of drones), and a prediction confidence (assessed by Anduril's proprietary algorithms) for each time step. In Section 3.3, we describe the pertinent quantitative features of each mature radar track. After an initial iteration of model development and testing using a sample taken every one hundred timesteps (1% sample size), we later performed another iteration of model development and testing using a random sample (10% sample size) of each bird and drone track. In the second iteration, we also derived three acceleration components (in the same directions as the velocity components) and added these numerical features to our model. In Section 3.4, we discuss our sampling method used in each iteration and assumptions made when training and testing our models.

After establishing and automating a method for steps one through three of the process, we spent the bulk of our time in model development and testing by looping through steps four, five, and six using our two-phase statistical learning approach presented in Section 3.5. Although the scope of this research lent itself to this methodology for systematically designing, training, testing, and redesigning our prototype models using exclusively radar data in steps four, five, and six, this workflow could also be adapted for data fused from a multi-sensor system (briefly discussed in Section 5.2.1) that could include vision (EO or IR), RF or SDR, and acoustic sensors.

In Section 3.3, we elaborate on our acquired radar track data, including the data composition and pre-processing performed in steps one through three prior to our model development.

3.3 Radar Track Data

To perform our methodology we acquired two batches of mature radar track data for aerial objects (birds or drones) that were detected and tracked by one or more sensors by Anduril's system. The data we received was considered mature because the detection system had already filtered out any noise and verified the presence of an aerial object. Although it is theoretically possible that the detected aerial object track could be something other than a bird or drone, our sponsor only provided radar track files for aerial objects that had been verified as a bird or drone using a secondary sensor (usually optical). In Section 5.2.3, we discuss ongoing research opportunities in this area where an analyst may have a combination of labeled and non-labeled radar track data. In our data pre-processing stage, we also ensured that the batches of files for both the CTS and NNSS locations were complete. In other words, we had all of the relevant features for the aerial object's track at every timestep and the binary categorical response variable. In our NNSS dataset, we discovered and removed timesteps and, in some cases, entire bird or drone tracks that were missing RCS values. Although other research, such as Medaiyese et al. (2021), has explored using a combination of labeled and non-labeled aerial track data by exploiting RF signals or other "fingerprints" of the aerial object, we had sufficient labeled data to be able to exclude aerial object tracks with an inconclusive categorical designation. In Section 5.2.3, we also briefly discuss the reality of collecting labeled track data in a combat environment and the potential for future research using semisupervised learning to build upon the research in this thesis.

Accounting for our sample data from Anduril, for the CTS location, we received more than 250,000 timesteps of bird track data and more than 150,000 timesteps of drone track data. The CTS data accounted for 230 unique bird tracks, ranging in length from 230 to 1,480 timesteps, and 144 unique drone tracks, ranging in length from 88 to 1,527 timesteps. For the NNSS location, we received more than 237,000 timesteps of bird track data and more than 386,000 timesteps of drone track data. The NNSS data included 79 unique bird tracks, ranging in length from 310 to 10,117 timesteps, and 112 unique drone tracks, ranging in length from 119 to 16,206 timesteps. However, for the NNSS location, after removing timesteps with incomplete data, we were left with approximately 206,000 and 295,000 timesteps respectively for bird and drone tracks. All of the recorded data had timesteps of 0.1 sec (10 Hz radar system), and a summary of the data can be seen in Figure 3.2 below.

		Pre-filtering		Post-filtering	
		Total Tracks	Total Timesteps	Total Tracks	Total Timesteps
CTS	Bird	230	251646	230	251646
	Drone	144	156517	144	156517
NNSS	Bird	79	237508	79	206090
	Drone	112	386775	94	295788

Figure 3.2. Summary of Bird and Drone Radar Track Data

We initially extracted eighteen fields and derived three fields (Table 3.1) relevant to our analysis. However, we relied exclusively on the three velocity components (East, North, and Up), three acceleration components (East, North, and Up), RCS, and altitude (meters above ground level) as the eight principal features and object class (bird or drone) as the binary categorical response variable for our model development. Although RCS is not the same as the area of the target, due to the intercepting surface of the object producing varying levels of reflected power back at the radar during flight, there is an area component of the measurement that is commonly measured in decibels relative to one square meter ($dBsm$). As a reference point in classifying aerial objects, we know that the average bird has a radar cross section of approximately -20 $dBsm$ (roughly 0.01 m^2) (GlobalSecurity 2021b). The six velocity covariance components account for noise in the radar's velocity calculations but were not used in this analysis. During the second iteration and sampling of our training data, we also derived the east, north, and up acceleration components by calculating the change in the respective velocity components from the prior timestep to the current timestep (roughly 0.1 sec). Lastly, although we did not have access to Anduril's proprietary algorithms for classifying the aerial objects, we did have access to their system's object class prediction, prediction confidence, and, in the case of drone predictions, the predicted drone group. This allowed us to compare Anduril's prediction accuracy with that of our preferred models.

After performing data pre-processing of the track data, we made one additional decision with respect to data sampling (Section 3.4) before going into model development.

Table 3.1. Radar Track Data. With the exception of the eastern, northern, and upward components of the acceleration (derived from the velocity components and timestamps between timesteps), we extracted the following bird and drone radar track data for analysis and statistical learning by our algorithms. The *objectClass* and *objectClassConfidence* variables refer to Anduril's predicted object class (bird, drone, or unknown) and the prediction confidence of its classifier respectively. The *objectDescriptor* variable refers to the actual drone group (1–2) of the drone (if known).

Data Name	Description	Units
timestamp	Timestamp of each timestep record in a track	$seconds^{-6}$
createTimestamp	Timestamp at the creation of a track	$seconds^{-6}$
lastMeasurementTimestamp	Timestamp at the end of a track	$seconds^{-6}$
enuVel_e	East(+)/West(-) Velocity Component	m/sec
enuVel_n	North(+)/South(-) Velocity Component	m/sec
enuVel_u	Up(+)/Down(-) Velocity Component	m/sec
enuVelCov_mxx	Velocity Covariance xx Component	N/A
enuVelCov_mxy	Velocity Covariance xy Component	N/A
enuVelCov_mxz	Velocity Covariance xz Component	N/A
enuVelCov_myy	Velocity Covariance yy Component	N/A
enuVelCov_myz	Velocity Covariance yz Component	N/A
enuVelCov_mzz	Velocity Covariance zz Component	N/A
enuAcc_e	East(+)/West(-) Acceleration Component	m/sec^2
enuAcc_n	North(+)/South(-) Acceleration Component	m/sec^2
enuAcc_u	Up(+)/Down(-) Acceleration Component	m/sec^2
altAgl	Altitude (Above Ground Level)	meters (m)
rcs	Radar Cross Section	decibels (dB)
objectClass	Predicted Object Class (Anduril Classifier)	N/A
objectClassConfidence	Prediction Confidence (Anduril Classifier)	N/A
objectDescriptor	Drone Group (1-2) *if known	N/A

3.4 Data Sampling

Due to our dataset being a set of time series tracks of birds and drones, we initially decided to sample 1% of the observations (once every 100 timesteps or about every 10 seconds) from all of our tracks in order to reduce correlation between observations in the development of our models. Although we could have chosen an even smaller sample size (i.e. every 500 or 1000 timesteps) to reduce correlation further, we balanced our decision with the consideration of not excluding some of our shorter bird and drone track lengths. The minimum bird and drone track lengths from the CTS dataset were 230 and 88 respectively. The minimum bird and drone track lengths from the NNSS dataset were 310 and 119 respectively. For the purposes of model development, we assumed that our sampling of the tracks every 100 timesteps reduced correlation sufficiently while also ensuring a representative sample of the bird and drone tracks we received from our sponsor.

Upon completing an initial iteration of training, testing and validating our models using a 1% sample (*Samp100*), we decided to contrast our results with a 10% random sample (*RandSamp*) from each of the bird and drone tracks in the second iteration of our methodology. By randomly selecting 10% of the timesteps from each bird and drone track, we acknowledge a degree of increased correlation in our dataset but wanted to investigate whether this larger sample dataset could allow our models to learn more from the feature space that also included the three derived acceleration components. Figure 3.3 shows the resulting sample sizes for both the initial iteration (*Samp100*) and second iteration (*RandSamp*) of our methodology.

		Samp100 (1% Sample; Every 100 Timesteps)		RandSamp 10% (10% Random Sample per Track)	
		Total Tracks	Total Timesteps	Total Tracks	Total Timesteps
CTS	Bird	230	2590	230	25160
	Drone	144	1495	144	15652
NNSS	Bird	79	2190	79	20610
	Drone	112	2830	112	29584

Figure 3.3. Summary of Bird and Drone Radar Track Samples Data

Having discussed our data sampling considerations and assumptions, Section 3.5 elaborates on our statistical learning approach for developing our models using those samples.

3.5 A Statistical Learning Approach

In this section, we present our thesis methodology (Figure 3.4) as a two-phase statistical learning approach performed separately using the bird and drone track data from each of the two training sites: CTS and NNSS. In the first phase (Section 3.5.1), we performed exploratory data analysis (EDA) and unsupervised learning methods to extract important variables, understand variable relationships, analyze outliers, and ultimately reveal any underlying structure in the data to provide insights and potentially gain marginal improvements in prediction accuracy during the second phase. In the second phase (Section 3.5.2), we trained, tested, and analyzed our supervised learning models both with and without an additional *cluster group* feature derived from the first phase. We concluded our two-phase model development by validating our models using the unseen track data from both of the CTS and NNSS training sites. We discuss our validation process and results in Chapter 4. Following the initial iteration (*Samp100*) to establish a baseline performance of our eight models (four models trained on each of the CTS and NNSS training site track data), we followed an abbreviated version (excluding some of the unsupervised learning methods) of the first phase of the two-step model development process in the second iteration (*RandSamp*) to contrast with the baseline performance of our eight models.

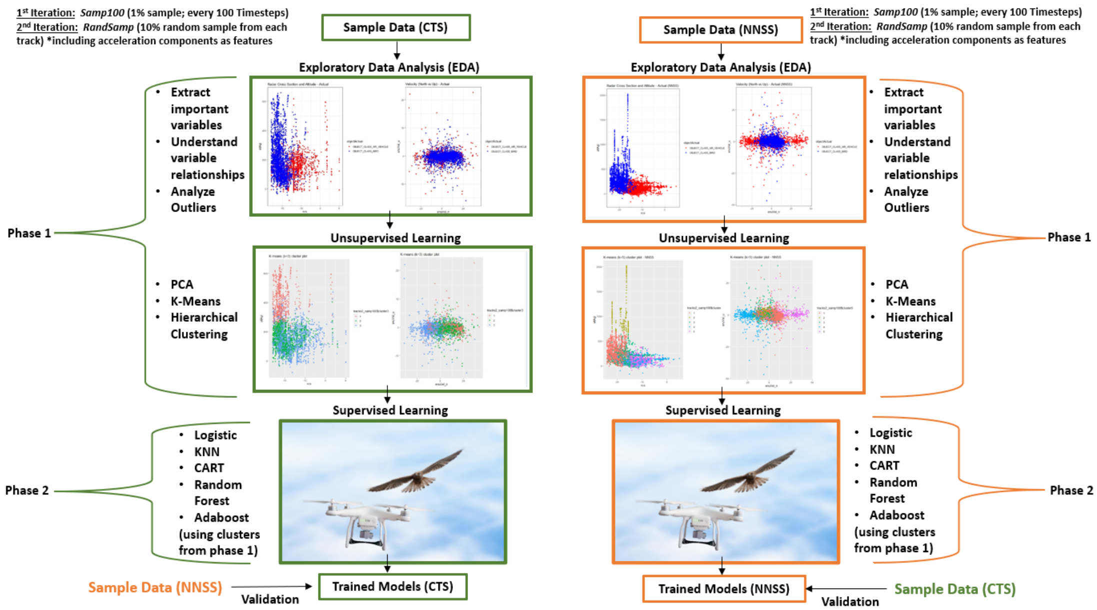
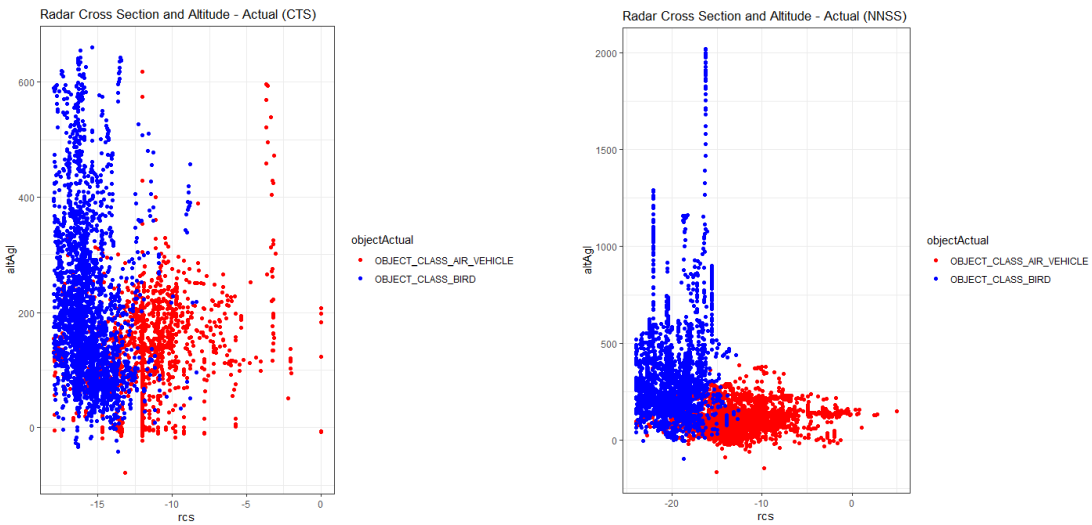


Figure 3.4. Our thesis methodology consisted of two iterations, each involving different data sampling techniques (Section 3.4) of a two-phase statistical learning approach (Section 3.5). We then analyzed and assessed each of our trained algorithms by comparing their respective performances from the two iterations by validating each algorithm's prediction accuracy using both the entire dataset from the training site on which the algorithm was trained and the entire dataset from the alternate training site.

In Section 3.5.1, we describe our insights from performing EDA on the track data from each of the two training sites and the unsupervised learning methods we used to reveal any inherent structure in the data and relationships between our features.

3.5.1 Model Development Phase 1 (Iteration 1 - *Samp100*)

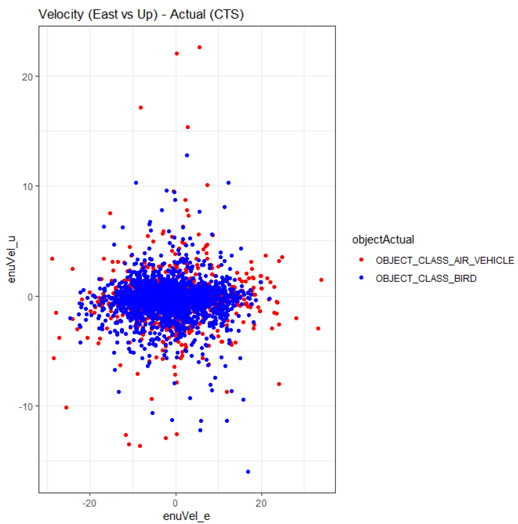
We began this methodology using our *Samp100* (1% sample described in Section 3.4) of the respective CTS and NNSS datasets. In our EDA, the plots of RCS vs. altitude for both the CTS and NNSS datasets (Figure 3.5) showed some natural separation in the data beyond which we might expect to find almost exclusively either drones or birds. At the CTS site (Figure 3.5a), aerial objects with a RCS of $-13dBsm$ or higher (less negative) are almost exclusively drones. At the NNSS site (Figure 3.5b), we can say the same for aerial objects with a RCS of $-15dBsm$ or higher (less negative). When examining altitude above ground level (AGL), we can see a similar phenomenon in which aerial objects above $300m$ AGL at both the CTS and NNSS sites are almost exclusively birds. Based on these plots alone, we can expect that our models will be able to exploit the separation in these features for predictive power with respect to birds and drones.



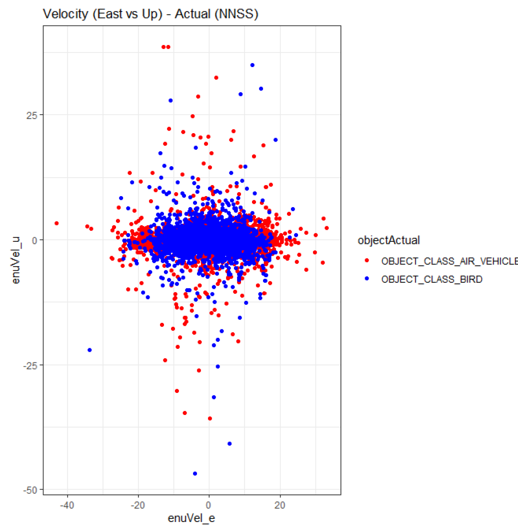
a. CTS b. NNSS

Figure 3.5. Two-dimensional plot of RCS (x-axis) vs. altitude (y-axis) for the *Samp100* (1st Iteration) of drones (red) and birds (blue) for CTS (left) and NNSS (right) data showing clear distinctions in RCS and altitude between birds and drones in both datasets.

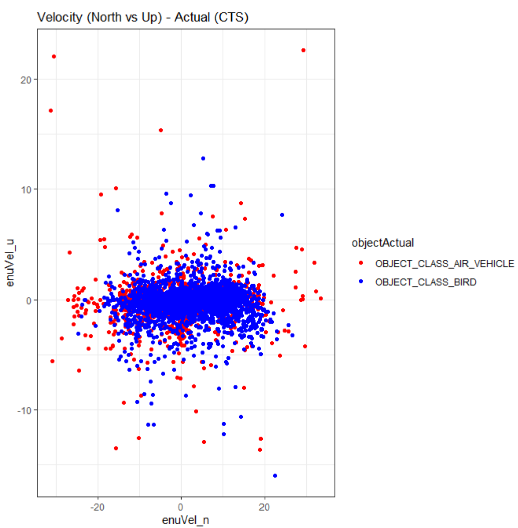
We performed similar two-dimensional plots of the eastern vs. upward velocity (Figures 3.6a and 3.6b) and northern vs. upward velocity (Figures 3.6c and 3.6d) components, and we found less clear separation in the data for discriminating birds and drones at the CTS site. However, at the NNSS site, we found that aerial objects traveling faster than $15m/sec$ upwards or downwards were almost exclusively drones. This distinction between the CTS and NNSS datasets is noteworthy because it could likely lead to overfitting by models trained and tested using the NNSS dataset during the alternate environment (CTS) validation.



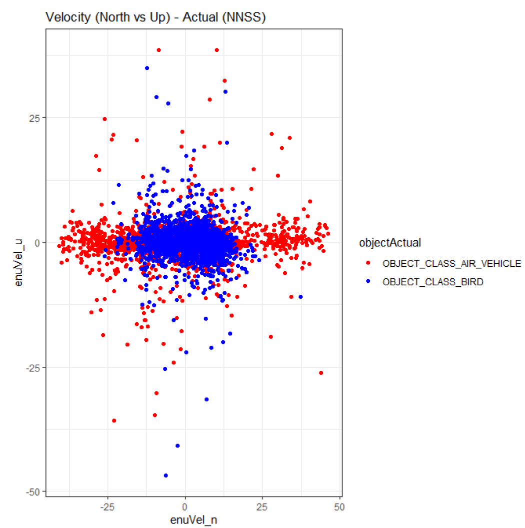
a. Velocity East vs. Up (CTS)



b. Velocity East vs. Up (NNSS)



c. Velocity North vs. Up (CTS)



d. Velocity North vs. Up (NNSS)

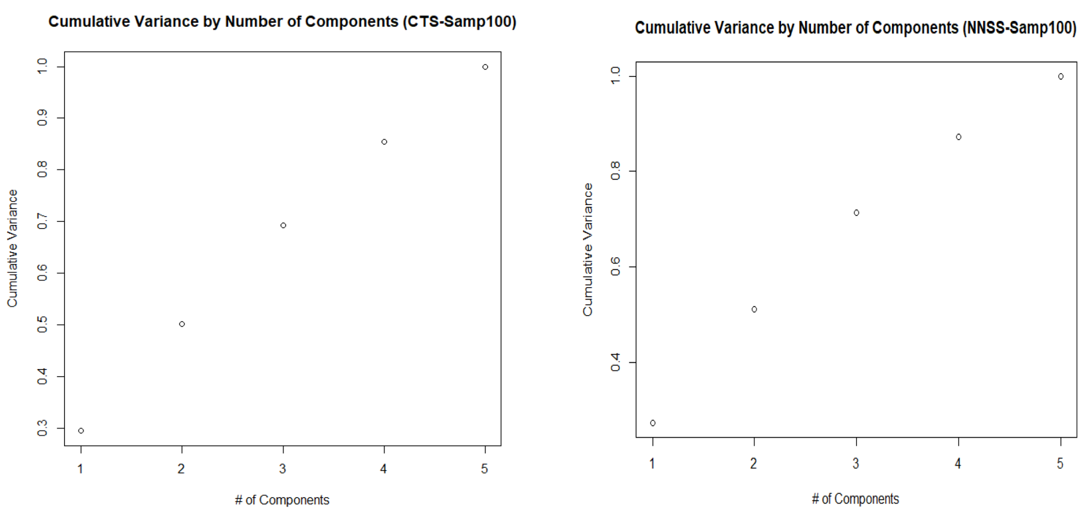
Figure 3.6. Two-dimensional plots of the eastern (x-axis) vs. upward (y-axis) and northern (x-axis) vs. upward (y-axis) components of the velocity for the *Samp100* (1st Iteration) of drones (red) and birds (blue) for CTS (left) and NNSS (right) data showing a clear separation between the two at speeds greater than $15m/sec$ in the upward or downward directions for NNSS dataset.

Principal Component Analysis (PCA) - *Samp100*

While the two-dimensional plots can help provide some superficial trends in the data, we next turned our attention to some unsupervised learning methods, beginning with principal component analysis (PCA), to garner some additional insights regarding our features.

Although analysts typically use PCA to reduce the dimensionality of their feature space to a handful of principal components that can explain a large percentage (usually above 85%) of the variance in a more simplified form (James et al. 2017), our analysis of the *Samp100* track data began with only five predominant features: RCS, altitude, and the northern, eastern, and upwards components of the velocity.

When we consider the number of components that can explain greater than 85% of the cumulative variance in the response variable of the CTS (Figure 3.7a) and NNSS (Figure 3.7b) *Samp100* datasets, we could have potentially reduced our feature space by one dimension (four dimensions instead of five) while being able to explain about 85% of the cumulative variance. However, we chose to keep our original feature space.



a. CTS - Samp100

b. NNSS - Samp100

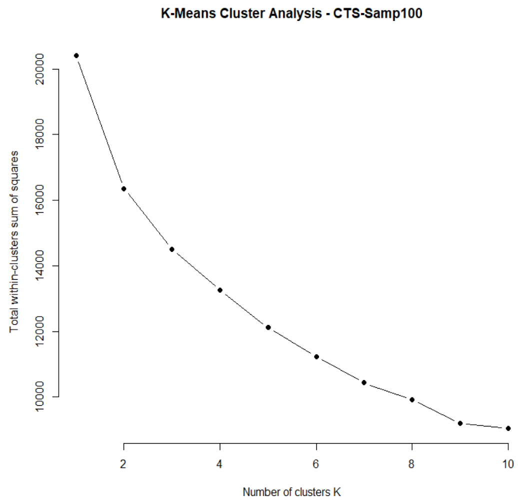
Figure 3.7. The graph of the cumulative variance proportion (y-axis) explained by number of principal components (x-axis) for the CTS (left) and NNSS (right) *Samp100* (first iteration) datasets did not provide a convincing case for reducing the number of principal components used in our models.

Clustering Analysis - *Samp100*

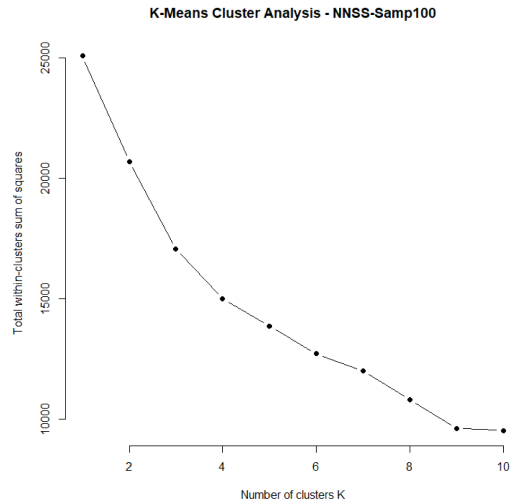
We then turned to clustering analysis to discover distinguishing groups of drones and birds in multi-dimensions beyond what we saw in the two-dimensional plots of RCS vs. altitude (Figure 3.5) and the upward velocity component vs. the northern and eastern velocity components (Figure 3.6). Our approach set forth to identify a handful of potentially promising insights in the first phase of our model development to use as a categorical input (i.e. by cluster or grouping) within some of the more promising supervised learning models for marginal improvements in prediction accuracy. This approach has proven successful in handling the classification of outliers, as in the credit card fraud detection research of Carcillo et al. (2021), and for efficiently handling new spatial-temporal data for human activity recognition in Budisteanu and Mocanu (2021).

To identify an optimal number of clusters using the sample data from each of our training sites, we began with a traditional approach using the k-means algorithm (with Euclidean distance) to establish a baseline assessment. We then proceeded to two other common clustering algorithms: a k-medoids algorithm, *pam* (partitioning around medoids), that is comparable to k-means but more resistant to outliers, and three different hierarchical clustering algorithms–*hclust*, *agnes* (agglomerative nesting), and *diana* (divisive analysis)–with two different linkage methods (*complete* and *Ward's*) using the *hcut* function featured in the *factoextra* package in R. We used the additional k-medoids and hierarchical clustering algorithms in order to corroborate our results using k-means by both observing the change in average silhouette width score and observing the number of significantly larger and distinct clusters that form while increasing the number of clusters k.

To analyze our k-means clustering performance, we began by looking at how increasing the number of clusters k affects the sum of square distances in clusters using an "elbow plot" to focus our search for an optimal number of clusters. In Figure 3.8, we see two promising *elbows* in the CTS plot (Figure 3.8a) at two and three clusters and two promising *elbows* in the NNSS plot (Figure 3.8b) at three and four clusters.



a. CTS - Samp100



b. NNSS - Samp100

Figure 3.8. Using an "elbow plot" for our k-means clustering analysis, we can see that either two or three clusters for the CTS *Samp100* (left) and either three or four clusters for the NNSS *Samp100* (right) may be promising clustering configurations.

To corroborate these visual results and aid us in selecting a preferred number of clusters, we can use a silhouette plot and look for the number of clusters that maximizes the average silhouette width across all of the clusters. For our CTS and NNSS *Samp100* datasets, we observed the average silhouette performance across number of clusters k (Figure 3.9) using *Pam*, *Hclust*, *Agnes*, and *Diana*.

Clustering Analysis - CTS Samp100 (Avg. Silhouette Widths)

# of Clusters	Kmeans	Pam	HC (Complete)	HC (Ward's)	Agnes (Complete)	Agnes (Ward's)	Diana
2	0.22	0.23	0.72	0.18	0.72	0.18	0.49
3	0.18	0.17	0.48	0.14	0.48	0.14	0.4
4	0.19	0.16	0.29	0.11			
5	0.18	0.17	0.25	0.12			
6	0.18	0.15	0.24	0.12			
7	0.18	0.15	0.22	0.12			
8	0.16	0.12	0.22	0.12			
9	0.16	0.12	0.22	0.12			
10	0.18	0.13	0.13	0.1			

Not calculated due to computation run time

a. CTS - Samp100

Clustering Analysis - NNSS Samp100 (Avg. Silhouette Widths)

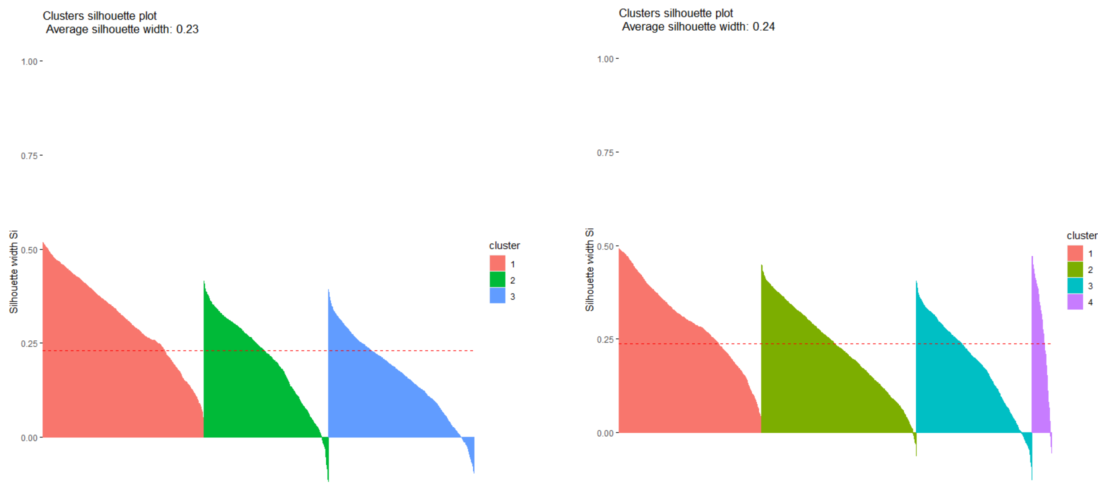
# of Clusters	Kmeans	Pam	HC (Complete)	HC (Ward's)	Agnes (Complete)	Agnes (Ward's)	Diana
2	0.2	0.2	0.59	0.17	0.59	0.17	0.47
3	0.23	0.23	0.59	0.19	0.59	0.19	0.44
4	0.24	0.19	0.33	0.2	0.33	0.2	0.39
5	0.22	0.2	0.34	0.21			
6	0.25	0.22	0.1	0.22			
7	0.23	0.2	0.1	0.21			
8	0.25	0.2	0.09	0.22			
9	0.24	0.21	0.06	0.2			
10	0.23	0.18	0.09	0.18			

Not calculated due to computation run time

b. NNSS - Samp100

Figure 3.9. Our clustering analysis of the average silhouette width for different numbers of clusters k for multiple algorithms led us to conclude that the optimal number of clusters for the CTS (left) and NNSS (right) *Samp100* datasets are two and three clusters respectively.

For the CTS *Samp100* dataset (Figure 3.9a), the highest average silhouette width across all clustering algorithms occurred with two clusters. For the NNSS *Samp100* dataset (Figure 3.9b), although the k-means, along with the *Hclust*, and *Agnes* algorithms using the *Ward's* linkage (minimizing within-cluster variance), resulted in marginally higher average silhouette widths above three clusters, we only see three prominent clusters and negative average silhouette widths among some of the clusters as we increase k beyond three clusters (Figure 3.10). Separately, the *Diana* algorithm, along with the *Hclust* and *Agnes* algorithms using *Complete* linkage (for more compact clusters), achieve their highest average silhouette widths at three clusters. In our assessment of average silhouette widths, only the *Diana* algorithm showed a slight preference for two clusters (rather than three clusters), while our *Hclust* and *Agnes* algorithms using *Complete* linkage scored the same for both two and three clusters. This tells us that our optimal cluster configuration would most likely be two or three clusters for the NNSS *Samp100* dataset.



a. NNSS - Samp100 (3 Clusters)

b. NNSS - Samp100 (4 Clusters)

Figure 3.10. Although the K-means silhouette plot of silhouette width (y axis) within each cluster (x axis) of the NNSS *Samp100* dataset with four clusters (right) achieves a slightly higher average silhouette width score than with three clusters (left), only three prominent clusters appear in both plots.

From our apparent identification of three distinct clusters in the NNSS *Samp100* dataset (as opposed to two prominent clusters in the CTS *Samp100* dataset), we hypothesize that our algorithms may be able to not only discriminate birds from drones, but potentially between different types of drones (i.e. rotary and fixed-wing) due to their unique flight patterns. Although we were unable confirm our hypothesis regarding different proportions of rotary wing and fixed-wing drones in the NNSS and CTS datasets, our observation of mostly rotary wing drones at the CTS site and understanding of the terrain differences—open desert terrain (NNSS) and coastal mountains (CTS)—leads us to believe that the difference in number of prominent clusters could reflect the proportional difference of UAS types at the two sites. In Section 5.2.4 we discuss the potential implications of such analysis.

After completing the first phase of our model development using the CTS and NNSS *Samp100* datasets, we next turned our attention to our supervised learning model development in phase two.

3.5.2 Model Development Phase 2 (Iteration 1 - *Samp100*)

After using unsupervised learning methods and clustering analysis in the first phase, phase two, of our first model development iteration, encompassed the training and testing of supervised learning models both with and without cluster groups (as categorical variables) derived from the first phase. Since our ultimate goal was the validation of our models on both the entire CTS and NNSS datasets, we sought to investigate and compare an assortment of models while maintaining interpretability and speed. We chose to investigate logistic regression, classification and regression tree (CART), k-nearest neighbors (KNN), random forest, and boosting (Adaboost) algorithms by using an 80/20 split between the training and testing sets of the sample data in order to derive our top performing CTS and NNSS models for validation in both their training environment and the alternate environment. In the first iteration, we performed this supervised modeling process (Figure 3.11) using five numerical features (excluding the acceleration components from Table 3.1) both with and without the categorical *cluster group* from the first (unsupervised) phase of our modeling.

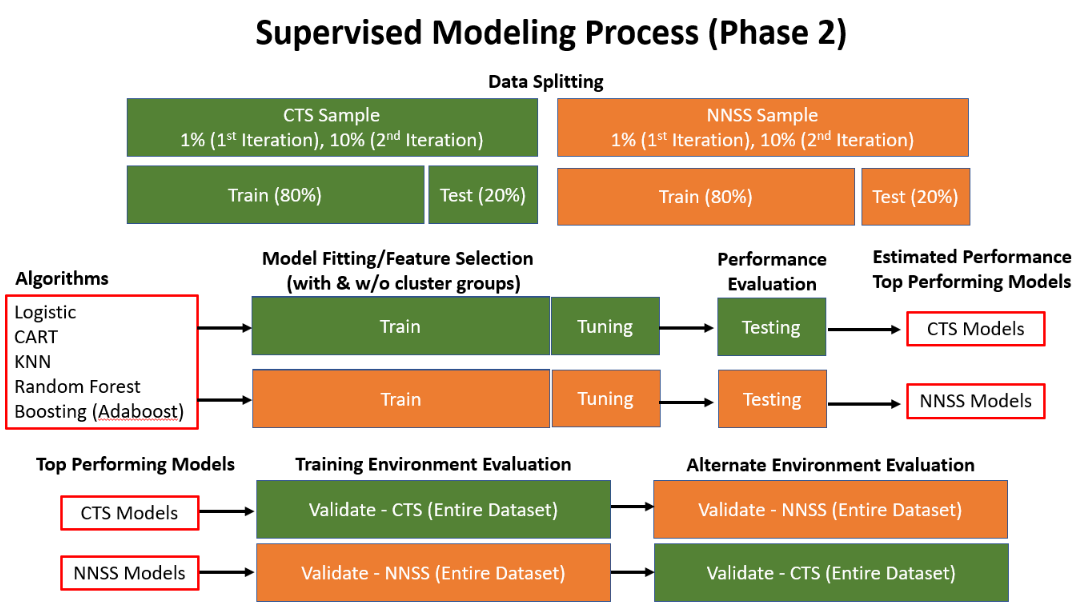
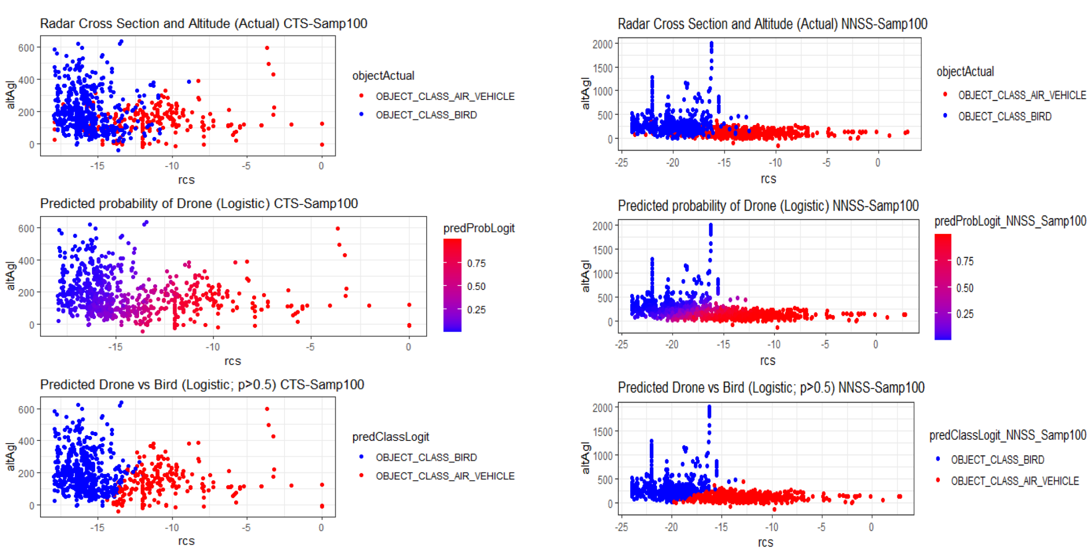


Figure 3.11. The supervised modeling process (phase 2) consisted of a competition among algorithms that were trained, tuned, and tested on the CTS (green) and NNSS (orange) samples to develop top performing models with and without the cluster groups (as categorical variables) from the first phase.

Logistic Regression - ***Samp100***

We began by fitting a logistic regression model after centering and scaling the five numeric predictors. For both the CTS and NNSS *Samp100* datasets, although the initial models only included RCS, altitude, and the northern velocity component as significant predictors, we added the upward velocity component as well after performing a stepwise model selection process, known as feature subsetting. Appendix A.1.1 shows a comparison of the logistic regression models. Using a fitted logistic regression model via stepwise model selection, we can see the strong influence of RCS and altitude and the nearly linear relationship in two-dimensional feature space that fairly accurately approximates what are birds (blue) and drones (red) (Figure 3.12). The comparable linear relationship between RCS and altitude, observed in the CTS (Figure 3.12a) and NNSS (Figure 3.12b) *Samp100* plots, indicates the potential *robustness* of the logistic regression models to different environmental conditions. However, we next turned to KNN for a simple instance-based machine learning algorithm that could potentially discover similarity among birds and drones in higher dimensions of our feature space without some of the rigidity of our logistic regression.

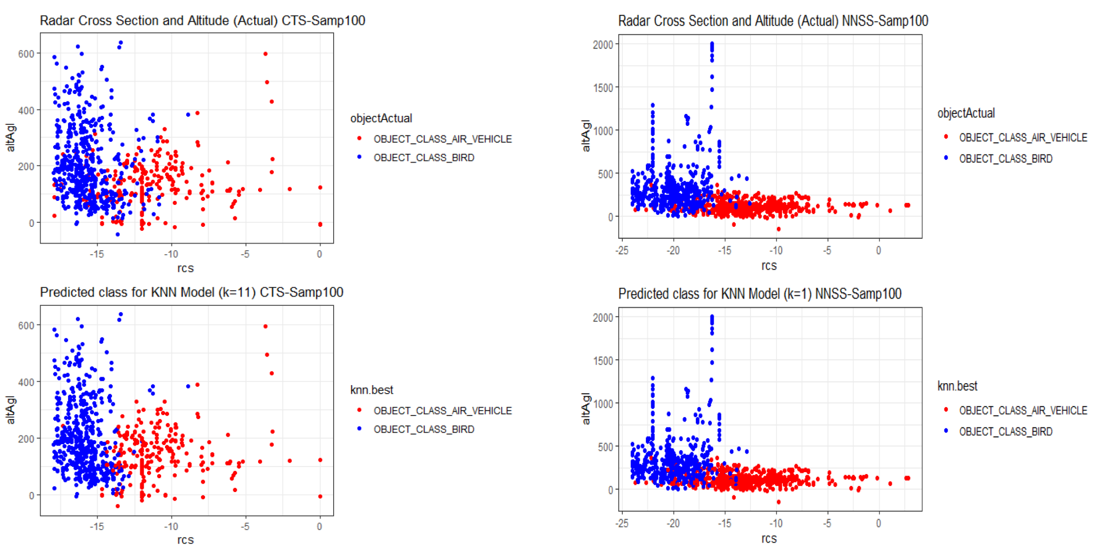


a. CTS - Samp100 (Logistic) b. NNSS - Samp100 (Logistic)

Figure 3.12. Although the fitted stepwise logistic regression model includes four predictors, we can see the strong influence of RCS and altitude within the models for both the CTS (left) and NNSS (right) *Samp100* datasets.

K-Nearest Neighbors (KNN) - *Samp100*

Similar to our logistic regression model development, we began by centering and scaling the five numeric predictors before training and tuning our models using different numbers of nearest neighbors k to find our best model. We applied both a 10-fold cross-validation and bootstrapping to thoroughly test our performance on the training set and discovered the best performance using $k = 5$ for both the NNSS and CTS KNN models. Appendix A.1.2 shows the comparison of our cross-validation and bootstrapping results. We can see the expected improvement of both the CTS (Figure 3.13a) and NNSS (Figure 3.13b) KNN models over their respective logistic regression models in their ability to understand the feature space using basic instance-based machine learning and indicated the potential promise of more sophisticated methods.



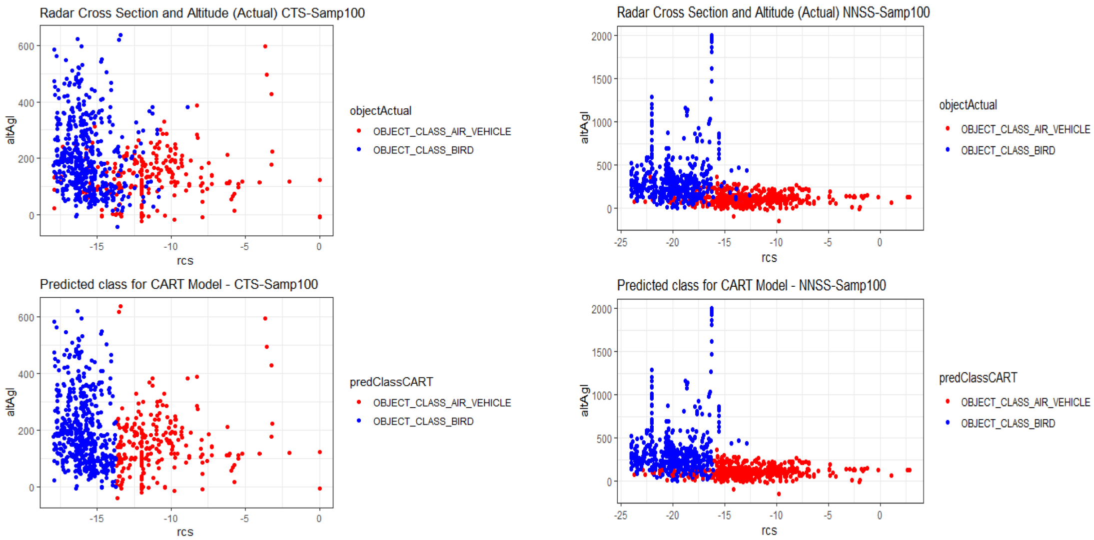
a. CTS - Samp100 (KNN) b. NNSS - Samp100 (KNN)

Figure 3.13. Our best performing KNN models for the CTS (left) and NNSS (right) *Samp100* datasets demonstrated an improved ability to understand the feature space using basic instance-based machine learning and the potential promise of more sophisticated methods.

After observing this improved performance, we then looked to compare the results with CART models that could allow us to interpret how our models were dissecting the feature space, including their improved handling of outliers and interactions between the predictors.

Classification and Regression Trees (CART) - *Samp100*

While our logistic regression models seemed to perform well using four out of the five predictors, their prediction performance exploited some natural separation between birds and drones mostly in terms of RCS and altitude alone to derive their models. Although the KNN models, as basic instance-based supervised learning algorithms, improved upon the performance of the logistic regression models, as an instance-based method, the model depends on its training set and would not generalize to other instances. In addition, since the KNN models treat all predictors equally, we cannot see which predictors are strongly correlated to the response variable. We then developed CART models to fine-tune our prediction capacity (over the logistic regression models) while also being able to interpret how our algorithm chose to categorize the feature space into a tree-based structure for improved classification of birds and drones. Although the best-performing CART models for the CTS and NNSS *Samp100* datasets (Figure 3.14) produced results comparable to their respective logistic regression models due to the weighted importance of both RCS and altitude in each of them, the resulting tree diagrams (Figures A.3 and A.4) and CART model summaries (Appendix A.1.3) offered some additional insights about the other three velocity component predictors and greater fidelity about the distinguishing features of birds and drones. Using a determination of RCS alone, our CTS and NNSS CART models would achieve a prediction accuracy near 80% and 78% respectively. For our NNSS CART model, also knowing the altitude would allow us to increase our prediction accuracy from 78% to nearly 91%, and also knowing the northern velocity component would increase our prediction accuracy above 93% (interpreting the respective CART model summaries in Appendix A.1.3). The achievement of this model performance using simple and interpretable CART models gave us confidence that the succeeding models developed using random forests and boosting, though less interpretable, would achieve superior performance.

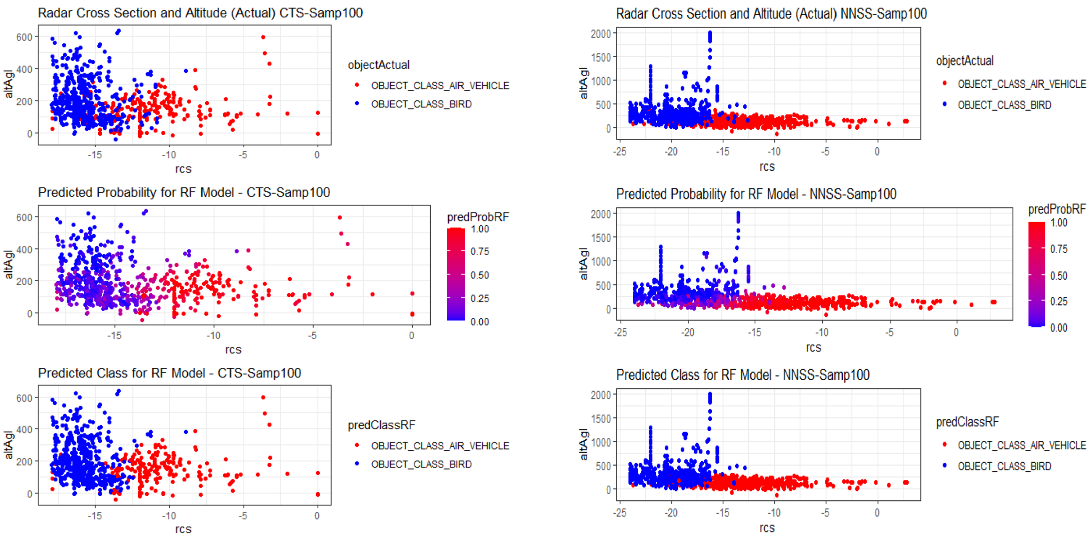


a. CTS - Samp100 (CART) b. NNSS - Samp100 (CART)

Figure 3.14. Our best performing CART models for the CTS (left) and NNSS (right) *Samp100* datasets performed comparably to their respective logistic regression models but provided insights in their respective tree diagrams (Figures A.3 and A.4) contrasting birds and drones at the two training sites.

Random Forests - *Samp100*

After gaining a better understanding of how birds could be discriminated from drones using the CART models developed using the *Samp100* datasets at each of the training sites, we then sought to improve the prediction performance further using random forest models with the five original features before adding a sixth cluster group feature from our best-performing unsupervised learning models during the first phase of our model development. As expected, our best-performing models derived from random forests using the original five features in our *Samp100* datasets (Figures 3.15 and 3.16) outperformed their respective logistic regression, KNN, and CART models.



a. CTS - Samp100 (Random Forest) b. NNSS - Samp100 (Random Forest)

Figure 3.15. Our best performing Random Forest models for the CTS (left) and NNSS (right) *Samp100* datasets improved upon the prediction accuracy of the logistic regression, KNN, and CART while losing some interpretability.

CTS - Samp100			
Accuracy: 0.9119		Actual Class	
		Drone	Bird
Predicted Class	Drone	252	25
	Bird	47	493

NNSS - Samp100			
Accuracy: 0.9751		Actual Class	
		Drone	Bird
Predicted Class	Drone	552	11
	Bird	14	427

a. CTS - Samp100 (Random Forest) b. NNSS - Samp100 (Random Forest)

Figure 3.16. Confusion Matrices for our best performing Random Forest models for the CTS (left) and NNSS (right) *Samp100* datasets show greater than 91% and 97% accuracy respectively.

Although we also tested our random forest models with an additional categorical cluster group feature, we observed no improvement by including them. We next looked at an adaptive boosting (AdaBoost) algorithm to gain any additional marginal improvements.

Boosting (AdaBoost) - *Samp100*

Boosting, a common technique for making minor improvements on models for binary classification problems, consists of improving the prediction power through the conversion of weaker learners to stronger ones. Adaptive Boosting (AdaBoost) uses decision trees of one-level to accomplish this. Using the *adabag* package and *boosting* method in R, we were able to accurately classify three additional aerial objects in both the CTS and NNSS *Samp100* datasets and achieve a marginal accuracy improvement (Figure 3.17).

CTS - Samp100			
Accuracy: 0.9155		Actual Class	
		Drone	Bird
Predicted Class	Drone	254	24
	Bird	45	494

a. CTS - Samp100 (AdaBoost)

NNSS - Samp100			
Accuracy: 0.9781		Actual Class	
		Drone	Bird
Predicted Class	Drone	555	11
	Bird	11	427

b. NNSS - Samp100 (AdaBoost)

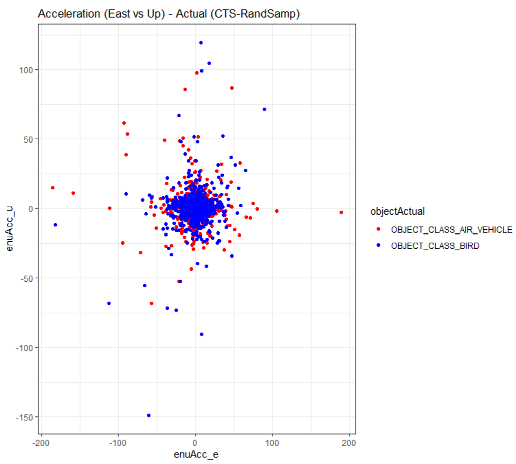
Figure 3.17. Confusion Matrices for our best performing Adaptive Boosting (AdaBoost) models for the CTS (left) and NNSS (right) *Samp100* datasets show a slight improvement over their respective random forest models.

Although our random forest and adaptive boosting models validated their performance (achieving comparable results as Anduril's classifier) on unseen data from their respective training environments, our initial evaluation of their performance in the alternate environment demonstrated a large degree of overfitting to their respective training environments. This initial *robustness* evaluation of the CTS and NNSS models generated using the *Samp100* datasets can be seen in Section 4.2.

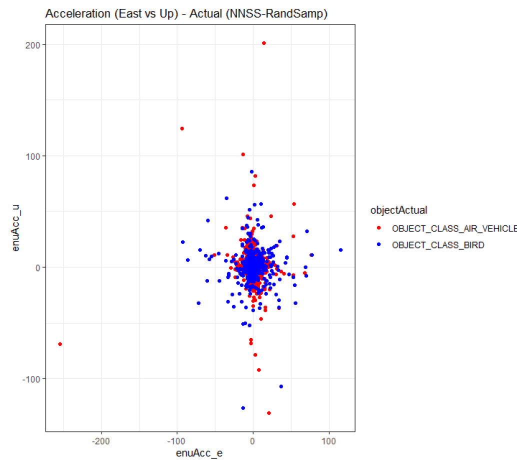
As a result of our dissatisfaction with the performance of our models in the alternate environment, we performed a second iteration of the methodology but modified our track sampling (Section 3.4) and added the derived acceleration components (northern, eastern, and upward) to our feature space with the goal of both reducing the overfitting we observed by seeking to discover any hidden components of the flight kinematics to more effectively differentiate birds from drones. In our second iteration, discussed in Sections 3.5.3 and 3.5.4, we revisited our methodology from the first iteration but with eight features (instead of five) in our *RandSamp* dataset. Due to the larger sample size and additional features (making the runtime of our unsupervised learning models cost prohibitive), we performed an abbreviated first phase in our second model development iteration and focused on contrasting our supervised learning models and results with those from the first iteration.

3.5.3 Model Development Phase 1 (Iteration 2 - *RandSamp*)

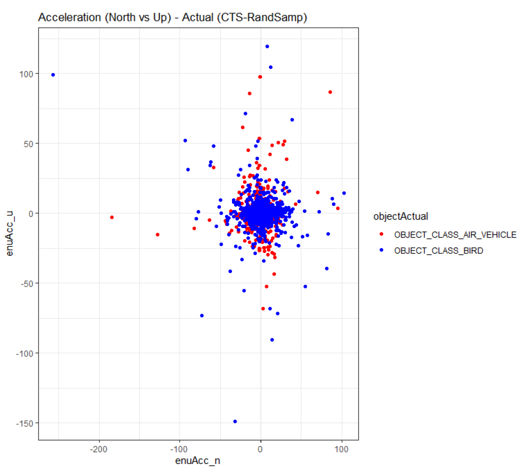
During the first phase of our *RandSamp* (Iteration 2) model development, we performed an abbreviated EDA by first looking for any observable separation between birds and drones among the added acceleration components. While we observed some minimal separation of drones around the periphery (higher acceleration values) of the CTS and NNSS acceleration component plots in Figure 3.18, we did not see the degree of separation between birds and drones that we had expected. We also observed some possible erroneous radar sensor track data (birds and drones with unnaturally high component acceleration values) that could not be resolved with our data sponsor after using our process of deriving the acceleration components. Since the data points (representing both birds and drones) with high acceleration values ($> 100m/sec^2$) can potentially be explained by sporadic wind gusts, we chose to not remove any data points from our models and generally acknowledge the impact that wind can have in obscuring a differentiation between birds and drones with respect to their velocity and acceleration components within the models. In Section 5.2.2, we elaborate on our recommendation to capture the real-time wind-speed and direction at the geographic location of the aerial object to improve the discriminatory capability of the models.



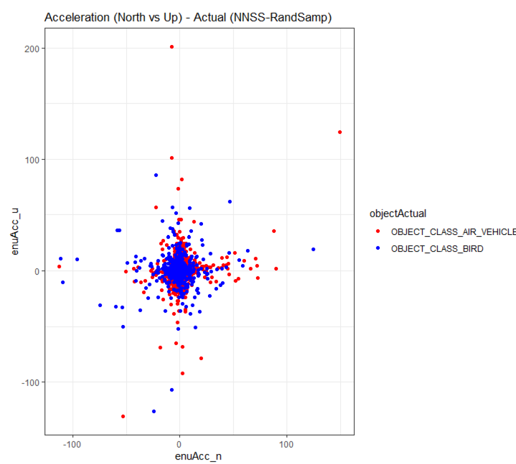
a. Acceleration East vs. Up (CTS)



b. Acceleration East vs. Up (NNSS)



c. Acceleration North vs. Up (CTS)



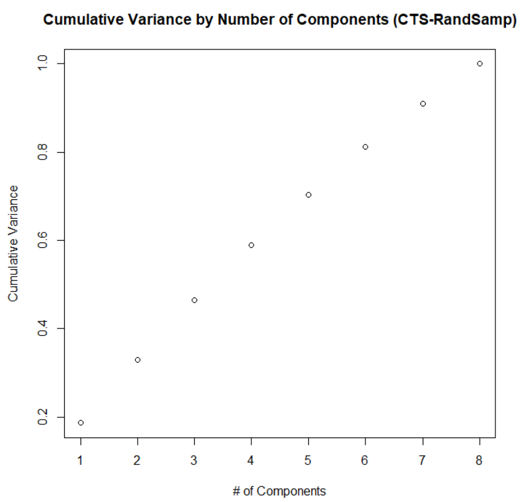
d. Acceleration North vs. Up (NNSS)

Figure 3.18. Two-dimensional plots of the eastern vs. upward and northern vs. upward acceleration components of drones (red) and birds (blue) for CTS (left) and NNSS (right) *RandSamp* datasets show minimal separation between birds and drones for either of the training sites.

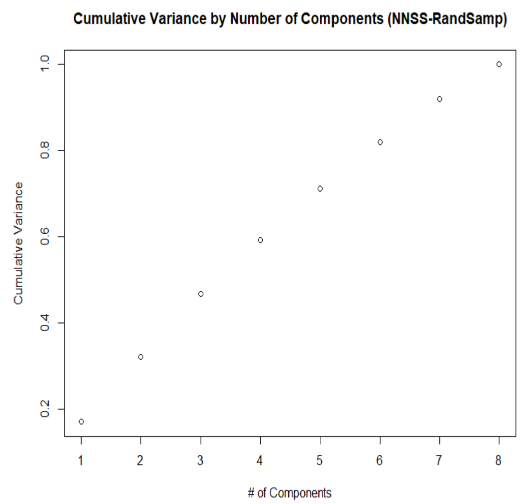
We then conducted PCA to determine whether our addition of the acceleration components had cluttered our feature space with any unnecessary noise and could be reduced in dimensionality.

PCA - *RandSamp*

Similar to our assessment of the cumulative variance explained by the principal components in the first iteration of our model development, we could have reduced the dimensionality by one (from eight to seven) due to the proportion of cumulative variance (0.9) explained by seven components in both the CTS (Figure 3.19a) and NNSS (Figure 3.19b) *RandSamp* datasets. However, we chose to keep the eight original features intact for our model development process in the second iteration.



a. CTS



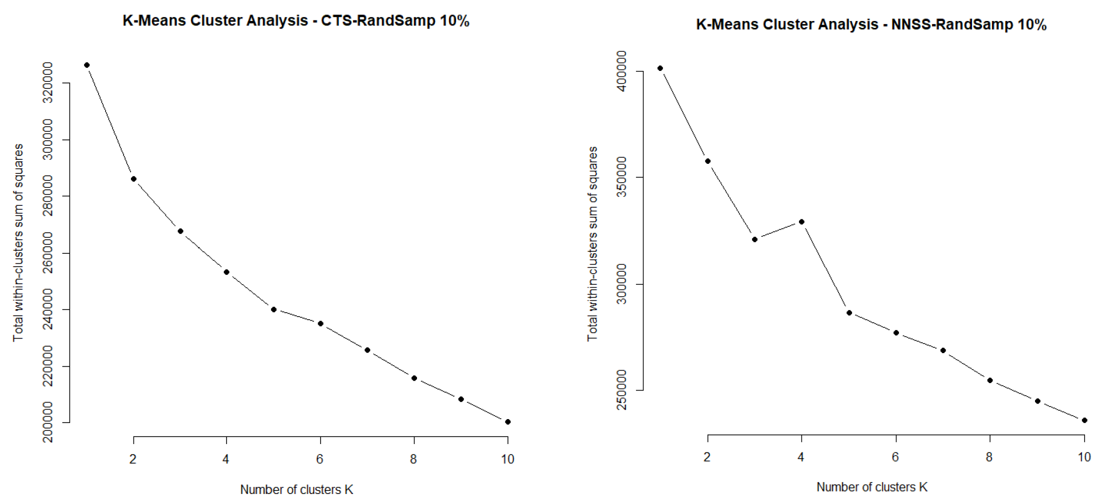
b. NNSS

Figure 3.19. We made the same determination not to reduce our number of principal components using the CTS (left) and NNSS (right) *RandSamp* datasets due to the lack of significant reduction in dimensionality (potentially from 8 to 7 dimensions) of the feature space.

Following our PCA, we revisited an abbreviated clustering analysis (using only k-means) to discover whether any new clustering may have emerged in our *RandSamp* datasets that were not evident in the *Samp100* datasets.

Clustering Analysis - *RandSamp*

As in the first iteration, we used our respective CTS and NNSS "elbow plots" of our total within cluster sum of squares to identify where our potentially optimal k-means number of clusters might occur (Figure 3.20). While our CTS *RandSamp* "elbow plot" (Figure 3.20a) seemed to indicate two clusters as optimal, our NNSS *RandSamp* plot (Figure 3.20b) provided a pronounced "elbow" at three clusters.



a. CTS - RandSamp

b. NNSS - RandSamp

Figure 3.20. Our k-means clustering "elbow plots" for our *RandSamp* datasets suggest a likelihood of two clusters for the CTS *RandSamp* dataset (left) and three clusters for the NNSS *RandSamp* dataset (right).

Due to the exponential increase in run-time for the unsupervised learning algorithms we used in the first iteration, we did not perform any additional clustering analysis beyond our observation of the k-means "elbow plots." However, if anything, the second iteration k-means clustering analysis provided an even clearer observed difference between the number of clusters between the CTS and NNSS training site datasets. After completing an abbreviated first phase of the second iteration, we then repeated the second phase of our model development to identify any noticeable differences in our model selection processes and performance of our CTS and NNSS models using the *RandSamp* datasets.

3.5.4 Model Development Phase 2 (Iteration 2 - *RandSamp*)

Logistic Regression - *RandSamp*

In the second iteration, despite our addition of the three acceleration components to the feature space, only the northern acceleration component (CTS) and upward acceleration component (NNSS) provide any additional predictive power to their respective logistic regression models. It is also noteworthy that all three velocity components (as opposed to only two of the velocity components using the *Samp100* datasets) are included in both the CTS and NNSS logistic regression models using the *RandSamp* datasets. Appendix A.2.1 provides additional comparative analysis of the resulting CTS and NNSS logistic regression step models using the *RandSamp* datasets. While the *RandSamp* models altered significantly from their respective *Samp100* models, their accuracy rates showed only marginal improvement (Figure 3.21).

Logistic Regression

CTS - Samp100			
Accuracy: 0.8739		Actual Class	
		Drone	Bird
Predicted Class	Drone	237	41
	Bird	62	477

CTS - RandSamp			
Accuracy: 0.8834		Actual Class	
		Drone	Bird
Predicted Class	Drone	2643	431
	Bird	521	4568

NNSS - Samp100			
Accuracy: 0.9223		Actual Class	
		Drone	Bird
Predicted Class	Drone	528	40
	Bird	38	398

NNSS - RandSamp			
Accuracy: 0.9226		Actual Class	
		Drone	Bird
Predicted Class	Drone	5574	420
	Bird	357	3688

Figure 3.21. The optimal CTS and NNSS logistic regression models developed using the *RandSamp* datasets both improved their prediction accuracy in comparison to their respective models developed using the *Samp100* datasets.

After observing the logistic regression models' marginal improvements in prediction accuracy with the inclusion of all three velocity components and a portion of the acceleration components, we also expected an improvement in the respective KNN models by using the *RandSamp* datasets.

KNN - *RandSamp*

While the CTS and NNSS KNN model training, using both bootstrapping and 10-fold cross validation methods, resulted in an optimal selection of $k = 5$ using the *Samp100* dataset, our application of the same methods using the *RandSamp* dataset (second iteration) resulted in an optimal selection of $k = 3$ for both the CTS and NNSS KNN models and an improved prediction accuracy. This decrease in the optimal k value (from $k = 5$ to $k = 3$) suggests that the KNN model using both the CTS and NNSS *RandSamp* datasets were able to capture the finer structure of their respective feature spaces better than their respective KNN models using the *Samp100* dataset. We also observed a large improvement in the prediction accuracy of the *RandSamp* KNN models (Figure 3.22).

KNN

CTS - Samp100			
Accuracy: 0.8923		Actual Class	
		Drone	Bird
Predicted Class	Drone	250	39
	Bird	49	479

CTS - RandSamp			
Accuracy: 0.9661		Actual Class	
		Drone	Bird
Predicted Class	Drone	3028	141
	Bird	136	4858

NNSS - Samp100			
Accuracy: 0.9552		Actual Class	
		Drone	Bird
Predicted Class	Drone	549	28
	Bird	17	410

NNSS - RandSamp			
Accuracy: 0.9785		Actual Class	
		Drone	Bird
Predicted Class	Drone	5902	120
	Bird	96	3921

Figure 3.22. The optimal CTS and NNSS KNN models developed using the *RandSamp* datasets both improved in prediction accuracy compared to their respective models developed using the *Samp100* datasets.

After observing the improved performance of both the logistic regression and KNN models developed using the CTS and NNSS *RandSamp* datasets, we next looked to their respective CART models to interpret any changes to the prominent features in their respective models by observing the tree diagram and model summary.

CART - *RandSamp*

In our second iteration of CART model development, we observed an improved prediction performance in comparison with the respective *Samp100* CART models. Although we observed both RCS and altitude as prominent features in both the *Samp100* and *RandSamp* CTS and NNSS CART models, if anything, those two features increased in importance in the respective CTS and NNSS CART models using the *RandSamp* datasets as seen in the CTS and NNSS model summaries (Appendix A.2.3). As with the logistic regression and KNN *RandSamp* models, our respective CART *RandSamp* models also improved their prediction performances (Figure 3.23).

CART

CTS - Samp100			
Accuracy: 0.8678		Actual Class	
		Drone	Bird
Predicted Class	Drone	243	52
	Bird	56	466

CTS - RandSamp			
Accuracy: 0.8879		Actual Class	
		Drone	Bird
Predicted Class	Drone	2759	510
	Bird	405	4489

NNSS - Samp100			
Accuracy: 0.9313		Actual Class	
		Drone	Bird
Predicted Class	Drone	525	28
	Bird	41	410

NNSS - RandSamp			
Accuracy: 0.9321		Actual Class	
		Drone	Bird
Predicted Class	Drone	5606	290
	Bird	392	3751

Figure 3.23. The optimal CTS and NNSS CART models developed using the *RandSamp* datasets both improved their prediction accuracy in comparison to their respective models developed using the *Samp100* datasets.

After observing the improvements in both model interpretability and performance for the CTS and NNSS CART models trained using the *RandSamp* datasets, we expected to observe the same for our random forest and boosting algorithms.

Random Forests - *RandSamp*

As in the first iteration of random forests model development, we trained and tested CTS and NNSS random forest models using our eight features before adding a ninth cluster group categorical feature from our best-performing unsupervised learning models during the first phase of the second iteration. During the second iteration, we not only observed improved performance in both the CTS and NNSS base random forests models (Figure 3.24), but we were also able to achieve an additional improvement using the additional cluster group feature using two, three, and four cluster k-means models. Appendix A.2.4 discusses the improved model performance with cluster groups, including our highest observed accuracy rate (99.2%) for the NNSS random forest model with four cluster groups.

Random Forest

CTS - Samp100			
Accuracy: 0.9119		Actual Class	
		Drone	Bird
Predicted Class	Drone	252	25
	Bird	47	493

CTS - RandSamp			
Accuracy: 0.9748		Actual Class	
		Drone	Bird
Predicted Class	Drone	3023	65
	Bird	141	4934

NNSS - Samp100			
Accuracy: 0.9751		Actual Class	
		Drone	Bird
Predicted Class	Drone	552	11
	Bird	14	427

NNSS - RandSamp			
Accuracy: 0.9895		Actual Class	
		Drone	Bird
Predicted Class	Drone	5889	63
	Bird	42	4045

Figure 3.24. The optimal CTS and NNSS Random Forest models developed using the *RandSamp* datasets both improved in prediction accuracy by comparison with their respective models developed using *Samp100* datasets.

After observing improved performance in our CTS and NNSS random forests models (both with and without cluster groups) trained using the *RandSamp* datasets, we then concluded our model development by revisiting our adaptive boosting algorithm.

Boosting (Adaboost) - *RandSamp*

To complete our model comparison, we trained and tested our adaptive boosting algorithm using the *RandSamp* datasets and improved upon both of our respective *Samp100* CTS and NNSS AdaBoost models (Figure 3.25). Although our *RandSamp* CTS and NNSS AdaBoost models both improved upon their respective *Samp100* models, their respective random forest models (Figure 3.24) achieved the highest prediction performances of any of our models during our testing.

AdaBoost

CTS - Samp100			
Accuracy: 0.9155		Actual Class	
		Drone	Bird
Predicted Class	Drone	254	24
	Bird	45	494

CTS - RandSamp			
Accuracy: 0.9331		Actual Class	
		Drone	Bird
Predicted Class	Drone	2795	177
	Bird	369	4822

NNSS - Samp100			
Accuracy: 0.9781		Actual Class	
		Drone	Bird
Predicted Class	Drone	555	11
	Bird	11	427

NNSS - RandSamp			
Accuracy: 0.9855		Actual Class	
		Drone	Bird
Predicted Class	Drone	5870	85
	Bird	61	4023

Figure 3.25. The optimal CTS and NNSS Adaptive Boosting (AdaBoost) models developed using the *RandSamp* datasets both improved in prediction accuracy by comparison with their respective models developed using the *Samp100* datasets.

After completing a second iteration of our two-phase model development, we brought forward sixteen total models for validation: four types of supervised models (logistic regression, CART, random forests, and AdaBoost), each trained and tested using two different datasets (*Samp100* and *RandSamp*) sampled from each of the two different training site environments (CTS and NNSS). In chapter 4, we contrast the performance of our respective models by validating their prediction accuracy both on unseen bird and drone tracks from their respective training environments and the alternate environment.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 4: Application, Results, and Analysis

Towards our goal of improving the prediction performance of our top models, we established a baseline prediction accuracy for four models (logistic regression, CART, random forests, and AdaBoost) at two unique training site environments (CTS and NNSS) using a sample from the bird and drone tracks every one hundred timesteps (*Samp100*). We originally performed unsupervised learning during the first phase of model development with the intent of either reducing dimensionality or identifying cluster groups that could be used to improve the performance of the supervised learning models in the second phase. However, we chose to keep the original set of features intact, rather than potentially reducing the dimensions by one. We also excluded the k-means *cluster group* feature in our baseline random forest models because we did not observe any improvement to our models when including the additional feature during our model testing. Despite not incorporating the unsupervised learning results from the first phase of our model development into our supervised learning models, our EDA aided our intuition regarding the supervised learning models, and our cluster analysis of the differences between the CTS and NNSS track data allowed us to speculate about the ability of our algorithms to not only discriminate between birds and drones but between different types of drones. We discuss the latter in Section 5.2.4 within the context of potential future work with drone discrimination.

Our top baseline prediction accuracy (models trained using the CTS and NNSS *Samp100* datasets) was 91.6% and 97.8% for the CTS and NNSS datasets respectively. Although our top-performing models performed as well or better than Anduril's classifier (as provided to us by the data sponsor) in discriminating birds from drones within the training environment, we were dissatisfied with the *robustness* of our models during the validation results of our CTS and NNSS models against the track data from the alternate training site. The drop in prediction accuracy indicated model overfitting to the track data from the respective training environments.

As a result of our findings from this initial iteration of model development and testing, we conducted a second iteration of model development and testing by modifying our training data sampling in two ways. We added three additional features, the three acceleration components (derived from the change in our velocity components per timestep), and randomly sampled 10% (as opposed to 1%) of the data from each of the bird and drone tracks. In Section 4.1 we evaluate the validation of our CTS and NNSS models developed using the (*RandSamp*) datasets to improve upon the performance of our respective models developed using the *Samp100* datasets. In Section 4.2, we compare the validation of those same models against the bird and drone track data from the alternate environment from which we trained them to assess any improved performance in the unseen environment.

4.1 Training Environment Validation

For our validation of model performance in the training environment, we ran our sixteen models—four types of supervised models (logistic regression, CART, random forests, and AdaBoost), using two different data sampling methods (*Samp100* and *RandSamp*) on the track data from each of the two training sites (CTS and NNSS)—on the complete dataset from the training environment on which the models were trained. Although we already had an indication of how our models would perform based on their testing during the second phase of both iterations of the model development, we evaluated a comparison of the prediction accuracy of the models and accounted for each model's receiver operating characteristic (ROC) curve by comparing the model area under the curve (AUC) scores. This would allow us to compare the models with one another by plotting each model's *sensitivity* (probability of correctly interpreting drones as drones) against its *specificity* (probability of correctly interpreting birds as birds). Although we still compared the models with one another (and the classifier that our data sponsor provided us), analysts also typically want models to achieve AUC scores above 0.8. While the accuracy rates and AUC scores against the training environment dataset were important to our analysis, our ultimate goal was improving upon our validation results against the alternate environment dataset using the respective models developed using the *RandSamp* dataset without sacrificing our prediction accuracy (or AUC scores) in the training environment. In Appendix A.3, we discuss the training environment performance using balanced prediction accuracy to compare the four model types for each of the training CTS and NNSS models. Although the comparison of balanced prediction accuracy gave us a strong sense of how the models compared with one another, we also compared the model ROC curves and AUC to visually and quantitatively evaluate the respective models. Figure 4.1 shows an AUC score comparison for each of the four model types using the *Samp100* and *RandSamp* datasets with the highlighted scores indicating the better performing model. Overall, we observed similar AUC scores in all eight of the model comparisons, with three out of the four CTS models and all four NNSS models using the *RandSamp* dataset improving their AUC scores. With the exception of the CTS logistic regression model, the improvements in AUC scores using the *RandSamp* dataset agreed with the balanced accuracy comparisons (Figure A.10). Although we ultimately sought improved performance in the alternate environment, in this effort, we had not sacrificed performance in the training environment.

	Training Environment Validation (AUC Scores)			
	CTS - Samp100	CTS - RandSamp	NNSS - Samp100	NNSS - RandSamp
Logistic	0.9338601	0.9341702	0.9798821	0.9799623
CART	0.8838985	0.8850222	0.9536846	0.9543016
Random Forest	0.9866965	0.9912645	0.9979097	0.9995565
AdaBoost	0.9850741	0.9826550	0.9983017	0.9990955

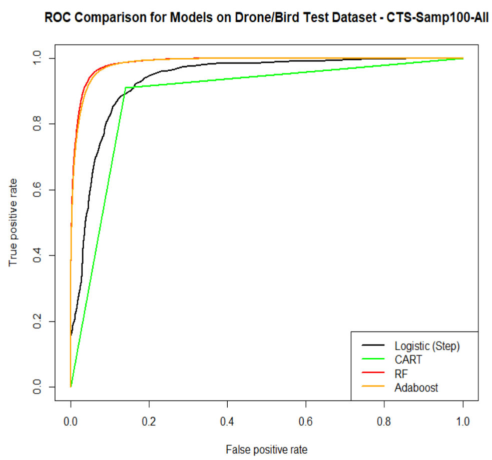
Figure 4.1. Using AUC scores to compare the performance of CTS and NNSS models built using the *RandSamp* datasets with their respective models built using the *Samp100* datasets, we observed improvement (highlighted in yellow) in three out of the four CTS and all four NNSS model types built using the *RandSamp* datasets.

After performing a quantitative evaluation of all four model types for our CTS and NNSS training environments using AUC scores, we also wanted to evaluate the visual representation of all four model types for each of the two training environments by comparing the respective ROC curves with one another.

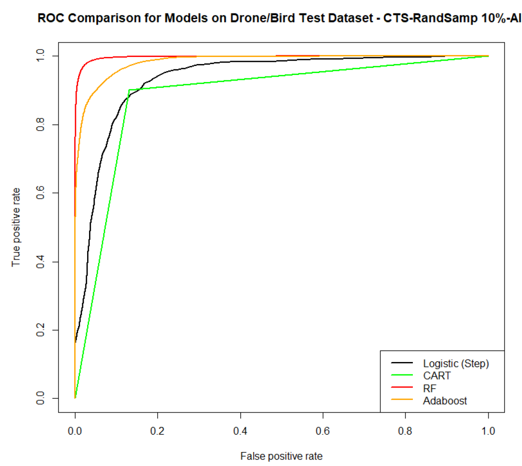
4.1.1 CTS Models - Training Environment Validation

In contrasting the CTS ROC curves from our training environment validation (Figure 4.2) of the models built using the *Samp100* (Figure 4.2a) and *RandSamp* (Figure 4.2b) datasets, we observed a marginal decrease in performance of the AdaBoost (orange) model and marginal increase in performance of the random forest (red) model. Meanwhile, we observed minimal changes to our logistics regression (black) and CART (green) models developed using the

RandSamp dataset. However, as we would see in our alternate environment validation using AUC scores (Figure 4.4), the CTS AdaBoost model using the *RandSamp* dataset improved in performance while the respective random forest model decreased in performance, exhibiting a trade-off between performance in the two environments.



a. CTS - Samp100 - Validation

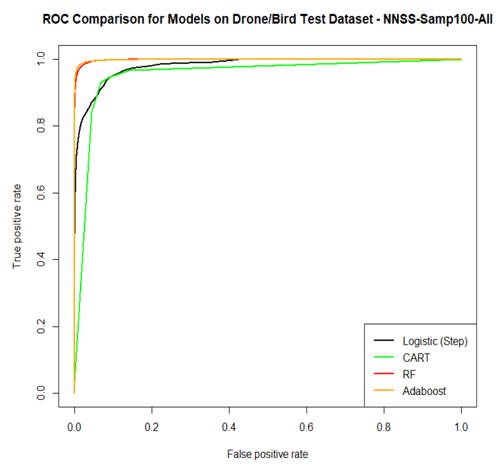


b. CTS - RandSamp - Validation

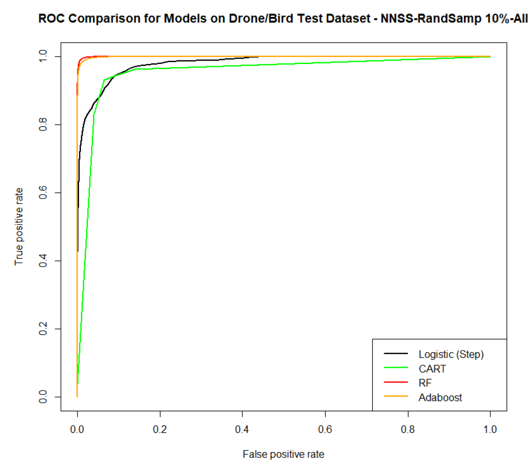
Figure 4.2. In our training environment comparison of the ROC curves for each of the model types trained using the CTS *Samp100* (left) and *RandSamp* (right) datasets, we can observe consistent performance between all of the models with the exception of the AdaBoost (orange) model whose ROC curve shows a marginal decrease in performance compared to the random forest model (red) showing a marginal increase performance.

4.1.2 NNSS Models - Training Environment Validation

When we contrasted our NNSS ROC curves from our training environment validation (Figure 4.3) of the models built using the *Samp100* (Figure 4.3a) and *RandSamp* (Figure 4.3b) datasets, we did not observe any discrepancies between our model types that differed from our expected performances. When comparing the CTS and NNSS models to one another, it is noteworthy that the NNSS logistic regression (black) and CART (green) models (Figure 4.3) were able to achieve superior performance in comparison with their respective CTS models (Figure 4.2). In fact, we observed that our NNSS logistic regression and CART models were able to discriminate birds from drones in the track data almost as effectively as the random forest and AdaBoost models, while their respective CTS models were far less effective. We believe this difference in the respective logistic regression and CART models between the two training sites may reveal an important environmental difference that we discuss further in Section 5.2.2.



a. NNSS - Samp100 - Validation



b. NNSS - RandSamp - Validation

Figure 4.3. In our training environment comparison, the models developed using the NNSS *Samp100* (left) and *RandSamp* (right) datasets showed comparable performance in terms of their ROC curves. Notably, the logistic regression (black) and CART (green) models appear to perform better than their respective models trained using the CTS datasets (Figure 4.2).

After evaluating the comparative training environment performance of the four model types between those developed using the *Samp100* and *RandSamp* datasets for both the CTS and NNSS training environments, in Section 4.2, we then evaluated the comparative *robustness* of the models by validating the models using the labeled track data from the alternate environment from which the models were trained.

4.2 Alternate Environment Validation

In our evaluation of the comparative model performance against bird and drone tracks in the alternate environment, similar to our performance evaluation, we were less focused on the raw balanced accuracy and AUC scores (although we did consider them) than on whether our new data sampling from the second iteration of our methodology had achieved improved performance (in terms of balanced accuracy and AUC scores). While we observed an improvement in all four of our CTS models and three out of the four NNSS models from our balanced accuracy comparison (Figure A.11), our comparison of AUC scores revealed the same improvement in all four of the CTS models but improvement in only one of the four NNSS models. Although we observed some disagreement between our balanced accuracy and AUC score comparison with respect to the NNSS models, it is clear that our adjusted sampling in the second iteration of our model development improved the *robustness* of our CTS models more than it did the NNSS models. It is also noteworthy that our random forest and AdaBoost models (with the exception of the NNSS random forest model), which are most likely to overfit to a particular training environment, achieved the most significant performance improvements during the alternate environment validation using the *RandSamp* dataset. Although we observed a balanced accuracy decrease by 20-25% in our top performing models when we compare their performance between the training environment (Figure A.10) and alternate environment (Figure A.11) AUC scores, all but the two CART models achieved AUC scores above 0.8 (Figure 4.4). These results suggest the importance of calibrating the models in a new environment, especially when the environments are drastically different. In Appendix A.4, we discuss the apparent trade-off in balanced accuracy between model calibration for a particular environment and the goal of increasing a model's *robustness* for multiple different environments. In Section 5.2.2, we discuss some of the environmental impacts on bird and drone track data and make some recommendations for improving a model's *robustness* to changing environmental conditions.

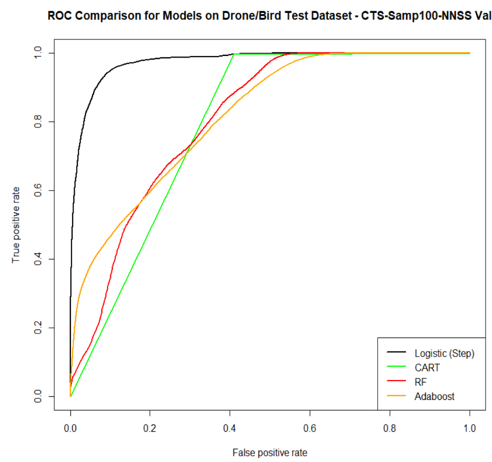
	Alternate Environment Validation (AUC Scores)			
	CTS - Samp100	CTS - RandSamp	NNSS - Samp100	NNSS - RandSamp
Logistic	0.9748060	0.9753221	0.9287062	0.9279182
CART	0.7929914	0.7969646	0.7212270	0.7055470
Random Forest	0.8071437	0.8787533	0.8937485	0.8796242
AdaBoost	0.8168372	0.8855789	0.8991405	0.9243978

Figure 4.4. Using AUC scores to compare the respective model types developed using the CTS and NNSS *RandSamp* and *Samp100* datasets, we observed improvement in all four of the CTS models but only one of the four NNSS models developed using the *RandSamp* datasets.

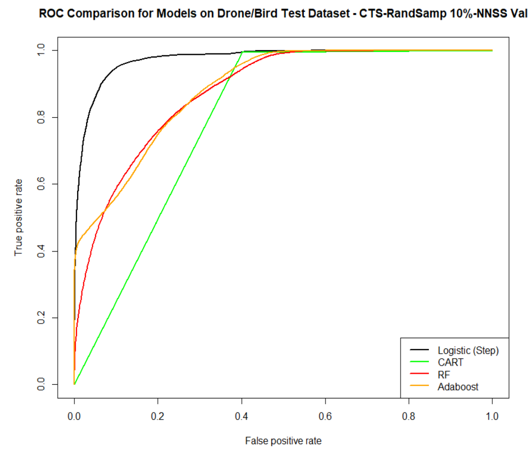
In addition to quantitatively comparing the alternate environment performance of our respective CTS and NNSS models using AUC scores, we also compared the visual representation of the model ROC curves in Sections 4.2.1 and 4.2.2.

4.2.1 CTS Models - Alternate Environment Validation

For our CTS models, aside from observing that each model's AUC score improved during the second iteration using the *RandSamp* dataset, we can observe that the ROC curves (Figure 4.5) for the random forest (red) and AdaBoost (orange) models become less erratic and more consistent. The apparent *robustness* of the logistic regression (black) model to environmental conditions indicates that the model's heavier reliance on an aerial object's RCS and altitude features as discriminators among the CTS track data proved most valuable in discriminating birds from drones in the NNSS training environment track data.



a. CTS - Samp100 - Robustness



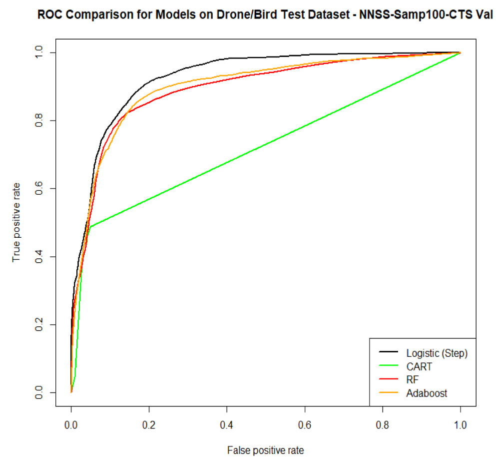
b. CTS - RandSamp - Robustness

Figure 4.5. In the alternate environment validation, the models developed using the CTS *RandSamp* dataset (right) demonstrated improved performance by comparison with the models developed using the *Samp100* dataset (left).

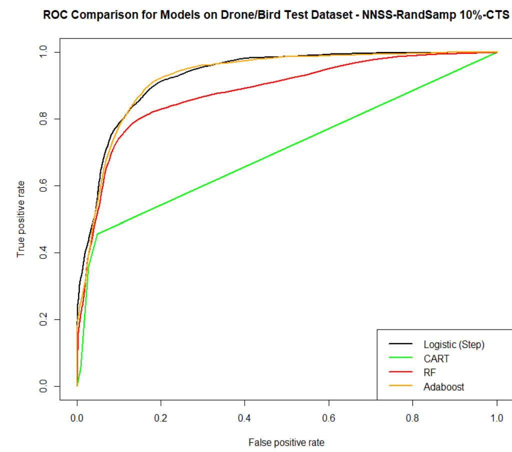
4.2.2 NNSS Models - Alternate Environment Validation

Since our alternate environment balanced accuracy (Figure A.11) and AUC score (Figure 4.4) model comparisons provided a mixed result regarding two of the four models developed using the NNSS training environment datasets, we took a closer look at the model ROC curves to better understand this phenomena.

In our observation of the ROC curves for the NNSS models (Figure 4.6), other than the AdaBoost (orange) model, the decrease in AUC score performance by the other three models is not immediately evident. For the random Forest and logistic regression *RandSamp* models that marginally improved (less than 1%) in balanced accuracy, both did so with marginal decreases in *sensitivity* (prediction rate against drones) and marginal increases in *specificity* (prediction rate against birds). The most significant overall observation from the ROC curve analysis is the dramatic improvement of the AdaBoost *RandSamp* (orange) model by comparison with the other three models. This improvement is largely due to the model's dramatic relative reduction in its false positive rate (predicting birds as drones). Nevertheless, our NNSS comparative analysis of the ROC curves, AUC scores, and balanced accuracy rates between the two iterations revealed only marginal success towards our goal of improving model *robustness* to conditions in the alternate environment.



a. NNSS - Samp100 - Robustness



b. NNSS - RandSamp - Robustness

Figure 4.6. For our NNSS *RandSamp* dataset (right) only the AdaBoost (orange) model improved upon its *Samp100* dataset (left) performance during the alternate training environment validation.

From our alternate environment comparison of NNSS and CTS ROC curves, although the CTS random forest and AdaBoost models made the most significant improvements in the second iteration, our NNSS random forest and AdaBoost models still performed better than their respective CTS models. Although we could not confirm any additional information about drone types in the specific track data batches we received for each training site, our observation of a consistent difference in the number of prominent clusters (two and three clusters in the CTS and NNSS datasets respectively), discussed in Sections 3.5.1 and 3.5.3, leads us to hypothesize that there may have been different proportional representations of rotary wing and fixed-wing drones in our track data between the two training sites. With the understanding that our more sophisticated learning algorithms (i.e. random forest and AdaBoost) rely more heavily on understanding the differences in flight phenomenology between birds and drones, an algorithm only able to identify one homogeneous drone type (as opposed to two) would be disadvantaged in discriminating birds from drones in an environment with a comparable proportion of rotary wing and fixed-wing drones. We discuss this topic and its implications for future research in Section 5.2.4.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 5:
Conclusion

5.1 Summary of Results

From our original consultation with Anduril Industries, we designed our research plan with the intent of accomplishing two goals. First, we wanted to apply a more systematic, methodical, and rigorous approach towards developing an improved model (or corroborating the performance of Anduril's existing classifier) for discriminating birds from drones. Second, we wanted to determine whether we could develop a model with greater *robustness* towards discriminating birds from drones in an unfamiliar environment from the one in which we trained it. The latter result would have an impact on the time spent calibrating a system's model in each new environment.

Towards the first goal, our top performing CTS model, using random forests, achieved a balanced accuracy above 97% using the *RandSamp* dataset (with additional acceleration components) in the second iteration. Our top performing NNSS models, using random forests and an adaptive boosting (AdaBoost) algorithm, both achieved a balanced accuracy above 98% from the second iteration. Both models improved upon the performance of Anduril's classifier (per the CTS and NNSS datasets provided by our data sponsor).

Towards our second goal of developing a more *robust* model, we did not develop a model whose performance in an unseen environment would have been acceptable to use without re-calibrating for that new environment. However, after completing a first iteration of model development, our modified sampling and inclusion of acceleration components in our models led to marginal improvements in most of our models in discriminating birds from drones in an unseen environment.

As a result of our systematic, comprehensive, and rigorous model development process, we derived insights about the learning process of our models to help improve the performance of existing models and identify areas for future model improvements in discriminating between birds and drones. In Section 5.2, we discuss several areas for future work.

5.2 Future Work

As a result of our systematic, comprehensive, and rigorous model development process, we were not only able to enhance intuition and interpretability of our models, but we were able to understand how our models adapted with the change in sampling techniques and the addition of features to the models. In doing so, we identified some additional opportunities for future work in this growing research area of C-sUAS. From our research, we identified four of these opportunities.

5.2.1 Combining Radar and Optical Sensor Data

Our research demonstrated an effective two-phase model development process using entirely radar data to first perform unsupervised learning and identify preferred *cluster groups* to add as categorical features to the supervised learning model development in the second phase. Although we observed some marginal improvement in the accuracy of random forest models in the second iteration of our model development using a two-cluster, three-cluster, and four-cluster categorical feature (Appendix A.2.4), we did not use any models with cluster group features in our validation process. However, we believe that this process could be used to include categorical *cluster group* features from other sensor data, to include EO/IR or RF. This hybrid semi-supervised learning approach has proven effective in other areas by producing *robust* models capable of accurately classifying outliers in the fraud detection research in Carcillo et al. (2021) and producing more efficient (in terms of speed and accuracy) classification models requiring the addition of new spatial-temporal data for human activity recognition research in Budisteanu and Mocanu (2021). We recommend continuing to use a two-phase process and employing the best combination of unsupervised and supervised learning methods while incorporating other sensor data that may contribute to enhancing the prediction accuracy. In addition to improving prediction accuracy, the inclusion of other sensor data less sensitive to environmental conditions will improve model *robustness* while reducing the vulnerabilities of any one type of sensor data. Siewert et al. (2019, p. 1) demonstrated the feasibility of reducing false positives using a combination of EO/IR, acoustic, and radar sensor data by adding features such as "target characteristic shape, texture, and spectral data." Corradino et al. (2021) offered a similar application of this technique involving machine learning classification algorithms with a combination of radar and optical satellite imagery data to improve the change detection performance for mapping lava flows. While continued improvement should be pursued using exclusively radar data, the scale of improvement necessary to stay ahead of the adversarial drone population will inevitably require the incorporation of other sensor data in models.

5.2.2 Environmental Impacts on Track Data

After observing the inherent challenge of developing a *robust* model that can sustain its discriminatory performance in different environments, we sought to first specify some of the aspects of the problem and then offer some potential future research to pursue. Due to the dynamic conditions in the environment (i.e. wind and precipitation), along with the unique landscape of the environment itself, a supervised learning methodology reliant on features of the flight patterns and phenomenology of birds and drones for model discrimination will inevitably have noisy datasets due to the changing environmental conditions or unique wind patterns or obstacles based on the particular landscape. Although drones will generally fly along pre-programmed or human-controlled routes without any regard for wind patterns or other environmental conditions, birds will adjust their flight pattern based on the conditions (such as drafting back and forth into a headwind). Liu et al. (2021, p. 2–3) also recognized the importance of this problem set as they performed “flight mechanic and behaviour analysis” and “motion characteristic modelling” to distinguish bird and drones under dynamic environmental conditions. Our recommendation for improving supervised learning model performance would be the inclusion of real-time wind speed and direction at the geographic location of the aerial object being tracked. By also identifying and mapping the geographic locations that may serve as obstacles or avoidance areas for drones (but not for birds), it may also be possible to add to the discriminatory performance by learning the behavior of aerial objects that may be affected differently by the landscape and obstacles within a particular environment. In an effort to improve the *robustness* of our models in different environments, the desire to pursue models that can account for the environmental conditions will ultimately need to be balanced with the time and effort involved in calibrating a system in a new environment. We recommend additional research studying how much track data may be necessary to train, test, and validate a new model to a given standard and how much of the track data must be labeled data. In Section 5.2.3 we discuss the potential future research with semi-supervised learning.

5.2.3 Semi-supervised Learning with Radar Track Data

Although our research was fortunate to have complete radar track data for hundreds of labeled bird and drone tracks, gaining access to such data in an austere or combat environment can be challenging, time-consuming, and costly. Consistent with the problem of knowing how much labeled track data is necessary to calibrate a system in a new environment, a related problem is what proportion of the tracks need to be labeled for adequate model development for a given standard. To answer this question, it is also important to understand the consequences of performing supervised learning with labeled datasets that do not accurately reflect a balanced sampling of the different types of drones and their respective flight phenomenologies to be learned by the models. In Section 5.2.4, we discuss the importance of this finding.

5.2.4 Drone Discrimination: Rotary Wing and Fixed-Wing

Through the unsupervised learning phase of our model development during both iterations, our clustering analysis clearly identified a difference in the number of prominent clusters (two and three respectively for the CTS and NNSS track data) that indicated the ability of our models to not only understand the different flight phenomenology of birds and drones, but between different types of drones (i.e. rotary wing and fixed-wing drones). In particular, during our clustering analysis for the NNSS track data, we believe our unsupervised learning methods were able to distinguish birds, rotary wing drones, and fixed-wing drones as three distinct clusters. Meanwhile, we hypothesize that our clustering analysis for the CTS track data resulted in only two prominent clusters because the CTS track data may not have included a sufficient number of fixed-wing drones, or the fixed-wing drone flight phenomenology was easily conflated with that of birds within the CTS training environment. Although the datasets Anduril Industries provided for this research did not include additional labeled information about the types of drones, our anecdotal observation of a predominance of rotary wing drones at the CTS training site and understanding of the distinct differences in the two training environments leads us to conclude that the proportional representation of drone types in a dataset and the environmental impact on the flight patterns of different drone types should be considered within the context of future research in this area.

5.3 Summary of Findings

In our *post mortem* of the research conducted in this thesis, while initially embarking on a mission to improve upon the performance of existing algorithms that discriminate birds from drones, we found that our methodology—including a comprehensive exploration of the data in two unique training environments and experimentation with data sampling and additional derived features—provided the most illuminating insights towards this problem set. From the time of our initial consultation with our data sponsor to the completion of this study, Anduril Industries has continued to improve upon its existing algorithms and radar system calibration to new environments, while also integrating other non-radar sensors within its *system-of-systems* design. During the same time period, the Russia-Ukraine War played out in real-time and extended beyond the one-year mark, in part due to Ukraine's ability to leverage some asymmetric advantages of sUAS to exploit vulnerabilities in what was once thought to be an insurmountable and overwhelming Russian military advantage. Although C-UAS defense technologies will inevitably gravitate towards protecting large installations and critical infrastructure against the threat of larger UAS (groups 3 and higher) with a greater destructive capacity, research efforts cannot ignore and must continue to pursue incremental improvements in statistical and machine learning models for radar data alone against the menacing challenge that sUAS (groups 1 and 2) will continue to be, especially at-scale. While the optimal C-sUAS solution lies in an "all of the above" approach that includes improved technological system capabilities, increased ingest and processing of real-time data, and the integration of different sensor types to reduce the vulnerabilities of any one sensor, rigorous testing and evaluation of statistical and machine learning models, as explored in this thesis, must keep pace with technological advancements to stay ahead of adversarial sUAS.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX: Additional Graphics and Outputs

A.1 Supervised Learning Modeling Outputs (*Samp100*)

A.1.1 Logistic Regression - Model Selection Comparison

For both the CTS and NNSS *Samp100* datasets, using the stepwise model selection process allows us to slightly improve our Akaike Information Criteria (AIC) score (by comparison with their respective baseline regression models) and to derive some additional insight regarding the additional importance of the upward velocity component as a predictor after eliminating the eastern component of the velocity from our models.

```
CTS - Samp100 (Logistic Regression Step Model Summary)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.3161  -0.4329  -0.1525   0.4255   3.1335

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)     -0.82804    0.06616 -12.515  < 2e-16 ***
scale(enuVel_n) -0.27878    0.06265  -4.450  8.6e-06 ***
scale(enuVel_u)  0.11820    0.06046   1.955   0.0506 .
scale(altAgl)   -0.87139    0.08383 -10.394  < 2e-16 ***
scale(rcs)       2.97347    0.10453  28.446  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)
Null deviance: 4362.2  on 3265  degrees of freedom
Residual deviance: 1991.3  on 3261  degrees of freedom
AIC: 2001.3
```

```
Number of Fisher Scoring iterations: 6
```

By comparing the CTS *Samp100* logistic regression step model summary (above) with the NNSS *Samp100* model summary (below), we can see that both the northern and upward velocity components offer greater predictive power in the NNSS *Samp100* model, and the NNSS model also provides a better balance of goodness of fit and complexity (measured by a lower AIC score).

```
NNSS - Samp100 (Logistic Regression Step Model Summary)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.8961  -0.1183   0.0324   0.1861   3.1690

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)      1.05295    0.09067  11.613  < 2e-16 ***
scale(enuVel_n) -0.38696    0.08310  -4.656 3.22e-06 ***
scale(enuVel_u)  0.16816    0.06435   2.613  0.00896 **
scale(altAgl)   -3.08305    0.17685 -17.434  < 2e-16 ***
scale(rcs)       3.93341    0.15646  25.140  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)
Null deviance: 5417.8  on 4014  degrees of freedom
Residual deviance: 1434.8  on 4010  degrees of freedom
AIC=1444.77

Number of Fisher Scoring iterations:  7
```

A.1.2 K-Nearest Neighbors (KNN) - Testing for k

Using both bootstrapping and a 10-fold cross-validation, we were able to quickly and thoroughly tune and test our KNN models using k values from 1 to 40 to corroborate our best-performing models using the CTS and NNSS *Samp100* datasets. Our optimal k values (lowest error) using both bootstrapping (brown) and cross-validation (gold), appear in the first ten using the NNSS *Samp100* dataset (Figure A.2) but for much higher k values for the CTS *Samp100* dataset (Figure A.1). We ultimately chose $k = 5$ for both the (CTS) and (NNSS) *Samp100* models by observing our earliest "knee" that seems to agree between the two algorithms and approximates the minimum error observed.

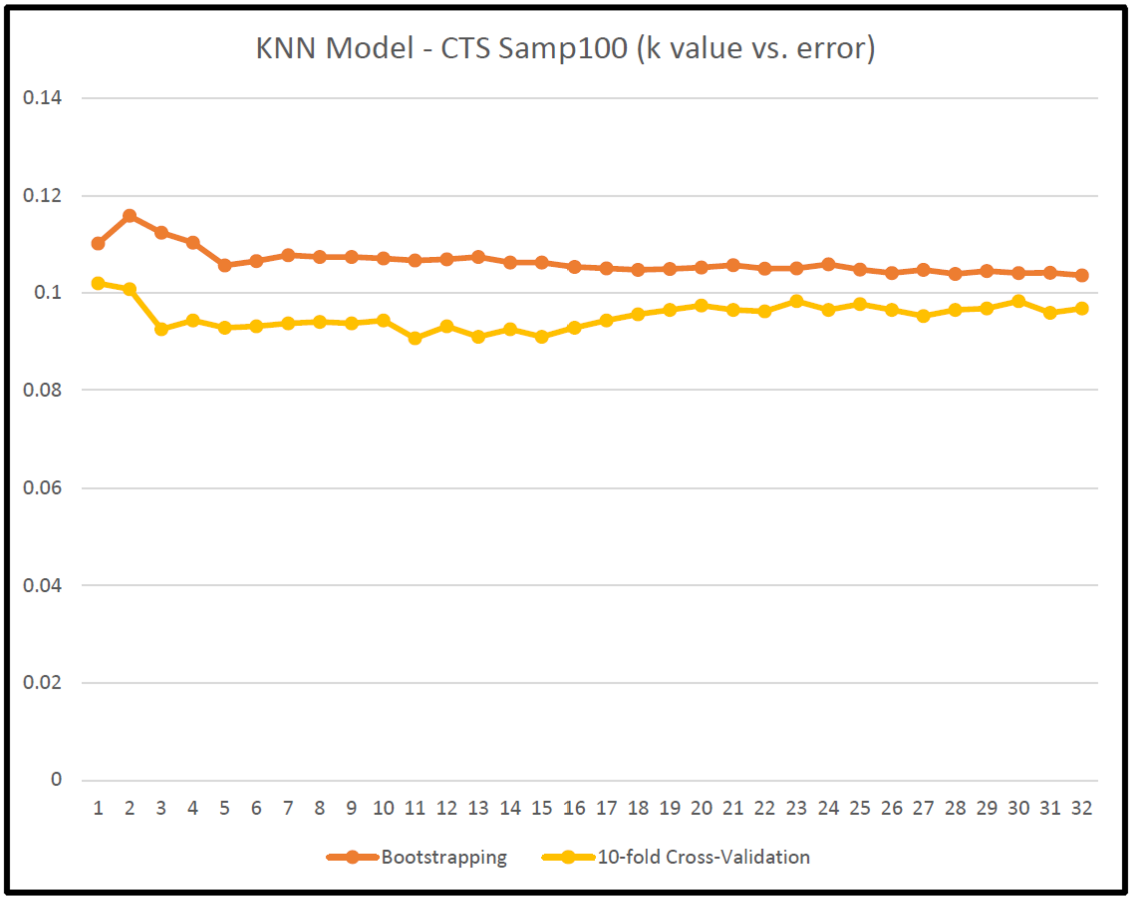


Figure A.1. For the CTS *Samp100* dataset, although we observed marginally lower error rates for some higher k values using both the bootstrapping (brown) and cross-validation (gold) methods, we ultimately chose $k = 5$ because of the "knee" that appears at an approximate minimal error rate using both methods.

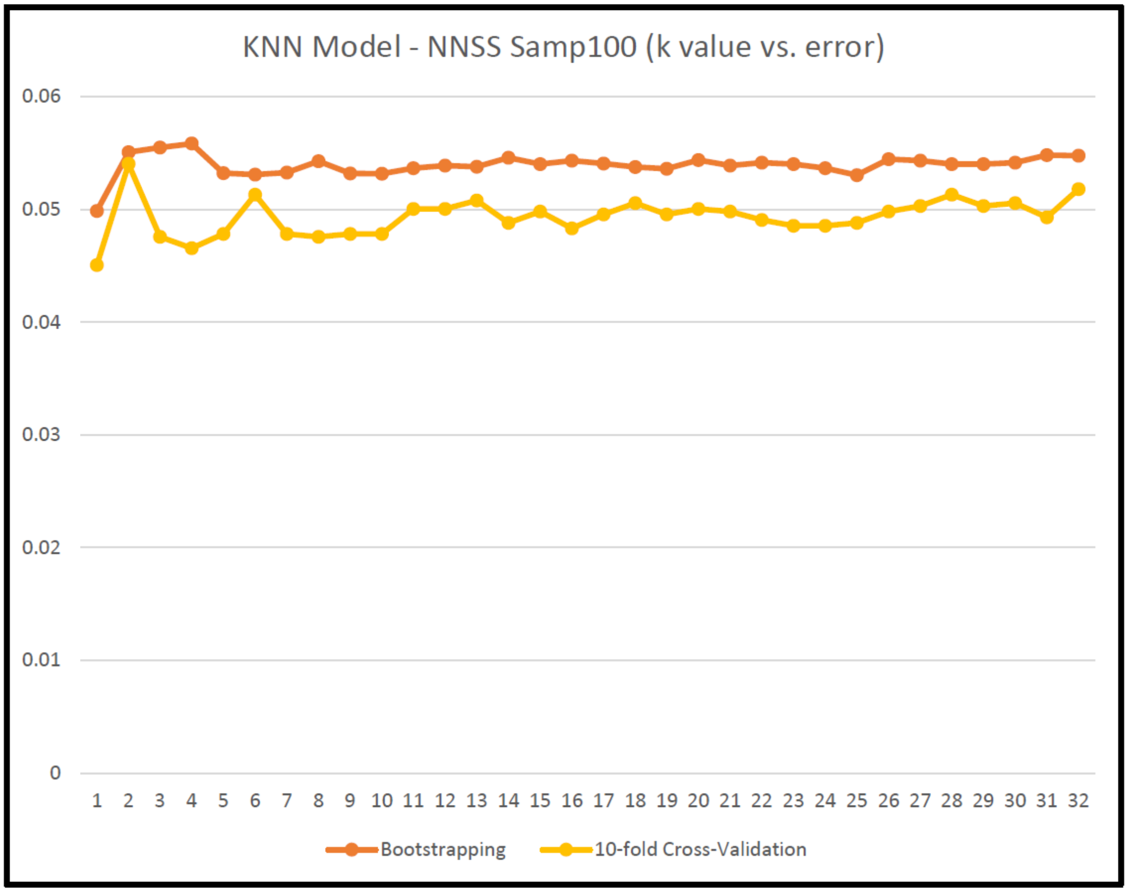


Figure A.2. For the NNSS *Samp100* dataset, while ignoring $k = 1$ due to the likelihood of overfitting to the training set, we chose $k = 5$ because of the "knee" in the bootstrapping (brown) graph that also approximates the minimum error rate observed for both methods.

A.1.3 Classification and Regression Trees (CART) - Model Diagrams

Although the best-performing CTS and NNSS CART models provided comparable results as their respective logistic regression models, the CART models offered some additional insights with respect to how we can distinguish birds from drones across all of the predictors using their model tree diagrams. Additionally, the tree diagrams and CART model summary data allow us to contrast the numerical differences between the distinguishing features of birds and drones observed in the CTS and NNSS training environments. Using the CTS

Samp100 dataset, our CART model summary (below) and CART model tree diagram (Figure A.3) tell us that nearly 92% of the aerial objects having a RCS less than -13.7 are birds and nearly 86% of the aerial objects having a RCS greater than -13.7 are drones. We can also quantitatively see the overwhelming importance of RCS and altitude by comparison with the three velocity components that have comparable importance with one another.

CTS - Samp100 (CART Model Summary)

```
         CP nsplit rel error    xerror       xstd
1 0.7277032      0 1.0000000 1.0000000 0.02197910
2 0.0100000      1 0.2722968 0.2762431 0.01395221

Variable importance
     rcs   altAgl  enuVel_u enuVel_e enuVel_n
      84       10         2        2        2

Node number 1: 3266 observations,    complexity param=0.7277032
  predicted class=OBJECT_CLASS_BIRD         expected loss=0.3879363  P(node) =1
    class counts:  1267  1999
   probabilities: 0.388 0.612
  left son=2 (1288 obs) right son=3 (1978 obs)
  Primary splits:
      rcs      < -13.71406  to the right, improve=939.50680, (0 missing)
      altAgl   < 255.2532   to the left,  improve=165.68750, (0 missing)
      enuVel_n < 1.012891   to the left,  improve=136.34710, (0 missing)
      enuVel_e < -0.2071627 to the right, improve= 72.16830, (0 missing)
      enuVel_u < 0.2295878  to the left,  improve= 22.41724, (0 missing)
  Surrogate splits:
      altAgl   < 17.55489   to the left,  agree=0.655, adj=0.124, (0 split)
      enuVel_u < -2.565018  to the left,  agree=0.614, adj=0.022, (0 split)
      enuVel_e < 13.4608    to the right, agree=0.614, adj=0.021, (0 split)
      enuVel_n < -18.82278  to the left,  agree=0.614, adj=0.021, (0 split)

Node number 2: 1288 observations
  predicted class=OBJECT_CLASS_AIR_VEHICLE  expected loss=0.1420807  P(node) =0.3943662
```

```
    class counts:  1105   183
   probabilities: 0.858 0.142

Node number 3: 1978 observations
  predicted class=OBJECT_CLASS_BIRD         expected loss=0.08190091  P(node) =0.6056338
    class counts:   162  1816
   probabilities: 0.082 0.918
```

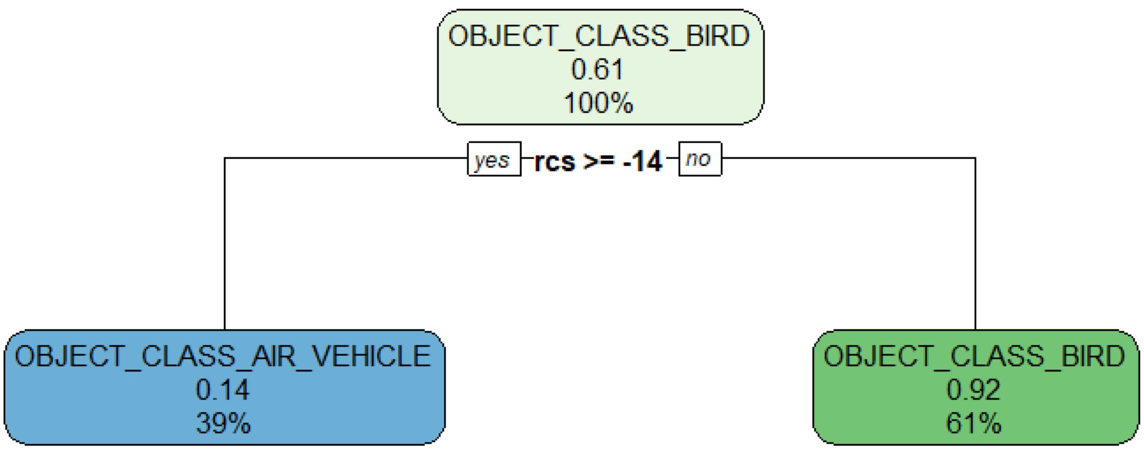


Figure A.3. Using *rpart* in R, the *Samp100* CART tree diagram only shows the primary split between birds and drones based on RCS alone.

Using the NNSS *Samp100* dataset, our CART model summary (below) and tree diagram (Figure A.4) tell us that nearly 95% of the aerial objects having a RCS greater than -16.1 are drones and nearly 82% of the aerial objects having RCS less than -16.1 are birds. The NNSS CART model tree diagram tells us that we can assert an even higher percentage of those respective drone and bird populations if we also know whether the aerial object is flying above or below $135m$ and $407m$ above ground level and faster or slower than $5.2m/sec$ to the south and $9m/sec$ to the north. Additionally, by comparison with the CTS *Samp100* CART model, the NNSS model attributes a greater relative importance to altitude (as opposed to RCS) and the velocity components.

NNSS - Samp100 (CART Model Summary)

```
          CP nsplit rel error    xerror       xstd
1 0.72027110      0 1.0000000 1.0000000 0.01915925
2 0.03696858      1 0.2797289 0.2797289 0.01236382
3 0.02926679      2 0.2427603 0.2501540 0.01177051
4 0.02156500      4 0.1842267 0.2045595 0.01075248
5 0.01000000      5 0.1626617 0.1817622 0.01018641

Variable importance
     rcs   altAgl enuVel_u enuVel_n
      56       37        4        3

Node number 1: 4015 observations,    complexity param=0.7202711
  predicted class=OBJECT_CLASS_AIR_VEHICLE  expected loss=0.4042341  P(node) =1
    class counts:  2392  1623
   probabilities: 0.596 0.404
  left son=2 (2174 obs) right son=3 (1841 obs)
  Primary splits:
      rcs      < -16.09467  to the right, improve=1161.31200, (0 missing)
      altAgl   < 240.1558   to the left,  improve= 792.89790, (0 missing)
      enuVel_n < -13.22583  to the left,  improve=  91.38170, (0 missing)
      enuVel_u < -1.404805  to the right, improve=  45.08646, (0 missing)
      enuVel_e < -14.64278  to the left,  improve=  23.31037, (0 missing)
  Surrogate splits:
      altAgl   < 175.6927   to the left,  agree=0.747, adj=0.448, (0 split)
      enuVel_u < -0.9123494 to the right, agree=0.570, adj=0.063, (0 split)
      enuVel_e < 5.219891   to the left,  agree=0.543, adj=0.003, (0 split)

Node number 2: 2174 observations,    complexity param=0.03696858
  predicted class=OBJECT_CLASS_AIR_VEHICLE  expected loss=0.05427783  P(node) =0.5414695
    class counts:  2056   118
   probabilities: 0.946 0.054
  left son=4 (2114 obs) right son=5 (60 obs)
  Primary splits:
```

```
      altAgl   < 406.8718   to the left,  improve=110.373000, (0 missing)
      rcs      < -15.33152  to the right, improve= 35.631710, (0 missing)
      enuVel_e < -3.335563  to the right, improve=  2.455990, (0 missing)
      enuVel_n < 8.308707   to the right, improve=  2.093409, (0 missing)
      enuVel_u < -1.788771  to the right, improve=  1.368066, (0 missing)

Node number 3: 1841 observations,    complexity param=0.02926679
  predicted class=OBJECT_CLASS_BIRD         expected loss=0.1825095  P(node) =0.4585305
    class counts:   336  1505
   probabilities: 0.183 0.817
  left son=6 (441 obs) right son=7 (1400 obs)
  Primary splits:
      altAgl   < 135.0366   to the left,  improve=142.380000, (0 missing)
      rcs      < -17.82246  to the right, improve= 75.468890, (0 missing)
      enuVel_n < -13.04793  to the left,  improve= 63.875410, (0 missing)
      enuVel_u < 1.361603   to the left,  improve=  9.870249, (0 missing)
      enuVel_e < 1.573629   to the left,  improve=  6.878949, (0 missing)
  Surrogate splits:
      enuVel_n < -14.18647  to the left,  agree=0.767, adj=0.027, (0 split)
      enuVel_u < -8.515464  to the left,  agree=0.761, adj=0.002, (0 split)

Node number 4: 2114 observations
  predicted class=OBJECT_CLASS_AIR_VEHICLE  expected loss=0.02743614  P(node) =0.5265255
    class counts:  2056    58
   probabilities: 0.973 0.027

Node number 5: 60 observations
  predicted class=OBJECT_CLASS_BIRD         expected loss=0  P(node) =0.01494396
    class counts:     0    60
   probabilities: 0.000 1.000

Node number 6: 441 observations,    complexity param=0.02926679
  predicted class=OBJECT_CLASS_AIR_VEHICLE  expected loss=0.4671202  P(node) =0.1098381
    class counts:   235   206
   probabilities: 0.533 0.467
```

```
  left son=12 (155 obs) right son=13 (286 obs)
  Primary splits:
      enuVel_n < -5.201084  to the left,  improve=35.774770, (0 missing)
      rcs      < -17.95087  to the right, improve=32.675870, (0 missing)
      enuVel_u < -1.200639  to the right, improve=11.465660, (0 missing)
      enuVel_e < 1.748426   to the left,  improve=10.088750, (0 missing)
      altAgl   < 103.1564   to the left,  improve= 6.543585, (0 missing)
  Surrogate splits:
      enuVel_e < -0.338431  to the left,  agree=0.667, adj=0.052, (0 split)
      rcs      < -17.16393  to the right, agree=0.664, adj=0.045, (0 split)
      altAgl   < 133.9505   to the right, agree=0.651, adj=0.006, (0 split)

Node number 7: 1400 observations
  predicted class=OBJECT_CLASS_BIRD         expected loss=0.07214286  P(node) =0.3486924
    class counts:   101  1299
   probabilities: 0.072 0.928

Node number 12: 155 observations
  predicted class=OBJECT_CLASS_AIR_VEHICLE  expected loss=0.1935484  P(node) =0.03860523
    class counts:   125    30
   probabilities: 0.806 0.194

Node number 13: 286 observations,    complexity param=0.021565
  predicted class=OBJECT_CLASS_BIRD         expected loss=0.3846154  P(node) =0.07123288
    class counts:   110   176
   probabilities: 0.385 0.615
  left son=26 (75 obs) right son=27 (211 obs)
  Primary splits:
      enuVel_n < 9.038784   to the right, improve=24.724270, (0 missing)
      rcs      < -17.17303  to the right, improve=20.353760, (0 missing)
      enuVel_e < -3.367993  to the right, improve=14.102560, (0 missing)
      altAgl   < 104.0111   to the left,  improve= 3.827577, (0 missing)
      enuVel_u < -1.200639  to the right, improve= 3.772067, (0 missing)

Node number 26: 75 observations
```

```
  predicted class=OBJECT_CLASS_AIR_VEHICLE  expected loss=0.2666667  P(node) =0.01867995
    class counts:    55    20
   probabilities: 0.733 0.267

Node number 27: 211 observations
  predicted class=OBJECT_CLASS_BIRD         expected loss=0.2606635  P(node) =0.05255293
    class counts:    55   156
   probabilities: 0.261 0.739
```

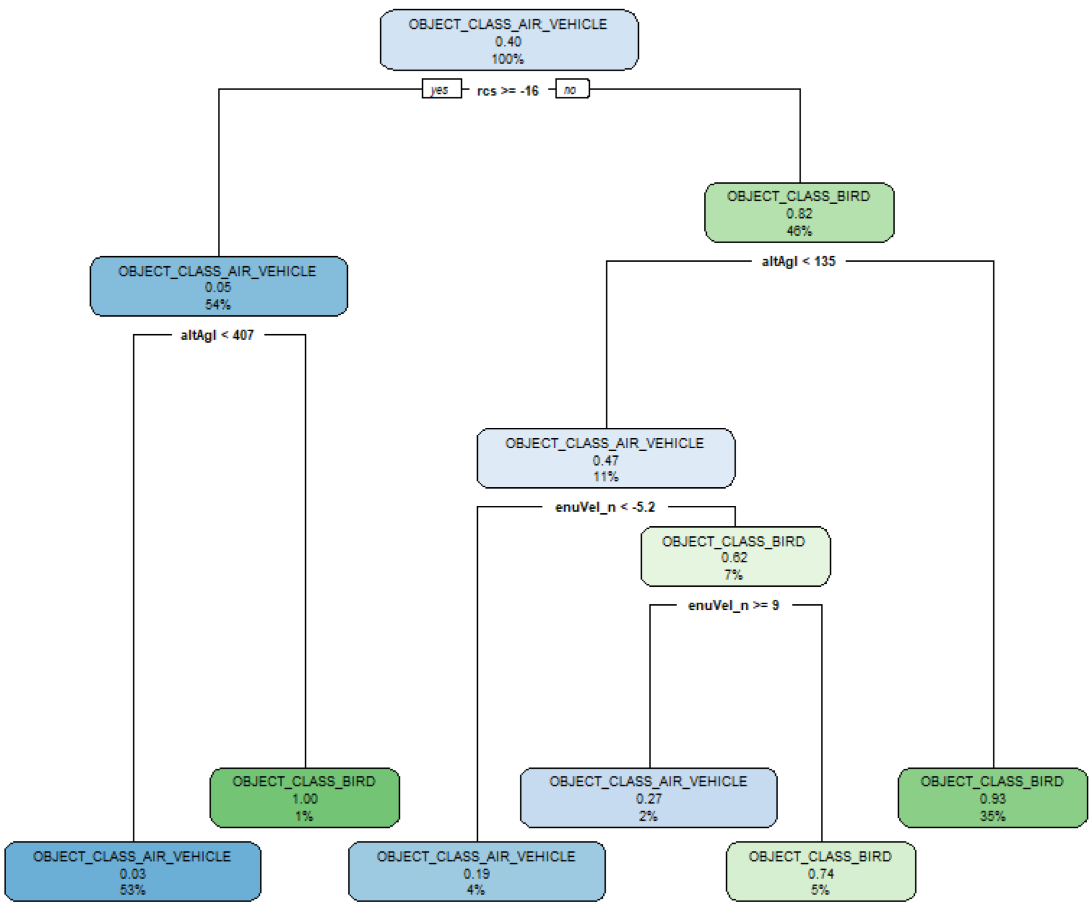


Figure A.4. By contrast with the *Samp100* CTS CART model tree diagram (Figure A.3), the *Samp100* NNSS tree diagram shows multiple levels of splits and distinguishes birds from drones based on RCS, altitude, and the northern velocity component.

A.2 Supervised Learning Modeling Outputs (*RandSamp*)

A.2.1 Logistic Regression - Model Selection Comparison

Using both the CTS and NNSS *RandSamp* datasets, our respective fitted logistic regression step models (below) indicate that RCS, altitude, all three velocity components, and only one of the acceleration components in each model is significant enough to be included: the northern acceleration component in the *RandSamp* CTS model and the upward acceleration component in the *RandSamp* NNSS model. The additional velocity and acceleration components in both the CTS and NNSS *RandSamp* logistic regression models tells us that these models can detect additional discriminatory behavior, between the velocity and acceleration of birds and drones, that their respective *Samp100* logistic regression models could not.

```
CTS - RandSamp (Logistic Regression Step Model Summary)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.3156  -0.4492  -0.1629   0.4221   3.1026

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)     -0.84941    0.02070 -41.040  < 2e-16 ***
scale(enuVel_e) -0.05258    0.01892  -2.779  0.00546 **
scale(enuVel_n) -0.30712    0.01979 -15.517  < 2e-16 ***
scale(enuVel_u)  0.10271    0.01780   5.770 7.93e-09 ***
scale(enuAcc_n) -0.03678    0.01823  -2.018  0.04361 *
scale(altAgl)   -0.89001    0.02626 -33.898  < 2e-16 ***
scale(rcs)       2.87620    0.03203  89.798  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)
    Null deviance: 43441  on 32648  degrees of freedom
Residual deviance: 20227  on 32642  degrees of freedom
AIC: 20241
```

```
Number of Fisher Scoring iterations: 6
```

In our comparison of the *RandSamp* CTS logistic regression step model summary (above) and NNSS model summary (below), the only big distinction between the two is the additional marginal importance of the northern acceleration component in the CTS model and the upward acceleration component in the NNSS model. It is also noteworthy that in both the CTS and NNSS models, the northern and upward velocity components are far more important than the eastern velocity component, suggesting that birds and drones can be more easily distinguished from one another by how fast they are flying north, south, upwards, or downwards than in the east/west direction. We can intuitively understand this distinction in terms of the migration patterns of birds (north/south) and unique upward and downward movement of drones (rotary wing, in particular).

```
NNSS - RandSamp (Logistic Regression Step Model Summary)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.1542  -0.1151   0.0298   0.1818   3.2961

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)      0.95022    0.02888  32.907  < 2e-16 ***
scale(enuVel_e)  0.07415    0.02313   3.205  0.00135 **
scale(enuVel_n) -0.41318    0.02730 -15.134  < 2e-16 ***
scale(enuVel_u)  0.13900    0.02221   6.258  3.9e-10 ***
scale(enuAcc_u) -0.04611    0.02346  -1.965  0.04940 *
scale(altAgl)   -3.24244    0.05885 -55.093  < 2e-16 ***
scale(rcs)       3.96723    0.05002  79.307  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)
    Null deviance: 54386  on 40154  degrees of freedom
```

```
Residual deviance: 14206  on 40148  degrees of freedom
AIC: 14220

Number of Fisher Scoring iterations: 7
```

A.2.2 K-Nearest Neighbors (KNN) - Testing for k

Using the *RandSamp* dataset, we observed similar shaped graphs between the respective CTS (Figure A.5) and NNSS (Figure A.6) bootstrapping (brown) and 10-fold cross validation (gold) graphs. For both the CTS and NNSS KNN models, we chose $k = 3$ as our optimal result due to the observed "knee" in both cross-validation (gold) graphs while approximating the minimum error rate using both methods. The observed change in preferred k value (from $k = 5$ to $k = 3$), between the first and second iterations of model development, meant that the respective CTS and NNSS KNN models developed using the *RandSamp* dataset (second iteration) were able to better capture finer structure of the space without having to increase k to reduce the model's sensitivity to noise present in the dataset.

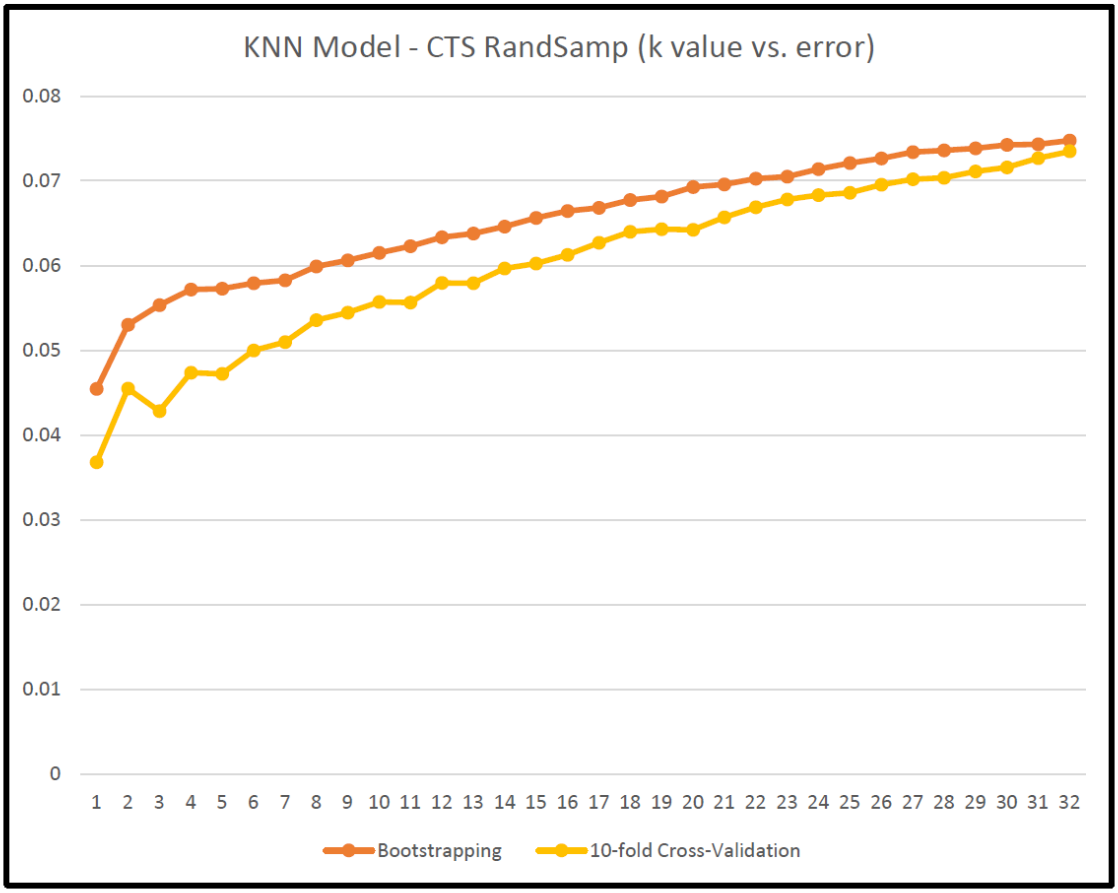


Figure A.5. For the CTS *RandSamp* dataset, we observed a lower optimal value ($k = 3$) than we observed for the CTS *Samp100* dataset (Figure A.1). (Note: we avoided choosing $k = 1$ for our optimal model due to the likelihood of overfitting to the training set.)

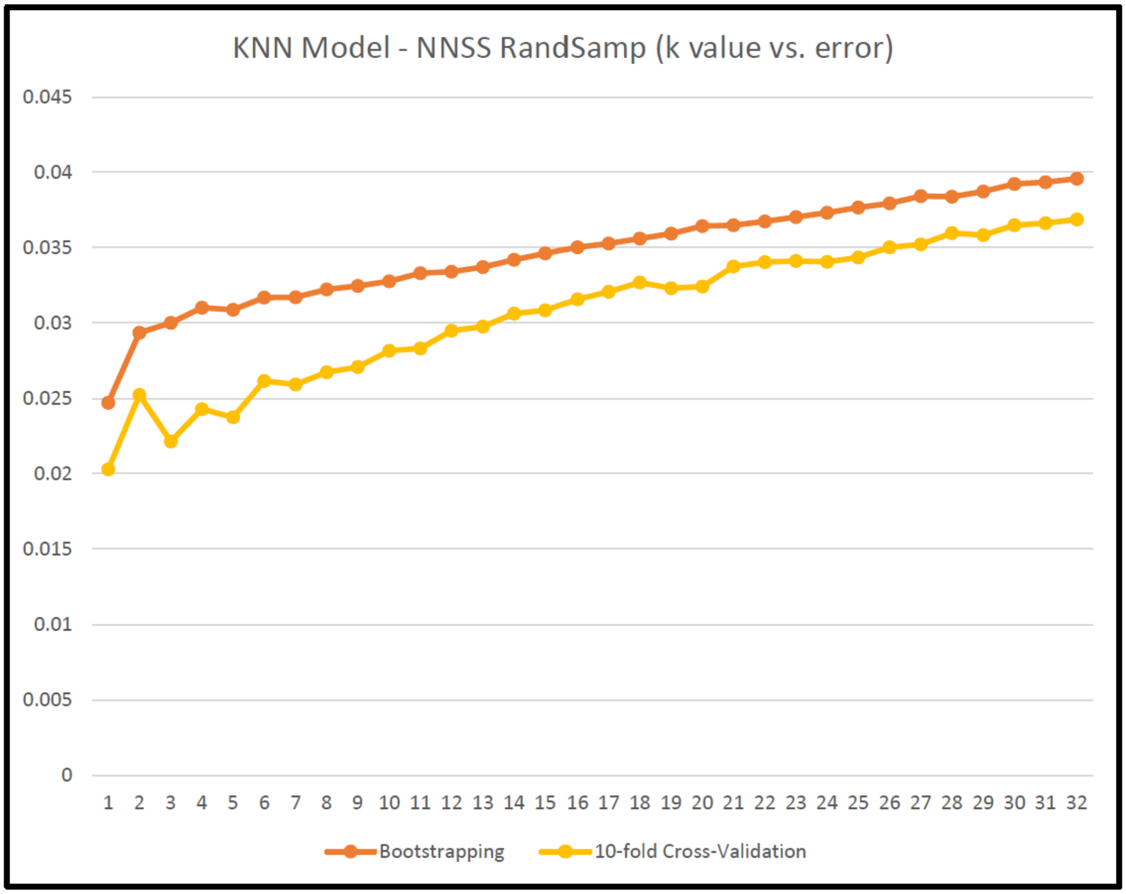


Figure A.6. For the NNSS *RandSamp* dataset, while ignoring $k = 1$ due to the likelihood of overfitting to the training set, we chose an optimal value of $k = 3$ (lower than the optimal value of $k = 5$ in Figure A.2).

A.2.3 Classification and Regression Trees (CART) - Model Diagrams

When comparing the *RandSamp* CART models with the *Samp100* models from the first iteration, we do not see significant differences in the overarching regression trees and only some small refinements in the relative importance of the top discriminating features: RCS and altitude. For the CTS datasets, our regression trees (Figures A.3 and A.7) are nearly identical, reinforcing the importance of determining whether an aerial object is a bird or drone at the CTS training environment almost exclusively on whether the RCS is greater than or less than -13.7. In our comparison of the *Samp100* (Appendix A.1.3) and

RandSamp (below) CTS CART model summaries, we only see a slight increase in the relative importance of altitude with respect to RCS and the velocity components in the *RandSamp* dataset.

```
CTS - RandSamp (CART Model Summary)

         CP nsplit rel error    xerror        xstd
1 0.7067585      0 1.0000000 1.0000000 0.007031932
2 0.0100000      1 0.2932415 0.2941224 0.004571954

Variable importance
     rcs   altAgl enuVel_n enuVel_e enuVel_u
      84       12        2        1        1

Node number 1: 32649 observations,    complexity param=0.7067585
  predicted class=OBJECT_CLASS_BIRD         expected loss=0.3824926  P(node) =1
    class counts: 12488 20161
   probabilities: 0.382 0.618
  left son=2 (12810 obs) right son=3 (19839 obs)
  Primary splits:
      rcs      < -13.7605   to the right, improve=8999.5450, (0 missing)
      altAgl   < 257.3084   to the left,  improve=1539.3980, (0 missing)
      enuVel_n < 1.022928   to the left,  improve=1233.0560, (0 missing)
      enuAcc_n < -0.1175795 to the right, improve= 713.4543, (0 missing)
      enuVel_e < -0.2942276 to the right, improve= 630.2046, (0 missing)
  Surrogate splits:
      altAgl   < 17.78213   to the left,  agree=0.664, adj=0.143, (0 split)
      enuVel_n < -18.39431  to the left,  agree=0.616, adj=0.020, (0 split)
      enuVel_e < 15.76142   to the right, agree=0.613, adj=0.013, (0 split)
      enuVel_u < -4.691854  to the left,  agree=0.610, adj=0.006, (0 split)
      enuAcc_u < -11.89433  to the left,  agree=0.608, adj=0.002, (0 split)

Node number 2: 12810 observations
  predicted class=OBJECT_CLASS_AIR_VEHICLE  expected loss=0.1555035  P(node) =0.392355
    class counts: 10818  1992
```

```
   probabilities: 0.844 0.156

Node number 3: 19839 observations
  predicted class=OBJECT_CLASS_BIRD         expected loss=0.08417763  P(node) =0.607645
    class counts:  1670 18169
   probabilities: 0.084 0.916
```

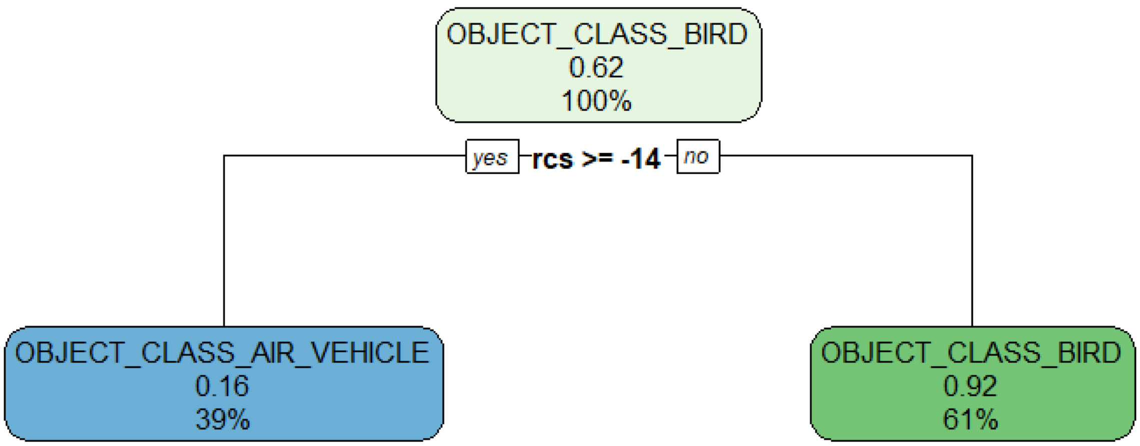


Figure A.7. Using *rpart* in R, the *RandSamp* CTS CART tree diagram is nearly identical to the *Samp100* tree diagram (Figure A.3), reinforcing the importance of RCS in discriminating birds from drones.

In our comparison of the *Samp100* NNSS CART model summary (Appendix A.1.3) with the *RandSamp* model summary (below), we can see a slight increase in the relative importance of RCS with respect to altitude and the velocity components. Otherwise, we still see the same relative importance of altitude and the northern velocity component in both the *Samp100* and *RandSamp* NNSS CART models.

NNSS - RandSamp (CART Model Summary)

```
          CP nsplit rel error    xerror        xstd
1 0.72900255      0 1.0000000 1.0000000 0.005974547
2 0.03811659      1 0.2709975 0.2714822 0.003823091
3 0.02742092      2 0.2328809 0.2331839 0.003574426
4 0.01672525      4 0.1780390 0.1786450 0.003167158
5 0.01000000      5 0.1613138 0.1627076 0.003033246

Variable importance
     rcs   altAgl enuVel_u enuVel_n
      58       37        3        1

Node number 1: 40155 observations,    complexity param=0.7290025
  predicted class=OBJECT_CLASS_AIR_VEHICLE  expected loss=0.4109575  P(node) =1
    class counts: 23653 16502
   probabilities: 0.589 0.411
  left son=2 (21691 obs) right son=3 (18464 obs)
  Primary splits:
      rcs      < -16.20122  to the right, improve=11762.9800, (0 missing)
      altAgl   < 238.409    to the left,  improve= 7930.2960, (0 missing)
      enuVel_n < -12.54041  to the left,  improve=  892.1239, (0 missing)
      enuAcc_u < -0.3179844 to the left,  improve=  388.1491, (0 missing)
      enuVel_u < 1.265286   to the left,  improve=  371.0078, (0 missing)
  Surrogate splits:
      altAgl   < 168.9208   to the left,  agree=0.744, adj=0.443, (0 split)
      enuVel_u < -1.314294  to the right, agree=0.563, adj=0.050, (0 split)
      enuAcc_e < 13.8791    to the left,  agree=0.541, adj=0.003, (0 split)
      enuAcc_n < 22.67206   to the left,  agree=0.541, adj=0.002, (0 split)
      enuVel_e < -42.7362   to the right, agree=0.540, adj=0.000, (0 split)

Node number 2: 21691 observations,    complexity param=0.03811659
  predicted class=OBJECT_CLASS_AIR_VEHICLE  expected loss=0.0578581  P(node) =0.5401818
    class counts: 20436  1255
   probabilities: 0.942 0.058
```

```
  left son=4 (21050 obs) right son=5 (641 obs)
  Primary splits:
      altAgl   < 390.3699   to the left,  improve=1149.41100, (0 missing)
      rcs      < -15.53665  to the right, improve= 440.28230, (0 missing)
      enuVel_e < -3.468423  to the right, improve=  31.77806, (0 missing)
      enuVel_u < 1.065537   to the left,  improve=  24.04432, (0 missing)
      enuVel_n < 7.65508    to the right, improve=  23.29457, (0 missing)
  Surrogate splits:
      enuVel_u < 83.60214   to the left,  agree=0.971, adj=0.003, (0 split)

Node number 3: 18464 observations,    complexity param=0.02742092
  predicted class=OBJECT_CLASS_BIRD         expected loss=0.1742309  P(node) =0.4598182
    class counts:  3217 15247
   probabilities: 0.174 0.826
  left son=6 (4526 obs) right son=7 (13938 obs)
  Primary splits:
      altAgl   < 138.9378   to the left,  improve=1337.26400, (0 missing)
      rcs      < -17.89272  to the right, improve= 675.12410, (0 missing)
      enuVel_n < -12.54558  to the left,  improve= 580.16880, (0 missing)
      enuVel_u < -0.8404117 to the right, improve=  92.46351, (0 missing)
      enuAcc_n < 0.7257759  to the right, improve=  90.01032, (0 missing)
  Surrogate splits:
      enuVel_n < -14.28176  to the left,  agree=0.760, adj=0.022, (0 split)
      enuVel_u < -11.97032  to the left,  agree=0.756, adj=0.003, (0 split)
      enuVel_e < -24.5335   to the left,  agree=0.755, adj=0.002, (0 split)
      enuAcc_n < -123.3815  to the left,  agree=0.755, adj=0.000, (0 split)
      enuAcc_u < 59.63613   to the right, agree=0.755, adj=0.000, (0 split)

Node number 4: 21050 observations
  predicted class=OBJECT_CLASS_AIR_VEHICLE  expected loss=0.02945368  P(node) =0.5242187
    class counts: 20430   620
   probabilities: 0.971 0.029

Node number 5: 641 observations
  predicted class=OBJECT_CLASS_BIRD         expected loss=0.009360374  P(node) =0.01596314
```

```
   class counts:     6   635
  probabilities: 0.009 0.991

Node number 6: 4526 observations,    complexity param=0.02742092
  predicted class=OBJECT_CLASS_AIR_VEHICLE  expected loss=0.491825  P(node) =0.1127132
    class counts:  2300  2226
   probabilities: 0.508 0.492
  left son=12 (1685 obs) right son=13 (2841 obs)
  Primary splits:
      rcs      < -18.10796  to the right, improve=363.96400, (0 missing)
      enuVel_n < -5.201592  to the left,  improve=311.70130, (0 missing)
      enuVel_u < -0.6997    to the right, improve=122.23860, (0 missing)
      altAgl   < 103.2503   to the left,  improve=100.91730, (0 missing)
      enuVel_e < 2.471117   to the left,  improve= 96.24622, (0 missing)
  Surrogate splits:
      enuVel_n < -13.16103  to the left,  agree=0.654, adj=0.071, (0 split)
      enuVel_e < -15.21119  to the left,  agree=0.638, adj=0.028, (0 split)
      enuAcc_e < -8.27922   to the left,  agree=0.629, adj=0.004, (0 split)
      enuVel_u < 18.14574   to the right, agree=0.628, adj=0.002, (0 split)
      enuAcc_n < 28.3576    to the right, agree=0.628, adj=0.001, (0 split)

Node number 7: 13938 observations
  predicted class=OBJECT_CLASS_BIRD         expected loss=0.06579136  P(node) =0.347105
    class counts:   917 13021
   probabilities: 0.066 0.934

Node number 12: 1685 observations
  predicted class=OBJECT_CLASS_AIR_VEHICLE  expected loss=0.231454  P(node) =0.0419624
    class counts:  1295   390
   probabilities: 0.769 0.231

Node number 13: 2841 observations,    complexity param=0.01672525
  predicted class=OBJECT_CLASS_BIRD         expected loss=0.3537487  P(node) =0.07075084
    class counts:  1005  1836
   probabilities: 0.354 0.646
```

```
  left son=26 (574 obs) right son=27 (2267 obs)
  Primary splits:
      enuVel_n < -9.18256   to the left,  improve=215.10040, (0 missing)
      enuVel_u < -0.3224806 to the right, improve= 81.52646, (0 missing)
      enuVel_e < 2.471117   to the left,  improve= 66.26289, (0 missing)
      rcs      < -20.60717  to the right, improve= 57.71213, (0 missing)
      altAgl   < 97.59573   to the left,  improve= 57.11493, (0 missing)
  Surrogate splits:
      enuAcc_n < -62.6169   to the left,  agree=0.799, adj=0.005, (0 split)
      enuAcc_u < -35.99866  to the left,  agree=0.799, adj=0.005, (0 split)
      enuAcc_e < 58.34052   to the right, agree=0.799, adj=0.003, (0 split)
      altAgl   < 138.7773   to the right, agree=0.798, adj=0.002, (0 split)
      rcs      < -18.11798  to the right, agree=0.798, adj=0.002, (0 split)

Node number 26: 574 observations
  predicted class=OBJECT_CLASS_AIR_VEHICLE  expected loss=0.2595819  P(node) =0.01429461
    class counts:   425   149
   probabilities: 0.740 0.260

Node number 27: 2267 observations
  predicted class=OBJECT_CLASS_BIRD         expected loss=0.2558447  P(node) =0.05645623
    class counts:   580  1687
   probabilities: 0.256 0.744
```

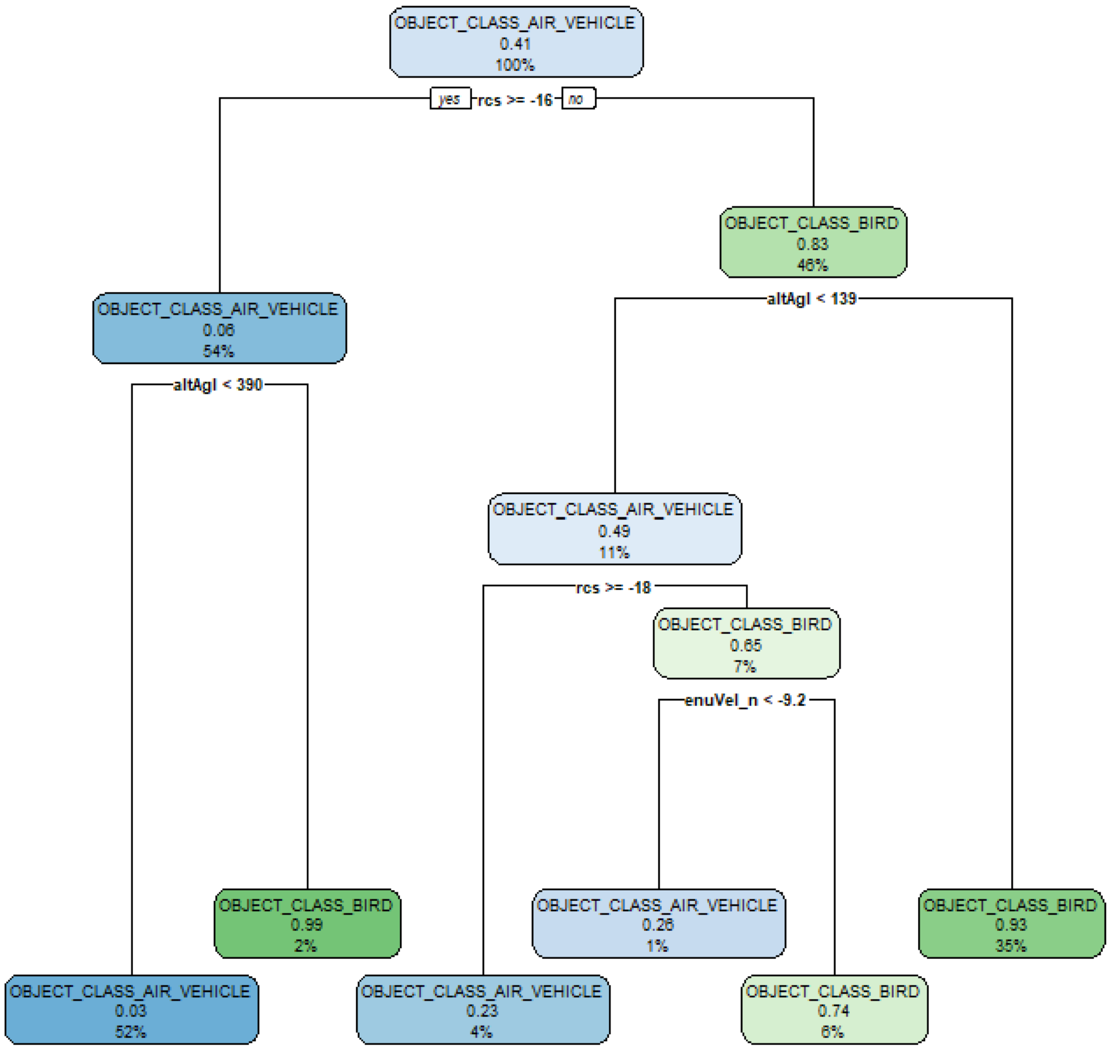


Figure A.8. Similar to the *Samp100* NNSS CART model, the *RandSamp* NNSS CART model primarily discriminates birds from drones based on a combination of RCS, altitude, and the northern velocity component. Although this tree diagram agrees with the first split in the *Samp100* tree diagram (Figure A.4), its subsequent splits in altitude, RCS and the northern velocity component are different and ultimately produced a better performing CART model for the NNSS dataset.

A.2.4 Random Forest Models with Cluster Group Feature

In addition to achieving improved performance in both our CTS and NNSS base random forest models using the *RandSamp* dataset, we observed an additional improved performance after adding a ninth cluster group categorical feature to the base random forest models. Using the cluster group assignments from the k-means algorithms for two, three, and four clusters, we saw marginal improvements in prediction accuracy. Figure A.9 shows the respective confusion matrices for the optimal *RandSamp* CTS and NNSS random forests models with cluster groups and their respective base random forest models.

Random Forest with Cluster Groups

CTS - RandSamp			
Accuracy: 0.9748		Actual Class	
		Drone	Bird
Predicted Class	Drone	3023	65
	Bird	141	4934

CTS - RandSamp (2 Clusters)			
Accuracy: 0.9778		Actual Class	
		Drone	Bird
Predicted Class	Drone	3040	57
	Bird	124	4942

NNSS - RandSamp			
Accuracy: 0.9895		Actual Class	
		Drone	Bird
Predicted Class	Drone	5889	63
	Bird	42	4045

NNSS - RandSamp (4 Clusters)			
Accuracy: 0.9922		Actual Class	
		Drone	Bird
Predicted Class	Drone	5903	50
	Bird	28	4058

Figure A.9. Although we did not observe any improved performance in our *Samp100* random forests models when we included a cluster group categorical feature, our random forests *RandSamp* CTS and NNSS models achieved improved prediction performance by adding a cluster group feature from our optimal k-means algorithms in the first phase of the second model development iteration.

A.3 Training Environment - Prediction Accuracy

Although we ultimately sought to improve the performance of our respective *RandSamp* models on the alternate environment dataset, we also observed an improved performance (measured by balanced accuracy) by all four of our *RandSamp* NNSS models and two of the four *RandSamp* CTS models in the training environment (Figure A.10). Our highest performing model for both our CTS and NNSS training environments were our random forest models, achieving 97.2% and 99.0% balanced accuracy respectively. The respective random forest models also improved the most (in terms of balanced accuracy percentage) between the first and second iterations. It is also noteworthy that, in the case of all four model types and in the use of both the *Samp100* and *RandSamp* datasets, the balanced accuracy rates against the NNSS training environment dataset were 2–6% better than against the CTS training environment dataset. We discuss the impact of the differences in the two environments on the model development in Section 5.2.2.

	Training Environment (Prediction Accuracy)											
	CTS Samp100			CTS RandSamp			NNSS Samp100			NNSS RandSamp		
	Balanced Accuracy	Sensitivity	Specificity	Balanced Accuracy	Sensitivity	Specificity	Balanced Accuracy	Sensitivity	Specificity	Balanced Accuracy	Sensitivity	Specificity
Logistic Regression	0.8774	0.8343	0.9206	0.8755	0.8336	0.9173	0.9181	0.9362	0.8999	0.9186	0.9357	0.9015
CART	0.8839	0.8594	0.9084	0.8850	0.8688	0.9012	0.9306	0.9320	0.9291	0.9328	0.9347	0.9309
Random Forests	0.9425	0.9101	0.9749	0.9720	0.9554	0.9886	0.9789	0.9862	0.9716	0.9903	0.9939	0.9867
AdaBoost	0.9415	0.9123	0.9706	0.9222	0.8778	0.9665	0.9836	0.9885	0.9787	0.9856	0.9905	0.9807

Figure A.10. Our use of balanced accuracy allowed us to not only compare the respective CTS and NNSS models between those developed using the *Samp100* and *RandSamp* datasets (higher balanced accuracy highlighted) but to also compare the models between the training sites despite the differences in the ratios of bird and drone track data between the environments.

A.4 Alternate Environment - Prediction Accuracy

In our evaluation of the comparative performance (in terms of balanced accuracy) between the respective CTS and NNSS models developed using the *Samp100* and *RandSamp* datasets, we observed improved performance (highlighted in yellow) in all four of our CTS models and three out of the four NNSS models (Figure A.11). We only observed a decrease in balanced accuracy with the NNSS CART model using the *RandSamp* dataset. Although all of our models had balanced accuracy rates below 80%, we observed a 1–5% improvement in balanced accuracy among our random forest and AdaBoost models using the *RandSamp* CTS and NNSS datasets. This is significant because it indicated that our models with the greatest propensity to overfit, showed the greatest improvement in balanced accuracy using the *RandSamp* datasets for building their models. Despite the strong improvements by the random Forest and AdaBoost CTS models using the *RandSamp* dataset, the CART and logistic regression models, with balanced accuracy rates of 79.7% and 78.9% respectively, performed the best of the four model types against the alternate environment track data.

	Alternate Environment (Prediction Accuracy)											
	CTS Samp100			CTS RandSamp			NNSS Samp100			NNSS RandSamp		
	Balanced Accuracy	Sensitivity	Specificity	Balanced Accuracy	Sensitivity	Specificity	Balanced Accuracy	Sensitivity	Specificity	Balanced Accuracy	Sensitivity	Specificity
Logistic Regression	0.7874	0.5772	0.9976	0.7890	0.5805	0.9976	0.6657	0.9852	0.3461	0.6713	0.9850	0.3575
CART	0.7930	0.5900	0.9960	0.7970	0.5980	0.9959	0.7054	0.9591	0.4517	0.6661	0.9731	0.3590
Random Forests	0.7363	0.5349	0.9376	0.7599	0.5370	0.9829	0.6894	0.9661	0.4126	0.6986	0.9647	0.4325
AdaBoost	0.7202	0.5653	0.8751	0.7741	0.5687	0.9794	0.7260	0.9586	0.4934	0.7358	0.9563	0.5153

Figure A.11. In our comparison of the performance of our respective CTS and NNSS models on the alternate environment track data, higher balanced accuracy (highlighted in yellow) between those models built using the *Samp100* and *RandSamp* datasets occurred in all four of our model types built using the CTS *RandSamp* dataset and three out of the four model types built using the NNSS *RandSamp* dataset.

List of References

Anduril Industries (2022) Counter UAS. Accessed May 2, 2022, https://www.anduril.com/.

Barrington L (2021) Drone strike on Riyadh oil refinery claimed by Houthis causes fire. Accessed May 2, 2021, https://www.reuters.com/world/middle-east/yemens-houthis-say-drone-attack-hits-saudi-aramco-riyadh-2021-03-19/.

Björklund S (2018) Target detection and classification of small drones by boosting on radar micro-doppler. *2018 15th European Radar Conference (EuRAD)*, 182–185 (IEEE).

Budisteanu EA, Mocanu IG (2021) Combining supervised and unsupervised learning algorithms for human activity recognition. *Sensors* 21(18):6309.

Carcillo F, Le Borgne YA, Caelen O, Kessaci Y, Oblé F, Bontempi G (2021) Combining unsupervised and supervised learning in credit card fraud detection. *Information Sciences* 557:317–331.

Corradino C, Bilotta G, Cappello A, Fortuna L, Del Negro C (2021) Combining radar and optical satellite imagery with machine learning to map lava flows at Mount Etna and Fogo Island. *Energies* 14(1):197.

Dempsey ME, Rasmussen S (2010) Eyes of the army–U.S. Army roadmap for unmanned aircraft systems 2010–2035. *U.S. Army UAS Center of Excellence, Ft. Rucker, Alabama* 9.

Detsch J (2022) Drones have come of age in Russia-Ukraine War. *Foreign Policy* 27.

Dixon R (2020) Azerbaijan's drones owned the battlefield in nagorno-karabakh — and showed future of warfare. *Washington Post* (November 1).

Elsayed M, Reda M, Mashaly AS, Amein AS (2021) Review on real-time drone detection based on visual band electro-optical (eo) sensor. *2021 Tenth International Conference on Intelligent Computing and Information Systems (ICICIS)*, 57–65 (IEEE).

Fry BJ (2004) *Computational information design*. Ph.D. thesis, Massachusetts Institute of Technology.

Ganti SR, Kim Y (2016) Implementation of detection and tracking mechanism for small UAS. *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, 1254–1260 (IEEE).

GlobalSecurity (2021a) Counter-sUAS / mobile force protection. Accessed May 9, 2022, https://www.globalsecurity.org/military/systems/ground/counter-suas.htm.

GlobalSecurity (2021b) Radar cross section (RCS). Accessed May 9, 2022, https://www.globalsecurity.org/military/world/stealth-aircraft-rcs.htm.

Henderson M (2020) Detection and classification of small UAS for threat neutralization. Accessed May 9, 2022, https://dsiac.org/articles/detection-and-classification-of-small-uas-for-threat-neutralization/.

Herrera GJ, Dechant JA, Green E, Klein EA (2017) Technology trends in small unmanned aircraft systems (sUAS) and counter-UAS: A five year outlook. Technical report, Institute for Defense Analyses.

Iddon P (2021) The new drone threat facing Iraqi Kurdistan's Erbil. Forbes. Accessed May 2, 2021, https://www.forbes.com/sites/pauliddon/2021/04/17/the-new-threat-of-drone-attacks-on-iraqi-kurdistans-erbil/?sh=7ad29bcc5c92.

IEEE Robotics and Automation Society (2016) ICUAS 2016 - International Conference on Unmanned Aircraft Systems. Accessed May 5, 2022, https://www.ieee-ras.org/component/rseventspro/event/767-icuas-2016-international-conference-on-unmanned-aircraft-systems.

James G, Witten D, Hastie T, Tibshirani R (2017) *An Introduction to Statistical Learning with Applications in R* (Springer).

Judson J (2021) New strategy wants to counter increasingly complex drone threats. DefenseNews. Accessed May 5, 2021, https://www.defensenews.com/pentagon/2021/01/07/joint-strategy-calls-for-common-architecture-to-counter-increasingly-complex-drone-threats/.

Liang J, Ahmad BI, Jahangir M, Godsill S (2021) Detection of malicious intent in non-cooperative drone surveillance. *2021 Sensor Signal Processing for Defence Conference (SSPD)*, 1–5 (IEEE).

Liu J, Xu QY, Chen WS (2021) Classification of bird and drone targets based on motion characteristics and random forest model using surveillance radar data. *IEEE Access* 9:160135–160144.

Medaiyese OO, Ezuma M, Lauf AP, Guvenc I (2021) Semi-supervised learning framework for uav detection. *2021 IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 1185–1190 (IEEE).

Mendis GJ, Randeny T, Wei J, Madanayake A (2016) Deep learning based doppler radar for micro uas detection and classification. *MILCOM 2016-2016 IEEE Military Communications Conference*, 924–929 (IEEE).

Miller CC (2021) Counter-small unmanned aircraft systems strategy. Technical report, U.S. Department of Defense, Washington, DC, https://apps.dtic.mil/sti/pdfs/AD1127557.pdf.

Poitevin P, Pelletier M, Lamontagne P (2017) Challenges in detecting uas with radar. *2017 International Carnahan Conference on Security Technology (ICCST)*, 1–6 (IEEE).

Samaras S, Diamantidou E, Ataloglou D, Sakellariou N, Vafeiadis A, Magoulianitis V, Lalas A, et al. (2019) Deep learning on multi sensor data for counter uav applications—a systematic review. *Sensors* 19(22):4837.

Scott MB (2021) Army counter-UAS. *Army University Press* (March-April 2021).

Sedivy P, Nemec O (2021) Drone RCS statistical behaviour. *Proceedings of the MSG-SET-183 Specialists' Meeting on "Drone Detectability: Modelling the Relevant Signature", North Atlantic Treaty Organization (NATO), Held Virtually (via WebEx)*, 4–1.

Shin S, Park S, Kim Y, Matson ET (2016) Design and analysis of cost-efficient sensor deployment for tracking small UAS with agent-based modeling. *Sensors* 16(4):575.

Siewert SB, Andalibi M, Bruder S, Rizor S (2019) Slew-to-cue electro-optical and infrared sensor network for small UAS detection, tracking and identification. *AIAA Scitech 2019 Forum*, 2264.

Sim J, Jahangir M, Fioranelli F, Baker CJ, Dale H (2019) Effective ground-truthing of supervised machine learning for drone classification. *2019 International Radar Conference (RADAR)*, 1–5 (IEEE).

Suits D (2020) Joint counter-sUAS strategy to address need for improved technology. Army News Service. Accessed May 3, 2021, https://www.army.mil/article/239593/joint_counter_suas_strategy_to_address_need_for_improved_technology.

Svanström F, Alonso-Fernandez F, Englund C (2022) Drone detection and tracking in real-time by fusion of different sensing modalities. *Drones* 6(11):317.

Wang B (2022) End of the tank era? Next Big Future. Accessed April 28, 2022, https://www.nextbigfuture.com/2022/03/end-of-the-tank-era.html.

Wang J, Liu Y, Song H (2021) Counter-unmanned aircraft system(s) (C-UAS): State of the art, challenges, and future trends. *IEEE Aerospace and Electronic Systems Magazine* 36(3):4–29.

Wilson B, Tierney S, Toland B, Burns RM, Steiner CP, Adams CS, Nixon M, Khan R, Ziegler MD, Osburg J, et al. (2020) *Small Unmanned Aerial System Adversary Capabilities*. Number RR-3023-DHS in 2020-2 (RAND Corporation).

Yasin R, et al. (2016) Countering the counter-UAS shortcomings. Technical report, C4ISR, North Syracuse, NY, https://policycommons.net/artifacts/1686508/editorial-whitepaper/2418153/.

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California