TELEMETRY DATA MINING FOR UNMANNED AIRCRAFT SYSTEMS

THESIS

Li Yu, Captain, USAF

AFIT-ENV-MS-22-M-275

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENV-MS-22-M-275

TELEMETRY DATA MINING FOR UNMANNED AIRCRAFT SYSTEMS

THESIS

Presented to the Faculty

Department of Systems Engineering & Management

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Systems Engineering

Li Yu, BS

Captain, USAF

March 2022

AFIT-ENV-MS-22-M-275

TELEMETRY DATA MINING FOR UNMANNED AIRCRAFT SYSTEMS

Li Yu, BS

Captain, USAF

Committee Membership:

Brent T. Langhals, Ph.D.
Chair

Michael R. Grimaila, Ph.D., CISM, CISSP
Member

Douglas D. Hodson, Ph.D.
Member

AFIT-ENV-MS-22-M-275

Abstract

With ever more data becoming available to the US Air Force, it is vital to develop effective methods to leverage this strategic asset. Machine learning (ML) techniques present a means of meeting this challenge, as these tools have demonstrated successful use in commercial applications. For this research, three ML methods were applied to a unmanned aircraft system (UAS) telemetry dataset with the aim of extracting useful insight related to phases of flight. It was shown that ML provides an advantage in exploratory data analysis and as well as classification of phases. Neural network models demonstrated the best performance with over 90% accuracy in classifying of UAS phases of flight. Categorical and Regression Trees (CART) also performed well, whereas C5.0 is less suited for this task. In addition, several interesting patterns were uncovered within the dataset, which can aid UAS operators in identifying mission anomalies and atypical system operation.

Acknowledgments

I would like to thank Dr. Langhals for showing me the beauty of relational databases. Perhaps more importantly, I am grateful for his support and guidance on data mining in general and on my thesis research in particular. I also want to thank Major Engle for crucial aid that allowed me to access the necessary data. Furthermore, I would like to thank everyone who previously worked on the Reaper Analysis Toolkit, who had done most of the hard work for me.

Li Yu

Table of Contents

List of Figures

Page

List of Tables

Page

TELEMETRY DATA MINING FOR UNMANNED AIRCRAFT SYSTEMS

I. Introduction

The rate of knowledge acquisition and the speed of decision-making processes may well determine the outcome of future armed conflicts. Although the Air Force has invested in advanced data analytics to support timely decision making, much work remains to be done. This research is intended to aid efforts in this area by exploring novel analysis methods on a relevant Air Force dataset. The desired outcome is an improved capability in data analytics for problems of immediate relevance to the USAF.

Data Analytics and Aircraft Maintenance

For today's Air Force, data is a strategic asset. However, the Air Force falls short in leveraging this asset (Geiger, 2017; Hamilton and Kreuzer, 2018). As the quantity, speed, and complexity of data continues to grow, so does its strategic value, and it becomes vitally important to develop the right tools and methods to harness the power of this data. At the 2018 Air Force Information Technology and Cyberpower Conference, Air Force Chief Data Officer Eileen Vidrine presented the following vision:

> The Air Force is a data-driven organization—one that purposely collects, creates, shares, and acts upon quality, authoritative data in and across mission areas and domains—empowering Airmen and the machines they rely upon with timely access to the data needed to enable advanced analytics, and accelerate decisive action. (Vidrine, 2018)

In the same presentation, Ms. Vidrine also summarized the five Air Force data goals: visible, accessible, understandable, linked, and trustworthy. In particular, the fourth goal

"linked" means linking data to gain new insights. This aim is echoed in the 2018 DoD Digital Engineering Strategy, which states, "DoD's vision is to build an enterprise capability that securely leverages data and analytics to enable insights and achieve faster and better data-driven decisions" (OUSD, 2018). Data analytics is powerful precisely because it converts raw data into insights and directly supports decision making.

One area that benefits from data-driven decisions is that of aircraft maintenance. As shown in Figure 1, maintenance activities can be divided into two broad categories: preventive maintenance, which is carried out at predetermined intervals or according to prescribed criteria, and corrective maintenance, which is carried out after a defect is found (BSI, 2010:13). For most aircraft, preventive maintenance plays a vital role due to the severe consequences of a system failure. One industry-standard maintenance strategy is known as MSG-3, and it employs a reliability approach to produce a scheduled maintenance program, where the maintenance intervals are optimized based on reliability data (ATA, 2002).

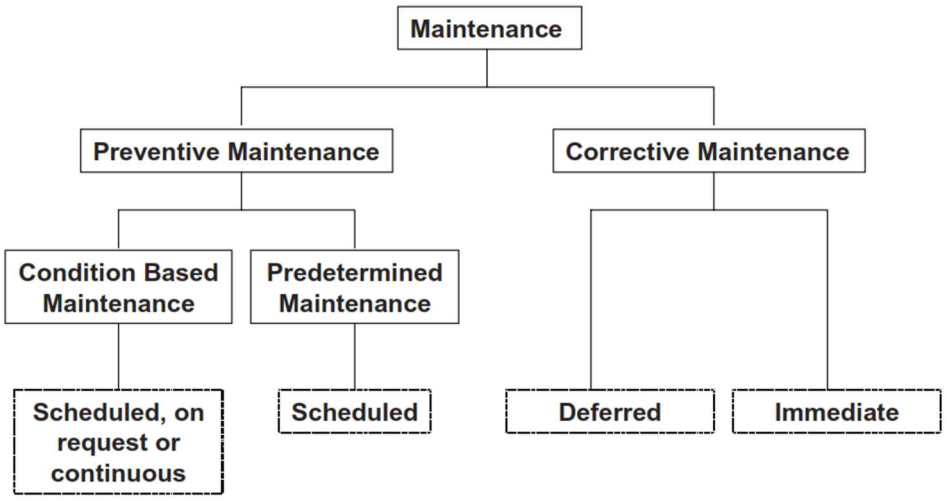


Figure 1. General categories of maintenance (BSI, 2010:20)

Although MSG-3 has worked well, it can be improved with the addition of predictive maintenance, which is possible with advanced data analytics. Predictive maintenance is a type of condition based maintenance, and it relies on continuous monitoring of system performance and forecasted failures (BSI, 2010:12). In effect, maintenance is no longer constrained to fixed intervals but adapts to known data such as sensor measurements and failure history. A recent systematic mapping of 155 predictive maintenance studies across a range of industries revealed increasing investment in predictive maintenance over the past 10 years (Ay Ture et al., 2021). In addition, these predictive efforts, many of which were based on machine learning, have generally provided functionality and financial gain to the relevant domains.

Machine Learning

Although datasets found in the Air Force have gotten larger, basic statistical techniques have not changed. Simple descriptive statistics continue to be the starting point for most types of analysis. Measures such as mean, median, range, and quartiles give basic information about a variable, and the histogram provides visualization of its distribution. With slightly more computational power, hypothesis testing and regression can offer the analyst with more useful information, which is sufficient to answer many questions of interest.

With very large datasets, however, it may be difficult to uncover hidden features and to gain new insights, or it may not be cost-effective to conduct a thorough statistical analysis. One common data processing task is classification: records of a dataset are sorted into one of several discrete categories, which typically represent high-level

features. For example, it may be desirable to classify radar readings as aircraft, missile, or something else. One statistical technique suitable for classification is multinomial logistic regression; however, it is adversely affected by multicollinearity (strong correlations) among predictor variables. On the other hand, several machine learning tools are available for classification tasks, which can be easier to use and offer better performance compared to multinomial logistic regression, especially for complex datasets with many variables (Levy and O'Malley, 2020).

Machine learning (ML) refers to the use of computer algorithms to accomplish various data analysis tasks after learning from examples. Put in a different way, the algorithm is trained on a set of known data, then the algorithm proceeds with the required task on other data. ML represents a sub-field of artificial intelligence. Two common classes of ML methods are decision trees and neural networks, both of which can be used for classification. A decision tree can be visualized as branches propagating from a root node, and each leaf would carry a classification: such a decision tree method processes data in a sequence of logical steps (Michie et al., 1994:2). A neural network consists of layers of interconnected nodes, where each node generates a signal based on the combination of inputs, and classification probabilities are read from the output nodes (Michie et al., 1994:84). A neural network can capture highly nonlinear behavior, but one downside is that a neural network model does not allow for simple interpretation.

In recent years, ML methods have demonstrated success over a wide range of applications, including cybersecurity and medicine (Shaukat et al., 2020; Li et al., 2020). Specifically, in the domain of autonomous flight, there is ongoing work using ML for

obstacle detection and collision avoidance, as part of the effort to build fully autonomous unmanned aircraft systems (Fraga-Lamas et al., 2019). Since ML methods are not domain specific, they are very relevant for Air Force data analytics problems. The research presented here is an attempt to improve understanding of ML and how it can best be used within the Air Force. In particular, as unmanned aircraft systems (UAS) have become a major capability, this was selected as the area of focus for this research.

Phases of Flight

The previous discussion covered data analytics and machine learning in general. The current research specifically focuses on data analysis of flight telemetry from a large-scale UAS. This type of time series telemetry is representative of data acquired from Air Force weapons systems. As USAF fields ever more sophisticated aircraft, there is an ever greater need for advanced analytics on such data.

For military UAS such as the MQ-9, flight telemetry takes the form of sequential time-stamped data with numerous variables similar to traditional human-occupied aircraft. The variables include various readings related to general flight, such as altitude, airspeed, angle of attack, as well as sensor readings from several subsystems. Significant subsystems include engines, electrical, communications. The position of aircraft control surfaces may be grouped with general flight parameters, or in a separate group as a collection of servo readings.

As previously stated, data analysis demonstrates value when it allows for extraction of new insights. In other words, effective data analysis should transform raw data into high level information, which should then be applied toward decision making.

Phase of flight represent a natural grouping for time-series flight telemetry. Typical phases include taxi, takeoff, climb, cruise, descent, approach, and landing, which are common descriptors for aviation states. However, since phase labels are not normally part of flight telemetry, they represent a higher level of meta-classification that adds value to the raw data. Therefore effective phase classification is sufficient to demonstrate the addition of value in the form of knowledge that is relevant to decision-making.

Additional insights can be generated once the phase of flight descriptor is added to a given set of telemetry. Phases are highly relevant for understanding flight risks, as there are distinct accident scenarios during each phase (Chidester, 2003). Analysis of flight data by phase enables identification of exceedances as well as atypical conditions. Phase labels also allow Air Force operators and maintainers to easily narrow the search space during post-mission analysis. This could aid in various tasks such as diagnosing system performance, visualizing trends, discovering inefficiencies, or understanding specific mission events.

Research Objectives

The general motivation of this research is to leverage ML to aid in predictive maintenance of Air Force systems. It is hoped that this effort may promote more effective data analytics in various Air Force domains. More specifically, the research is focused on phase of flight classifications on UAS telemetry data. In this way, specific insights obtained through the analysis may provide immediate value for UAS operators.

The research objectives can be stated in terms of two following data mining tasks.

- Task one: apply ML methods to aid exploratory data analysis (EDA) related to phases of flight, where acquisition of new data insights indicates success
- Task two: apply ML methods to build classification models for phases of flight, where accurate classifications indicate success

For each task, the success condition signifies potential applications of specific ML methods for Air Force data analytics.

In this chapter, high-level motivation was given for research in the subsequent chapters. Data analytics is useful when it produces new insights, therefore telemetry data mining is useful when it produces high-level information, such as phase of flight classifications. Improved understanding of ML through this task may aid AF data analytics in general. The remainder of the document is laid out as follows: Chapter 2 provides proper context for both UAS and ML methods, Chapter 3 describes the telemetry dataset and the pre-classification process, Chapter 4 delivers the data mining results, and Chapter 5 gives a discussion with respect to the relevance of acquired insights.

II. Background on ML and UAS

Broadly, machine learning (ML) refers to a class of computer algorithms that can generalize from experience. "Machine learning algorithms build a model based on sample data, known as training data, in order to make predictions or decisions without being explicitly programmed to do so" ("Machine Learning," 2022). ML algorithms are able to adapt to data and to work on unseen samples, which make such methods highly versatile. In addition, ML has demonstrated success in applications where conventional algorithms proved inadequate. The first use of the term "machine learning" is attributed to Arthur Samuel, who developed an early self-learning checkers program (Samuel, 1959). In recent years, a dramatic increase in available data has unleashed the power of ML. Today ML methods are heavily used in web recommender systems, speech recognition, autonomous vehicles, among many other domains.

This chapter provides on overview of unmanned aircraft systems (UAS), followed by the discussion of previous ML work related to aircraft maintenance. Three ML methods are then explained in detail, including Categorical and Regression Trees, C5.0, and neural network.

Unmanned Aircraft Systems

UAS have seen significant advancement and growing applications in recent years. UAS are commonly referred to as unmanned aerial vehicles (UAV), but here the term UAS is preferred as it accounts for a closely related system of systems of which the air vehicle is the most prominent part. Components of a full system are shown in Figure 2.

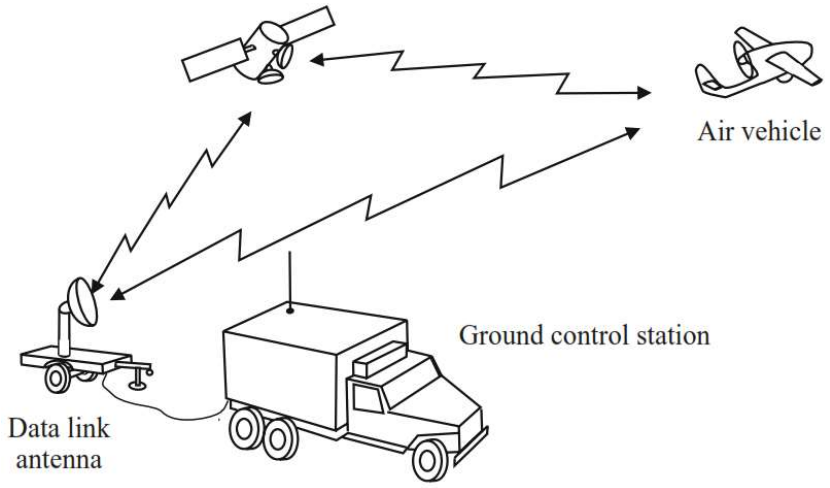


Figure 2. Generic UAS configuration (Fahlstrom and Gleason, 2012:8)

Typical UAS consist of air vehicles, payloads, ground control stations (GCS), data links, and other ground support equipment (Falhstrom and Gleason, 2012:8). Similar to traditional aircraft, the air vehicle consists of several subsystems including airframe, propulsion, flight control, electrical power system, etc. However, an air vehicle that constitutes part of an UAS has no human pilot on board and is further differentiated from missiles by its reusability (Falhstrom and Gleason, 2012:29). Payloads can be seen as distinct from the air vehicle, as they are typically interchangeable from one vehicle to the next. An UAS may carry such payloads as surveillance packages, electronic warfare systems, or weapons. The GCS consists of a variety of equipment necessary for remote operations, and it can range in size from a laptop computer to a large permanent structure (Falhstrom and Gleason, 2012:9). Human operators may also be considered part of the GCS. Next, the data link is the primary data interface between the air vehicle and the GCS. This includes the air data terminal on the air vehicle, the ground terminal connected

to the GCS, as well as intermediaries such as communication satellites (Falhstrom and Gleason, 2012:10). Finally, ground support equipment such as maintenance equipment, spares, fueling, and transportation are necessary for any complex GCS. In a sense, hardware and software used for post-mission data analytics may be considered part of ground support and thus part of the total UAS.

The 2009 USAF Unmanned Aircraft Systems Flight Plan explicitly refers to "a family of unmanned aircraft consisting of small man-portable vehicles, including micro and nano-sized vehicles, medium 'fighter sized' vehicles, large 'tanker sized' vehicles, and special vehicles with unique capabilities, all including autonomous-capable operations" (Department, 2009:3). The USAF has used UAS extensively for a variety of tasks including intelligence/ surveillance/ reconnaissance (ISR), close air support, combat search and rescue, precision strike, overwatch, and more. For example, the MQ-9 Reaper (Figure 3) is an armed, multi-role, long endurance UAS which supports a variety of

Figure 3. MQ-9 Reaper launching an AGM-114 Hellfire missile

payloads. It has a maximum altitude of 50,000 ft, a maximum airspeed of 240 knots, and an endurance of 18 hours. The Reaper is flown by an USAF pilot at the GCS, with options for semi-autonomous and pre-programmed flight. It can be controlled by line of sight within 100 miles of base, or beyond line of sight via satellite datalinks (Department, 2009:27).

ML Methods for Aircraft Maintenance

For manned aircraft, sustainment costs constitute a significant portion of the total system lifecycle cost, and the same holds true for UAS. The annual maintenance and operating cost for the MQ-9 has been estimated at $5.1 million in 2012 dollars, compared to unit acquisition cost of $30.2 million (Wheeler, 2012). Thus sustainment costs dominate total spending with the expected 20-year service life. And if UAS lifespan is extended, as is typical of USAF aircraft, maintenance costs can be expected to grow in a nonlinear manner.

Indeed, the high cost of maintenance poses a significant challenge for the aerospace industry in general. Given the large quantity of data available today, there has been significant interest in applying ML methods for predictive maintenance, which is reflected by a substantial quantity of literature on this topic. Notably, an Airbus sponsored survey paper identified a wide array of ML tools with direct relevance for aircraft maintenance, including both supervised methods and unsupervised methods (Adhikari et al., 2018). These algorithms are shown in Figure 4. The paper then presented an ML-based diagnostics and prognostics framework incorporating a host of methods. In this context, supervised methods, such as decision trees and neural networks, are applied

to datasets with a defined target variable, which is known in a training dataset. Unsupervised methods, such as k-means clustering and self-organizing maps, can be applied with no predefined target variable, but additional analysis is usually required to extract desired data patterns. This research employs supervised methods, and phase of flight labels represent the target variable.

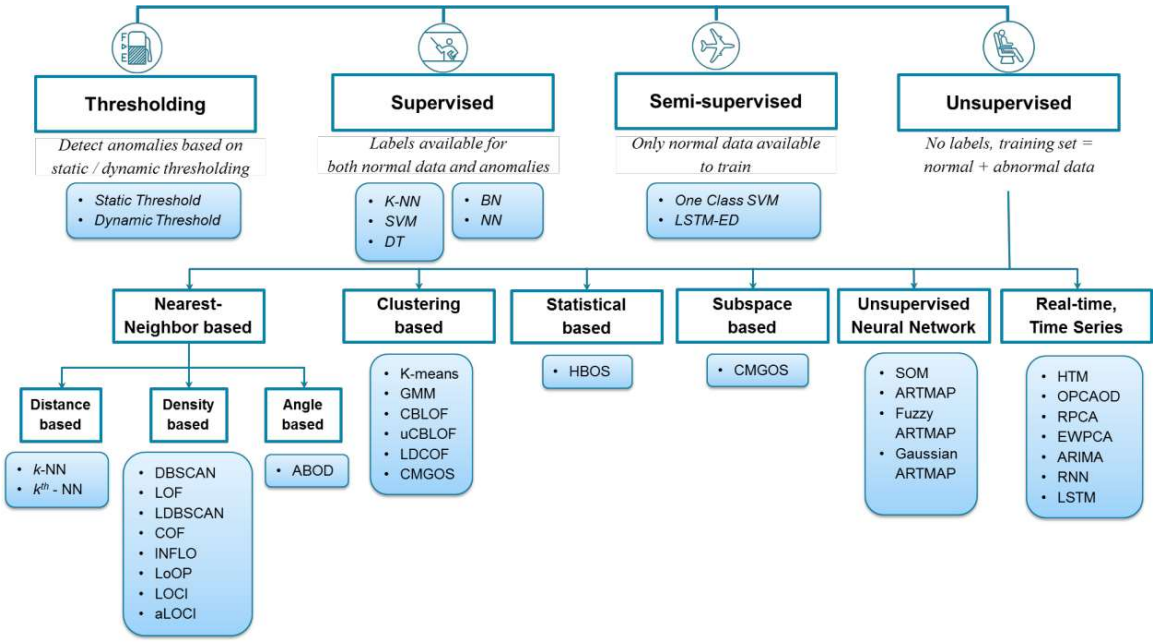


Figure 4. Overview of ML tools applicable to maintenance (Adhikari et al., 2018)

A review of literature related to aircraft maintenance shows that neural network, also called artificial neural network (ANN), is the most commonly used ML method in this domain. Early work conducted by Trani et al. (2004) applied ANN for prediction of aircraft fuel consumption for climb, cruise, and descent conditions. This ML model was able to achieve lower error when compared to Eurocontrol's Base Data of Aircraft model, a widely-used analytical model designed for the same task. Later work by Kozik and Sep (2012) employed ANN to forecast parts demand during helicopter engine overhaul; in

this case, ML performed significantly better than the existing process which was a moving averages method. More recently, research was conducted by Altay et al. (2014) to predict time to failure using data from 60 commercial aircraft. ANNs were built using both standard back propagation and back propagation aided by a genetic algorithm. The model variants produced similarly satisfactory results, achieving near 90% correlation with test data.

A recent conference paper by Tarik and Jebari (2020) provided a summary of 13 ML papers on maintenance problems outside aviation. Among ML methods surveyed, support vector machine (SVM) was most common, follow by decision tree and random forest, then neural network. SVM entails the construction of a hyperplane to differentiate between two classes of known data. Random forest can be seen as an enhanced decision tree algorithm that rely on several decision trees instead of one. Beyond the survey, Tarik and Jebari applied three ML methods for failure prediction using an aircraft engine dataset. SVM demonstrated the best results with 91% accuracy, followed by decision tree at 88% and neural network at 82%.

Two other published works bear relevance to the current research. The PhD dissertation by Korvesis (2017) analyzed three types of aircraft data, including post-flight reports, logbook data, and sensor measurements. A variety of methods were applied with the aim of modeling component degradation. Specifically, for sensor measurements, Korvesis applied a novel algorithm to fit a Gaussian mixture model, which generated risk estimates. Unfortunately, the model was only validated on synthetic data, as true risk values were unavailable. A separate paper by Carson et al. (2020) carries more direct

significance. This paper described the creation of a hybrid decision tree-neural network model using a Boeing 737 maintenance dataset. In this hybrid method, a decision tree serves to classify the type of defect, and several neural networks serve as split criteria at each decision node. Alternative activation functions were considered, and the best was the tanh activation function with approximately 93% accuracy. In addition, it was shown that model performance using all 72 input variables was similar to using just the top 11 weighted variables.

From the previous, it is seen that existing literature on ML and aircraft maintenance emphasized failure prediction. Neural network methods were frequently applied but infrequently tested against other methods. Phase of flight was only mentioned by Trani in connection to fuel consumption; also it was not applicable to many of the aviation datasets used in previous research. Therefore, this research opens an unexplored area by classifying phase of flight instead of predicting failure, which may yield new insights to support operation and maintenance activities. The current work is also distinct in the use of data from an UAS instead of conventional manned aircrafts.

This data mining effort employs two decision tree methods and a neural network method. Neural network was chosen based on its frequent use in existing literature. Although decision tree methods were less frequently used for maintenance problems, they offer advantages of simplicity and ease of interpretation. Moreover, two decision tree variants (CART and C5.0) are considered due to their distinct implementations.

Classification and Regression Trees

One of the earliest and most well-known ML methods is Classification and Regression Trees (CART), which was developed by researchers at UC Berkeley and Stanford (Breiman et al., 1984). As it is a decision tree algorithm, it operates on the principle of recursive partitioning of a known dataset. The partitioning process can be visualized as branches forming at a series of decision nodes, starting at the root node and ending at the leaf nodes. For classification tasks, each leaf node would be associated with a class or category. The completed decision tree then represents a decision model that can be used to classify new data.

At each decision node, CART splits the data into exactly two pieces, which populate the left and right child nodes. The optimal split is determined by the contrast in class proportions. Stated more precisely, the optimal split is one that maximizes the goodness metric Φ_s (equation adapted from Larose and Larose, 2015).

$$\Phi_s = 2P_L P_R \sum_j |P(j|t_L) - P(j|t_R)| \quad (1)$$

In this equation, P_L is the proportion of records in the left child node, and P_R is the proportion in the right child node. The index j refers to a particular class, and the summation is performed over all classes, which are the classifications of interest. $P(j|t_L)$ is the proportion of class j records within the left child node, and $P(j|t_R)$ the proportion in the right. Therefore, the difference represents contrast in class proportions.

CART can continue to perform binary partitions until no further splits are possible, which occurs upon reaching a pure node, or when predictor variables become

indistinguishable. However, such fully expanded trees can be quite large, and may have overfitted the data. Therefore, CART also performs pruning, where subtrees are removed and replaced with a single leaf node. Specifically, CART employs minimal cost-complexity pruning, which seeks to reduce both classification errors and number of leaf nodes (Breiman et al., 1984:71-75).

CART is a widely used ML algorithm, and it is well suited for classification tasks. Due to its simplicity, it is faster than many other ML methods. Decision trees produced by CART also offer ease of interpretation. Each split in a decision tree indicates a possible break point for a predictor variable, and could provide immediately useful insight.

C5.0

C5.0 is also a decision tree algorithm. It is an extension of C4.5 presented by Quinlan (1993), and it offers improved efficiency and additional functionality. However, basic splitting and pruning processes remain the same. Whereas CART performs binary splits based on class contrast, C5.0 can perform multiple splits via the concept of entropy reduction. Stated more precisely, the optimal split is one that minimizes the entropy metric H_s (equation adapted from Larose and Larose, 2015).

$$H_s = -\sum_i \sum_j p_i p_{ij} \log_2(p_{ij}) \quad (2)$$

In this equation, index i refers to a child node or partition, and index j refers to a class. Therefore the double summation sums over all classes at a partition, then over all partitions. The term p_i is the proportion of records placed in partition i, and p_{ij} is the

proportion of class j records within partition i. If the proportion of classes are unchanged after partitioning, then entropy is also unchanged. If each partition contains just a single class, then entropy has been reduced to 0.

Similar to CART, C5.0 also relies on pruning to avoid overfitting. Both algorithms initially create an oversized tree, then prunes subtrees to improve a given metric. In the case of C5.0, it seeks to minimize predicted error rate, which is based on confidence limits for a binomial distribution (Quinlan, 1993:37-41). Unlike CART, C5.0 does not explicitly take into account complexity (number of leaf nodes) during pruning, therefore C5.0 decision trees can be expected to be somewhat larger. C5.0 decision trees also have a wider shape due to multiple splits.

C5.0 is well suited for classification tasks and since it is still a fairly simple ML algorithm, it is also a fast method. Because C5.0 produces a decision tree model, the results are easy to interpret by a human.

Neural Network

Neural networks represent a type of ML algorithm that is quite different from decision tree methods. Although there are many implementations of neural networks, they are generally based on work published by B. D. Ripley (1996). The neural network model typically consists of several interconnected layers, including an input layer, an output layer, and one or more hidden layers (Figure 5). Each layer consists of one or more nodes. Nodes in the input layer take on values of predictor variables, and nodes in the output layer hold estimates for target variables. Nodes in hidden layers takes in the sum of input values, then produces an output by means of an activation function. The

sigmoid activation function is commonly used, as it mimics nonlinear behavior of biological neurons (Larose and Larose, 2015).

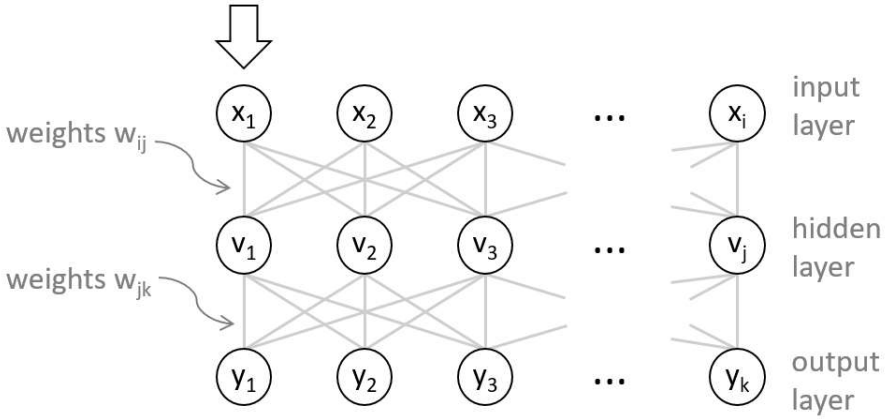


Figure 5. Neural network layout with one hidden layer

In a neural network, each node in a particular layer is connected to every node in adjacent layers. Predictor variables enter the model at the input layer, which pass signals to the hidden layers, then to the output layer where results are read. A weight is associated with each connection, and the basic goal of this ML method is to find the optimal weights, of which there can be many. The initial set of weights may be pre-specified, or generated at random. Then the algorithm runs an optimization procedure to minimize the error of model outputs; in other words, the weight values are adjusted to best fit the known data. The process of adjusting weight values based output errors is referred to as back-propagation. Various numerical optimization procedures can be applied here, such as gradient descent, or Nelder-Mead optimization (downhill simplex), which is the default in the R language.

Neural networks are effective in modeling highly nonlinear behavior, and they have been shown to be robust against noise or errors in training data (Michie et al., 1994:3). However, since the weight vector grows rapidly with additional nodes, neural network algorithms can be slow, so a prespecified number of iterations may be used as the termination criterion. Compared to decision trees, neural network models tend to be opaque to human interpretation, as individual weights do not have particular significance. Since the output nodes hold continuous values, neural networks are suited for estimation problems, but they can be adapted for classification as well.

Validation and Conclusion

ML methods learn from examples; that is, all ML methods require training data where the target variable is known. ML models are built from the training data, and once built, these models are capable of the desired task such as classification. In order to validate model performance, it should also be tested on known data. This test data, also referred to as holdout data, is distinct from the training data, but the target variable is known and can be compared with model outputs.

Typically, a known dataset is randomly partitioned into two parts, one for training and one for testing. However, model performance may vary depending on the split. To account for this variation, the random partitioning may be repeated several times. Each time the model is trained on a different set of data then tested on the remaining data. The aggregate performance is then taken to represent the model performance. This approach is referred to as k-fold validation, and it is used when evaluating ML classification performance in the later sections.

In this chapter, back background was provided for both ML and UAS. Current literature on ML and aircraft maintenance was examined to substantiate subsequent research. The three ML methods used for this research are CART, C5.0, and neural network, as implemented using the R language. The next chapter discusses details of the telemetry dataset and the assignment of phase of flight labels via a pre-classification algorithm.

III. Dataset Description and Phase Pre-Classification

This chapter describes the UAS telemetry dataset and the assignment of phases of flight. A human designed pre-classification algorithm is used to label telemetry data with suitable phases, and these are assumed to be true classifications suitable for training ML models. A plan is then presented for the data mining effort.

Data Overview and Previous Database Work

UAS telemetry data used for this research was provided by the 432d Air Expeditionary Wing at Creech AFB. Typical of raw data in the Air Force, the original MQ-9 telemetry files were fragmented and difficult to use. Over the course of a single mission, telemetry was split by multiple ground control stations, then further split into single hour sections. Moreover, the data structure evolved over time with changes to software versions, and there is little to no metadata to aid in variable identification. This made post-mission analysis exceedingly tedious. To tackle the problem, UAS operators sought help from researchers at the Air Force Institute of Technology, which led to the creation of Reaper Analysis Toolkit. This is a set of Python algorithms that provides a number of capabilities, the most significant of which is the automation of extract, transform, and load operations of raw telemetry data into a MySQL relational database. Within this database, telemetry is consolidated and well-organized, and various analysis tasks can be easily accomplished using standard SQL queries.

This research takes advantage of previous efforts and leverages the accessibility of the relational database to perform more extensive data analysis. The database schema consists of 42 tables, including metadata tables such as "mission," "gcs," and "tail,"

subsystem tables with telemetry collected at 1Hz, a high-rate data table "Adata" collected at 20Hz, as well as time reference table "DMtime." (The 20Hz telemetry is outside the scope of this research; only 1Hz telemetry is used for subsequent analysis.) Despite having many tables, the database schema is not complex. All telemetry for a given mission can be considered as a single large table: rows correspond to times of reading, and columns correspond to flight parameters and sensor readings.

This research uses telemetry data from 30 complete missions associated with a single air vehicle (A4033). These missions were flown from years 2010 to 2012, and mission durations range from about 4.5 hours to 12 hours, with a mean of 6 hours and 36 minutes. This provides a total of 713,044 records (1Hz telemetry). 390 data variables were gathered from 15 subsystem tables, as noted in Table 1. The "general performance"

Table 1. Subsystem tables used for data analysis

	SQL Table	Var Count
1	aircraft comm	31
2	aircraft payload	22
3	attitude	5
4	electrical	87
5	engine data	44
6	fuel system	13
7	general performance	57
8	hold modes	19
9	landing gear	25
10	nav autopilot	22
11	overrides	8
12	pilotinput	12
13	servofailures	10
14	servofeedback	21
15	videomux	14

table has the second highest count of variables and contains a number of key attributes, including redundant measures of key flight parameters such as altitude and airspeed. Taken together, the subsystem variables represent nearly all relevant measures captured by the air vehicle.

Phase of Flight Pre-classification

The previously described dataset is assumed to be of very high quality, with few errors and inconsistencies. While some variables contain null records, or are entirely unpopulated, a majority has no missing entries. The dataset has no phase of flight labels, as this state was unknown to the UAS itself. Since ML methods require examples, it is necessary to assign phase labels to the dataset, in effect establishing "ground truth." In principle, this would be accomplished by a domain expert, such as the UAS pilot; however, this resource was not available, especially considering the tedious nature of manual assignments. Instead, a pre-classification algorithm was employed as an efficient surrogate for human judgment.

The pre-classification algorithm is a program written in the R language: it takes a set of complete mission data as input and adds phase label to each record. In other words, pre-classification creates a new variable with one of eight phase of flight labels. The eight phases used in this research are based on those published by the International Civil Aviation Organization (ICAO), but were adapted for to meet specific needs. Of the thirteen phases given by ICAO, six are not applicable, which are: pushback/towing, maneuvering (at low altitude), emergency descent, uncontrolled descent, and unknown. The remaining seven are: standing, taxi, takeoff, initial climb, en-route, approach, and

landing; these served as reference for pre-classification (CICTT, 2013). Definitions for the eight pre-classification phases of flight are shown in Table 2. Using these definitions, the pre-classification algorithm scans the time-series mission and divides telemetry data into phase blocks, after which each record can be assigned the appropriate label.

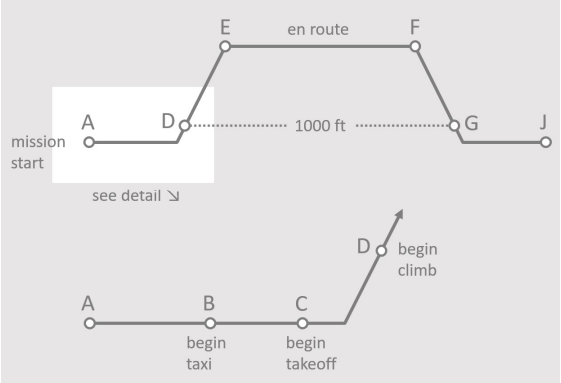
Table 2. Phase of flight definitions

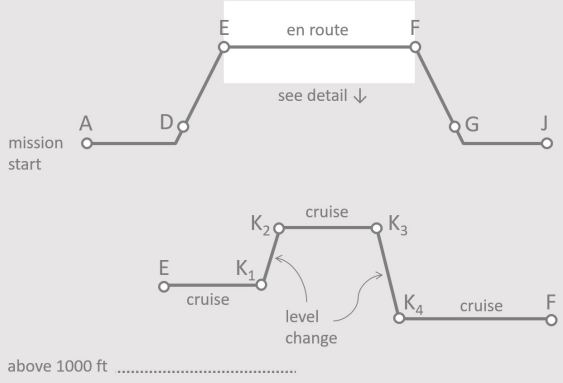
ID	Phase	Definition	ICAO Reference
1	Standing	On the ground, stationary, at beginning and end of mission	Same as ICAO standing
2	Taxi	On the ground, moving, before takeoff and landing	Same as ICAO taxi
3	Takeoff	Acceleration on ground until reaching 1000 ft altitude	Combines ICAO takeoff and initial climb
4	Landing	From 1000 ft until deceleration complete on the ground	Combines ICAO approach and landing
5	Climb	From takeoff to steady cruise altitude	Based on ICAO en-route subphase "climb to cruise"
6	Descent	From steady cruise altitude to landing	Based on ICAO en-route subphase "descent"
7	Cruise	Steady level flight at least 5 minutes	Based on ICAO en-route subphase "cruise"
8	Level-Change	Climb or descent between cruise levels	Based on ICAO en-route subphase "change of cruise level"

Researchers at Purdue University previously considered a similar problem, and they created an automatic phase identification program designed for general aviation aircraft (Goblet et al., 2015). That algorithm used four variables: ground speed, airspeed, altitude, and engine RPM. Variants of the algorithm were tested with 16 flights of the Cirrus SR20 aircraft, and achieved accuracy between 89% and 94% when compared to

manual classifications. Unfortunately, this algorithm cannot be applied to the UAS telemetry, as the engine RPM variable is not available, and airspeed measures follow a different convention.

Whereas the Purdue researchers intended to build a widely applicable algorithm, the pre-classification scheme for the current research is intended only to assign reasonable phase labels to this specific dataset, and for the specific usage case of testing ML methods. Only two variables were used for pre-classification: altitude above ground level (AGL), and ground speed. Data is processed by individual missions and assumed to be sequential and contiguous. The basic pre-classification procedure is shown in Figure 6.





1 – Forward Pass	2 – Backward Pass	3 – En-route Differentiation
1.1 Assign first data point as A (mission start) 1.2 Starting at A and parsing forward, check for ground speed > 0 over 10s, assign point B (taxi start) 1.3 Check forward acceleration > 1 kt/s and ground speed ≥ 12 kt, assign point C (takeoff start) 1.4 Check altitude ≥ 1000 ft over 10s, assign point D (climb start) 1.5 Check altitude change < 1 ft/s and altitude stable over 5m, assign point E (climb end)	2.1 Assign last data point as J (mission end) 2.2 Starting at J and parsing backward, check ground speed > 0 over 10s, assign point I (taxi end) 2.3 Check forward acceleration > 1 kt/s and ground speed ≥ 12 kt, assign point H (landing end) 2.4 Check altitude ≥ 1000 ft over 10s, assign point G (descent end) 2.5 Check altitude change > -1 ft/s and altitude stable over 5m, assign point F (descent start)	3.1 Starting at E, check altitude change ≥ 2 ft/s or ≤ -2 ft/s, then check altitude increasing or decreasing over 2m, assign K1 3.2 Check for altitude change < 2ft/s and > -2 ft/s, then check altitude stable over 5m, assign K2 3.3 Continue parsing data for level changes until reaching point F

Figure 6. Altitude illustration and basic steps of the pre-classification algorithm

During the forward pass, the algorithm starts at the first data point and proceeds to identify four key events during the early mission: begin taxi, begin takeoff, begin climb, and end climb. During the backward pass, the algorithm works similarly but in reverse. At this point, there is a large en-route block, which is then subdivided into alternating cruise and level-change blocks, as need. Transitions into and out of cruise segments proved somewhat challenging: the algorithm must identify clear changes in altitude but avoid rapid transition between phases. To achieve this, a combination of techniques were used, including local averaging and variance calculations. The full algorithm is given in Appendix A.

Pre-classification Outcomes

Since the objective of the pre-classification algorithm is to produce results that match human judgment, its performance can be checked by visualizing phases of flight on altitude plots. When discrepancies were found, parameters within the algorithm were adjusted to eliminate those discrepancies, then the plots are checked again. This iterative process continued until results became satisfactory for all 30 missions in the dataset. Figure 7 shows altitude over time for four missions, and each plot is colored by phases assigned by the pre-classification algorithm. These plots demonstrate that pre-classifications match human judgment. During mission 1, there was a single cruise altitude, which is colored light blue. A few touch-and-go maneuvers were performed near the end of flight, which were grouped with descent, and only the final segment was colored purple for landing. During mission 5, there were four distinct cruise altitudes,

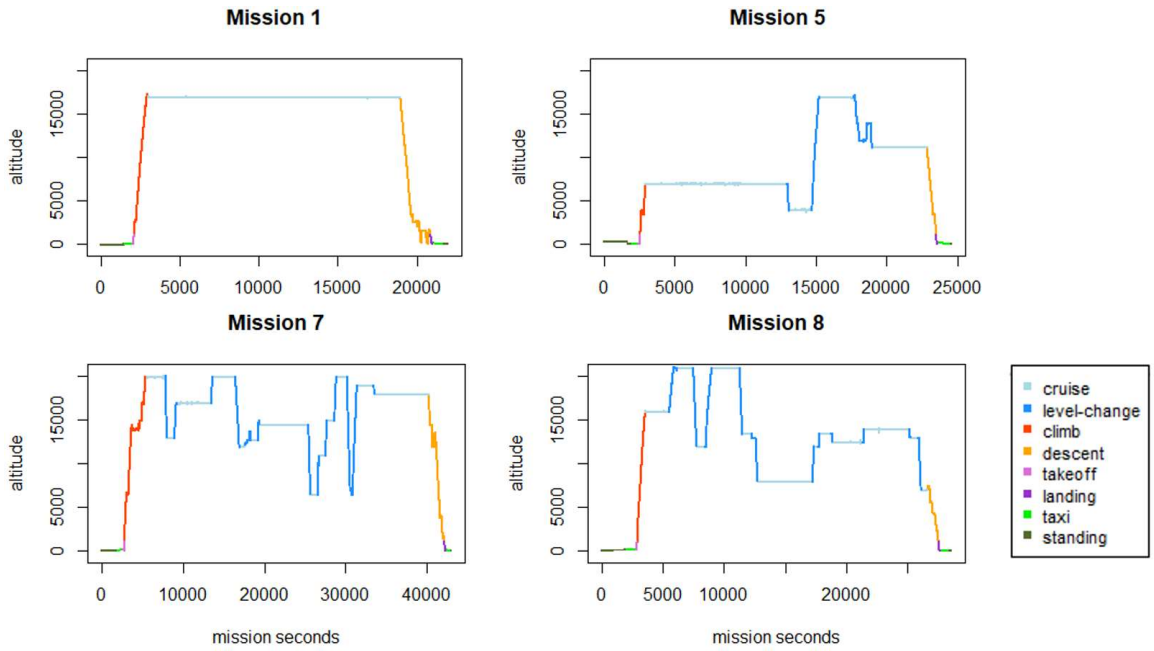


Figure 7. Pre-classification outcomes on four select missions

which were correctly labeled. However, two additional holding altitudes of short duration were labeled level-change, which is consistent with phase definitions. The situation was even more complex for mission 7, which was a lengthier mission. During climb, the aircraft held at about 14,000 ft, then continued climbing to about 20,000 ft. However, this could have been alternatively labeled as a short cruise phase followed by level-change. This represented an ambiguity that would require determination by a domain expert. In this case, the pre-classification labels were accepted as correct. Since these areas of ambiguity represent a small portion of total data, the final pre-classification results were deemed sufficient to allow research to proceed into the next phase.

Shorter phases at the beginning and end of each mission were also appropriately labeled. Figure 8 provides closer views of mission 8 start and end, with ground speed

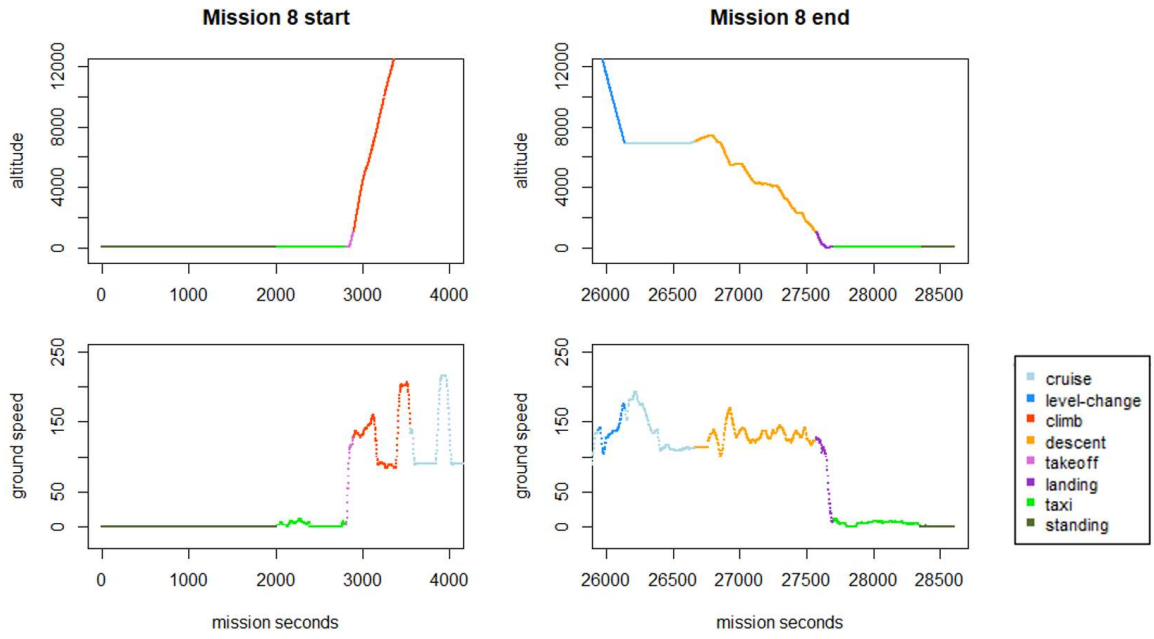


Figure 8. Detail views of pre-classifications on mission #8

Table 3. Frequency of pre-classification phase labels within dataset

ID	Phase of Flight	N Labeled	Proportion
1	Standing	73,284	10.3%
2	Taxi	39,518	5.5%
3	Takeoff	2,726	.4%
4	Landing	3,764	.5%
5	Climb	32,120	4.5%
6	Descent	33,865	4.7%
7	Cruise	477,969	67.0%
8	Level-change	49,798	7.0%
	TOTAL	713,044	100%

plotted in addition to altitude. Rapid acceleration was well captured within takeoff (dark pink), and rapid deceleration was well captured within landing (purple). From the counts of phase assignments in Table 3, it can be seen that takeoff and landing phases represent very small proportions (.4% and .5% of total records), which can be challenging for ML. Similar numbers of records were labeled as climb and descent, with somewhat more labeled as level-change. Cruise at steady altitude is the largest phase by far, representing 67% of all records. These proportions are reasonable for UAS missions.

Data Mining Procedure

With the addition of pre-classifications phase labels, the dataset is now complete. The two previously defined ML data mining tasks are implemented by means of the following procedure:

1. Using telemetry from mission 1 only, apply descriptive statistics to dataset variables to gain better understand these variables
2. Apply CART to the full dataset to produce a decision tree for phase classification, then interpret decision tree splits for useful patterns
3. Based on previous outcomes, identify a subset of predictor variables for use in subsequent models
4. Apply CART, C5.0, and neural network for phase classification. The dataset is randomly split by mission with 25 missions assigned to the training set and 5 missions assigned to the test set.
5. Compare ML performance of the three methods via 10-fold validation.

This research employs three ML methods by means of the R language packages "rpart," "C50," and "nnet." More specifically, the neural network model has the following structure: an input layer with 14 nodes corresponding to select attributes, a single hidden layer with 8 nodes with sigmoid activation function, and an output layer with 8 nodes corresponding to the phases of flight. Complete connectivity of the neural network entails 192 weight values.

In this chapter, eight phases of flight were defined, and a pre-classification algorithm was formulated as substitute for expert judgment. This algorithm produced phase labels to complete the telemetry dataset, and this categorical variable serves as target variable for ML methods. At this point the main data mining effort can begin.

IV. Data Mining Using ML

This chapter covers the primary data mining effort. First, initial exploratory data analysis (EDA) was performed on a subset of data to improve understanding of the data. Then a CART model was constructed using the full dataset. By comparing EDA findings, a reduced set of 14 variables was selected for ML classification. CART, C5.0, and neural network models were then compared in terms of classification accuracy.

Initial Exploratory Data Analysis

Prior to applying ML methods, it is necessary to develop a better understanding of the many variables in the dataset. To this end, basic descriptive statistics were used to examine to data from a single mission (mission 1), on as many variables as feasible. Of the 15 subsystem tables in the database, "general performance" is perhaps the most important. Table 4 provides a list of all 57 variables in "general performance." Of these, 2

Table 4. List of variables in "general performance"

mission_id	Tail_No	AP1_Roll	Predator_Fuel_Tank_Select
row_num	Stall_Spd	AP1_R_Rate	Use_Secondary_Airspeed
Prop_P_Pi	Wind_Dir	AP1_P_Rate	Use_Secondary_Alpha
Mag_Hdg_Cmd	Wind_Spd	AP1_Y_Rate	Use_Secondary_Altitude
Alt_MSL_Cmd	Lift_Coeff	AP2_Roll	Airspeed_To_Throttle_Mode
Norm_Accel	Beta	AP2_R_Rate	Presets_Are_Being_Sent_To_AV
Compass_Hdg	Alt_MSL_2nd	AP2_P_Rate	Altitude_To_Elevator_Mode
VSI	Sec_AS	AP2_Y_Rate	Automatic_De_Ice_Mode_Select
AOA	Sec_AOA	Flt_Dir_Gspd	Payload_Joystick_Trigger_Button
Pri_AS	Course	Trim_X	Reset_MTS_Ball
Alt_MSL	Rel_Humidity	Trim_Y	Emergency_Mission_NOT_Loaded
Roll_Rate		L_D	Stall_Protection_On_Off
OAT		Alt_AGL	Left_Side_Pitot_Heat_Status
Ground_Spd		Hdg	Right_Side_Pitot_Heat_Status
True_AS		Total_AV_Wt	Ice_Detected
Density_Alt		Spec_Range	
		Spec_Endur	

serve as primary key, 40 are numerical variables, and 15 are Boolean variables. With statistical software, it was quickly found that the variable "Prop_P_Pi" consist of entirely null values, and the variable "Spec_Range" take only discrete values of 0 and 1. Further EDA using other mission data would reveal that this variable takes integer values from 0 to 3, which may reflect a limitation in the UAS.

Six of the variables in "general performance" are closely associated with aircraft altitude, these are: "Alt_AGL," "Alt_MSL," "Alt_MSL_2nd," "Alt_MSL_Cmd," "Density_Alt," and "OAT." It may be inferred that above-ground-level (AGL) and mean-sea-level (MSL) give redundant measures for this critical metric. As shown in Figure 9, "Alt_AGL," "Alt_MSL," and "Alt_MSL_2nd" have perfect correlation, and outside air

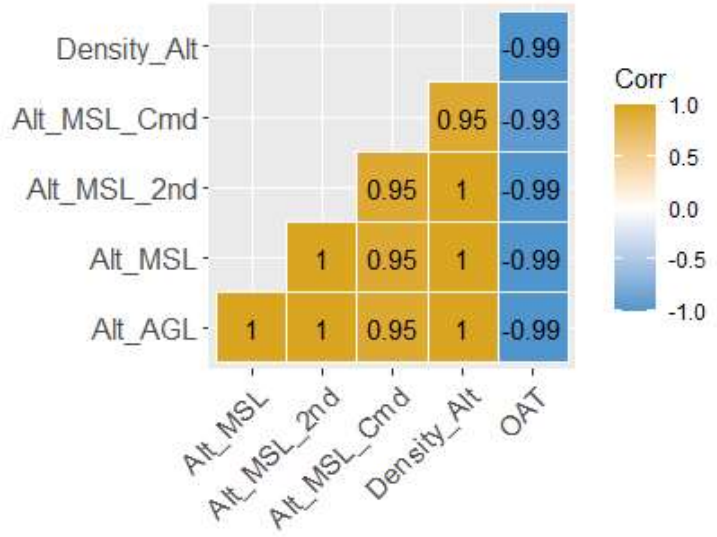


Figure 9. Correlation matrix for the altitude variables

temperature (OAT) has near perfect negative correlation with the previous. However, the variable "Alt_MSL_Cmd" shows weaker correlation with the rest. To better understand

this variable, it was useful to examine the altitude trend over the course of mission 1, as shown in Figure 10. It can be seen that the "command" value matched actual MSL during cruise, possibly serving as reference for the autopilot, and it may also provide guidance during climb and descent.

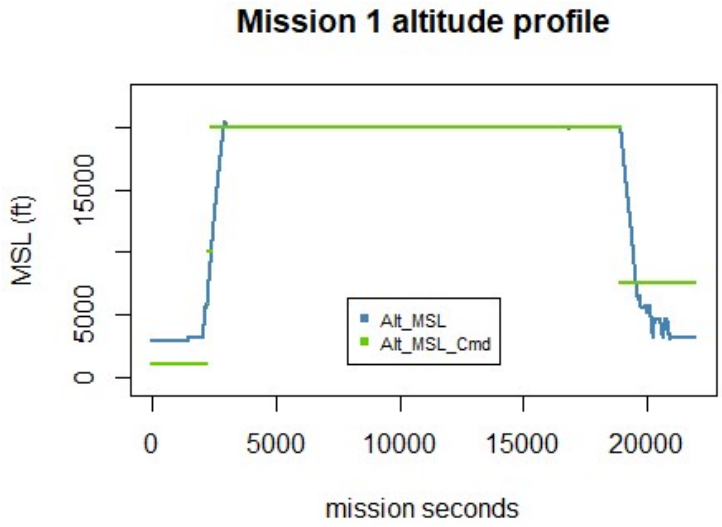


Figure 10. MSL altitude compared to "Alt_MSL_Cmd" during mission 1

The previous discussion served to illustrate activities performed during initial EDA, where the aim was to explore relationships among variables and improve understanding of the dataset. With meticulous EDA, it is possible to uncover significant patterns and insights from data; however, this may not represent a time-efficient approach with complex datasets.

Decision Tree Creation

As previously described, the telemetry dataset used for analysis is fairly large, consisting of 30 missions with 713,044 per-second records. There are also 390 predictor variables, plus the phase of flight generated by the pre-classification algorithm, which is

used as target variable. Given the number of variables, comprehensive EDA and traditional statistics can be quite laborious. Typically as part of EDA, one would attempt to reduce the number of dimensions through principal component analysis or factor analysis, which guides subsequent model-building. Instead, for the current analysis, ML will be applied directly to the full dataset.

The dataset is given to the CART algorithm with no additional processing, using all 390 predictor variables plus phase of flight as target variable, which represents known classifications. The resulting decision tree is shown in Figure 11. This decision tree is

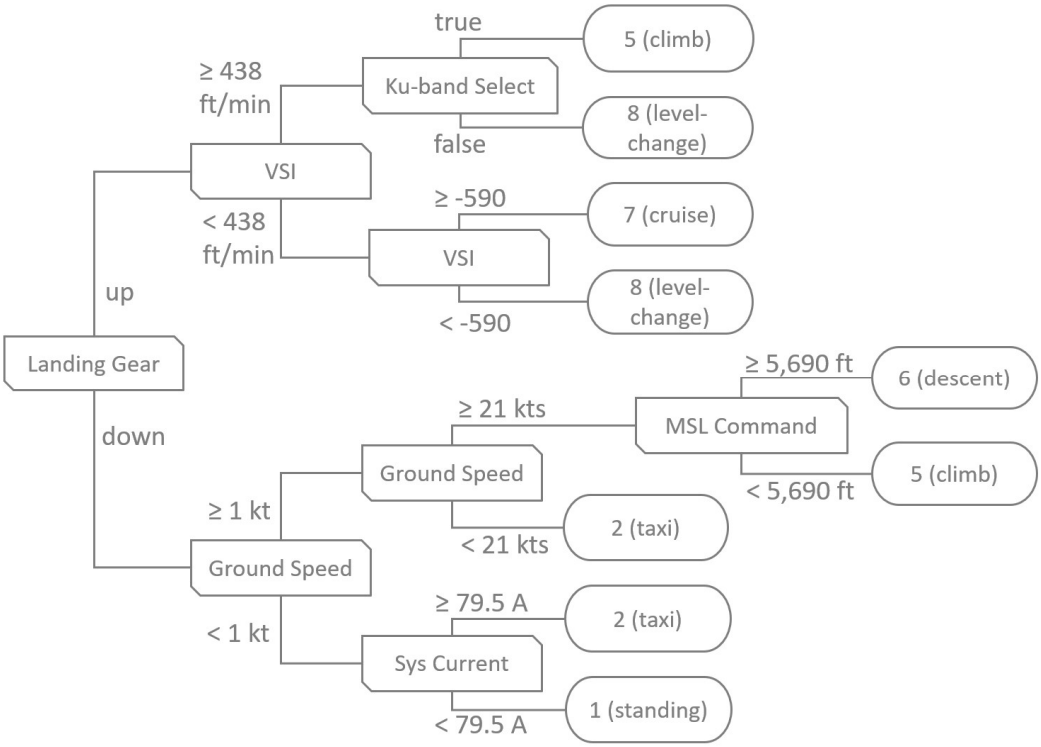


Figure 11. CART decision tree produced from the full dataset

surprisingly small, with only nine terminal nodes. However, none of the terminal nodes are classified as phase 3 takeoff or phase 4 landing, meaning that all such data would be

misclassified under this model. Only six predictor variables were used in the decision tree, and all others (384) were discarded. This is reasonable as there are significant correlations among many of the variables. In a way, these six variables serves in lieu of factors which would be produced by factor analysis. It is worth noting that altitude is absent among the six variables, even though it was one of two variables used for pre-classification (together with ground speed). However, MSL command is among the six variables.

Closer examination of the decision tree in Figure 11. reveals several interesting findings. Specifically, each of the six decision tree variables chosen by CART provides a degree of useful insight.

1. Landing gear: this binary variable indicates whether landing gear is up or down. The decision tree splits on landing gear at the root, which is intuitive for a human, since the UAS must be in flight if landing gear is up. However, it demonstrates CART is capable of capturing high-level information.
2. VSI (vertical speed indication): this numerical variable gives a reading in feet per minute—positive for climb and negative for descent. The decision tree splits on VSI twice, effectively bracketing a range for cruise, or steady-level flight. CART indicates that VSI ranges from -590 to 438 during cruise, which is corroborated by the distribution shown in Figure 12. However, it would be difficult for a human to identify precise break points using the histogram.

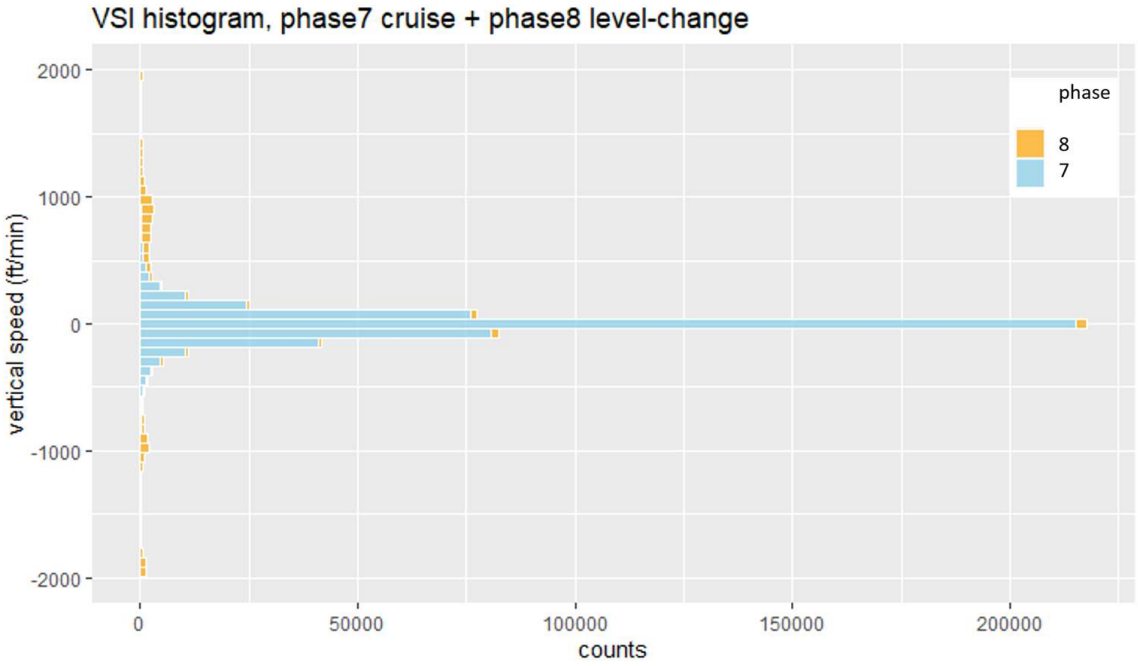


Figure 12. Comparison of vertical speed for cruise and level-change

3. Ku-band select: this binary variable indicates the Ku-band satellite link is active. CART uses this variable to differentiate climb from level change. This captures specific knowledge about UAS operation.
4. Ground speed: this numerical variable gives forward ground speed in knots, and it is strictly positive. This variable was used for pre-classification and CART recovers some domain knowledge within the pre-classification algorithm.
5. MSL command: this numerical variable is represents the pilot-specified altitude level for use by auto-pilot. It typically takes on discrete levels within the dataset. CART found that it is set lower during climb than it is during descent, which is quite counterintuitive and captures a pattern in human pilot behavior.

6. Sys current: this numerical variable indicates the total electrical current draw by the UAS, in amps. CART found that higher current draw differentiates standing and taxi while the UAS is on the ground. Although intuitive, this would not be known a priori to a non-expert.

From the previous discussion, it is clear that the naïve application of CART to the full dataset has generated insights related to phases of flight. Most of this information could not have been revealed by principal component analysis or factor analysis.

Variable Selection

Although CART is fast and thus capable of working with all variables in the dataset, a reduced dataset is preferred for other ML methods. Therefore, subsequent analysis will use a reduced set of just 14 predictor variables. These consist of 6 used for the CART decision tree and 8 others that show potential for differentiating phases of flight. Summary statistics for all 14 variables are given in Table **5**. Variables "Landing_Gear_Up" and "Ku_RF_Mode" take binary values, while the rest are numerical. Table **6** provides the mean of each variable by phase of flight. It can be seen that each variable shows significant variation between phases, so each has predictive value in differentiating the phases.

Table 5. Summary statistics for selected variables

#	Variable	Units	SQL Table	mean	stdev	min	1st qt	medi	3rd qt	max
1	Alt_AGL	ft	general performance	10.7k	6.2k	-166	6.9k	12.7k	15.4k	21.0k
2	Ground_Spd*	kt	general performance	123	55.9	0	110	128	156	304
3	VSI*	ft/min	general performance	-.005	443	-5k	-46.4	0	44.0	5k
4	Alt_MSL_Cmd*	ft	general performance	13.1k	6.0k	1k	10k	15k	18k	21k
5	Lift_Coeff	-	general performance	.824	.305	.28	.64	.78	.86	1.70
6	Spec_Range	nm/lb	general performance	.800	.415	0	1	1	1	3
7	System_Amp*	A	electrical	110	28.3	20	102	116	128	174
8	PPDM1_SW2_Amp	A	electrical	7.02	3.63	0	4.8	9.0	9.2	9.8
9	PPDM1_SW6_Amp	A	electrical	2.34	2.39	0	0	2.4	3.6	14.8
10	PPDM1_SW9_16Amp	A	electrical	10.5	3.84	0	10.5	11.6	12.1	21.1
11	Oil_Temp	deg	engine data	71.1	14.4	10	73	73	74	104
12	Oil_Press	psi	engine data	97.8	25.2	0	103	105	107	112
13	Landing_Gear_Up*	0/1	landing gear	.802	.399	0	-	-	-	1
14	Ku_RF_Mode*	0/1	aircraft comm	.739	.439	0	-	-	-	1

variables labeled with (*) are from CART decision tree

Table 6. Contrast between phases of flight for selected variables

#	Variable	Units	Overall Mean	Mean reading by phase							
				1	2	3	4	5	6	7	8
1	Alt_AGL	ft	10.7k	30	36	295	283	6115	5435	13378	12702
2	Ground_Spd	kt	123	0	5	89	89	127	125	143	147
3	VSI	ft/min	-.005	1	0	688	-463	431	-477	0	16
4	Alt_MSL_Cmd	ft	13.1k	1970	4201	1000	8309	8287	8390	15708	14876
5	Lift_Coeff	-	.824	1.51	1.56	1.15	0.98	0.81	0.71	0.71	0.70
6	Spec_Range	nm/lb	.800	0.00	0.00	0.00	0.53	0.41	0.77	0.99	0.77
7	System_Amp	A	110	36	77	97	86	98	99	121	125
8	PPDM1_SW2_Amp	A	7.02	0.03	1.55	3.81	0.52	3.39	2.23	8.78	9.10
9	PPDM1_SW6_Amp	A	2.34	0.72	1.01	1.54	0.92	2.06	1.46	2.67	2.88
10	PPDM1_SW9_16Amp	A	10.5	0.42	4.00	8.56	10.07	10.72	10.73	11.91	12.25
11	Oil_Temp	deg	71.1	33	86	84	80	78	77	73	73
12	Oil_Press	psi	97.8	23	85	108	106	107	108	105	105
13	Landing_Gear_Up	0/1	.802	0.00	0.00	0.20	0.00	0.60	0.28	1.00	1.00
14	Ku_RF_Mode	0/1	.739	0.00	0.03	0.09	0.00	0.26	0.20	0.94	1.00

More detailed visualization can be shown via box plots, as shown in Figure 13. For system current, there is a clear separation in values during standing and taxi, as suggested by the CART decision tree. PPDM (power-pass distribution module) switch 2 represents a subset of electrical components, and this typically draws about 9A during cruise and level change, but significantly less power during other phases.

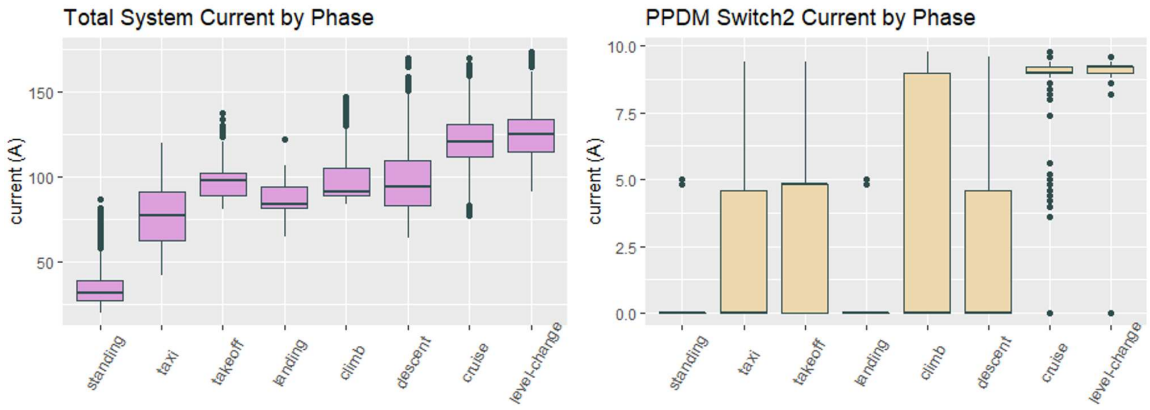


Figure 13. Box plots by phase for system current and PPDM switch 2

Comparison of Decision Tree Models (CART and C5.0)

At this point, EDA has provided a good level of understanding of the dataset and variables, therefore it is time to apply ML methods for phase classifications. The human designed pre-classification algorithm has previously generated phase labels, which are assumed to be the true values. The dataset is now split into a training set and a test set. The training set, consisting of 25 randomly selected missions, retains the phase information and is used to build classification models. The models are then applied to the test set, which consists of the remaining 5 missions, and where the pre-classification

labels are hidden. The models demonstrate their classification performance by generating phase labels for the test set, and the results are compared to the true values.

As previously mentioned, this process is repeated several times with different random splits, known as k-fold validation. However, it may be instructive to first examine the outcome of one such trial. It should be noted that in each trial, the data is randomly split by mission, but the same split is used for each ML method. For trial 1, CART produced a decision tree very similar to the one shown in Figure 11, with identical decision variables and leaf nodes. This model does not classify any records as takeoff or landing (phases 3 and 4), meaning that all such records are misclassified. However, since those phases account for a small portion of records, CART was able to deliver an overall accuracy of 92.1% on the test set. The contingency matrix is shown in Table **7**. This shows standing and taxi (phases 1 and 2) classifications were more accurate compared to

Table 7. Contingency matrix for CART (trial 1)

ML Accuracy 92.1%		CART Classifications							
		1	2	3	4	5	6	7	8
Pre-classifications	1	14192	24	0	0	0	0	0	0
	2	323	5616	0	0	0	0	0	0
	3	0	39	0	0	447	0	10	0
	4	0	38	0	0	0	580	0	0
	5	0	0	0	0	7588	40	1188	1105
	6	0	0	0	0	0	3697	363	351
	7	0	0	0	0	10	413	59870	525
	8	0	0	0	0	47	279	2682	7748

the in-flight phases (5-8). The single most frequent error was misclassification of level-change (phase 8) as cruise (phase 7), accounting for 2682 errors.

C5.0 produced a very large decision tree in trial 1. This decision tree consists of 2339 leaf nodes, compared to the CART decision tree with 9 leaf nodes. The first three levels of the C5.0 decision tree are shown in Figure 14. The decision tree continues

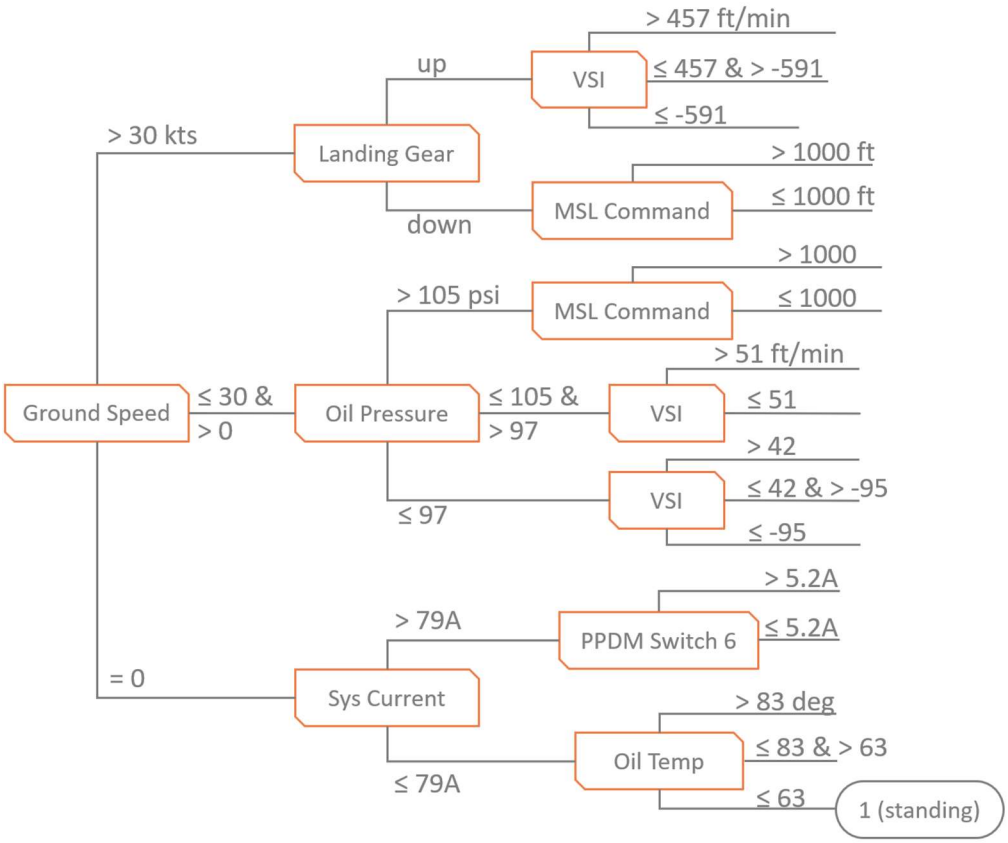


Figure 14. Top 3 levels of C5.0 decision tree (trial 1)

branching for several more levels, except the bottom branch which terminates at standing (phase 1). Despite the differences, this top portion of the C5.0 model utilized five of the six variables used by CART. In addition, C5.0 identified the VSI interval associated with cruise: between -591 ft/min and 457 ft/min. These are similar to the values found with

CART. For this trial, C5.0 resulted in an accuracy of 80.9%. Despite a much larger tree, C5.0 had lower accuracy, indicating that the model overfit the data. The advantage demonstrated by CART is likely due to minimal cost-complexity pruning, which seeks to reduce tree size as well as the number of errors. This pruning method appears better suited for the current classification task.

The C5.0 contingency matrix is given in Table 8. Unlike CART, C5.0 classified all eight phases, including takeoff and landing (phases 3 and 4). In fact, 81% of these

Table 8. Contingency matrix for C5.0 (trial 1)

ML Accuracy 80.9%		C5.0 Classifications							
		1	2	3	4	5	6	7	8
Pre-classifications	1	13794	422	0	0	0	0	0	0
	2	110	5823	0	6	0	0	0	0
	3	0	6	488	0	2	0	0	0
	4	0	20	0	419	0	179	0	0
	5	0	0	114	0	8375	55	479	898
	6	0	0	0	157	0	4064	89	101
	7	0	0	0	0	620	692	46732	12774
	8	0	0	0	0	751	388	2640	6977

records were correctly classified. Unfortunately, a number of records from climb were misclassified as takeoff, and records from descent misclassified as landing. The single most significant type of error for C5.0 is misclassification of cruise (phase 7) as level-change (phase 8), with 12,774 records incorrectly classified. Numerous records for level-change were also misclassified as cruise. As cruise is by far the largest phase, this accounts for overall poor performance from C5.0.

Comparison of Decision Trees with Neural Network

When the neural network model is trained and tested using the trial 1 split, it resulted in classification accuracy of 91.0%. This is higher than C5.0 and slightly below CART. As shown by the contingency matrix (Table 9), neural network performance was very similar to CART for the two largest phases, cruise and standing (phases 7 and 1). Unlike CART, neural network classified many records as takeoff (phase 3) and a few as landing (phase 4). Unfortunately, the majority of these were misclassified. In particular, 1638 records from climb (phase 5) were classified as takeoff (phase 3). This is greater than the total number of records from climb and descent.

Table 9. Contingency matrix for neural network (trial 1)

ML Accuracy 91.0%		Neural Network Classifications							
		1	2	3	4	5	6	7	8
Pre-classifications	1	14061	153	0	2	0	0	0	0
	2	100	5837	0	2	0	0	0	0
	3	0	114	239	0	143	0	0	0
	4	0	154	0	24	0	440	0	0
	5	0	304	1638	0	6363	149	587	880
	6	0	1	0	0	176	3696	368	170
	7	0	1	0	4	35	483	59796	499
	8	0	0	1	0	543	285	2439	7488

Unlike decision trees, neural network models do not allow for convenient interpretation. Individual weights associated with each network connection do not have particular significance. However, by varying the value of each input variable while holding others constant, one can rank variables by model sensitivity. In Table 10, this

ranked list from neural network is compared to the decision variables from CART and the ranked variable usage from C5.0. Three variables appear in the top six for all methods, and these are: "Landing_Gear_Up," "VSI," and "Ku_RF_Mode." While landing gear position and vertical speed are clearly relevant to phase of flight, the significance of Ku-band satellite link is unexpected. For CART, this was used to differentiate climb from level-change. One interpretation supported by data is that UAS pilots tend to switch from line of sight control to satellite link upon reaching cruise altitude.

Table 10. Comparison of significant variables for each method

CART Decision Variables	*C5.0 Variable Usage*	*Neural Network Variable Sensitivity*
Landing_Gear_Up	Ground_Spd	Landing_Gear_Up
VSI	VSI	Oil_Press
Ground_Spd	Oil_Temp	VSI
Ku_RF_Mode	Landing_Gear_Up	PPDM1_SW2_Amp
System_Amp	Ku_RF_Mode	System_Amp
Alt_MSL_Cmd	Alt_AGL	Ku_RF_Mode
	Spec_Range	PPDM1_SW9_16Amp
	PPDM1_SW9_16Amp	Alt_AGL
	Lift_Coeff	Spec_Range
	Alt_MSL_Cmd	Alt_MSL_Cmd
	PPDM1_SW6_Amp	PPDM1_SW6_Amp
	Oil_Press	Ground_Spd
	System_Amp	Lift_Coeff
	PPDM1_SW2_Amp	Oil_Temp

Overall Classification Performance

For a more thorough comparison of classification performance, all three ML methods underwent 10-fold validation. As was the case for trial 1, of 30 total missions,

25 were selected at random as the training set, and the remaining 5 were used for the test set. The random partitioning of data was conducted 10 times, and a set of 10 accuracy measures were obtained during this process. In each trial, the same split was used for the three ML methods: CART, C5.0, and neural network. The code used for ML modeling is found in Appendix C.

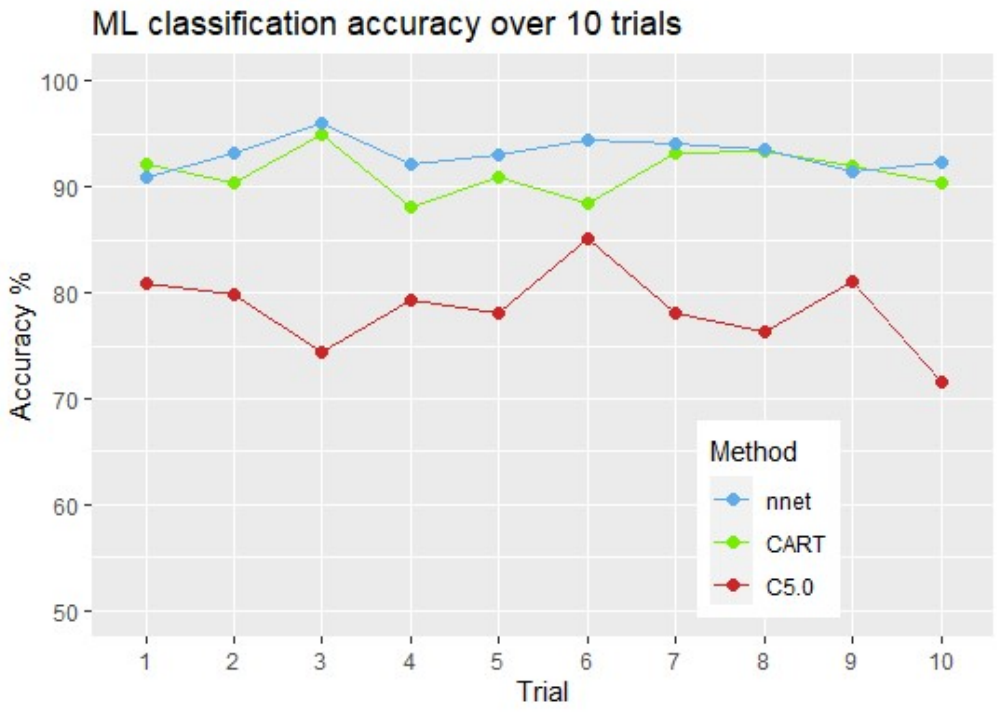


Figure 15. 10-fold validation using three ML methods

ML classification results are shown in Figure 15 and summarized in Table 11. Any algorithm should achieve a minimum of 67.0% accuracy, since this is the proportion of records labeled cruise during pre-classification. All three ML methods met the minimum threshold for all validation trials. CART achieved a mean classification accuracy of 91.4%, compared to C5.0 at 78.5% and neural network at 93.1%. Applying t-

test for paired differences, there is strong evidence that CART outperforms C5.0 in this task ($p<.001$). There is also evidence showing that neural network outperforms CART ($p=.032$). It is worth noting that the best algorithm variant used by Goblet et al. (2015) achieved an accuracy of 93.9%. Although these performance metrics are not directly comparable, there is evidence to suggest that ML methods can achieve performance on par with human-designed domain-specific algorithms, at least in the area of phase classifications. The demonstrated level of accuracy also compares favorably with neural network models in existing literature.

Table 11. Summary of ML classification accuracy

ML Method	classification accuracy (%)		
	min	max	mean
CART	88.2	94.9	91.4
C5.0	71.6	85.1	78.5
Neural Network	90.9	95.9	93.1

In this chapter, the previously outlined data mining plan was carried out. Application of CART to the full dataset uncovered useful data patterns and also aided in the selection of the most important variables. Among the three ML methods, neural network outperformed others in phase classification, although C5.0 showed an advantage in takeoff and landing phases.

V. Conclusions and Recommendations

This chapter provides a summary of data mining results. The preceding application of ML methods to UAS telemetry produced both specific insights and general understanding of data analysis. Given an efficient classification method, phases of flight can play an effective role in predictive maintenance.

Data Mining Outcomes

The application of ML methods in this research demonstrated that ML can be effective for Air Force data analytics. For both previously defined data mining tasks, success conditions were met. Data mining task one was the application of ML methods to aid EDA. For this task, CART was directly applied to the full dataset, which represented an expedient alternative to other dimension reduction techniques. The resultant decision tree produced direct insights on the dataset, including typical range for VSI during cruise, and identification of system current and Ku-band selection as useful variables in phase classification. Data mining task two was the application of ML methods to classify phase of flight. Three ML methods were applied to a reduced dataset with 14 predictor variables, and each method constructed distinct classification models. Both CART and neural network resulted in reasonable accuracy (91.4% average for CART, 93.1% average for neural network) . C5.0 did not perform as well; however, it is potentially useful in identifying the smaller phases of takeoff and landing. It may be possible to combine CART and C5.0 in a hybrid method: one could run both algorithms, then use C5.0 classifications for takeoff and landing, and CART classifications for all the remaining records.

Specific insights from this research may be of benefit for UAS operators. For example, total current draw as well as distribution switch current draw can be used to identify atypical operation, either by exceedances for a particular phase, or by unusual usage patterns during the course of a mission. Additionally, the Ku-band satellite link is typically activated upon reaching the first cruise altitude, even though it can be activated earlier. This could be an unintended pattern of operation. More generally, a pre-classification algorithm similar to what was used in this research can be a powerful enhancement to post-mission analysis. Assuming an external program was used to build these classifications, phase labels can be easily written back into the telemetry database then accessed via SQL queries.

In terms of Air Force data analytics, this research has shown that CART is an effective tool for both EDA and classification tasks. Neural network has been known to work well for maintenance problems, and this method demonstrated an advantage for phase classification as well. However, neural network models can take much longer to train, as they require numerical optimization of potentially large numbers of weights. It should be noted that data insights were gained by means of a set of known phase classifications. For ML methods to be effective, some domain knowledge is required, and further domain knowledge is needed to properly interpret models created by ML tools. Moreover, ML tools only serve to extract patterns within the data, and these patterns can also be found by traditional statistical techniques, given sufficient effort. Therefore ML should never be applied indiscriminately, but rather reserved for cases where they can provide a clear advantage.

Usage Scenarios

The addition of phase labels to telemetry datasets allow for much higher fidelity data analysis. Identifying exceedances in any critical system parameter would certainly be desirable for UAS maintainers and operators. As an example, it can be useful to examine in-flight fuel flow, which might be visualized using a scatterplot as shown in Figure 16. As expected, there is a strong correlation between fuel usage and vertical speed. With the data colored by phase of flight, it can be seen that the highest fuel use occurred during climb, at about 450 lbs/hr. Fuel use during cruise is generally between 150 and 280 lbs/hr. However, there is a cluster of points classified as cruise with

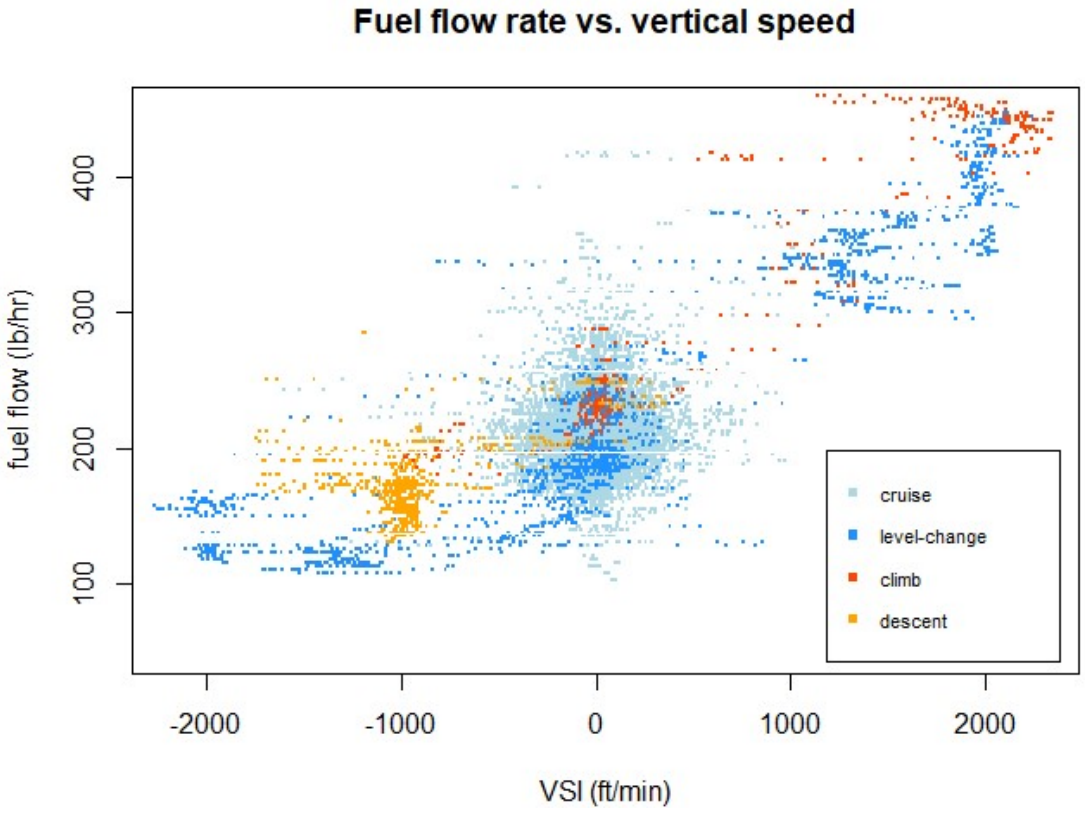


Figure 16. Scatterplot of fuel flow and VSI for mission 5, colored by phase

unusually high fuel flow (>400 lbs/hr). Further examination of flight data reveals that this occurred immediately after transition from climb to cruise, therefore it is likely explained by maneuvers at that time. Another slightly more sophisticated example is related to the electrical system. It was previously noted that overall current draw differs depending on phase of flight. Furthermore, electrical subsystems also exhibit distinct current ranges depending on phases. Therefore, the phase-based electrical usage profile can be compared with baseline to determine and isolate anomalous behavior.

Given a full mission dataset, a domain-specific software tool such as the pre-classification algorithm can provide reliable phase labels. Although ML models are less accurate, ML classifications may suffice in cases where only the instantaneous telemetry is available. This could be during a mission, or when data has become fragmented. Moreover, the CART algorithm can be used in a novel manner. With typical telemetry, CART produces decision trees that are largely similar, thus significant changes may be indicative of data anomalies.

Limitations and Future Work

Although this research used a substantial dataset, it is still quite small compared to many modern datasets. More data with more aircraft no doubt holds potential for additional insights. Furthermore, the dataset used here is static, whereas many Air Force data analytics problems involve constantly changing data. In terms of ML, the number of available methods is large and rapidly growing, and only three were applied to this research. This provides opportunity for future work using the same data. For example, Kohonen networks are widely employed for dimension reduction, and this method may

produce constructed predictor variables that are more effective than the 14 basic variables used here. Finally, this work can benefit from validation by UAS domain experts.

Beyond UAS maintenance, advanced data analytics is critical to the future success of the Air Force, and speed is essential for data-driven decisions. Currently, many barriers exist, including lack of access, lack of tools, and lack of knowledge. This research project discovered several useful patterns exist in an Air Force dataset, which had been inadequately utilized. Furthermore, it was shown that insights can be obtained even with limited resources, both through basic statistics and by leveraging established ML methods. It is hoped that this research and similar efforts can erode barriers and encourage all Airmen to find the means to make data work for them.

Appendix A. Pre-classification Algorithm (R code)

```
# labels phases on a single mission
# labels phases on a single mission
# takes AGL and groundSpeed, returns phaseLabels

# 1 = standing
# 2 = taxi
# 3 = takeoff
# 4 = landing
# 5 = climb
# 6 = descent
# 7 = cruise
# 8 = level-change


preclassification2 <- function(AGL, groundSpeed) {

 lineMax = length(AGL)
 diffAGL = c(diff(AGL), 0)

 phaseMarkers = integer(10)
 names(phaseMarkers) = c("start", "taxi_start", "takeoff_start",
                         "climb_1k",
                         "cruise_start", "cruise_end",
                         "descent_1k",
                         "landing_end", "taxi_end", "end")


 ##marker 1: start at 1
 phaseMarkers["start"] = 1

 ##marker 2: taxi_start
 for (i in 1:lineMax) {
  if ((groundSpeed[i] > 0) &&
      (groundSpeed[i+5] > 0) &&
      (groundSpeed[i+10] > 0)) {
    break
  }
 }
 phaseMarkers["taxi_start"] = i

 ##marker 3: takeoff_start
 for (j in i:lineMax) {
```

```
  if ((groundSpeed[j+10]-groundSpeed[j+5] >= 5) &&
      (groundSpeed[j+5]-groundSpeed[j] >= 5)) {
    if (groundSpeed[j+5] >= 12)
      break
  }
}
phaseMarkers["takeoff_start"] = j

##marker 4: climb_1k, reached 1k AGL
for (k in j:lineMax) {
  if ((AGL[k] >= 1000) &&
      (AGL[k+5] >= 1000) &&
      (AGL[k+10] >= 1000)) {
    break
  }
}
phaseMarkers["climb_1k"] = k

##marker 5: cruise_start, reached cruise altitude
for (ii in k:lineMax) {
  if (mean(diffAGL[(ii):(ii+10)]) < 1) {
    # verify level flight over 5min
    if (sd(AGL[(ii):(ii+300)]) < 30)
      break
  }
}
phaseMarkers["cruise_start"] = ii


# now we search in reverse starting from the end
##marker 10: start at the end
phaseMarkers["end"] = lineMax

##marker 9: taxi_end
for (i in lineMax:1) {
  if ((groundSpeed[i] > 0) &&
      (groundSpeed[i-5] > 0) &&
      (groundSpeed[i-10] > 0)) {
    break
  }
}
phaseMarkers["taxi_end"] = i

##marker 8: landing_end
```

```
for (j in i:1) {
  if ((groundSpeed[j-10]-groundSpeed[j-5] >= 5) &&
      (groundSpeed[j-5]-groundSpeed[j] >= 5)) {
    if (groundSpeed[j-5] >= 12)
      break
  }
}
phaseMarkers["landing_end"] = j

##marker 7: descent_1k, reached 1k AGL
for (k in j:1) {
  if ((AGL[k] >= 1000) &&
      (AGL[k-5] >= 1000) &&
      (AGL[k-10] >= 1000)) {
    break
  }
}
phaseMarkers["descent_1k"] = k

##marker 6: cruise_end, start final descent
for (ii in k:1) {
  if (mean(diffAGL[(ii):(ii-10)]) > -1) {
    # verify level flight over 5min
    if (sd(AGL[(ii):(ii-300)]) < 20)
      break
  }
}
phaseMarkers["cruise_end"] = ii
##marker complete


# assign labels to all records
phaseLabels = integer(lineMax)

phaseLabels[1:(phaseMarkers["taxi_start"]-1)] = 1
phaseLabels[(phaseMarkers["taxi_start"]):
            (phaseMarkers["takeoff_start"]-1)] = 2
phaseLabels[(phaseMarkers["takeoff_start"]):
            (phaseMarkers["climb_1k"]-1)] = 3
phaseLabels[(phaseMarkers["climb_1k"]):
            (phaseMarkers["cruise_start"]-1)] = 5
phaseLabels[(phaseMarkers["cruise_start"]):
            (phaseMarkers["cruise_end"])] = 7
phaseLabels[(phaseMarkers["cruise_end"]+1):
```

```
            (phaseMarkers["descent_1k"])] = 6
  phaseLabels[(phaseMarkers["descent_1k"]+1):
            (phaseMarkers["landing_end"])] = 4
  phaseLabels[(phaseMarkers["landing_end"]+1):
            (phaseMarkers["taxi_end"])] = 2
  phaseLabels[(phaseMarkers["taxi_end"]+1):
            lineMax] = 1

  # differentiate between cruise and level-change
  cruiseFlag = 7
  AGL60s = 125 #alt change cutoff value

  for (jj in (phaseMarkers["cruise_start"]+60):
      (phaseMarkers["cruise_end"]-60)) {

    if (cruiseFlag == 7) {
      # AGL trend up/down means level change
      if (abs(mean(diffAGL[(jj):(jj+15)])) >= 2) {
        if ((((AGL[jj+45]-AGL[jj+15]) >= .75*AGL60s) &&
             ((AGL[jj+120]-AGL[jj-30]) >= 2.5*AGL60s)) ||
            (((AGL[jj+45]-AGL[jj+15]) <= -.75*AGL60s) &&
             ((AGL[jj+120]-AGL[jj-30]) <= -2.5*AGL60s)))
          cruiseFlag = 8
      }
    }
    else if (cruiseFlag == 8) {
      # AGL constant means cruise
      if (abs(mean(diffAGL[(jj):(jj+15)])) < 2) {
        if (sd(AGL[(jj):(jj+300)]) < 35)
          cruiseFlag = 7
      }
    }
    phaseLabels[jj] = cruiseFlag
  }

  return(factor(phaseLabels))

}
```

Appendix B. Import Data From MySQL (R code)

```
missionMax = 30
durations = integer(missionMax)

library(RMariaDB)
dbMQ <- dbConnect(RMariaDB::MariaDB(),
                  user='basicuser',
                  password='whatever',
                  dbname='mq',
                  host='localhost')


for (mm in 1:missionMax) {
  missionSpec = paste0("=", mm)

  query <- paste("SELECT * FROM GeneralPerformance",
                 "WHERE mission_id",
                 missionSpec)
  rs = dbSendQuery(dbMQ, query)
  gp = dbFetch(rs)
  dbClearResult(rs)

  query <- paste("SELECT * FROM Electrical",
                 "WHERE mission_id",
                 missionSpec)
  rs = dbSendQuery(dbMQ, query)
  ele = dbFetch(rs)
  dbClearResult(rs)

  query <- paste("SELECT Oil_Temp, Oil_Press",
                 "FROM EngineData",
                 "WHERE mission_id",
                 missionSpec)
  rs = dbSendQuery(dbMQ, query)
  eng = dbFetch(rs)
  dbClearResult(rs)

  query <- paste("SELECT Landing_Gear_Up FROM LandingGear",
                 "WHERE mission_id",
                 missionSpec)
  rs = dbSendQuery(dbMQ, query)
  ldg = dbFetch(rs)
  dbClearResult(rs)
```

```
  query <- paste("SELECT Ku_RF_Mode FROM AircraftComm",
                 "WHERE mission_id",
                 missionSpec)
  rs = dbSendQuery(dbMQ, query)
  comm = dbFetch(rs)
  dbClearResult(rs)

  telemetry = cbind(gp[,c(40,14,8,5,21,43)],
                    ele[,c(3,55,59,62),],
                    eng, ldg, comm)


  # create mission data frame
  durations[mm] = dim(telemetry)[1]
  phaseLabel <- preclassification2(telemetry$Alt_AGL,
                                   telemetry$Ground_Spd)
  dataCmd = paste0("mission", mm, "<-cbind(telemetry,phaseLabel)")
  eval(str2lang(dataCmd))
}

# end data collection
dbDisconnect(dbMQ)
message("Total collected: ", sum(durations))
```

Appendix C. Phase Classification Using ML (R code)

```
library(rpart, C50, nnet)

# generate random train/test splits by mission number
set.seed(7)
splitTable = integer()

for (ii in 1:10) {
  randomDraw = order(runif(30))
  splitTable = rbind(splitTable, randomDraw)
}
row.names(splitTable)=c("trial1","trial2","trial3","trial4",
                        "trial5","trial6","trial7","trial8",
                        "trial9","trial10")

trainSize = 25
testSize = 5
accuracyList = double()
accuracy <- function(x){sum(diag(x)/(sum(rowSums(x)))) * 100}

test.cl <- function(true, pred) {
  true <- max.col(true)
  cres <- max.col(pred)
  table(true, cres)
}

# k-fold validation loop
for (kk in 1:10) {

  # assemble train set
  bindCmd = "trainset = rbind("
  for (ii in 1:trainSize) {
    bindCmd = paste0(bindCmd, "mission", splitTable[kk,ii])
    if (ii < trainSize) bindCmd = paste0(bindCmd, ", ")
    else bindCmd = paste0(bindCmd, ")")
  }
  eval(str2lang(bindCmd))

  # assemble test set
  bindCmd = "testset = rbind("
  for (jj in (31-testSize):30) {
    bindCmd = paste0(bindCmd, "mission", splitTable[kk,jj])
    if (jj < 30) bindCmd = paste0(bindCmd, ", ")
```

```
  else bindCmd = paste0(bindCmd, ")")
 }
 eval(str2lang(bindCmd))


 # CART model
 cartModel <- rpart(phaseLabel ~ .,
                 data=trainset, method="class")

 prediction <- predict(cartModel, testset[,-15], type="class")
 contingency <- table(testset[,15], prediction)
 accuracyList[kk] = accuracy(contingency)

 # C5.0 model
 C50Model <- C5.0(phaseLabel ~ ., data=trainset)

 prediction <- predict(C50Model, testset[,-15])
 contingency <- table(testset[,15], prediction)
 accuracyList[10+kk] = accuracy(contingency)


 # neural net model
 trainsize = dim(trainset)[1]
 testsize = dim(testset)[1]

 combinedd = rbind(trainset, testset)
 scaledd <- scale(combinedd[,1:14])
 phaseMatrix <- class.ind(combinedd$phaseLabel)

 neuralModel <- nnet(scaledd[1:trainsize,],
                  phaseMatrix[1:trainsize,],
                  size=8, decay=.005, maxit=100,
                  Wts=refWeights)

 prediction <- predict(neuralModel,
                    scaledd[(trainsize+1):(trainsize+testsize),])
 myt = test.cl(phaseMatrix[(trainsize+1):(trainsize+testsize),],
             prediction)
 accuracyList[20+kk] = accuracy(myt[as.integer(colnames(myt)),])

 # end validation trial
 message("validation trial ", kk, " complete")
}
```

Appendix D. Topic Paper Presented at CSCE 2021

Identifying Phases of Flight for Unmanned Aircraft Systems through Machine Learning

Li Yu, Brent T. Langhals, Ryan D.L. Engle, Michael R. Grimaila, & Douglas D. Hodson

***Abstract*—Machine learning techniques provide the means to leverage large quantities of data to aid in operational decision making and autonomy. The proposed research will apply machine learning to unmanned aircraft systems (UAS) telemetry data, which has been made available through a relational database but remains underutilized. One or more prediction models will be created for UAS phases of flight by means of algorithms such as decision trees and neural networks, based on various system performance parameters, and these models will then be validated to assess machine learning performance.**

***Keywords*—machine learning, artificial intelligence, autonomy, unmanned aircraft systems (UAS), data analytics, data modeling.**

1. MOTIVATION

In today's Air Force, both hardware and informational systems continuously generate huge quantities of data. Unfortunately, most of this data is either discarded, ignored or, at best, only marginally used. Recognizing this fact, the Air Force Chief Data Officer offered a vision to make better use of data with the goal of improving operational decision making [1]. Automated machine learning represent one feasible path for reaching this objective.

Machine learning (ML) methods and algorithms have been successfully applied to a wide range of disciplines, including cyber threat detection and COVID-19 diagnostics [2] [3]. Within the domain of unmanned aircraft systems (UAS), there has been ongoing research focused on obstacle detection and collision avoidance through machine learning [4]. The proposed research would utilize an existing UAS dataset to answer two questions: one, whether there is significant hidden value within the data; and two, whether this hidden value can be efficiently extracted via ML methods.

In addition to improving the use of data, these answers may have significant implications for UAS autonomy. Numerous UAS are currently in use by the Air Force, ranging from small drones to multi-million-dollar weapons systems [5]. Although the majority of existing UAS are either remote controlled or semi-autonomous (human in the loop), future missions will likely utilize greater levels of autonomy, as shown by a recent Delphi study [6]. That study also indicated greater levels of autonomy would rely on expanded exploitation of UAS-generated data through machine learning and artificial intelligence technologies. Specifically, UAS would have to become self-aware of its environment as well as internal states.

In other words, if the UAS is to take on the tasks of a human operator, it must be able to synthesize high level state variables using low level sensor data. One such high level state variable is the phase of flight (e.g. taxi, take-off, climb), which is not directly captured by UAS sensors yet is highly relevant to UAS operation. Meaningful insights on phase of flight produced via ML would demonstrate hidden value within the dataset.

2. DATASET

This research utilizes real-world data collected by a commercially produced, large-scale Air Force UAS. The UAS captures vast amounts of flight telemetry, aircraft to ground communication, and internal systems messaging from each mission, which can last well over 24 hours. In its raw state the data is not easily accessible, making real-time data analysis impossible. However, previous work resulted in an indirect mapping algorithm, which automated extract, transform, and load (ETL) operations to a relational database [7]. The data is now available in a relational database that supports fleet-wide data views across multiple missions.

While the database represents a dramatic improvement in usability compared to the unprocessed data format, much of the data remain underutilized. The data for each mission spans the course of many hours, both on the ground and in the air. In all there are over 3000 variables, including location data, flight characteristics, general performance as well as subsystem parameters. Some variables have Boolean values (e.g. landing gear up) while others are double precision (e.g. airspeed). All variables are sampled once per second, with a subset sampled at 20 times per second.

This research will focus on the identification of UAS phase of flight (or flight segment). Such identification is not available in the existing data which is comprised of low level system variables. Differentiation between various phases of flight has immediate utility for UAS operators and maintainers, as this allows them to analyze a range of data corresponding to a particular operating environment. Specifically, unique indicators associated with each phase of flight become apparent.

3. METHODOLOGY

Since the UAS does not directly capture phases of flight, an objective ground truth must first be established. However, since the UAS is largely operated by a human pilot, several reference variables are available that are directly associated with pilot-directed actions. The phase of flight can then be determined by means of a decision tree, as shown in Figure 1, and a classification is assigned to each second of data. UAS experts will be enlisted to help design the decision tree and evaluate the results to ensure accuracy. Once phase of flight is known, a prediction model will be generated using machine learning methods on the training dataset without using the reference variables. At this point, model performance can be evaluated using a test dataset. Traditional statistical methods may also be applied to gain greater insight on the problem.

Figure 1 represents a basic deterministic classification model. Four possible reference variables are as follows: landing gear position is a simple binary status indicator; pilot inputs represent stick inputs from the remote pilot; altitude can be one of several double precision variables; ground speed is another double precision variable, and acceleration is calculated from changes in ground speed. The eight phase of flight classifications are as follows:

- Flight-active – when the UAS is at altitude and maneuvering by remote
- Flight-inactive – when the UAS is at altitude and on auto-pilot
- Ground-stationary – when the UAS is on the ground and not moving
- Ground-taxi – when the UAS is on the ground and moving at a low speed
- Take-off – when the UAS is accelerating prior to take-off rotation
- Landing – when the UAS is decelerating after touch down
- Climb – when the UAS is climbing after take-off rotation
- Descent – when the UAS is descending for landing

Initial classification is likely to have errors and would require manual corrections. In addition, classification may be indeterminate for certain periods; for example, the UAS may fly level partially through its descent or climb. After ground truth is manually determined, ML algorithms such as decision trees, neural networks, etc. will be applied without the use of the

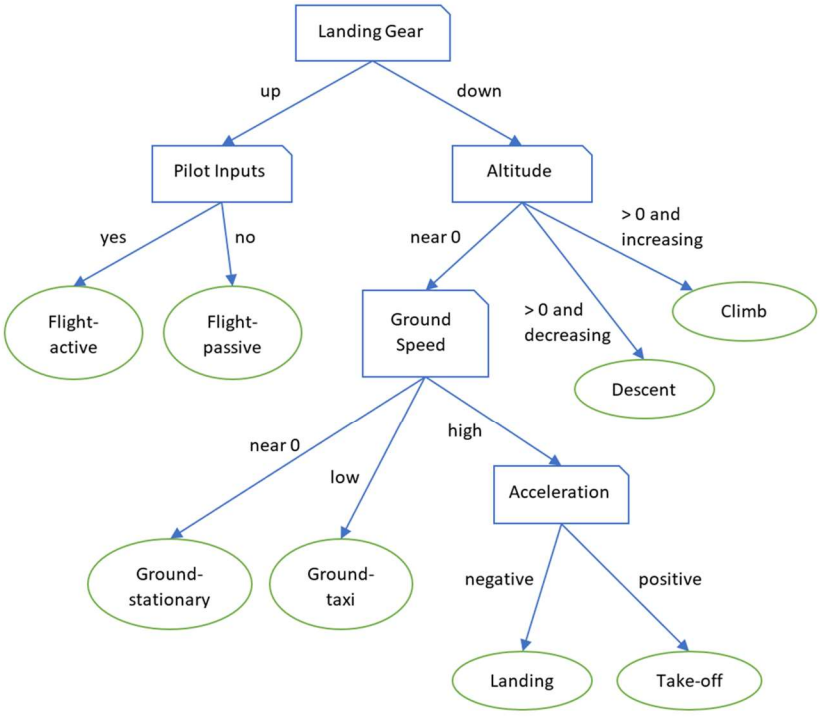


Figure 1. A priori decision tree for assigning phases of flight

reference variables. Instead, the ML models will be constructed using general performance parameters and subsystem performance parameters (to include electrical, engine, fuel, servos, etc.). Performance of predictive models can be measured and compared through classification accuracy on a test dataset and by means of receiver operating characteristic curves.

Machine learning (i.e. using categorical and regression trees) has been found to outperform traditional logistic regression, especially when there are strong non-linearities and/or discontinuities in the relevant variables [8]. However, traditional statistical techniques may still offer useful insight on relationships between variables. Applicable techniques would include factor analysis on subsystem parameters, correlation analysis, as well as multinomial logistic regression for phase determination. Identification of previously unknown correlations or interactions can supplement insights derived from ML methods and may provide a specific benefit to UAS operators and maintainers.

4. ANTICIPATED OUTCOMES

The available UAV data, accessible within a relational database, provides a test case to examine hidden value within existing data. Specifically, two methods for identifying phases of flight can be compared: first through a decision tree based on a priori knowledge and reference variables, second by means of ML model using other system variables. Given the large number of variables and additional derived variables, it is likely that the ML model can achieve reasonable accuracy in predicting phases of flight. If so, then phase of flight state information is encapsulated twice within the dataset, which demonstrates hidden value within underutilized data.

Additionally, if the ML model can reveal new and useful relationships between system variables and phases of flight, then ML is effective in extracting hidden value. An example of ML application to data analytics would be the creation of an improved decision tree that can resolve errors and ambiguities of the basic model. An example of ML application to UAS autonomy would be determining when to retract and extend the landing gear in the absence of a human pilot.

5. REFERENCES

[1] E. Vidrine, "The Power of Data," 27 August 2018. [Online]. Available: https://www.afitc-event.com/wp-content/uploads/CDO-Update.pdf. [Accessed 18 May 2021].

[2] K. Shaukat, S. Luo, V. Varadharajan, I. A. Hameed, S. Chen, D. Liu and J. Li, "Performance Comparison and Current Challenges of Using Machine Learning Techniques in Cybersecurity," *Energies,* vol. 13, no. 10, 2020.

[3] W. T. Li, J. Ma, N. Shende, G. Castaneda, J. Chakladar, J. C. Tsai, L. Apostol, C. O. Honda, J. Xu, L. M. Wong, T. Zhang, A. Lee, A. Gnanasekar, T. K. Honda, S. Z. Kuo and M. A. Yu, "Using Machine Learning of Clinical Data to Diagnose COVID-19: A Systematic Review and Meta-analysis," *BMC Medical Informatics and Decision Making,* vol. 20, no. 1, 2020.

[4] P. Fraga-Lamas, L. Ramos, V. Mondéjar-Guerra and T. M. Fernández-Caramés, "A Review on IoT Deep Learning UAV Systems for Autonomous Obstacle Detection and Collision Avoidance," *Remote Sensing,* vol. 11, no. 18, 2019.

[5] Danish Technological Institute, "Global Trends of Unmanned Aerial Systems," 2019. [Online]. Available: https://02f09e7.netsolhost.com/AUVSIDocs/Global%20Trends%20for%20UAS.pdf. [Accessed 1 May 2021].

[6] A. Sigala and B. Langhals, "Applications of Unmanned Aerial Systems (UAS): A Delphi Study Projecting Future UAS Missions and Relevant Challenges," *Drones,* vol. 4, no. 1, 2020.

[7] R. D. Engle, B. T. Langhals, M. R. Grimaila and D. D. Hodson, "A Methodology for Storing Log Data with Changing Structure in a Relational Database," *Journal of DoD Research and Engineering,* vol. 4, no. 1, 2021.

[8] J. J. Levy and A. J. O'Malley, "Don't Dismiss Logistic Regression: The Case For Sensible Extraction of Interactions in The Era of Machine Learning," *BMC Medial Research Methodology,* vol. 20, no. 1, 2020.

Bibliography

Adhikari, Patha and others. "Machine Learning Based Data Driven Diagnostics & Prognostics Framework for Aircraft Predictive Maintenance," *10th International Symposium on NDT in Aerospace*. Dresden, Germany: DGZfP, 2018.

Air Transport Association of America. *ATA MSG-3 Operator/Manufacturer Scheduled Maintenance Development, Revision 2002.1*. Washington: ATA, 2002.

Altay, Ayca, Omer Ozkan, and Gulgun Kayakutlu. "Prediction of Aircraft Failure Times Using Artificial Neural Networks and Genetic Algorithms," *Journal of Aircraft*, vol. 51, no. 1 (2014).

Ay Türe, Begüm, and others. "Techniques for Apply Predictive Maintenance and Remaining Useful Life: A Systematic Mapping Study," *BSEU Journal of Science*, Vol. 8, No. 1 (2021).

Breiman, Leo and others. *Classification and Regression Trees*. Belmont CA: Wadsworth International Group, 1984.

British Standards Institution. *Maintenance Terminology*. BS EN 13306:2010. London: BSI Standards Publication, 2010.

Buede, Dennis M. and William D. Miller. *The Engineering Design of Systems: Models and Method* (3rd edition). Hoboken NJ: Wiley, 2016.

Carson, Jarrod and others. "A Hybrid Decision Tree-Neural Network (DT-NN) Model for Large-Scale Classification Problems," *2020 IEEE International Conference on Big Data*. 4103–11. Atlanta GA: IEEE, 2020.

Celikmih, Kadir and others. "Failure Prediction of Aircraft Equipment Using Machine Learning with a Hybrid Data Preparation Method." *Scientific Programming*, vol. 2020, no. 1 (August 2020).

Chidester, Thomas R. "Understanding Normal and Atypical Operations through Analysis of Flight Data," *Proceedings of the 12th International Symposium on Aviation Psychology*. Dayton OH: Wright State University Press, 2003.

Commercial Aviation Safety Team/International Civil Aviation Organization Common Taxonomy Team (CICTT). "ICAO Phase of Flight Definition and Usage Notes (1.3)." *IntlAviationStandards.org*. April 2013. Retrieved on 30 March 2021.

Department of The Air Force. *USAF Unmanned Aircraft Systems Flight Plan 2009–2047*. Washington: HQ USAF, May 2009.

Fahlstrom, Paul G., and Thomas J. Gleason. *Introduction to UAV Systems* (4th edition). Chichester, West Sussex: John Wiley & Sons, 2012.

Fraga-Lamas, Paula and others. "A Review on IoT Deep Learning UAV Systems for Autonomous Obstacle Detection and Collision Avoidance," *Remote Sensing*, vol. 11, no. 18 (September 2019).

Geiger, Stacey. "Data Analytics Can Help Air Force Become More Effective." *wpafb.af.mil*. 88th Air Base Wing Public Affairs, USAF. 11 May 2017. Retrieved 14 February 2022.

Goblet, Valentine and others. "Identifying Phases of Flight in General Aviation Operations," *15th AIAA Aviation Technology, Integration, and Operations Conference*. Dallas TX: American Institute of Aeronautics and Astronautics, 2015.

Hamilton, Shane P. and Michael P. Kreuzer. "The Big Data Imperative: Air Force Intelligence For The Information Age," *Air & Space Power Journal*, vol. 32, no. 1 (2018).

Korvesis, Panagiotis. *Machine Learning for Predictive Maintenance in Aviation*. Ph.D. dissertation. Paris-Saclay University, Paris, 2017 (2017SACLX093).

Kozik, Piotr and Jaroslaw Sep. "Aircraft Engine Overhaul Demand Forecasting Using ANN," *Management and Production Engineering Review*, vol. 3, no. 2 (2012).

Larose, Daniel T. and Chantal D. Larose. *Data Mining and Predictive Analytics* (2nd edition). Hoboken, NJ: Wiley, 2015.

Levy, Joshua J., and A. James O'Malley. "Don't Dismiss Logistic Regression: The Case for Sensible Extraction of Interactions in the Era of Machine Learning," *BMC Medical Research Methodology*, vol. 20, no. 1 (December 2020).

Li, Wei Tse and others. "Using Machine Learning of Clinical Data to Diagnose COVID-19: A Systematic Review and Meta-Analysis," *BMC Medical Informatics and Decision Making*, vol. 20, no. 1 (December 2020).

"Machine Learning." *Wikipedia.org*. Wikimedia Foundation, Inc. 15 January 2022. Retrieved 15 January 2022.

Michie, Donald and others. *Machine Learning, Neural and Statistical Classification*. New York: Ellis Horwood, 1994.

Office of the Deputy Assistant Secretary of Defense for Systems Engineering. *DoD Digital Engineering Strategy*. Washington: OUSD A&S, June 2018.

Quinlan, J. R. *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann Publishers, 1993.

Ripley, Brian D. *Pattern Recognition and Neural Networks*. Cambridge: Cambridge University Press, 1996.

Samuel, Arthur L. “Some Studies in Machine Learning Using the Game of Checkers.” *IBM Journal of Research and Development*, vol. 44, no. 1.2 (1959).

Shaukat, Kamran and others. “Performance Comparison and Current Challenges of Using Machine Learning Techniques in Cybersecurity.” *Energies*, vol. 13, no. 10 (May 2020).

Tarik, Mouna and Khalid Jebari. "Maintenance Prediction by Machine Learning: Study Review of Some Supervised Learning Algorithms," *Proceedings of the 2nd African International Conference on Industrial Engineering and Operations Management*. Harare, Zimbabwe: IEOM Society International, 2020.

Trani, A. A. and others. “A Neural Network Model to Estimate Aircraft Fuel Consumption,” *Proceedings of the AIAA 4th Aviation Technology, Integration, and Operations Forum*, vol. 2. Chicago IL: AIAA, 2004.

Vidrine, Eileen, Chief Data Officer, USAF. "The Power of Data." Presentation at Air Force Information Technology and Cyberpower Conference (AFITC), Montgomery AL. 27 August 2018.

Wheeler, Winslow. "The MQ-9's Cost and Performance." *Time.com*. 28 February 2012. Retrieved 15 January 2022.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From – To)*
24-03-2022	Master's Thesis	January 2021 – March 2022

TITLE AND SUBTITLE

Telemetry Data Mining For Unmanned Aircraft Systems

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Yu, Li N., Captain, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)
Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way, Building 640
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENV-MS-22-M-275

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Office of Scientific Research
Erik Blasch, Ph.D.
Air Force Research Laboratory, Arlington, VA 22203
(erik.blasch.1@us.af.mil)

10. SPONSOR/MONITOR'S ACRONYM(S)

AFOSR

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
DISTRUBTION STATEMENT A. APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES
This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

14. ABSTRACT
With ever more data becoming available to the US Air Force, it is vital to develop effective methods to leverage this strategic asset. Machine learning (ML) techniques present a means of meeting this challenge, as these tools have demonstrated successful use in commercial applications. For this research, three ML methods were applied to a unmanned aircraft system (UAS) telemetry dataset with the aim of extracting useful insight related to phases of flight. It was shown that ML provides an advantage in exploratory data analysis and as well as classification of phases. Neural network models demonstrated the best performance with over 90% accuracy in classifying of UAS phases of flight. Categorical and Regression Trees (CART) also performed well, whereas C5.0 is less suited for this task. In addition, several interesting patterns were uncovered within the dataset, which can aid UAS operators in identifying mission anomalies and atypical system operation.

15. SUBJECT TERMS
Data Mining, Machine Learning, Unmanned Aircraft Systems, Phase of Flight

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Brent T. Langhals
U	U	U	UU	76	19b. TELEPHONE NUMBER *(Include area code)* (937) 255-3636, ext. 7402 (brent.langhals@afit.edu)