Enhanced Filtering of False Alarm for the 5th Generation Fighter Aircraft using Machine Learning Classification

by Nathan Her

B.S. in Aerospace Engineering, May 2007, University of Illinois at Urbana Champaign
M.Eng. in Aerospace Engineering, May 2012, University of Maryland at College Park

A Praxis submitted to

The Faculty of
The School of Engineering and Applied Science
of The George Washington University
in partial fulfillment of the requirements
for the degree of Doctor of Engineering

August 31, 2019

Praxis directed by

Michael Grenn
Professorial Lecturer of Engineering Management and Systems Engineering

Amir Etemadi
Associate Professor of Engineering and Applied Science

The School of Engineering and Applied Science of The George Washington University certifies that Nathan Her has passed the Final Examination for the degree of Doctor of Engineering as of July 22, 2019. This is the final and approved form of the Praxis.

Enhanced Filtering of False Alarm for the 5th Generation Fighter Aircraft using Machine Learning Classification

Nathan Her

Praxis Research Committee:

Michael Grenn, Professorial Lecturer of Engineering Management and Systems Engineering, Praxis Director

Ebrahim Malalla, Visiting Associate Professor of Engineering and Applied Science, Committee Member

Amir Etemadi, Associate Professor of Engineering and Applied Science, Committee Member

© Copyright 2019 by Nathan Her
All rights reserved

Dedication

I dedicate this praxis to my wife who has endured and sacrificed her precious time and energy so I can concentrate on finishing the coursework and this praxis. The past two years have been a challenge for my wife to take care of one toddler son and an infant son every Saturday without my support from 9am to 5pm while I was taking the classes. Finishing this degree would not have been possible without my wife's support. I am very grateful that my wife has been cheering me up and encouraging me to complete this praxis.

Acknowledgements

I wish to express my sincere thanks to my advisors Dr. Michael Grenn and Dr. Amir Etemadi for giving me the guidance in developing and writing this praxis. When I initially came up with a praxis topic that I was interested in but there was not enough data to support the topic. Both advisors strongly recommended that I come up with a new topic and that literally saved my degree because I definitely would not have been able to collect all the necessary data in time to complete the praxis with that original topic. Ever since the day that I came up with a new topic for this praxis, both advisors have provided very good feedbacks during the monthly meetings, which helped me tremendously in developing the hypotheses.

I also would like to thank the 5th generation fighter aircraft Prognostics and Health Management (PHM) Integrated Product Team (IPT) lead for allowing me to write on this topic and reviewing this praxis for public release. Special thanks to another PHM engineer from the 5th generation fighter aircraft team for the valuable brain storming sessions and help with parsing the data. This praxis could not have been possible to start without the help from the PHM engineer.

Abstract of Praxis

Enhanced Filtering of False Alarm for the 5th Generation Fighter Aircraft using Machine Learning Classification

Prognostics and Health Management (PHM) has become an important capability for the 5th generation fighter aircrafts such as F-35 Lightning II or F-22 Raptor to ensure the safety of flight and reduce the life cycle costs. The on-board PHM System of a 5th generation aircraft uses sensors, software and computing capability to detect component or system faults and generate health reporting codes. Similar to the legacy military aircraft platforms, the PHM system also generated a large number of false alarms, which can reduce the aircraft availability and increases the Operating and Support (O&S) costs. Studies have been conducted in the past to identify the root causes of false alarms (Ungar 2015) and presented various mitigation strategies including supervised machine learning techniques (de Padua Moreira & Nascimento Jr 2012) to reduce the number of false alarm at the source. However, the goal of the research in this praxis is to train machine-learning classification models using the PHM data and the maintenance records to create an enhanced filtering capability based on the lessons-learned from the operating units. The results of the research have shown that the machine-learning classification model called, RUSBoostedTrees, exceeded the human performance by 4% and 9% in identifying false alarm and faults, respectively, along with the savings of 11 minutes per maintenance event. The results also have shown that the trained algorithm will reduce the diagnostics performance gaps between different operational sites. Lastly, the model can be trained with a fewer number of maintenance events than what it would take human to achieve the equal or better performance in distinguishing false alarms and faults.

Table of Contents

Dedication iv

Acknowledgements v

Abstract of Praxis vi

List of Figures x

List of Tables xi

List of Symbols xii

List of Acronyms xiii

Chapter 1—Introduction 1

1.1 Background 1

1.2 Research Motivation 7

1.3 Problem Statement 11

1.4 Thesis Statement 12

1.5 Research Objectives 13

1.6 Research Questions and Hypotheses 13

1.7 Scope of Research 14

1.8 Research Limitations 15

1.9 Organization of Praxis 16

Chapter 2—Literature Review 18

2.1 Introduction 18

2.2 Root Causes of False Alarms 18

2.3 Machine Learning Classification Models 21

2.3.1 Logistic Regression 23

2.3.2 Support Vector Machine (SVM) 24
2.3.3 Decision Tree 25
2.3.4 Ensemble 26
2.4 Use of Machine Learning to reduce False Alarms 27
2.5 Summary and Conclusion 31
Chapter 3—Methodology 32
3.1 Introduction 32
3.2 Data Preparation 32
3.3 Input Parameters (Predictors) 34
3.3.1 Type/Version Effectivity (TVE) 35
3.3.2 Software Data Load (SDL) 36
3.3.3 Total Number of HRCs from the same power cycle 36
3.3.4 Health Reporting Code (HRC) 37
3.3.5 Logistics Control Number (LCN) Group 37
3.3.6 Health Assessment Method Indicator (HAMI) 38
3.3.7 Type of Test 38
3.3.8 # of HRCs within +/- 2 min window of HRC assertion 38
3.3.9 Time Occurred 39
3.3.10 SET, RES and RST Counter 39
3.3.11 Max Nz, Max Hp and Max Vcas 40
3.4 Research Procedures 40
3.5 Data Analysis 43
3.5.1 H1: Accuracy in Identifying False Alarm 43

3.5.2 H2: Accuracy in Identifying Fault 45
3.5.3 H3: Time to identify False Alarm 45
3.5.4 H4: Performance Variance by Operational Sites 46
3.5.5 H5: Accuracy Improvement Rates 47
Chapter 4—Results 49
4.1 Introduction 49
4.2 ML vs. Human in identifying false alarm 49
4.3 ML vs. Human in identifying Fault 51
4.4 Time Wasted on Troubleshooting False Alarms 53
4.5 Performance Variance by Operational Sites 55
4.6 Machine Learning Improvement Rate 59
4.7 Other Studies to Find the Optimal ML Algorithms and Settings 64
Chapter 5—Discussion and Conclusions 68
5.1 Discussion 68
5.2 Conclusions 75
5.3 Contributions to Body of Knowledge 76
5.4 Recommendations for Future Research 76
References 78
Appendix A 81

List of Figures

Figure 1-1. PHM Data Flow. 4

Figure 1-2. Discrepancy between Initial and Final FID Codes by Operational Sites 8

Figure 2-1 Causes of False Alarms 19

Figure 2-2 Two-Dimensional Pattern Recognition 22

Figure 2-3. SVM Maximum Margin (Kotsiantis 2007) 24

Figure 2-4. Decision Tree Pruning 25

Figure 2-5. Multi-Layer Perceptron 29

Figure 3-1. k-fold Cross-Validation 41

Figure 4-1. Machine Learning vs. Human in Identifying FA 50

Figure 4-2. Machine Learning vs. Human in Identifying Fault 53

Figure 4-3. Levene's Test for human performance for FA across operational sites 56

Figure 4-4. Levene's Test for human performance for fault across operational sites 57

Figure 4-5. Machine-Learning vs. Human in identifying Faults by Operational Sites 58

Figure 4-6. Machine-Learning vs. Human in identifying FA by Operational Sites 59

Figure 4-7. Incorrect Assessment Rate per 100 Maintenance Events 60

Figure 4-8. Incorrect Assessment Rate per 100 ME's prior to May 2016 61

Figure 4-9. Incorrect Assessment Rate per 100 ME's post May 2016 62

Figure 4-10. Incorrect FID Scoring Rates (Fault) vs. Number of data points 63

Figure 4-11. Incorrect FID Scoring Rate (FA) vs. Number of data points 64

Figure 4-12. Fault Identification Accuracy vs. Data Sampling Ratios 65

Figure 4-13. FA Identification Accuracy vs. Data Sampling Ratios 65

Figure 5-1. Trend Change in FID Code Scoring in 2016 73

List of Tables

Table 1-1. FID Code Samples 6

Table 1-2. Total Maintenance Man-Hours (MHH) Wasted on False Alarm 10

Table 3-1. Exported PHM Data Elements 33

Table 3-2. Exported FRACAS Data Elements 34

Table 3-3. Processed Data for ML Training and Validation 35

Table 3-4 List of Classification Models 42

Table 3-5. Terms used to define sensitivity, specificity and accuracy 45

Table 4-1. Human Performance Level from FRACAS 49

Table 4-2. Machine Learning Performance Level 51

Table 4-3. Machine-Learning vs. Human Accuracies 52

Table 4-4. Total Maintenance Man-Hours spent on False Alarms 54

Table 4-5. Time required to train and test the machine-learning algorithms 55

Table 4-6. Top 5 RUSBoostedTrees Results with Different Combinations of Predictors 67

Table 5-1. RUSBoostedTrees vs. BaggedTree Results 69

Table 5-2. Human Performance with Corrected Number of FA events 71

Table A-1. Number of Discrepancies between FID (I) and FID (U) 81

Table A-2. Various FA to Fault Data Sampling Ratio 81

Table A-2. Various FA to Fault Data Sampling Ratio (Cont'd) 83

Table A-3. Various Data Sample Sizes 84

Table A-4. Top 20 RUSBoostedTrees Results with Various Predictors 85

Table A-5. Sample Machine Learning Input Data 86

List of Symbols

Hp	Pressure Altitude
$N2$	High Pressure Compressor Speed
Nz	Vertical G-load
$Vcas$	Cockpit Airspeed

List of Acronyms

AV	Air Vehicle
AID	Automatic Interaction Detection
CART	Classification and Regression Trees
CEL	Check Engine Light
DoD	Department of Defense
DNA	Deoxyribonucleic acid
ECG	Electrocardiogram
FN	False Negative
FP	False Positive
HAMI	Health Assessment Method Indicator
HMC	How Malfunctioned Code
HRC	Health Reporting Code
ICU	Intensive Care Unit
IFTTT	If This, Then That
IPP	Integrated Power Package
IPT	Integrated Product Team
LCN	Logistics Control Number
MCR	Mission Capable Rate
ML	Machine Learning
MLP	Multi-layer Perceptron
MTBF	Mean Time Between Failures

OBD-II	On-Board Diagnostic II
PHM	Prognostics and Health Management
R&R	Remove and Replace
RUS	Random Under-sampling
SDD	System Development and Demonstration
SDL	Software Data Load
SVM	Support Vector Machine
TVE	Type/Version Effectivity
TN	True Negative
TP	True Positive
UAI	Unique Aircraft Identifier
US	United States
VBA	Visual Basic for Applications
WO	Work Order

Chapter 1—Introduction

1.1 Background

Prognostics and Health Management (PHM) has become an important capability for the 5th generation fighter aircrafts such as F-35 Lightning II or F-22 Raptor to keep ensuring the safety of flight and reduce the life cycle costs (McCollum 2011). PHM provides the fault detection and isolation, and life management capabilities for aircraft maintainers to easily identify and address the current and impending failures of the aircraft. The 5th generation aircrafts are equipped with a suite of on-board sensors such as pressure sensors, voltmeters, flow meters, temperature sensors, strain gauges, corrosion sensors, accelerometers, etc. to detect anomalies and malfunctions and to measure the aircraft performance parameters in order to analyze the degradation status of the aircraft. Based on the information captured by the on-board PHM system, the aircraft generates a series of fault codes called Health Reporting Codes (HRCs) to let the pilot and the maintainers know the health of the aircraft and what components to troubleshoot or replace.

The concept of an HRC is similar to the On-Board Diagnostic II (OBD-II) trouble codes, also known as Check Engine Light (CEL) codes, which cars manufactured since 1996 (US Environmental Protection Agency 2003) can generate to provide the information about the various conditions of the car. A car can typically generate CEL codes in the order of a few hundred, which the total numbers can vary depending on the model and the manufacturer of the car, whereas an 5th generation aircraft has thousands of different combinations of HRCs. An HRC is consisted of 13 digit numbers that can identify what reporting entity generated the fault code. While the large number of HRCs

enables the 5th generation aircraft to have the high degree of fault detection coverage, it could also generate a large number of HRCs that incur manpower in determining no action is needed. Maintainers often spend hours troubleshooting and adjudicating HRCs just to realize that there was nothing wrong with the components. This is called a false positive or a false alarm. One of the root causes for false alarms is that a component can have its HRC triggering mechanism set up so sensitive that it reports a problem whenever a subtle anomaly in the component is detected or if the monitored parameter exceeds a very conservative threshold while the component could still be in perfectly working condition. This approach will increase the fault detection rate at a cost of generating more false alarms.

It may be sometimes desirable to have a high fault detection rate even it takes the false alarm rate to be high in order to keep the aircrafts safe in the sky, especially if the aircraft acquisition program is still going through the development phase. These triggering points can be fine-tuned and matured throughout the life of the program. In 2015, one of the 5th generation aircraft platform had the fault detection coverage of 84% and the fault detection rate of 85% (Gilmore 2015). The fault detection rate has been continuously improving since the performance was reported in the 2015 DOT&E report through the PHM maturation processes and as the newer on-board and off-board software were implemented to fix the known software issues.

The high false alarm rate problem is not just limited to the 5th generation aircrafts but also various military aircraft platforms have suffered from having high false alarm (FA) rates. F/A-18C and V-22, have the false alarm rates of 88% (Shannon & Knecht 2010) and 92% (Westervelt 2006), respectively. According to the transcript from the US

Congressional Hearing in 2015 regarding one of the two 5th generation aircraft acquisition program status, roughly 90% of the HRCs generated on the earlier Block 1B/2A aircrafts were false alarms (U. S. Cong. House. Comm. on Armed Services House of Representatives 2015). The false alarm rate for the latest Block 3F aircrafts has greatly improved since 2015 but due to the sensitivity nature of the 5th generation aircraft program, the latest and actual diagnostics performance numbers were not available for public release. Having a high false alarm rate has three major adverse impacts as follows:

- Increases the O&S cost due to maintainers having to spend time on troubleshooting and removing & replacing (R&R) components that are in perfectly working condition.
- Creates a crying-wolf situation where maintainers lose their trust in the PHM system and start ignoring fault codes that could lead to a safety issue.
- Creates the supply shortage in the spares pool since good working parts were either scrapped or sent to the supplier for a further analysis after being removed and replaced with another part.

There are many possible causes of false alarms. In order to understand the causes of false alarms on the 5th generation air system, one must understand how HRCs are generated. HRC are generated in either of the two ways. One source of HRCs is air vehicle (AV), also known as on-board generated HRCs, and the other source is called off-board generated HRCs. The 5th generation aircraft has a suite of on-board sensors that collect the necessary information to diagnose the health of aircraft components and systems. Many of the electronic components have a capability called Built-in Test (BIT) to check for shorts in the circuit or the exceedance of the parameters such as voltages,

temperatures, etc. beyond the designed operating limits. If an anomaly or exceedance is detected, the BIT generates a message indicating the fault and the message is then validated by the on-board PHM software and translated into a corresponding HRC. A large number of the BIT functionality is embedded in the electronics controller of the component and the BIT assessment results are sent to the data recorder as binary values, where "0" indicates that no fault was detected and "1" indicates that a fault was detected.

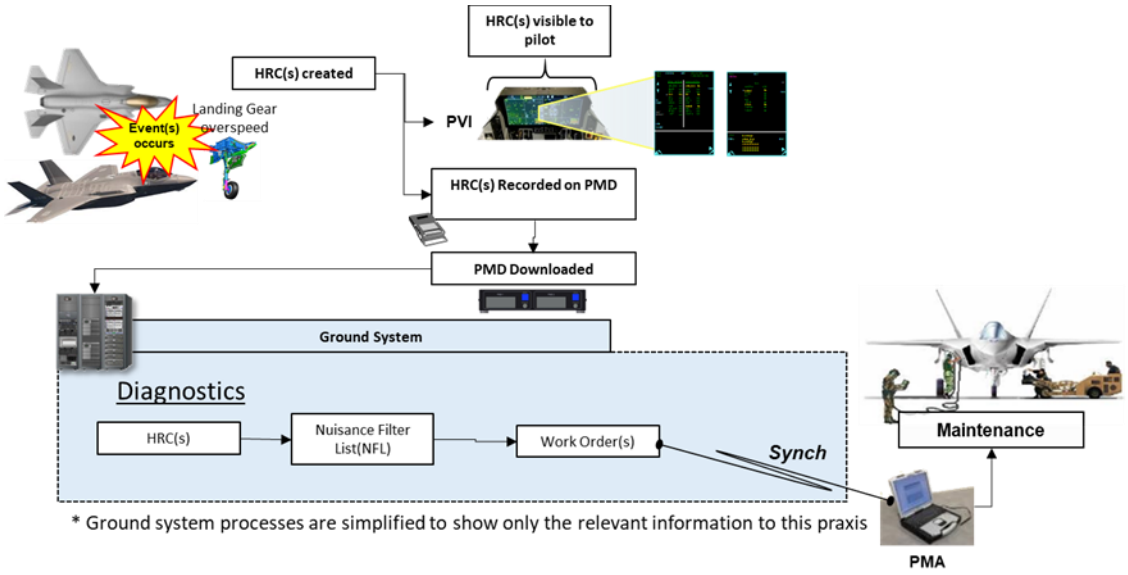


Figure 1-1. PHM Data Flow.

As shown in Figure 1-1, once an aircraft is powered down, PHM data is downloaded via the portable memory device (PMD) into the ground system. The off-board PHM runs a series of analyses to determine the integrity of the downloaded PHM data and calculate the usage and the exceedances beyond the operating envelopes to trigger additional HRCs that will create work orders for maintenance actions. The PHM data is consisted of engineering data, life expended data and maintenance data. The engineering data is the collection of sensor readings (e.g. altitudes, pressures, temperatures, G-loadings, strain gauge readings, etc.) and event-driven snap shots of the

data related to HRCs to aid engineering investigations or analyses as needed. The life-expended data is the data used to calculate the accrued usage or the condition of life-tracked components. The maintenance data includes the information about the software configuration of the aircraft, HRCs and maintenance relevant information to trigger maintenance actions. The off-board PHM can also generate HRCs based on the analyses using the downloaded PHM data. In this praxis, the enhanced filtering will focus on reducing the number of air vehicle generated HRCs that are presented to the maintainers for action.

Once the portable memory device (PMD) is downloaded and the HRCs are made available in the ground system, the off-board PHM system filters certain HRCs that are known and confirmed to be always false alarms using a list called the nuisance filter list (NFL). For each HRC that is not filtered through the NFL, work orders are automatically created to drive maintenance actions to address the detected faults. A typical work order contains the list of items to inspect and replace as well as the references to the step-by-step instructions called Joint-Services Technical Data (JTD). After completing the maintenance actions prescribed in the work order, maintainers enter the maintenance logs into the ground system as to whether the prescribed maintenance actions has resolved the issues or not using the performance rating system called fault isolate and detect (FID) codes. A few samples of FID codes are shown in the Table 1-1.

Table 1-1. FID Code Samples

FID	Definition
D10	PHM automatic correct detection and correct isolation to a single LRC
D20	PHM automatic correct detection and correct isolation to two or three LRCs
D30	PHM automatic correct detection but incorrect isolation or isolation to great than three LRCs
D32	PHM automatic correct detection but no correct solution found
D35	PHM automatic correct detection but no solution available
D40	Diagnostic False Alarm
D42	Diagnostics False Alarm with known solution forthcoming
D50	Diagnostics False Alarm with confirmed fix (validated on aircraft)
D80	PHM correct automatic detection and correct isolation to one propulsion module
D82	PHM correct automatic detection, but incorrect isolation or isolation to greater than one propulsion module
P40	Prognostic Predicted Time to Failure False Alarm
X99	PHM Non-Pertinent

These FID codes are sent to the maintenance action recording system called Failure Reporting, Analysis and Corrective Action System (FRACAS), which is used to assess the performance of the PHM system. There are approximately 17 different FID codes including the ones for prognostics efficiency but for the purpose of the study in this Praxis, only the FID codes relevant to true positive (D10, D20, D30, D32, D35, D80 and D82) and false positive (D40, D42, D50, and P40) will be used and renamed as “Fault” and “False Alarm”, respectively. FID codes are initially entered in by the maintainers and captured in the FRACAS as FID (I) and then a group of engineers and subject matter experts (SMEs) routinely gather together to review the historical records to correct and update the FID (I) codes as FID (U) as necessary. This is a very valuable information to

understand what percentage of the maintenance actions were related to false alarm and what percentage of the initial assessment by the maintainers were different from the updated FID codes, which will be used to prove one of the hypotheses in the upcoming chapters. FID (U) will be used as the response (a.k.a. output parameter) for the supervised machine learning algorithms.

1.2 Research Motivation

As it was stated in the 2015 DOT&E report and the transcript from the 2015 congressional hearing on one of the two 5th generation aircrafts, approximately 90% of the AV generated HRCs were false alarm from the earlier 5th generation aircrafts. Imagine that a car owner sees a Check Engine Light (CEL) on. The owner decides to take a day off from work to take the car to a repair shop and the car mechanic finds nothing wrong with the car. The repair shop mechanic clears the CEL codes. The next day, the car owner sees the CEL lit again and takes another day off from work to take the car back to the repair shop and the shop finds no issues again. If the car owner has to go through this multiple times, would the car owner take the car back to the repair shop when the CEL is lit again? This same behavior was developing in the 5th generation aircraft maintenance community. A large percentage of the HRCs were reported in the ground system for the maintainer to take actions and later turned out to be false alarms after spending their time and effort to adjudicate the HRCs. After a few iterations of wasting time on false alarm HRCs, the operational sites started developing their own list of HRCs to ignore and started cancelling or deferring the work orders based on the lessons-learned in their operating units. The reason for deferring work orders is that the ground system automatically groups the work orders if there is an open work order associated with the

same HRC from the previous flights or ground operations and does not create a new work order since the existing work order already exists. The reason for doing this is that it takes a few minutes just to cancel a work order created from a known false alarm. Using locally developed HRC cheat sheet was partially a contributing factor for the variances in PHM performance throughout the different operational units. The following Figure 1-2 shows the percentage of FID (I) that was different from the updated FID (U) and grouped by the operating sites to show that diagnostics performance numbers varied across the operational sites while the aircraft and software configurations were similar or identical.

Figure 1-2 shows how many maintenance events were incorrectly assigned with FID (I) codes and then were later switched from non-FA to FA or from non-fault to fault by the SMEs where the data is from January 2015 to May 2018.

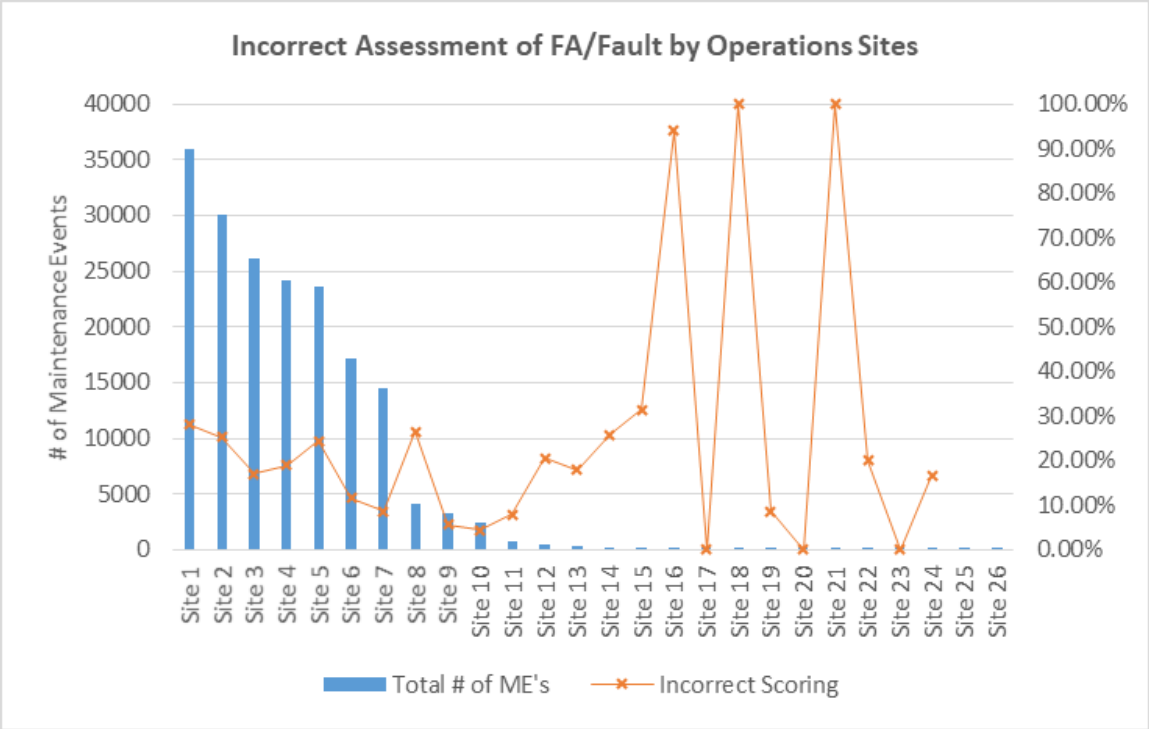


Figure 1-2. Discrepancy between Initial and Final FID Codes by Operational Sites

The names of the operational sites were renamed as Site 1, 2 and so on to avoid disclosing the program sensitive information. There are a couple of developmental test (DT) flight sites included in the data where they do not follow the similar maintenance practices in terms of using FID codes because the DT aircrafts are often instrumented with real-time telemetries and also there are engineers on site to diagnose in real-time. Excluding the data points from the two DT sites, the average percentage of the discrepancy (i.e. switching from non-fault to fault, non-FA to FA) between the initial and final FID codes is 19.3%. One of the 5th generation fighter aircraft program currently relies on manual labor to review and update the discrepancies in the FRACAS data. Without the manual review of the FRACAS data for accuracy, the PHM performance metrics calculated using the FRACAS data would be inaccurate and may not reflect the true performance of its diagnostics capabilities.

On top of the cost associated with hiring engineers to review the FRACAS data, another motivation for this praxis is that the maintenance man-hours (MMH) spent on troubleshooting faults before realizing that nothing was wrong with the component or the aircraft can be directly translated into the avoidable cost in terms of the dollar amount. Based on the 2018 USAF reimbursement rates for military maintainers from E-4 ($74,663) to E-6 ($109,320) and the standard available time of 167.9290 hours/mo (Tanner 1988), the total MMHs associated with spending time on false alarms were worth 15.55 Man-Years. Using the average reimbursable rate between E-4 and E-6, one of the 5th generation aircraft fleet could have saved approximately $1.43M in the operating and support (O&S) expense as shown in Table 1-2. The expense associated with spending

labor hour on false alarms will increase sharply as more 5th generation aircrafts are delivered and fielded in the upcoming months.

Table 1-2. Total Maintenance Man-Hours (MHH) Wasted on False Alarm

Total # of Maintenance Events (MEs)	183,984
# of False Alarms	175,305
Total Maintenance Man Hours (MMHs) Spent on troubleshooting false alarms	31,334.86
Maintenance Man-Year	15.55
Avoidable O&S Cost ($)	$1.43M

Over the last decade, there has been numerous studies published in the machine-learning field and the algorithms are now readily available in commercial statistical tools like R or MATLAB. The classification problem using machine-learning algorithms is a great heuristic approach to solve the false alarm mitigation issue. Theoretically, it would be ideal to tackle each of the false alarm issues at its root cause by investigating the sensor data and the algorithms at the source of BIT, but due to the enormous amount of false alarms, there is constantly a backlog of issues to be resolved because engineering and financial resources are limited. It is not often realistically possible to allocate the necessary resources immediately to find the root cause and resolve the issue for every false alarms due to the other competing higher priorities.

In addition, certain software fixes cannot be implemented via an ad-hoc distribution of software update as if Microsoft Windows security patches are distributed on demand. Certain fixes will have to be rolled into the next major software release that

could take up to six months to a year. Whereas machine learning algorithms can be trained weekly or even daily and incorporate the latest lessons-learned.

Most importantly, from having many false alarms, there could be a situation where a true positive HRC could be mistakenly ignored from the habit of ignoring certain set of known false alarm HRCs without realizing that the false alarm issue had been fixed in the latest software release. This could lead to a safety issue.

The motivation for the research in this praxis is driven by the necessity for a better way to capture the lessons learned not just from local units but also from all the operating sites and to apply the lessons learned immediately rather than waiting for months for the next major software to come. The goal is to implement the lessons-learned in way that is more systematic so that all operational units can utilize the most update-to-date information in order to accurately filter true false alarm HRCs without affecting the safe operations of the 5^{th} generation aircrafts.

1.3 Problem Statement

Approximately 90% of the fault codes that the 5^{th} generation On-board PHM system generated in 2015 were false alarms, which generated unnecessary work orders for maintainers to either spend the unnecessary time to inspect for faults or administratively close the work order (U. S. Cong. House. Comm. on Armed Services House of Representatives) that has resulted in a decrease in aircraft availability (Vandawaker et al 2015) and an increase in the operating and support (O&S) costs (Hoyle et al 2015).

According to the transcript from the Congressional Hearing (U. S. Cong. House. Comm. on Armed Services House of Representatives), the Program Executive Officer (PEO) of a 5th generation aircraft program in 2015 stated that 50% of the HRCs generated on the Block 1B/2A aircrafts were automatically flagged as invalid and filtered using the Nuisance Filter List (NFL) in the off-board system. Of the 50% that made through the NFL, 80% of the HRCs were false positives or false alarms, which means that 90% of the aircraft generated HRCs were false alarms.

1.4 Thesis Statement

False alarms can be filtered by utilizing machine learning algorithms in the ground-based PHM system to identify and assess the signs of false alarms from the 5th generation aircraft Prognostics and Health Management (PHM) data and Failure reporting, analysis, and corrective action system (FRACAS) data and automatically classify the detected health indications as false positive or true positive before presenting all Air Vehicle-generated fault codes to aircraft maintainers.

In this praxis, the purpose of the machine learning algorithms is not to eliminate the false alarms at the source where the aircraft components generate binary BIT messages, because the majority of the source data or the algorithm for the BIT is the suppliers' proprietary information and readily not available to the end users. Instead, the machine learning algorithms will be used to train the classification model to mimic what an experienced maintainer would do to avoid spending any maintenance hours on false alarms with the knowledge of the history of the fault codes and the false alarm indicators but to apply the knowledge in more consistent and systematic way using machine learning models.

1.5 Research Objectives

The main goal of the praxis is to show that machine learning can be used to train a classification model that can distinguish faults and false alarms based on various contributing factors derived from the PHM data and FRACAS data. The first objective of the praxis is to understand the benefits of using machine learning classification models to identify false alarms. The second objective is to understand the data sampling requirements for training the machine-learning algorithms efficiently. The last objective is to find out what machine learning technique is best suited for solving the classification problem for the 5^{th} generation aircraft false alarm issue.

1.6 Research Questions and Hypotheses

In order to understand the benefits of using machine-learning classification algorithms, the following research questions were asked.

RQ1: Will machine-learning classification model yield a better result at identifying false alarms using the contributing factor elements from the PHM data?

RQ2: Can machine-learning algorithms be trusted for safe flight operations?

RQ3: How much maintenance man-hours will be saved from not having to spend the time on troubleshooting false alarms?

RQ4: Can machine-learning algorithms be used to decrease the performance variance in correctly identifying false alarms between maintainers at each operational site?

The second objective will be investigated by asking the following research questions.

RQ5: How many data points do machine-learning algorithms need in order to surpass the human performance in distinguishing false alarms and faults?

From the research questions above, the following hypotheses were derived to test the research questions.

H1: Trained classification models using machine-learning algorithms will identify false alarm at a higher accuracy than human maintainers will.

H2: Trained classification models using machine-learning algorithms will identify faults at a higher accuracy than human maintainers will.

H3: Trained classification models using machine-learning algorithms will take less time to identify whether an HRC is a false alarm or not.

H4: Trained classification models using machine-learning algorithms will decrease the performance variance in distinguishing faults and false alarms between each operational site while maintaining a better performance than human.

H5: The accuracy in distinguishing faults and false alarm using machine-learning classification algorithms will improve at a faster rate than the rate at which human maintainers will improve over time.

1.7 Scope of Research

The scope of the study is to train the classification models using a variety of different machine learning algorithms to determine which algorithm yields the most accurate predictions of faults and false alarm. The best performing algorithm will be used to check the sensitivity of each input parameter to see how varying input parameters will

affect the accuracy of the classification model. Then, the best performing algorithm along with the best set of input parameters will be used to calculate the final performance metrics for machine learning algorithms and the metrics will be compared to the human performance metrics.

However, the detailed and qualitative investigation and understanding as to why certain input parameters and certain combinations worked better or worse than the others are not part of the scope of the praxis. This praxis will cover the high-level rationales as to why certain input parameters were chosen and why certain machine learning algorithms worked better in the chapter 2.

1.8 Research Limitations

One of the biggest limitations of this study is the accuracy of the labeled data for the supervised learning. FID (U) codes are the output parameter that is used to assign each data point as fault or false alarm during the training of machine-learning algorithms. Generally, FID (U) codes are more accurate than FID (I) because the FID (U) codes go through the SME reviews and the FID codes are updated if they were incorrectly assigned initially. However, the SME reviews also involve human mistakes and not all FID (I) codes may go through the same rigorous reviews. Machine learning algorithms can only be as good as the accuracy of the training data.

Another limitation is that the machine learning algorithms will not leverage every single PHM sensor data to find the optimal set of input parameters due to the enormous size of the PHM data that needs to be processed. A single flight can generate a few hundred Mega-bytes (MBs) of PHM data. In this praxis, a large number of power cycles (e.g. flights, ground runs, etc.) were processed which can easily go over tens of Tera-

bytes (TBs) if every signal was parsed out. Due to the limited computing power and the required storage size, only a small set of pre-determined PHM parameters were parsed and processed.

In addition, there was a limitation in choosing the set of input parameters due to the availability of the necessary documentation. It would have been ideal to tackle the issue at the source level where BIT signals are generated. At least the full understanding of how each HRC is linked to which sensor readings would have helped in running anomaly detection algorithms and use the outcome as the input parameters for the classification model. However, only the partial mapping between HRCs and the source data was available at the time of writhing this praxis so the partial information could not be leveraged in training machine learning algorithms to avoid generating a bias in the classification model. One of the input parameters that was initially considered was the reliability number such as the mean time between failures (MTBF) or the accrued usage of a component. It was going to be a good indicating factor for a true fault because there would be a higher chance for a part to fail as the accrued usage of the part approaches the end of its life limit. However, not all components are life-tracked parts, which means that the reliability numbers or the accrued life usage is not readily available. It can be manually calculated by tracking each part number for the installation and the removal dates but this is a very cumbersome and time-consuming task.

1.9 Organization of Praxis

The rest of the praxis is organized as follows: The chapter 2 will provide the review of the previous studies to understand the nature of the aircraft false alarms. Different applications of machine learning will be reviewed as well as its application on

reducing false alarms. Different machine learning algorithms were tested to find the optimum algorithm for this classification problem, which will be covered in the next chapters. The praxis will cover briefly how each algorithm works. The chapter 3 will present the methodology used for this praxis to show how machine-learning algorithms can improve in distinguishing false alarms and faults. This chapter will also discuss in detail how and what PHM data was prepared and processed as the input parameters for the machine learning algorithms. The results of the machine learning algorithms will be presented and discussed in the chapters 4 and 5. Lastly, the chapter 5 will end with the recommendations for future research.

Chapter 2—Literature Review

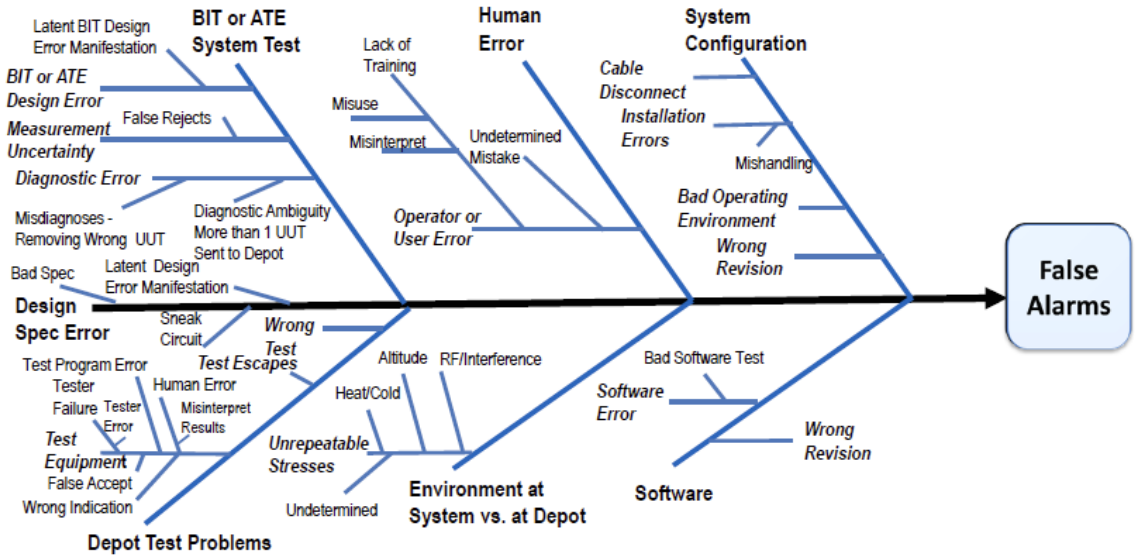
2.1 Introduction

There are substantial researches in the past to identify the causes of aircraft generated false alarms (Ungar 2015) and presented various mitigation strategies including supervised machine learning using maintenance data to detect anomalies (de Padua Moreira & Nascimento Jr 2012). While these studies offer valuable insight for engineers and component suppliers in detecting false alarms at the individual component level, they will not help maintainers to distinguish false alarms for the entire set of HRCs, because it will be very challenging to implement the anomaly detection accurately and consistently throughout the entire aircraft components to reduce false alarms. Not every components are from a single supplier so each company might have its own way of solving problems and the proprietorship might hinder from implementing certain knowledge and technologies consistently throughout the different suppliers. In addition, not all components were designed to have the built-in component controller to support the same level of complexity in detecting anomalies. Since a large portion of the source data used in the BIT is not recorded on the PMD, there is not enough data to run such anomaly detection algorithms off-board for every components. Even if the data is available, HRC triggering mechanisms and limits are often proprietary information that is not well known to the end users.

2.2 Root Causes of False Alarms

As it was mentioned earlier in the chapter 1, a false alarm occurs when the aircraft reports a problem with a component while there is actually nothing wrong with the component. There is a term called no-fault-found (NFF) which FA is a part of NFF.

However, NFF also includes intermittent problems where the aircraft reports a problem and the maintainers could not find anything wrong with it because the failure mode was intermittent and can only occur in a certain operating environment such as loose cables or pulling high-G maneuvers. These intermittent problems are often reported as FA because the maintainer did not find any discrepancies and the component supplier may not be able to duplicate (cannot duplicate, CND) the fault and returns the component as retest OK (RTOK). The RTOK information was not available at the time of writing this praxis, so the number of false alarms was solely based on the FID information and How Malfunctioned Code (HMC) of 799 (i.e. No defect found or cannot duplicate) from the FRACAS data.



Note. Reprinted from “Causes and Costs of No Fault Found Events”, by Ungar, L., (2015), IPC APEX EXPO 2015.

Figure 2-1 Causes of False Alarms

The causes of false alarms are well summarized in the fishbone diagram in Figure 2-1. Ungar categorized false alarms into six major threads. They are system configuration, human error, BIT system test, Depot test problems, environment and software. The 5th generation aircraft program had also experienced the false alarms driven by these six main threads.

As for faults related to system configuration, the 5th generation aircraft program has a few number of aircrafts dedicated for flight tests only and they often carry special test equipment that a normal operational aircrafts would not carry. False Alarms were often triggered for this special configuration. The 5th generation aircraft fleet has also experienced disconnected or loose cables that could have contributed in creating false alarm as well. The F-18 program also had experienced the similar experience where the circuit cards in the infrared linescanner (IRLS) were loose and false alarms were generated (Westervelt 2006).

The next main thread is the human error. Ungar even suggests that human error is the greatest contributor for FAs (Ungar 2015). A complex air system requiring human interaction will be prone for human to make mistakes such as pressing the wrong buttons at the wrong time, misinterpreting it as anomaly and reporting it as a fault.

Under the BIT system test thread, diagnostic ambiguity is also a big problem for the legacy and the 5th generation aircraft programs. Often times multiple components are removed due to the inherent ambiguity by design. If the suppliers cannot duplicate the problem on these good parts, the parts may come back as RTOK and the HRC can potentially be scored as false alarm. As mentioned earlier, the HRC triggering threshold

could have been set very conservatively so that it will trigger at a subtle change in the tracked parameters in order to increase the fault detection rate.

Depot test problem is also a contributing factor for false alarms. Human errors can occur not only on the flight line but also at the supplier's facility or depots. Sometimes, RTOK items could have been tested OK using the wrong configuration or test setup.

Environment factors also affect the false alarms. The 5^{th} generation aircrafts have experienced anomalies and data recording issues when the aircrafts were flying at a high altitude. After the investigation, the root cause was discovered to be the cosmic rays corrupting the electronic components on-board. According to the research done by Blattau and Hillman, hard drive, batteries, capacitors, liquid crystal display, memory and CPU were found to be sensitive to high altitude failure because of the following two conditions, reduced air density and increase in cosmic ray (Blattau & Hillman 2004).

2.3 Machine Learning Classification Models

Humans are great at recognizing complex patterns in an image such as recognizing nature scenes or an object in a distorted picture. However, trained computer models outperform humans in rapid categorization tasks since the discriminative information is present but humans often do not have enough short-term memory power or time to extract the necessary information to make the pattern recognition seamlessly (Borji & Itti 2014). For example, humans are great at recognizing patterns in a simple two or three-dimensional data such as an example shown in Figure 2-2.

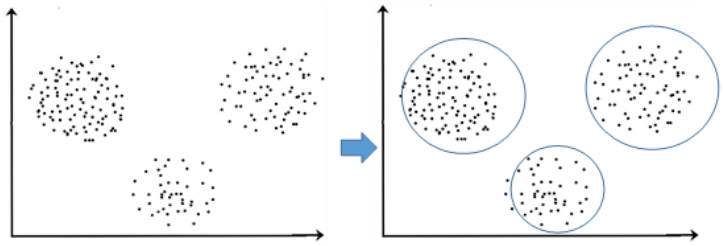


Figure 2-2 Two-Dimensional Pattern Recognition

As the number of interdependent variables increases, it quickly becomes challenging for humans to see the patterns. Likewise, humans would not be good at recognizing patterns in deoxyribonucleic acid (DNA) sequencing as compared to machine learning algorithms.

Machine learning is used to train computers how to process the data more efficiently. Sometimes it is not possible to recognize the pattern or extract information from the data by looking at the data. In this case, machine learning is used (Richert & Coelho 2013). Machine learning algorithms uses a set of features in the dataset. The features can be continuous, categorical or binary. If the machine learning algorithms are given with known labels, which is the corresponding true outcome, it is called supervised learning (Kotsiantis, 2007). In contrast, if known labeled data is not provided in hopes to discover new and useful classes of items is called unsupervised learning (Jain et al., 1999). Another type of machine learning that uses external trainer to measure how well the system operates is called reinforcement learning (Barto & Sutton, 1997). A review of historical survey of different types of machine learning classifiers including logic and instance based algorithms are well presented by De Mantaras and Armengol (1998).

One of the biggest challenge with supervised learning algorithms is collecting and preparing the dataset. If the data contains noise and missing values, then the algorithm

requires a significant pre-processing (Zhang el al. 2002). Different ways to detect outliers or noise have been presented by Hodge & Austin (2004). The outlier detection can be used to detect human errors, fraudulent activities, changes in system behavior, etc. For a specific classification problem regarding false alarm, it can be used to detect anomalies in the data and use them as predictors in machine learning algorithms. In this praxis, 13 different machine-learning algorithms were used to compare the results and find the best algorithm to use for the 5th generation aircraft application. These 13 algorithms can be categorized into the following four types of machine-learning techniques.

2.3.1 Logistic Regression

Logistic regression works similar to linear regression. The difference is that logistic regression has a binomial response variable. This algorithm uses a logistics function or sigmoid function to classify between two classes as shown in Figure 2-3. Since the purpose of this praxis to classify between two classes, this algorithm would be a great algorithm to test.

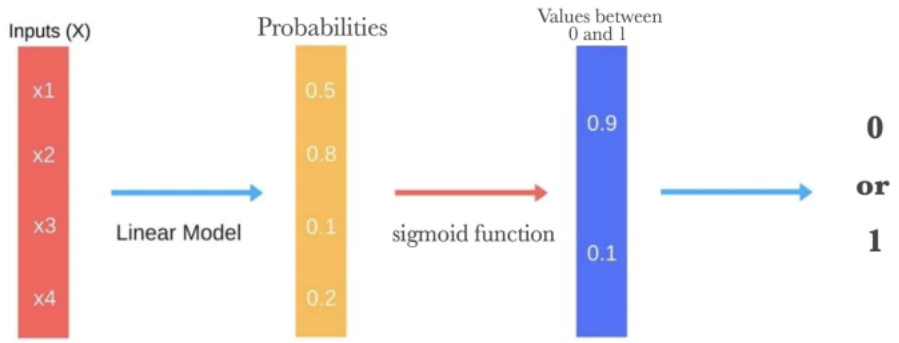


Figure 2-3. Logistics Regression

The main advantage is that the logistics regression algorithm does not require many computational resources. In addition, the results are highly interpretable since the

meaning of the probabilities are explicit and meaningful. However, the disadvantage is that the data has to be clean and well balanced and prepared. The algorithm does not handle variables that highly correlated and is known to produce over-fitting problem.

2.3.2 Support Vector Machine (SVM)

Support vector machine revolves around the idea of margin, which is the hyperplane that separates two data classes. The goal of SVM is to maximize the margin between the two data classes and creates the largest distance between separating hyperplanes as shown in Figure 2-3.

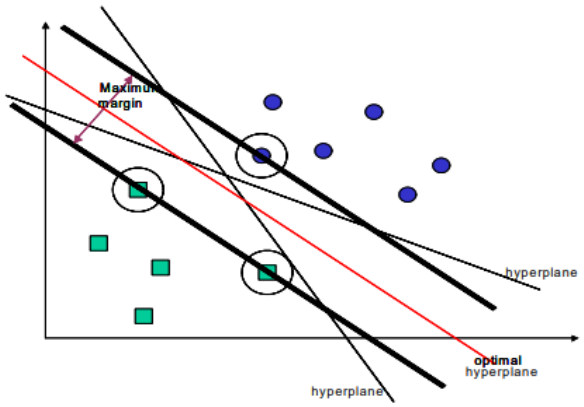


Figure 2-3. SVM Maximum Margin (Kotsiantis 2007)

The greatest advantage of using SVM is that it provides a good out-of-sample generalization, which yields a good prediction accuracy for previously unseen data (Auria & Moro 2008). By choosing the appropriate parameters in the case of a Gaussian kernel, SVM can be robust even if the dataset sample size is biased. Another advantage is that SVM solves a convex problem where there is only one optimal solution whereas Neural Network algorithms are prone to finding local minima. However, the disadvantage of

SVM is that it cannot be represented as a simple parametric function because its dimension may be very high.

2.3.3 Decision Tree

The publication of the first regression tree algorithm dates back to Morgan & Sonquit in 1963. Automatic Interaction Detection (AID) splits the data into two nodes recursively. Each node in a decision tree represents a feature to be classified. Each branch represents a value that the feature can assume. The AID algorithm had some weaknesses and did not attract much interest in the community. The weakness were that AID can severely overfit the data (Einhorn, 1972), only one variable can be used if there are two or more highly correlated variables (Doyle, 1973), and AID cannot handle sampling variability of the data (Bishop et al, 1975). Decision tree regained its attraction in 1984 when Classification and Regression Trees (CART) was introduced (Breiman et al., 1984). CART solved the under-fitting and over-fitting problems of AID by implementing pruning of the tree as shown in Figure 2-4. The CART algorithm is the one that is implemented in commercial software like R or Matlab.

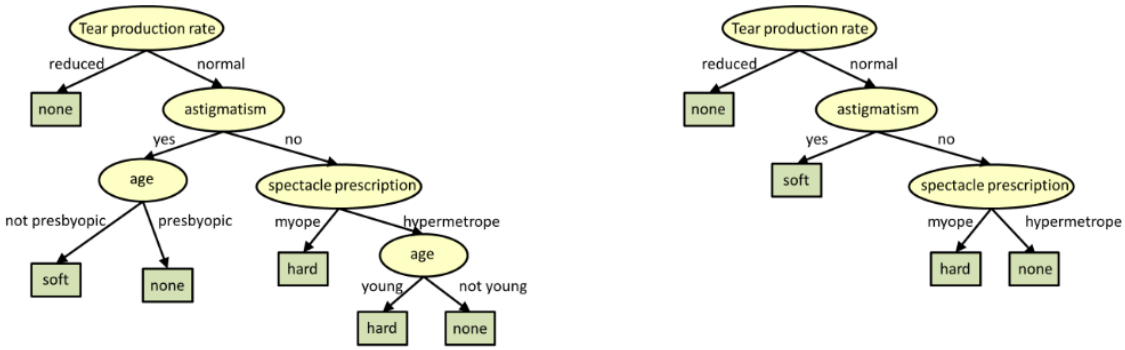


Figure 2-4. Decision Tree Pruning

The advantages of using the decision tree methods are that the algorithms implicitly perform feature selection, which means that the input data can include parameters that are both less and more important and the algorithms will figure out which ones are more important automatically. Decision tree algorithms require less time for data preparation since the algorithms can handle missing data and unbalanced datasets. This is a great algorithm to test the hypothesis in the praxis because there will be a lot more data points associated with false alarms than faults so the data will be imbalanced. One of the strongest point about this algorithm is that it can handle both less important and more important input automatically. Not all components have the same failure modes so certain input parameters for machine learning algorithms may be important for some failures but not for the entire set of failure modes.

2.3.4 Ensemble

Ensemble uses a set of classifiers and individual decisions are combined using weighted voting to classify new examples (Dietterich, 2000). In Matlab, the following three ensemble methods were available: 1) Boosted Trees, 2) Bagged Trees and 3) RUSBoosted Trees.

Boosted and bagged trees in Matlab are CART with the boosting and bagging techniques, respectively. Boosting is an iterative procedure where it applies a weighted average of results obtained from the predictions uing various samples. Whereas bagging uses a simple averaging of results to obtain the overall prediction. These techniques were introduced by Breiman (1996). RUSBoosted trees is a type of boosted trees with the technique called random under-sampling (RUS). RUS randomly removes samples from the majority class until the dataset are balanced. (Seiffert et al., 2010)

2.4 Use of Machine Learning to reduce False Alarms

Use of machine learning to reduce false alarms while correctly detecting faults is the main goal of this praxis. In 2012, de Padua Moreira and Nascimento Jr implemented the maintenance log in a supervised machine learning to predict the health of aircraft bleed valve as HEALTHY or UNHEALTHY. The goals of this praxis and the Moreira's research are almost identical in a sense that this praxis is about detecting false alarm using the aircraft data and the maintenance records and Moreira's study was to detect impending failure using the aircraft data and the maintenance records. Moreira used the SVM classifier to predict the binary response. The main difference is the input parameters. Moreira's research used the parameters that are known to be the contributing factors in assessing the health of bleed valve and the data was available such as manifold pressure, manifold temperature and N2 (high pressure compressor speed) of the engine where the bleed is taken from. This study was feasible because the failure mode of the bleed valve was well known and the author knew what input parameters were useful and meaningful. In this praxis, input parameters were chosen without the deep knowledge behind the failure modes of each component, because this praxis focuses on reducing false alarms at the aircraft level and not at each component level. In another word, the purpose of the praxis is not to eliminate the false alarms from the source but rather to create an algorithm that can automatically distinguish faults and false alarms to help human as if it was an experienced maintainer helping less experienced maintainers. While the method presented by Moreira's research offers a good way to filter false alarm at one component level or one HRC, it would take an enormous amount of time to investigate and find the best way to detect the anomalies in the source data for the emtire

thousands of HRCs. As it was previously mentioned in the chapter 1, a large amount of the HRCs are triggered based on the binary data and how the binary information is created is often proprietary and readily not available to the end users.

There are also several studies on using machine learning to reduce false alarms in intensive care units (ICU) (Fahmi et al., 2012 and Wang et al., 2015). ICU collects real-time patient vital signs data including blood pressure and electrocardiogram (ECG). If abnormalities are detected, the monitoring system triggers the alarm to notify the nurses and physicians. However, approximately 86% of the ICU alarm was false alarms (lawless, 1994) and the nurse and doctors were making too many trips based on the false alarm information. Both the ICU and the 5^{th} generation aircraft PHM are in the same dilemma where if the false alarms are ignored without a good reason, the health of the patient or the aircraft could be in danger.

Fahmi used two different machine learning technique to reduce false alarm rates. The two machine learning algorithms were SVM and Multi-layer perceptron (MLP). MLP is a family of neural networks, which consists of multiple layers including inputs, hidden layers and outputs as shown in Figure 2-5.

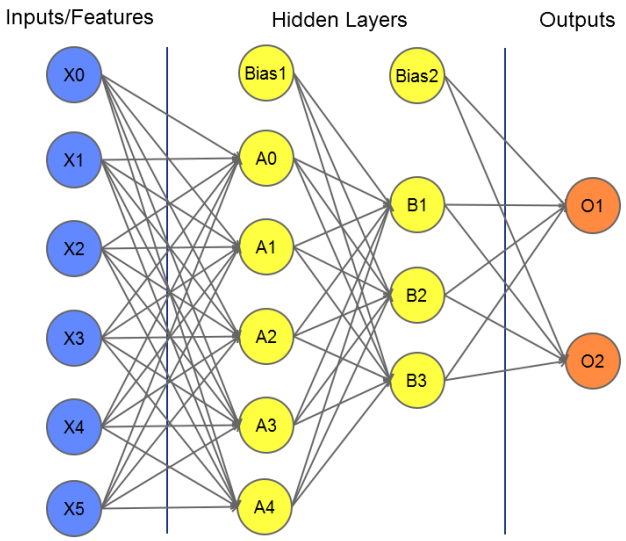


Figure 2-5. Multi-Layer Perceptron

Each connection between the layers is weighted and the weight is varied through the iterations to find the optimal combination that matches the labeled outputs. Other than using two different ML approaches and using a different set of continuous data, the research methodology was almost identical to Moreira's work on reducing FA on bleed air valve. They both used SVM and binary-labeled outputs to train the models as supervised learning. The intention of this research is also similar to the purpose of this praxis. This praxis and the study done by Wang et al. have several things in common. First, both studies experienced high false alarm rates of approximately 90%. The high false alarm rate creates an unbalanced ratio between the false alarm dataset and fault or abnormality dataset. Choosing the right machine-learning algorithm that can handle unbalanced dataset was vital for the good result. Wang's best result at suppressing false alarm rate was by 13.96%. That means the nurses or doctors saved approximately one trip out of seven false alarms if the machine-learning algorithm was used. The major difference between this ICU study and this praxis is the input parameters again. Similar

to the study on the bleed air valve, the ICU study also used continuous sensor readings as input parameters and the supervised learning was to filter out the noise or anomalies based on the labeled data. While the technique presented in the ICU studies can be leveraged to reduce the false alarms for a particular set of HRC where the continuous data is available and the failure mechanisms are well known, it will be very challenging to apply this approach for thousands of HRCs.

Another study to reduce false alarm using machine learning has been conducted in the field of data breach detection (Vasseur 2018). There two types of data breaches. The first type is external cyberattacks where there have been numerous researches in the past to detect and reduce false alarms for network intrusion. Vasseur's research focuses on the second type of data breach, which is the malicious internal access of data. Vasseur used the Iterative Dichotomiser 3(ID3) decision trees method to train a normal behavior using the sample data that have been vetted and confirmed by the subject matter experts as true or false alarms. Some of the predictors were the information such as what data was accessed, who (or what role) accessed it, when accessed it and the number of accesses. The purpose of the machine-learning model is to capture the knowledge of the SMEs who can distinguish between the normal behavior and malicious access of data by investigating the predictor values. The purpose of Vasseur's research is identical to the purpose of this praxis. The labeled data vetted by the SME is essentially the FID (U) from the FRACAS. The main difference between the method used in Vasseur's study and what is presented in this praxis is the use of only one machine learning algorithm, ID3 decision trees. Different combinations of predictors as well as the imbalance in the training data set can influence the outcome of the ID3 decision tree models. Also, the

ID3 decision tree does not guarantee an optimal model as it can converge on local optima because the ID3 decision tree uses a greedy algorithm where it searches for the locally optimal choice at each stage. This method also has a disadvantage of over-fitting the training data. Lastly, the ID3 decision tree method is not efficient at handling continuous data as there are many places to split the continuous data and it can be time consuming.

2.5 Summary and Conclusion

There are so many applications of using machine learning to reduce false alarms in many different industries such as the aerospace industry, e-commerce and even in medical industries. Many studies have been done in the past to suggest a better way to classification models with unbalanced data, lack of large data, missing data, etc. Great algorithms can definitely help in handling poor data issues but the most important contributing factors to achieve a high level of accuracy in the classification problem are choosing the right input parameters or features and the accuracy of the labeled data. Certain machine learning algorithms are more tolerable and forgiving in dealing with imbalanced, uncleaned and highly correlated data than the other but if the labeled data is not accurate, then the algorithms will only get as good as the bad data that they were trained with. In the next chapter, the rationale for selecting the input parameters will be explained.

Chapter 3—Methodology

3.1 Introduction

In this praxis, the historical false alarm rates from the FRACAS data will be quantitatively compared to the outcome of the machine learning classification algorithms using the PHM and FRACAS data. During the data preparation phase, the PHM data and the FRACAS data were encoded using a random alphanumeric key generator to avoid disclosing any 5th generation aircraft program's sensitive information (e.g. names of operating base, aircraft identifiers, HRCs, etc.). This information is not releasable to the public in its original form not to disclose any information that can be used to trace back to any specific aircrafts or configuration.

3.2 Data Preparation

First, the PHM data was collected with the dates ranging from January 2014 to May 2018. Preparing the PHM data into a usable format was a challenging and time consuming task since there are dozens of different definition files that have accumulated over the course of numerous AV software updates. Once the data was converted into human-readable spreadsheet format, Visual Basic for Applications (VBA) scripts were run to further extract the required information such as the number of HRCs asserted within the +/– 2 minute window. Another challenge was in handling lots of power cycles, which resulted in over 2 million rows. Microsoft Excel 2007 or later only supports up to 1,048,576 rows on a worksheet so the data had to be split into multiple files for faster processing. Over 2 million rows of PHM data was then reduced to 183,984 rows by filtering out the HRCs that are not present in the FRACAS database, which were mostly filtered by the NFL and were not made visible to the maintainers. The following data

elements shown in Table 3-1 were parsed out from the PHM data and processed for each of the 183,984 rows.

Table 3-1. Exported PHM Data Elements

Data	Values
Date	Date (MM/DD/YYYY)
Type/Version Effectivity (TVE)*	7 Digit Alpha-numeric Identifier
Power Cycle #	Unique set of numbers e.g. XX-XXXXXXX
Total # of HRCs	Number
HRC*	13 digit numbers (####### ## ####)
LCN Group*	First 2 Digits of HRC (XX##### ## ####)
HAMI*	Middle 2 Digits of HRC (####### XX ####)
Type of Test	Text (Flight, Engine Run, PIT Test, IPP Run, Flight Abort, Ground Abort, Taxi, Other)
# of HRCs set within +- 2 min window	Number
Time occurred	Timestamp of when HRC was asserted (HH:MM:SS)
Unique Aircraft Identifier (UAI)*	Text
SDL*	Text
RST Count	Number
SET Count	Number
RES Count	Number
Maximum G-load in Z-direction (Nzmax)**	Numbers
Maximum Pressure Altitude (HPmax)**	Numbers (ft)
Maximum Cockpit Airspeed (Vcasmax)**	Number (kts)

* Encoded with alpha-numeric random key generator
** Numbers were multiplied by random factors

The FRACAS data was extracted using database queries to extract the following data elements as shown in Table 3-2. A total of 273,219 maintenance events were exported. Out of 273,219 maintenance events, 183,985 ME's were assigned with the FID (U) that is either fault or false alarm by excluding the FID codes such as X99 (i.e. non-PHM pertinent) used for servicing oil or routine inspections.

Table 3-2. Exported FRACAS Data Elements

Data	Values
WO Discover Date	Date (MM/DD/YYYY)
TVE*	Text
UAI*	Text
HRC*	Each HRC is a 13 digit number
FID Code Initial (I)	Text (e.g. D10, D20, etc.)
FID Code Updated (U)	Text (e.g. D10, D20, etc.)
How Malfunction Code	3-Digit Numbers (e.g. 000, 799, etc.)
Total Maintenance Man Hours (Total MMH)	Number (Hrs)

* Encoded with alpha-numeric random key generator

The most important part of the data preparation was to find the matching entries between the PHM data to FRACAS data to be able to assign the appropriate FID codes to the PHM data. By searching for the matching dates, unique aircraft identifiers (UAIs), power cycle numbers and HRCs, 12,122 matches were discovered and each row of the 12,122 rows in the PHM data was assigned with the FID codes from the matching FRACAS data. Out of 12,122 maintenance events, only 892 maintenance events were found to be true positive or faults and the rest of the 11,230 ME's were false alarms. This praxis will also look into the effects of training algorithms with unbalanced data.

3.3 Input Parameters (Predictors)

The input parameters are the collection of potential contributing factors in determining whether an asserted HRC is a false alarm or not. The following list of input parameters and the predictor data type as shown in Table 3-3 were selected for training the classification models and the rationale for choosing each parameter is explained in the following sections.

Table 3-3. Processed Data for ML Training and Validation

Data	Predictor Data Type
TVE*	Categorical
SDL*	Categorical
Total # of HRCs in the same power cycle	Double
HRC*	Categorical
LCN Group*	Categorical
HAMI*	Categorical
Type of Test	Categorical
# of HRCs set within +- 2 min window	Double
Time occurred	Double
FID Code (U)	Response (2 Classes: Fault and FA)
RST Count	Double
SET Count	Double
RES Count	Double
NzMax**	Double
HPmax**	Double
VcasMax**	Double

* Encoded with alpha-numeric random key generator
** Numbers were multiplied by random factors

3.3.1 Type/Version Effectivity (TVE)

The type/version effectivity is the unique identifier for a specific set of hardware and software configuration. The TVE in this praxis was synonymously used as Unique Aircraft Identifier (UAI) that identifies a specific aircraft by its tail number. The reason for using the TVE as a predictor is that each aircraft can often behave differently from the other aircrafts. For example, one aircraft may experience more false alarms triggered by loose electrical connections or maintenance induced anomalies that could be unique condition of the aircraft. If there are certain HRCs that consistently assert from a particular aircraft, this predictor will increase the accuracy of the classification models.

3.3.2 Software Data Load (SDL)

Similar to the TVE where the TVE identifies the aircraft configuraiton, the software data load (SDL) is the software configuration of the aircraft. Similar to the operating system (OS) for a personal computer or a cell phone, the SDL has the essential software that enables the controlling of the aircraft as well as PHM capabilities. If there are known anomalies or false alarms associated with a certain SDL, then often times the fix will be implemented via the release of the next SDL. Until the aircraft receives the next SDL with the fixes, the false alarm HRCs might assert numerous times. These false alarm HRCs can also be put in the NFL so that maintainer will not have to see the nuisance HRCs but the root cause of the false alarm has to be fully understood and the fix has to be identified before the HRC can be added to the NFL.

3.3.3 Total Number of HRCs from the same power cycle

In the earlier software and configuration of the 5^{th} generation aircraft, some aircrafts generated a serval hundred HRCs per flight. The majority of the asserted HRCs was false alarm. Unless there is a significant failure on the aircraft, zero to a few HRCs are typically expected from a normal flight. In order for several hundred HRCs to be true faults, the aircraft has to be in such a serious condition that the pilot will notice and report the problems and the aircraft will not carry out another flight until all problems are adjudicated. As the number of HRCs from the same power cycle increases, the probability that one of the many HRCs being a false alarm will decrease assuming that the aircraft did not

experience a catastrophic failure that impact various components all at once which the pilot should be able to tell.

3.3.4 Health Reporting Code (HRC)

HRC itself was used as a predictor. By treating it as a categorical input, the machine learning will be able to learn which particular HRCs have been false alarms in the past under what particular software and aircraft configuration. If this input parameter were used as the only parameter, it would defeat the purpose of this praxis in identifying HRCs as fault or false alarm using machine learning but since there are other contributing factors that can supplement this predictor, it is almost as if creating NFL lists for various configuration automatically.

3.3.5 Logistics Control Number (LCN) Group

The first two digits of an HRC is called Logistics Control Number group number. The two numbers tell which system (e.g. landing gear system, flight control system, etc.) is experiencing the problem. Similar to a certain aircraft having a consistent false alarm problem, particular aircraft components or systems could also consistently cause false alarms. The systems or sub-systems components are manufactured by various suppliers and certain systems are more mature in the diagnostic capability than the other systems. In addition, certain systems may behave differently depending on the AV software version. Therefore, the LCN group could be a good contributing factor that can have an impact on the outcome of the machine learning classification models.

3.3.6 Health Assessment Method Indicator (HAMI)

These middle two-digit numbers after the first 7 digits of an HRC is called Health Assessment Method Indicator (HAMI). It tells which fault detection method was used. The exact definition of the different methods is not releasable to the public but the data is still be a valuable predictor for training the classification models because the algorithms do not need to know the exact meaning of different values. All it cares is that different values as a categorical input clearly influence the outcome.

3.3.7 Type of Test

If a certain HRC was asserted while the aircraft was on the ground where by design the particular HRC can only be asserted during flight, it is definitely a false alarm. Theoretically, this false alarm should not ever happen if the PHM system was designed correctly and working as designed but these problems do occur. These type of issues are fixed as they are discovered through the release of software updates or the advanced filtering which can incorporate a simple if-this-then-that (IFTTT) logic.

3.3.8 # of HRCs within +/- 2 min window of HRC assertion

Certain components are interdependent of each other and may not function correctly if the other component is not working. For example, the integrated power package (IPP) provides the electrical power or the operating pressure to certain components when the aircraft is operating. If the IPP fails abruptly during an engine run, a series of components that lost the electrical power may report a problem. However, there may be nothing wrong with the other components that

reported the problem, since the root cause of the problem is the IPP failure. The group of HRCs that are asserted within the same window of time may only have one root cause where the rest of the HRCs sympathetic HRCs but they can be recorded as false alarms if the maintainer does not know about the root cause and could not duplicate the fault for the rest of the components.

3.3.9 Time Occurred

This predictor was included to investigate how the time of HRC assertion contributes to accurately distinguishing faults and false alarms. For example, certain HRCs may assert in the first few seconds of aircraft start up where they are rescinded after the system is fully up and running. This could also be a misleading indicator and degrade the accuracy of certain machine learning models since the actual timing of a component failure is often independent of the operating hours since engine start-up. Whether this predictor will actually degrade the performance of the algorithms or not will be investigated by turning on and off all different combinations of the 15 predictors to see which combination will yield the best results.

3.3.10 SET, RES and RST Counter

The same HRC can be asserted or rescinded multiple times throughout the duration of a flight or an engine run, because the on-board PHM system is continuously monitoring the health of the aircraft. Intermittent problems such as loose wiring connection can trigger an HRC during a high G-load maneuver and when the aircraft is level cruising, the HRC might be rescinded because the wiring connection is intact again. A large number of toggling could mean that there

might be some intermittent problems. Intermittent failures are often caused by design defects, repeatable test voids and loose/contaminated interconnections such as solder joints or connector pins (Ungar 2015).

3.3.11 Max Nz, Max Hp and Max Vcas

The maximum vertical G-load, pressure altitude and cockpit airspeed were included as the predictors. Originally, the snapshot values of these parameters at the time of HRC assertion were going to be used but it was challenging to extract them because the same HRC could assert multiple times making several different values to choose from but a consistent logic to automatically choose the right values could not be easily created. Instead, the maximum Nz, Hp and Vcas were extracted in order to investigate whether the aircraft experiencing the high G-maneuvers, high altitudes or high speeds would be prone to generating false alarms more often or not.

3.4 Research Procedures

Once the data was prepared, it was imported into the MATLAB Classification Learner app by assigning the correct predictor data type as shown in Table 3-3. In the import setting, the k-fold cross-validation scheme was chosen. The k-fold cross validation scheme is useful when the training data set is limited and cannot afford to set aside a percentage of the data for the validation purpose only. Even though, there were 12,122 data points available for the training, the majority of the data was false alarm, which could potentially make the classification model biased if the algorithms are not carefully chosen. The problem of imbalanced dataset and the methods for dealing with the imbalanced dataset is explained in Jayasree & Gavya (2014). One of the way to deal

with the imbalanced dataset is to either oversampling or under-sampling method. The oversampling method is to balance the dataset by replicating the minority class samples. This method increases the overfitting occurrences. The under-sampling method is to remove samples from the majority class samples. In order to keep as many "fault" data possible, the 5-fold cross-validation scheme was used. This method partitions the data into five disjoint sets and trains the model using the out-of-fold observations and assess the performance using in-fold data as shown in Figure 3-1.

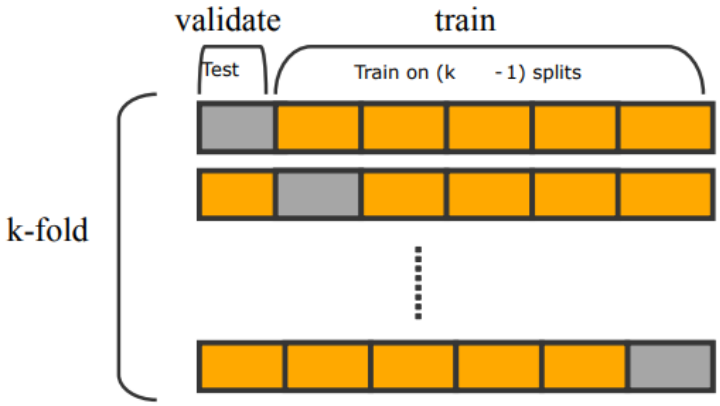


Figure 3-1. k-fold Cross-Validation

The test error is then calculated using the average over all folds. The disadvantage of using the k-fold cross-validation scheme is that it takes longer to train models than using the holdout validation scheme where a percentage of the data is set aside for validation. Since it only took few seconds up to half an hour at the most to train the models using different algorithms, the k-fold cross-validation scheme was the optimal choice.

After the entire 12,122 data points were imported and the validation scheme was selected, the following models in Table 3-4 were trained to show which of the algorithms yields the best result. The set-up parameters are listed for each classification model.

Table 3-4 List of Classification Models

List of Classification Models	Set up Parameters
1. Logistic Regression	Maximum # of splits: 100 Split criterion: Gini's diversity index Surrogate decision splits: Off
2.Linear SVM	Kernel function: Linear, Quadratic, Cubic Kernel scale: Automatic Bix constraint level: 1 Multiclass method: One-vs-One Standardize data: true
3.Quadratic SVM	
4.Cubic SVM	
5.Fine Gaussian SVM	Kernel function: Fine Kernel scale: 0.97 Box constraint level :1 Multiclass method: One-vs-One Standardize data: true
6.Medium Gaussian SVM	Kernel scale: 3.9
7.Coarse Gaussian SVM	Kernel scale: 15
8 Fine Decision Tree	Maximum # of splits: 100 Split criterion: Gini's diversity index Surrogate decision splits: Off
9.Medium Decision Tree	Maximum # of splits: 20
10. Coarse Decision Tree	Maximum # of splits: 4
11. Ensemble Boosted Trees	Ensemble method: AdaBoost Learner type: Decision tree Maximum number of splits: 20 Number of learners: 30 Learning rate: 0.1
12. Ensemble Bagged Trees	Ensemble method: Bag Learner type: Decision tree Number of learners: 30
13. Ensemble RUSBoosted Trees	Ensemble method: RUSBoost Learner type: Decision tree Maximum number of splits: 20 Number of learners: 30 Learning rate: 0.1

Using the same 13 classification models, the impacts of imbalanced input data was investigated by varying the ratios between "Faults" and "False Alarms". Since there is a fewer data points for faults, the entire 892 fault data was included and the false alarm

data was randomly chosen to have the ratios between FAs and Faults as follows: 0.2:1, 0.4:1, 0.6:1, 0.8:1, 1:1, 1.5:1, 2:1, 2.5:1, 3:1, 13:1.

Using the algorithm that yielded the best results, the impact of varying the number of input data was investigated. The ratio between fault and false alarm was kept at 5% to 95% ratio, respectively, to compare the results against the human performance from the FRACAS where the 95% of the available data was false alarm. The results of these different test cases will be presented in the next chapter.

3.5 Data Analysis

In the following sections, the test set-ups and the methods of statistical analysis will be explained for each hypothesis.

3.5.1 H1: Accuracy in Identifying False Alarm

In order to prove that machine learning can distinguish faults and false alarm with a higher accuracy than the human performance, the results from the trained machine-learning algorithms were compared against the human performance. The human performance numbers were calculated by counting the number of maintenance events that had both FID (I) and FID (U) with FA codes against the total number of FA events in the FRACAS data. This number shows what percentage of the FA events were maintainers able to identify the HRCs correctly as FA. Note that even if the maintenance events were initially scored as false alarm, maintainers often had to spend some time troubleshooting or confirming before realizing that the HRCs were false alarms. In another word, maintainers could not determine whether the HRCs were truly false alarm or not if they had to spend the time to confirm. Therefore, the human performance level

in distinguishing is actually worse that what it shows in the FRACAS data. The time spent to identify or confirm a false alarm will be discussed as part of hypothesis #3.

The machine learning classification results are presented in the form as shown in Table 3-5. True positive (TP), true negative (TN), false positive (FP) and false negative (FN) are the commonly used terms to describe the sensitivity, specificity, positive and negative predictive values and accuracy of the results.

Sensitivity = TP/(FN+TP) = What is the probability that the algorithm will correctly identify fault when a fault is present?

Specificity = TN/(TN+FP) = What is the probability that the algorithm will correctly identify FA when a FA is present?

Positive Predictive Value (PPV) = What is the probability that the HRC is actually a fault when the algorithm predicts a fault?

Negative Predictive Value (NPV) = What is the probability that the HRC is actually FA when the algorithm predicts FA?

Accuracy = (TN + TP)/(TN+TP+FN+FP) = Overall Prediction accuracy in determining FA and Fault

The equation for the accuracy above is to calculate the overall accuracy of the algorithm including faults and FA. Since hypothesis #1 is to prove that the algorithm can identify FA better than human, the specificity is the correct measure of value to test the hypothesis #1.

Table 3-5. Terms used to define sensitivity, specificity and accuracy

Observed Condition	Fault	TP	FN	Sensitivity = TP/(FN+TP)
	FA	FP	TN	Specificity = TN/(TN+FP)
		PPV = TP/(FP+TP)	NPV = TN/(TN+FN)	Accuracy = (TP+TN)/ (TP+TN+FP+FN)
		Fault	FA	
		Predicted Condition		

3.5.2 H2: Accuracy in Identifying Fault

As discussed in the previous section, the sensitivity is the number that will be used to prove that the machine learning algorithms can identify faults at a higher accuracy than the human performance level.

3.5.3 H3: Time to identify False Alarm

As it was briefly mentioned in the earlier section, maintainers often spend the time to adjudicate HRCs just to realize that the HRCs were false alarms. The maintainers then record these maintenance events as false alarms in the ground system and the FID (I) would show up as false alarm in the FRACAS database. After the SME reviews, if the FID (U) codes stay as false alarm, these maintenance events would be counted as correctly assessed by the maintainers. However, if the maintainers spent any time on troubleshooting or confirming false alarms, then the maintenance events should not have been counted towards the number of correct assessment. While the hypothesis #1 was to measure how many maintenance events were correctly assessed by the maintainers, the purpose of

this hypothesis is to show that the machine learning classification algorithm can save even more time not having to spend as much time as on these false alarms events that the maintainers had already assessed as false alarm.

In order to calculate the total maintenance man-hours (MMH) spent for FA events, the FRACAS data was filtered for "FA" and the total MMH each filtered row was summed up. The total MMH was then divided the number of FA events to show the wasted hours per event for troubleshooting FA. The unnecessary MMH spent per event was then compared against the time it took for the machine learning algorithms to be trained and the time it takes for the ML algorithms to make a prediction with a new set of data.

3.5.4 H4: Performance Variance by Operational Sites

As brand new operational sites are receiving the aircrafts for the first time, their maintainers typically lack the corporate knowledge when dealing with false alarms. The maintainers in these new locations will have to go through the learning curve to figure out what HRCs tend to be false alarms and when and under what conditions, HRCs would normally assert as false alarms. By leveraging the machine-learning algorithm that has been trained on the enterprise data, new operational sites can benefit from the same corporate knowledge applied in the ground system as if they have been operating just like the other sites, which can aid the maintainers in their decision-making. First, each maintenance event will be marked with 1 or 0 depending on whether the FID(I) matched FID(U) or not. The binary data will go through the Levene's test to show that each operational site had a statistically significant variance in correctly

identifying FA and faults. To compare the diagnostic performance by operational sites, the true positive rates and true negative rates were grouped by the operational sites. The results were sorted by the number of maintenance events from highest to lowest. This data will show how the newer operational sites with less experience performs differently as compared to the sites with many maintenance events.

If the machine learning algorithms were trained on the entire fleet users' data assuming each participating countries would allow their flight information to be used in the training, every ground system users will have the same performance level in distinguishing faults and false alarms. The cumulative standard deviations by adding one operation site to the entire fleet will be calculated to show if the introduction of a new site will yield a larger variance within the fleet. Since the operational sites are sorted by the number of maintenance events from largest to smallest, each additional site will represent lesser experienced operational site as compared to the rest of the fleet. The slope of the trend line for the cumulative standard deviation is a good indicator that the human performance varies as the number of maintenance changes between the operational sites.

3.5.5 H5: Accuracy Improvement Rates

Typically, machine learning is known for requiring "Big Data" in order to accurately train the algorithms because with less data to train, results may be easily influenced with subsequent data (Vennapusa & Bhyrapuneni, 2019). As it was mentioned, the ratio between faults and false alarms captured in the FRACAS

is heavily biased and the number of "fault" data is limited to be truly qualified as "Big Data". Of course, the PHM data itself is a big data considering there are thousands of recorded parameters and information. However, the number of data available for training and validating is somewhat limited due to the limited number of maintenance events with verified "fault" FID codes. Therefore, it is important to prove that the machine-learning algorithm can quickly learn and its performance can improve at a higher rate than the improvement seen in the operational sites given the same amount of data points or experience. In order to compare the performance improvement between human and machine-learning algorithms side-by-side, the incorrect assessment rate per 100 maintenance events were calculated using the FRACAS data and the trend line was fitted over the scattered plots to derive the rate at which the performance of the maintainer was increasing. This slope was then compared against the slope at which the machine learning was improving as a function of number of input data to mimic varying the number of maintenance events. For example, if there were 200 maintenance events, the performance improvement of human would show up as two points that gives the slope. Using machine-learning algorithms, the number of training data sample size was varied gradually to display the improvement in performance as a function of the number of inputs. In order to compare apples to apples, the ratio of the input data was set up to mimic the fault to false alarm ratio that the maintainers would experience in the field.

Chapter 4—Results

4.1 Introduction

In this chapter, the results of the analyses are presented without interpretation. The interpretation of the results is reserved for the chapter 5. The results are presented in tables and figures to show the statistical analyses to answer research questions or hypotheses.

4.2 ML vs. Human in identifying false alarm

The very first research question was whether the machine-learning classification model will be more accurate in distinguishing false alarms than the human maintainers.

H1: Trained classification models using machine-learning algorithms will identify false alarm at a higher accuracy than human maintainers will.

Table 4-1 shows the human performance levels in distinguishing faults and false alarms, which were calculated from the FRACAS data. There were total of 183,894 maintenance events.

Table 4-1. Human Performance Level from FRACAS

Final Disposition FID (U)	Fault	4018	4661	46%
	FA	33076	14229	81%
		Fault	FA	Accuracy
		11%	97%	79%
		Initial Disposition FID (I)		

Out of 175,305 observed FA events (FID (U)), 81% of the observed false alarms were accurately assessed as FA by the maintainers (FID (I)). The negative predictive value (NPV) for identifying FA accurately was 97%. In other words, when a maintainer assess an HRC as a false alarm, there is 97% chance that the HRC is indeed a false alarm. However, there is only 11% chance that a fault is truly present when a maintainer assesses an HRC as a fault. In the next chapter, the reason for the low PPV for identifying fault will be explained.

The following Figure 4-1 shows the results from the machine learning classification algorithms that were trained with 12,122 data points using all 15 predictors.

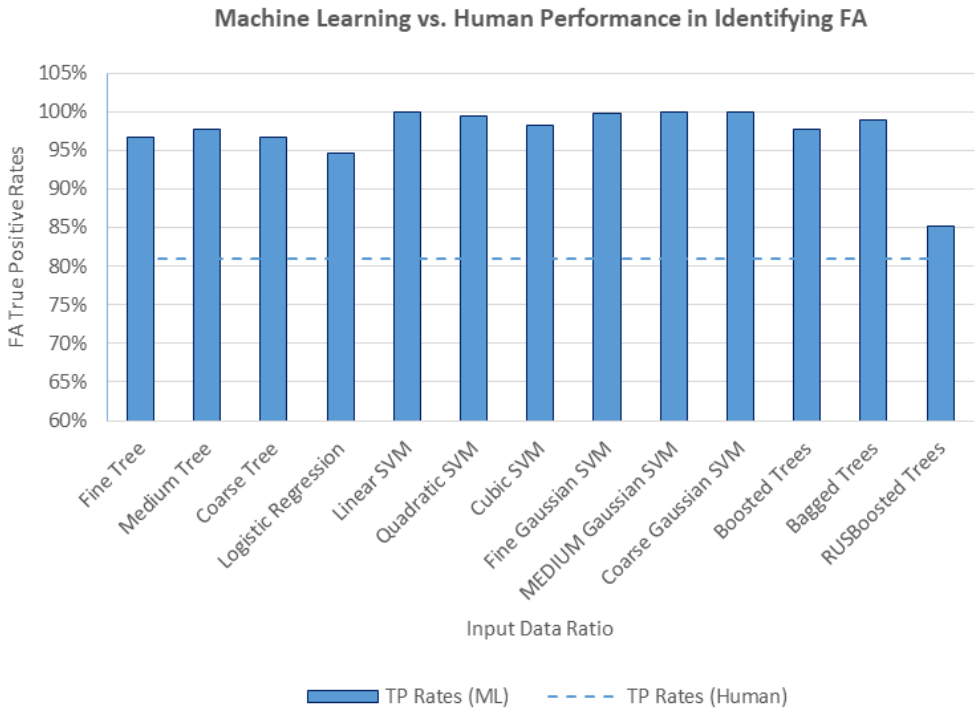


Figure 4-1. Machine Learning vs. Human in Identifying FA

All 15 machine-learning classification models exceeded the human performance level of 81% and accurately identified false alarms as false alarms. While the results are promising, the results for identifying fault using machine-learning algorithms have to be

considered at the same to understand the overall accuracy of the ML predictions. The reason for the high accuracies of machine learning algorithms will be discussed in chapter 5.

The following Table 4-2 shows the performance values of the RUSBoostedTrees since it is the only algorithm that performed better than human in identifying both false alarms and faults. The result of the machine learning algorithms in identifying faults will be discussed in the following chapter.

Table 4-2. Machine Learning Performance Level

Final Disposition FID (U)				
	Fault	493	399	55%
	FA	1658	9572	**85%**
		Fault	FA	Accuracy
		23%	96%	83%
		Machine Learning Prediction		

4.3 ML vs. Human in identifying Fault

The next research question was whether the predictions of the machine-learning algorithms can be trusted for safe flight operation or not. Even if false alarms can be accurately identified, if faults are not mistakenly predicted as FA and the maintenance actions were skipped, it could lead in to a flight safety issue.

H2: Trained classification models using machine-learning algorithms will identify faults at a higher accuracy than human maintainers will.

As shown in table 4-3, the human maintainers were only capable of identifying faults accurately 46% of the times. 12 of the 13 different trained machine-learning algorithms performed worse than the human maintainers in identifying true faults. The only machine-learning classification model that yielded a higher accuracy in identifying faults than the human maintainers was RUSBoosted Trees with the sensitivity of 55%.

Table 4-3. Machine-Learning vs. Human Accuracies

Model	TP	FP	FN	TN	TN Rates (Human)	TP Rates (Human)	TN Rates (ML)	TP Rates (ML)	FA NPV	Fault PPV	Accuracy
Fine Tree	10863	367	608	284	81%	46%	97%	32%	95%	44%	92.0%
Medium Tree	10975	255	617	275	81%	46%	98%	31%	95%	52%	92.8%
Coarse Tree	10861	369	574	318	81%	46%	97%	36%	95%	46%	92.2%
Logistic Regression	10623	607	530	362	81%	46%	95%	41%	95%	37%	90.6%
Linear SVM	11220	10	802	90	81%	46%	100%	10%	93%	90%	93.3%
Quadratic SVM	11167	63	700	192	81%	46%	99%	22%	94%	75%	93.7%
Cubic SVM	11028	202	647	245	81%	46%	98%	27%	94%	55%	93.0%
Fine Gaussian SVM	11198	32	867	25	81%	46%	100%	3%	93%	44%	92.6%
MEDIUM Gaussian SVM	11226	4	839	53	81%	46%	100%	6%	93%	93%	93.0%
Coarse Gaussian SVM	11230	0	892	0	81%	46%	100%	0%	93%	0%	92.6%
Boosted Trees	10979	251	623	269	81%	46%	98%	30%	95%	52%	92.8%
Bagged Trees	11102	128	651	241	81%	46%	99%	27%	94%	65%	93.6%
RUSBoosted Trees	9572	1658	399	493	81%	46%	85%	55%	96%	23%	83.0%

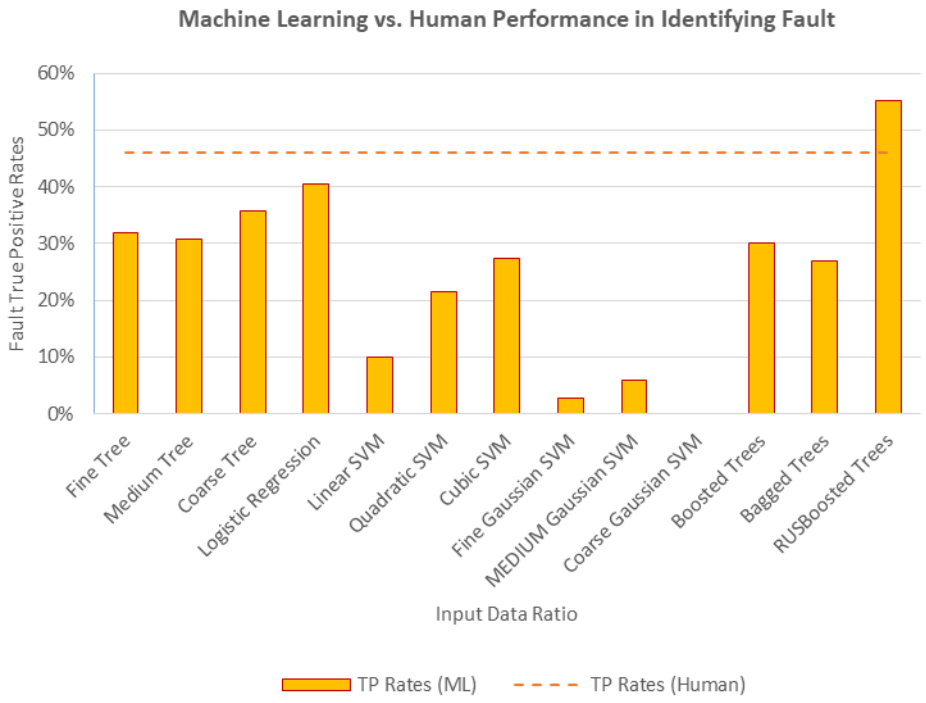


Figure 4-2. Machine Learning vs. Human in Identifying Fault

Overall, the RUSBoosted Trees algorithm scored 85% accuracy in identifying FA and 55% accuracy in identifying fault. The overall accuracy of the machine learning prediction is 83% as compared to 79% for human.

4.4 Time Wasted on Troubleshooting False Alarms

The research question # 3 was to understand the savings in maintenance man-hours from not having to spend any time on troubleshooting or confirming false alarms if machine-learning algorithms were used filter out false alarm.

H3: Trained classification models using machine-learning algorithms will take less time to identify whether an HRC is a false alarm or not.

On average, approximately 11 minutes were spent on each false alarm event as shown in Table 4-4, Out of 175,305 FA maintenance events, 30,147 events had at least 0.1 hours or

more spent on troubleshooting false alarms. Approximately 1 hour was spent on each of the 30,147 false alarm events.

Table 4-4. Total Maintenance Man-Hours spent on False Alarms

175,305 ME's	Hrs	Min
Sum	31334.86	1880091.9
Max	139.12	8347.0
SteDev	1.51	90.5
Avg/ME	0.18	10.7

As shown Table 4-5, the algorithm that took the longest time was Fine Gaussian SVM and it took approximately 26 minutes to train with 12,122 data points on a desktop computer with Intel i5-6600K CPU and 16 GB of RAM. Theoretically, it would take 6.7 hours for the same computer system to process all available 183,984 rows of FRACAS data. Running machine learning algorithms support multiple CPUs and the powerful computing resources such as cloud computing services can be utilized to reduce the training time to a fraction of the time it took for a typical desktop to train the data. The training the model can also done in parallel using the computer resources outside of the normal ground system which means the training time will not impact any aircraft turnaround time or the computing resources to do the normal aircraft maintenance. According to the prediction speed in Table 4-5, once the modes are trained, it is almost instantaneous to predict with a new data. Therefore, the only time needed by the machine-learning algorithm to be able to distinguish faults and false alarm is the training time which can easily be done overnight where there is less activity on the flight line or use the external computing resource to train the model and import the trained model to the ground system.

Table 4-5. Time required to train and test the machine-learning algorithms

	Fine Tree	Medium Tree	Coarse Tree	Logistic Regression	Linear SVM	Quadratic SVM	Cubic SVM	Fine Gaussian SVM	MEDIUM Gaussian SVM	Coarse Gaussian SVM	Boosted Trees	Bagged Trees	RUSBoosted Trees
Training Time (sec)	5.7807	5.2189	5.0272	2866.9	501.65	520.31	575.77	1584.8	1071.8	1007.8	1034.7	1051.9	1059.5
Prediction Speed (obs/sec)	89000	90000	96000	14000	290	240	220	150	230	300	19000	16000	22000

4.5 Performance Variance by Operational Sites

The fourth research question resolves around the idea of reducing the performance gaps between the operational sites that have been operating for a long time and newly standing up sites with less experienced maintainers.

H4: Trained classification models using machine-learning algorithms will decrease the performance variance in distinguishing faults and false alarms between each operational site.

In order to prove the hypothesis #4, one must first show that there is a statistically significant variance between the operational sites. Using the Leven's test, the null hypothesis that the variances of human performance are equal across the operational sites was formulated. If the null hypothesis can be rejected by having a low p-value (i.e. $p<0.05$), it can be statistically proven that the human performance indeed varies between the operational sites. The figure 4-3 shows that the average accuracy of human performance in identifying FA correctly at each operation site and the standard deviation.

Group Summary Table

Group	Count	Mean	Std Dev
1	33769	0.74595	0.43533
2	28887	0.7503	0.43285
3	25035	0.83667	0.36968
4	23792	0.81805	0.38581
5	22162	0.77877	0.41509
6	16456	0.90247	0.29669
7	13545	0.94315	0.23156
8	3951	0.74614	0.43527
9	3258	0.94751	0.22304
10	2356	0.9652	0.18332
11	666	0.93844	0.24054
12	392	0.79592	0.40354
13	344	0.81977	0.38494
14	198	0.76263	0.42655
15	197	0.68528	0.46559
16	21	0.47619	0.51177
17	158	1	0
18	19	0	0
19	31	0.90323	0.30054
20	4	0	0
21	24	1	0
22	15	0.93333	0.2582
23	10	0.9	0.31623
24	12	1	0
25	1	0	0
26	2	1	0
Pooled	175305	0.81132	0.38512
Levene's statistic (absolute)	1226		
Degrees of freedom	24, 175279		
p-value	0		

Figure 4-3. Levene's Test for human performance for FA across operational sites

The following Figure 4-4 shows that the average accuracy of human performance in identifying faults correctly at each operation site. The calculated p-values for both FA and faults were zeroes, which means that the null hypothesis can be rejected.

Group Summary Table

Group	Count	Mean	Std Dev
1	2216	0.31814	0.46586
2	1100	0.67545	0.46842
3	1155	0.69351	0.46124
4	422	0.34834	0.47701
5	1394	0.38307	0.48631
6	690	0.43188	0.4957
7	1008	0.52083	0.49981
8	191	0.51832	0.50098
9	58	0.75862	0.43166
10	86	0.72093	0.45117
11	22	0.36364	0.49237
12	11	0.72727	0.4671
13	14	0.85714	0.36314
14	15	0.46667	0.5164
15	4	0.75	0.5
16	144	0	0
17	9	1	0
18	24	0	0
19	4	1	0
20	5	1	0
21	99	0	0
22	5	0.4	0.54772
23	2	0.5	0.70711
25	2217	0.31845	0.46598
Pooled	10895	0.4335	0.46843
Levene's statistic (absolute)	130.66		
Degrees of freedom	23, 10871		
p-value	0		

Figure 4-4. Levene's Test for human performance for fault across operational sites

The percent error ranges from 5.58% to 31.34% for the Sites 1 through 15 as shown in Table A-1 in the Appendix. As the number of maintenance decreases, the percent errors spike up and down which ranges from 0% to 100%. The cumulative standard deviation clearly shows that the performance variance increases as the sites with less number of maintenance events are added to the overall fleet.

The 26 operational sites had the average error rate of 40.64% for identifying faults whereas the error rate of the RUSBoostedTrees algorithm in identifying faults was 45% as shown in Figure 4-5. The average error rate of 40.64% between the operational sites as

compared to 45% for machine learning algorithms could easily mis-lead people to think that machine-learning algorithm is performing worse than human but the average error rate of 40.64% was calculated using non-weighted error rate for each site. For example, almost one fifth of the entire maintenance events were from Site 1 and Site 1 had the error rate of 68.19%. Site 1 should have been the most influential site when it comes to calculating the overall error rate but it was only counted as one value along with the other 25 sites for calculating the average error rate between the operational sites. As it was shown in Table 4-1 earlier, the true weighted error rate for human throughout the entire fleet was 54%, which is 9% higher than the machine-learning algorithm. Again, the cumulative standard deviation of the percent errors shows that the performance variance increases as the newer sites with less experience are introduced to the overall fleet.

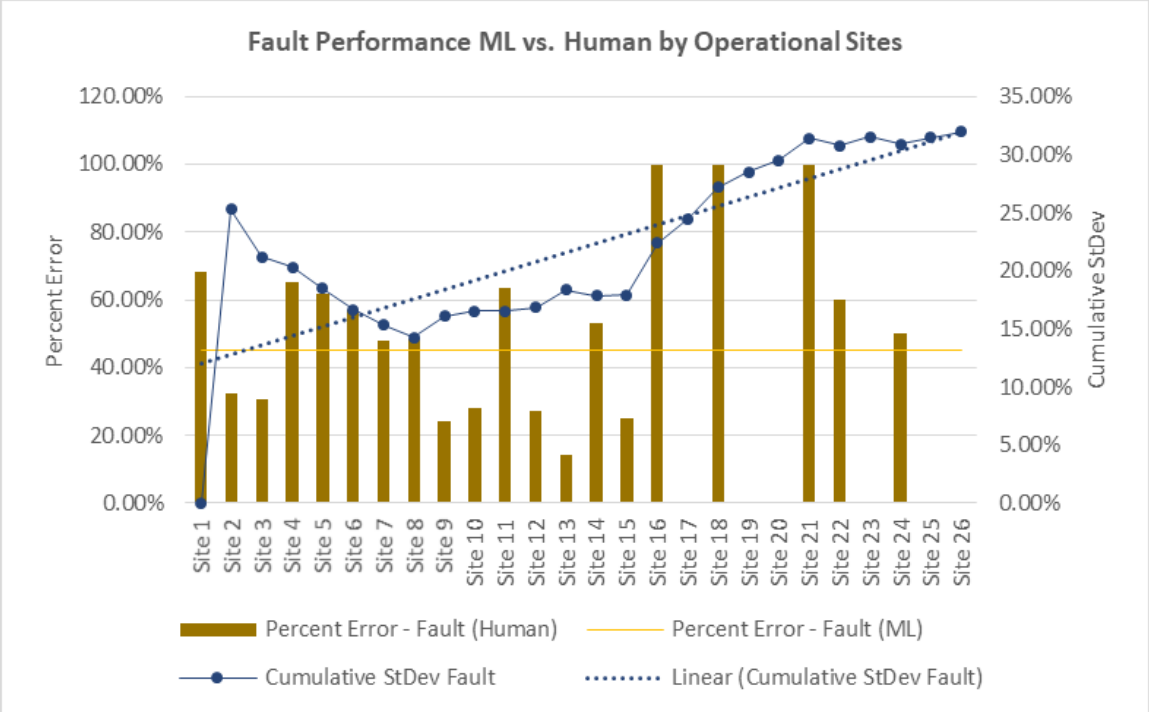


Figure 4-5. Machine-Learning vs. Human in identifying Faults by Operational Sites

As shown in Figure 4-6, the human maintainers had the error rate of 24.43% across the different operational sites in identifying false alarms. The RUSBoosted Trees algorithm yielded the error rate at 15%. The cumulative standard deviation again shows the similar trend as seen in the cumulative standard deviation numbers for faults.

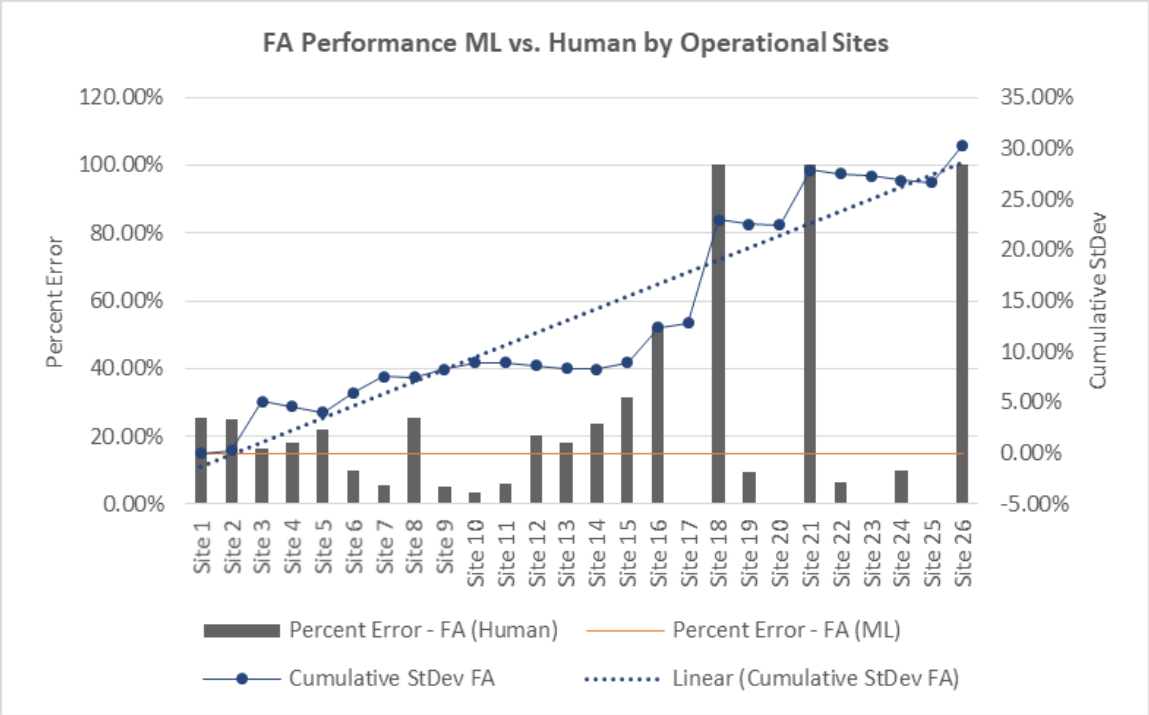


Figure 4-6. Machine-Learning vs. Human in identifying FA by Operational Sites

Both Figure 4-5 and Figure 4-6 had the trend lines for the cumulative standard deviation with positive slopes. In other words, as newer operations sites are added to the overall fleet, the standard deviation in the entire fleets also increases.

4.6 Machine Learning Improvement Rate

The fifth research question is the requirement on the number of input data in order for machine-learning algorithms to surpass the human performance in distinguishing false alarms and faults.

H5: The accuracy in distinguishing faults and false alarm using machine-learning classification algorithms will improve at a faster rate than the rate at which human maintainers will improve over time.

Figure 4-7 shows how many FID (I) codes were incorrectly assessed per 100 maintenance events. A step decrease is observed in the incorrect assessment rate for FA around May 2016. The possible rationale for the step improvement in identifying FAs will be explained in the next chapter. The performance level in identifying faults ranges from 0% to 100% and are distributed all over the charts without apparent trends. However, there are the slightly concentrated groups along the horizontal lines at the percent errors of 0%, 50% and 100%. The possible explanation for these three concentrated groups will be explained in the chapter 5.

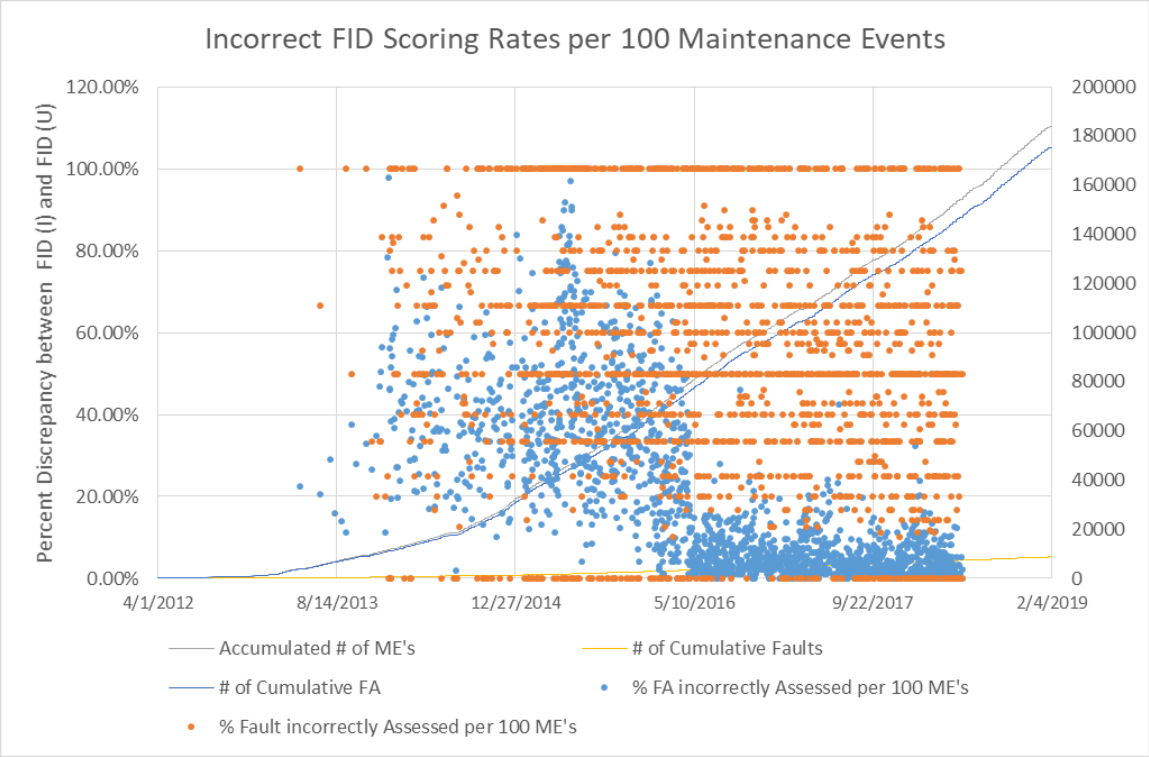


Figure 4-7. Incorrect Assessment Rate per 100 Maintenance Events

Since there is a step decrease around the May 2016 timeframe, Figure 4-7 was broken down into two different timeframes to fit trend lines over the scattered data points as shown in Figure 4-8 and Figure 4-9.

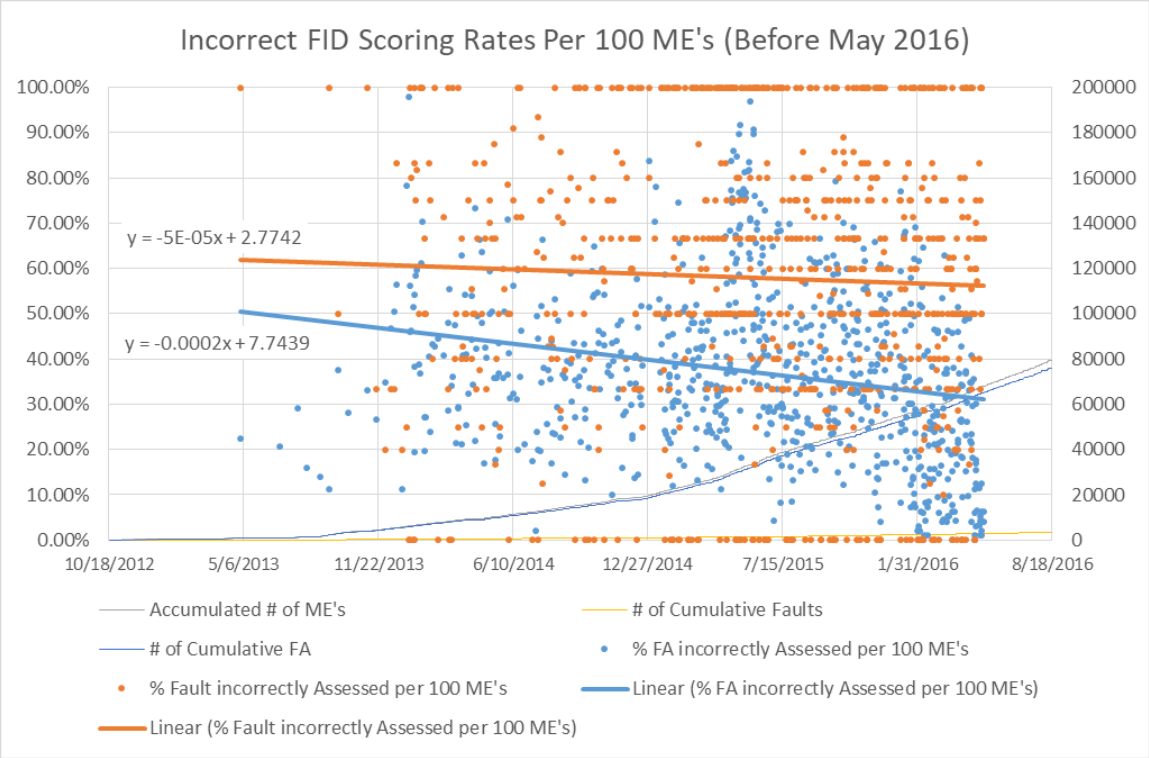


Figure 4-8. Incorrect Assessment Rate per 100 ME's prior to May 2016

The trend line shows a positive improvement in identifying false alarms over time. It has improved from 50% in 2013 to 30% in May 2016. However, the trend line for faults shows a very little improvement over the 3 years. As of May 2016, the incorrect FID scoring rate for faults is still over 50%.

There is a significant decrease in the incorrect FID scoring rate for false alarms post May 2016 as compared to years prior to 2016. As sown in Figure 4-9, there are approximately 100,000 data points between May 2016 and May 2018. The incorrect FID scoring rate for faults has improved from approximately 61% error to 42% according to

the linear trend line. According to the trend line, the accuracy would increase 0.019% over 100 maintenance events on average.

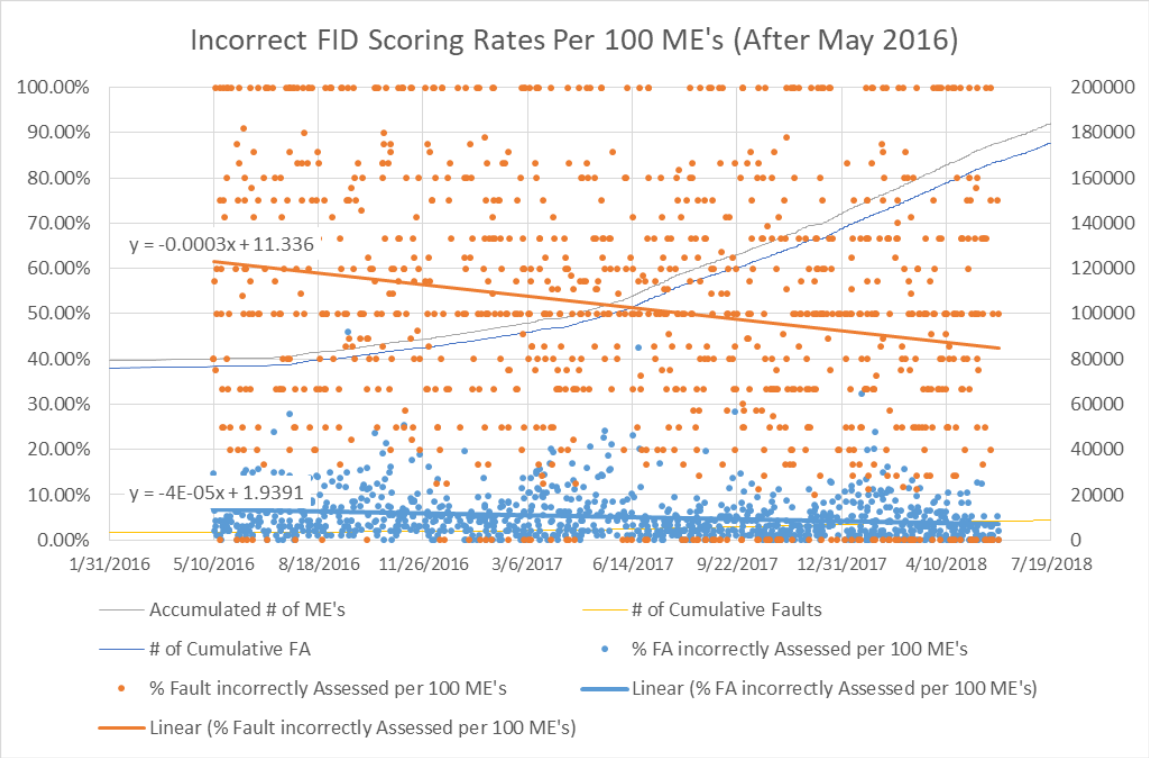


Figure 4-9. Incorrect Assessment Rate per 100 ME's post May 2016

By applying the factor of 0.019% improvement per 100 data points for human, the performance level in identifying fault was compared between human and ML as a function of number of data points. The machine-learning algorithm showed the improvement from 87% error to 44% error just in approximately 7000 data points. The Figure 4-10 shows both the actual percent errors for human maintainers and averaged percent errors based on the slope of the trend line. The machine learning algorithm starts at a higher percent rate with the limited data set of 606 maintenance events but improves at a faster rate and surpasses the human performance at approximately 3000th maintenance events.

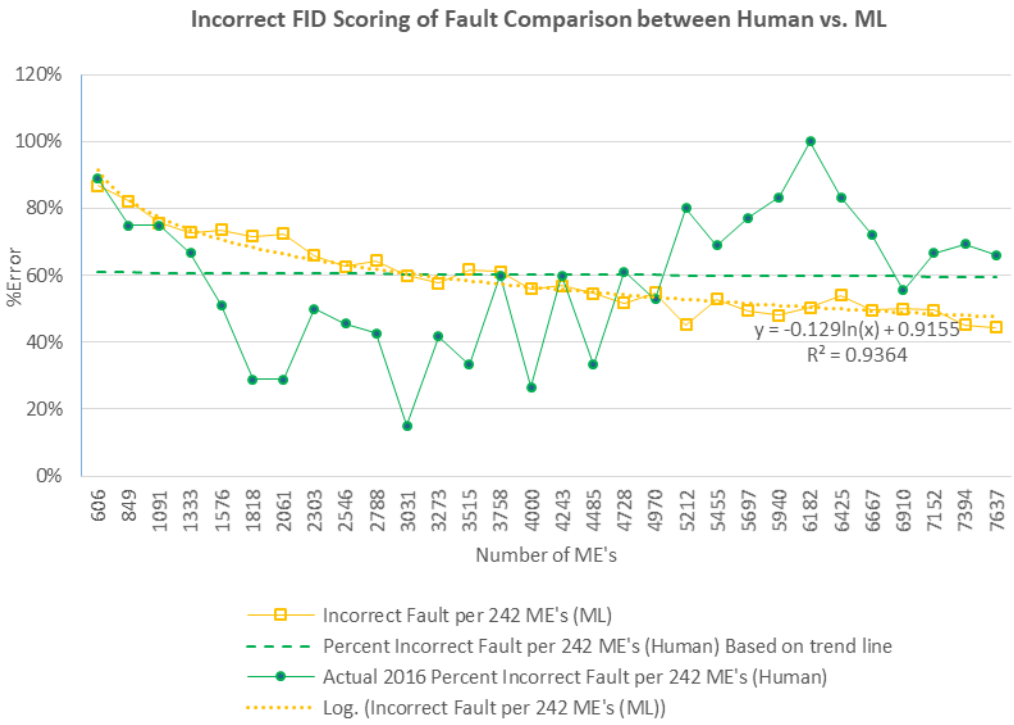


Figure 4-10. Incorrect FID Scoring Rates (Fault) vs. Number of data points

As shown in Figure 4-11, the percent error in identifying false alarms the human maintainers averages around 7% with a slow of -0.00004% per 100 maintenance events. Unlike the human performance where the percent errors decreased as the number of maintenance events increased, the error rate for the machine-learning algorithm actually increase from 5% to 14%. While the 14% error rate is still better than the overall human performance of 19% error rate for the entire data points, the probable cause for the better performance for human post May 2016 will be explained in the next chapter.

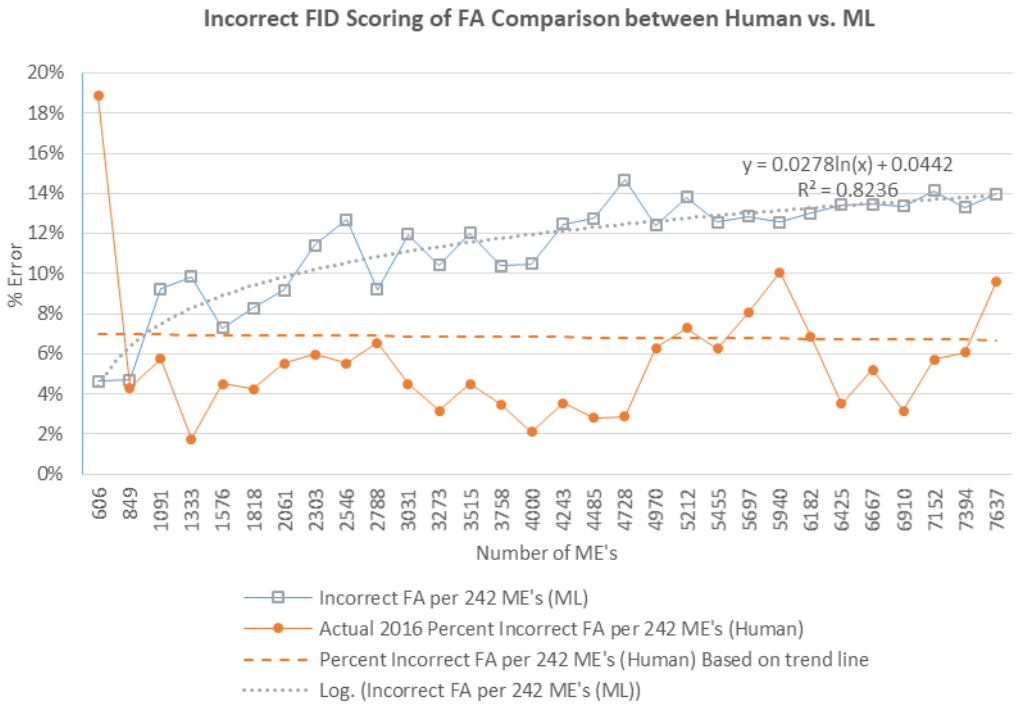


Figure 4-11. Incorrect FID Scoring Rate (FA) vs. Number of data points

4.7 Other Studies to Find the Optimal ML Algorithms and Settings

For better understanding of the imbalance in the input data, the following research was conducted. 13 different machine-learning algorithms were run with the different sets of input data that had various FA-to-Fault ratios. RUSBoostedTrees was the only model that was insensitive to the imbalance in the input data because RUSBoostedTrees has a built-in under-sampling technique to handle the imbalance in the data. The other models quickly lost their accuracy as the FA-to-Fault ratio went past 1:1 as shown in Figure 4-12.

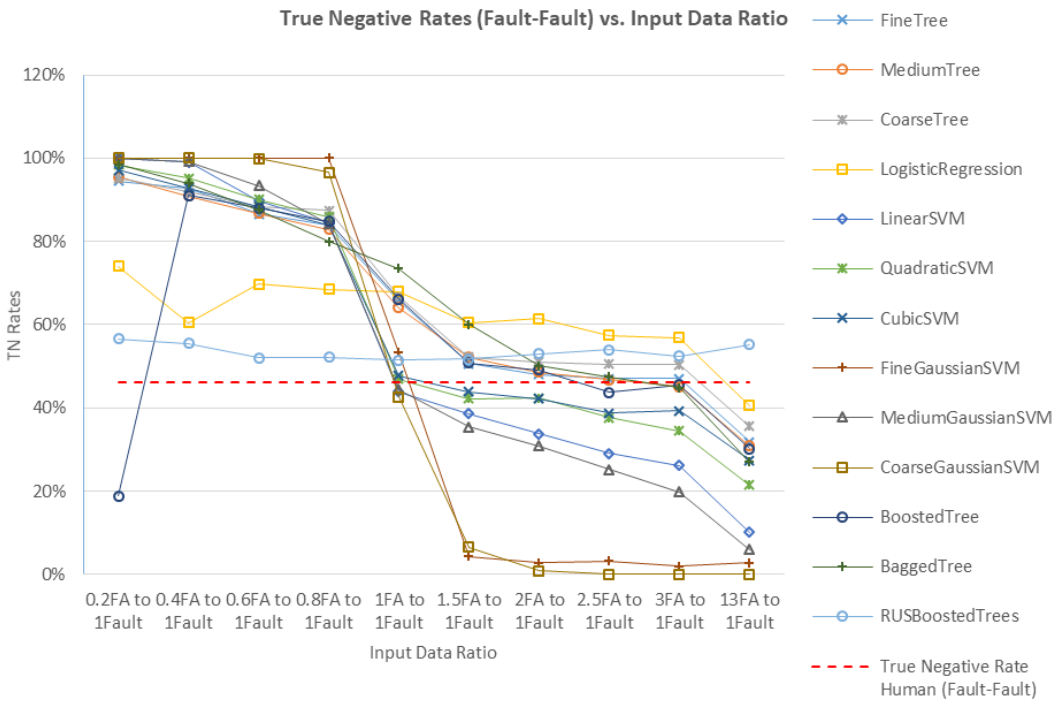


Figure 4-12. Fault Identification Accuracy vs. Data Sampling Ratios

Figure 4-13 shows that for all 13 algorithms, the accuracy of identifying FA increased as the FA-to-Fault ratio increased. Again, RUSBoostedTrees was not as sensitive to the imbalance in the input data as the other machine learning algorithms.

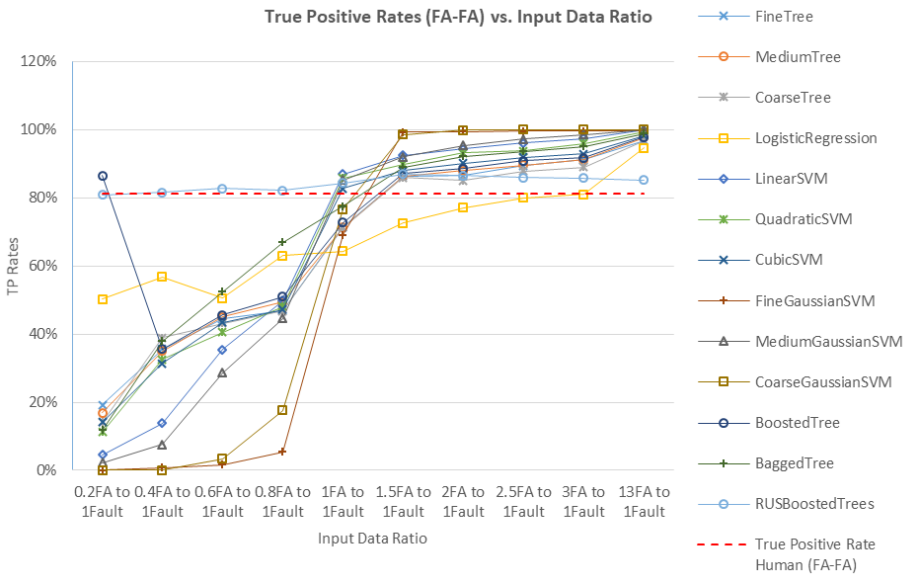


Figure 4-13. FA Identification Accuracy vs. Data Sampling Ratios

Out of the 130 different combinations tested to find the optimal input data ratio and the best algorithms, 27 cases had better results than the human performance accuracy of 81% for FA and 46% for Fault. The 27 cases were then sorted by the overall accuracy including both the fault performance and false alarm performance. The RUSBoostedTrees ranked as #1 with the overall accuracy of 83%. One notable observation is that BaggedTrees at 2.5 FA-to-Fault ratio yielded the 94% accuracy in identifying FA and 47% accuracy in identifying fault. If the goal were to reduce as much FA as possible as long as the fault detection rate is equal or better than the human performance level, the BaggedTrees algorithm would have chosen as the best algorithm. The rationale for choosing the best machine-learning algorithm will be explained in the chapter 5.

The next study was to examine the impacts of using different combinations of predictors by turning on and off certain predictors. Since the RUSBoostedTrees yielded the best result in the previous section, the RUSBoostedTrees algorithm was run with the different sets of input data with the FA-to-Fault ratio of 2:0 instead of using the full 12,122 data points to save the computing time. There were a total of 32,768 combinations using 15 different predictors. The top 20 results are listed in Table A-4 in the Appendix and the top five combinations are listed in Table 4-6. As seen in Table 4-6, the FA identification accuracy (TN rate) was 88% on average and Fault identification accuracy (TP rate) was 55% on average.

Table 4-6. Top 5 RUSBoostedTrees Results with Different Combinations of Predictors

TVE	SDL	Total # of HRCs	HRC	LCN Group	HAMI	Type of Test	# of HRC within 2 Min Window	Time Occurred	RST Count	Set Count	RES Count	Nz Max	Hp Max	Vcas Max	True Negative Rates (FA-FA)	True Positive Rates (Fault-Fault)	FA Negative Predictive Values	Fault Positive Predictive Values	Accuracy	ML Better than Human	Rank (Based on Accuracy)
1	0	1	1	0	1	1	0	0	0	0	1	1	0	1	88%	54%	79%	69%	76.7%	Yes	1
1	0	0	1	1	1	1	1	1	0	0	0	1	0	1	88%	55%	79%	69%	76.6%	Yes	2
1	1	0	1	0	0	0	1	1	0	0	0	1	0	1	87%	55%	80%	68%	76.5%	Yes	3
1	0	1	1	0	0	0	1	0	1	0	0	0	1	1	88%	53%	79%	69%	76.5%	Yes	4
1	0	1	1	0	1	0	1	1	0	1	0	1	1	1	87%	54%	79%	68%	76.5%	Yes	5
1	0	1	1	1	1	0	1	0	1	0	1	0	1	1	87%	56%	80%	68%	76.5%	Yes	5

As it was mentioned in the chapter 2, varying input parameters or predictors had little to no effect on the overall outcome of the RUSBoostedTrees results because the decision trees algorithms are tolerable and flexible with imbalanced data and input parameters that are less important.

Chapter 5—Discussion and Conclusions

5.1 Discussion

The machine learning algorithm results are promising and exceeded the human performance in both the determination speed and the accuracy. As it was mentioned in the chapter 4, it is very important to look at both the accuracy of false alarm identification and fault identification at the same time. As it is shown in Table A-2 in the Appendix, multiple algorithms yielded the FA identification accuracy of 100%. If these numbers were to be looked at separately independent of the fault identification accuracy, the result would have been ideal. However, the accuracy in identifying fault was often 0% for these algorithms. This happens when the algorithm is not flexible in handling imbalanced data. In other words, the algorithms predicted every single event as false alarms. Since approximately 95% of the actual maintenance events were false alarm, the overall accuracy of the machine learning came out to be also in the 93-95% range. This accuracy is a misleading indicator if the fault identification accuracy is not considered together. Misclassifying a fault as false alarm could affect the flight safety. Therefore, sacrificing the fault identification accuracy to increase the FA identification accuracy is not desirable and will not be accepted by the airworthiness authorities. As the bare minimum, the lowest threshold for the fault identification accuracy of machine learning algorithms will be set at the same or better performance than the human maintainers before it can be said that machine-learning algorithms can perform better than human can.

The following Table 5-1 shows the top two algorithms that exceeded the human performance in both false alarm identification and fault identification. The two results offer the benefits of using machine-learning algorithms from the two different ends of the

spectrum. RUSBootstedTrees had a better Fault identification accuracy than the BaggedTree model while the BaggedTree model had a much better accuracy in identifying false alarm. The tiebreaker was the overall accuracy. Even though, the FA identification accuracy for the BaggedTree was much higher, the FA-to-Fault ratio of the input data that was used to train the BaggedTree algorithm was low at 2.5 to 1 ratio. In other words, the inaccuracy in identifying faults would affect the overall accuracy more heavily.

Table 5-1. RUSBoostedTrees vs. BaggedTree Results

	True Negative Rates (FA-FA)	True Positive Rates (Fault-Fault)	FA Negative Predictive Values	Fault Positive Predictive Values	Accuracy
RUSBoostedTrees (13:1 FA-to-Fault Ratio)	85.24%	55.27%	96.00%	22.92%	83.03%
BaggedTree (2.5:1 FA-to-Fault Ratio)	93.68%	47.42%	81.67%	75.00%	80.46%

Not only the machine learning algorithm offers the better accuracy in distinguishing faults and false alarms than the human but it also takes less time to decide whether an HRC is a false alarm or not. Another cost benefit of using machine learning is more than just not having to spend the maintenance man-hours on but also the aircraft components are removed as part of the FA troubleshooting reduces the spare posture which impacts the aircraft downtime awaiting for parts and the mission capable rate (MCR). The MCR is very important key performance parameters for warfighters.

In section 4.2, the FRACAS data showed that 81% of the observed false alarms were accurately assessed as FA by the maintainers and 46% of the observed faults were

correctly assessed as faults. The negative predictive value (NPV) for identifying FA accurately was 97%. In other words, when a maintainer assessed an HRC as a false alarm, 97% if the times, it was indeed a false alarm. However, when the maintainers recorded the maintenance events as fault, only 11% of the ME's were actually faults and the 89% was false alarm which means that the maintainers must have spent some time troubleshooting and removed parts that were in good conditions. Out of 183,984 maintenance events related to either FA or Faults in the FRACAS database, approximately 95% of the ME's were false alarms but the maintainers reported 79.84% of the HRCs as false alarms and scored 20.16% of the HRCs initially as faults and spent the time troubleshooting FA events that once they thought were faults. This is the reason why knowing PPV values is as important as understanding the meaning of the true positive and negative rates. Just looking at the true positive and true negative rates do not paint the full picture.

According to the data shown in Table 4-1, 175,305 maintenance events were false alarms. Out of the 175,305 FA events, 30,147 events had more than 5 minutes spent on troubleshooting the false alarms. The average time spent on these 30,147 FA events was 62.4 minutes per FA. These FA events with maintenance man-hours are normal for the FRACAS database because what FID (I) captures is not maintainer's initial guess at whether the HRC would be a false alarm or not, but rather their final disposition after spending time to troubleshoot and remove & replace parts to resolve the asserted HRC.

For the purpose of the praxis where the goal is to reduce the number of false alarms as well as the maintenance man-hours associated hours, these false alarm events should not have been scored as FA under FID (I). If the machine-learning algorithm had

told the maintainers that the HRC was a false alarm, the maintainers would not have spent the time on troubleshooting. If these FA events were initially scored as faults, then the human performance in identifying false alarms would have been reduced from 81% to 64% as shown in Table 5-2. The PPV value for identifying fault would have been also lowered to from 11% to 6%.

Table 5-2. Human Performance with Corrected Number of FA events

Final Disposition FID (U)	FA	4018	4661	46%
	Fault	63223	112082	64%
		FA	Fault	Accuracy
		6%	96%	79%
		Initial Disposition FID (I)		

Instead of adjusting the FRACAS database, the total maintenance hours spent on false alarm events was treated as the time it takes for human to come to a conclusion as to whether an HRC is a false alarm or not. In other words, human maintainers will take approximately 11 minutes of their time before they can decide whether the HRC is a false alarm or not. 11 minutes per HRC may not sound like a big deal but there are thousands of these FA events that can quickly add to cost the program $1.43M. The labor hours of the maintainers can be translated into the avoidable cost. Considering this dollar figure was derived from just 4 years' worth of data and the total number of aircrafts will increase by 10 folds in the upcoming years, the impact from spending 11 minutes per FA

HRC will exponentially accumulate as more aircrafts are delivered and operate in the next 50 years.

Figure 4-7 shows that in 2016, the error rates for identifying false alarms by the maintainers decreased significantly. Prior to 2016, the error rates for identifying false alarms were typically between 30% and 50% but there is a big step decrease in May of 2016 which brought the error rates down to around 7%. Two possible factors could have contributed to the sudden change from FID (I) to FID (U). The first factor is the maintainers themselves. FID (I) is assigned by a maintainer so the maintainer is one-half of the equation. The 5th generation fighter aircraft program carries out routine meetings with the operational units to provide maintainers the information about recent trends and lessons-learned as well as to emphasize the importance of assigning FID codes correctly. However, this factor can be ruled out easily by looking at the steep decrease in the error rates for false alarm shown in Figure 4-7. Even if the routine maintenance awareness meetings were held, the impacts to FID codes would have shown gradually instead of all at once over one night. The other possible factor is the SME review that assigns FID (U). They are the second-half of the equation in the FID(I)-FID(U) relationship. While maintainers assign the FID (I) after individual maintenance event, FID (U) is assigned after SME reviews, which happens at a longer interval such as every three months. If there was any change in the guidance as to how SMEs were supposed to score FID codes differently, the impact would indeed show up all at once during the next round of FID (U) assessment.

The ratios between the number of false alarms and faults was compared against the ratio between number of incorrect FA assessment and the number of incorrect faults

assessment. Figure 5-1 shows that the moving average trend line for the FA-to-Fault ratio is consistent before and after May 2016. However, the trend line for the incorrect assessments has a step decrease around in May 2016. This is the sign that the similar ratio of false alarms to faults are asserting on the aircrafts but the way that the FID codes are assigned have changed in 2016.

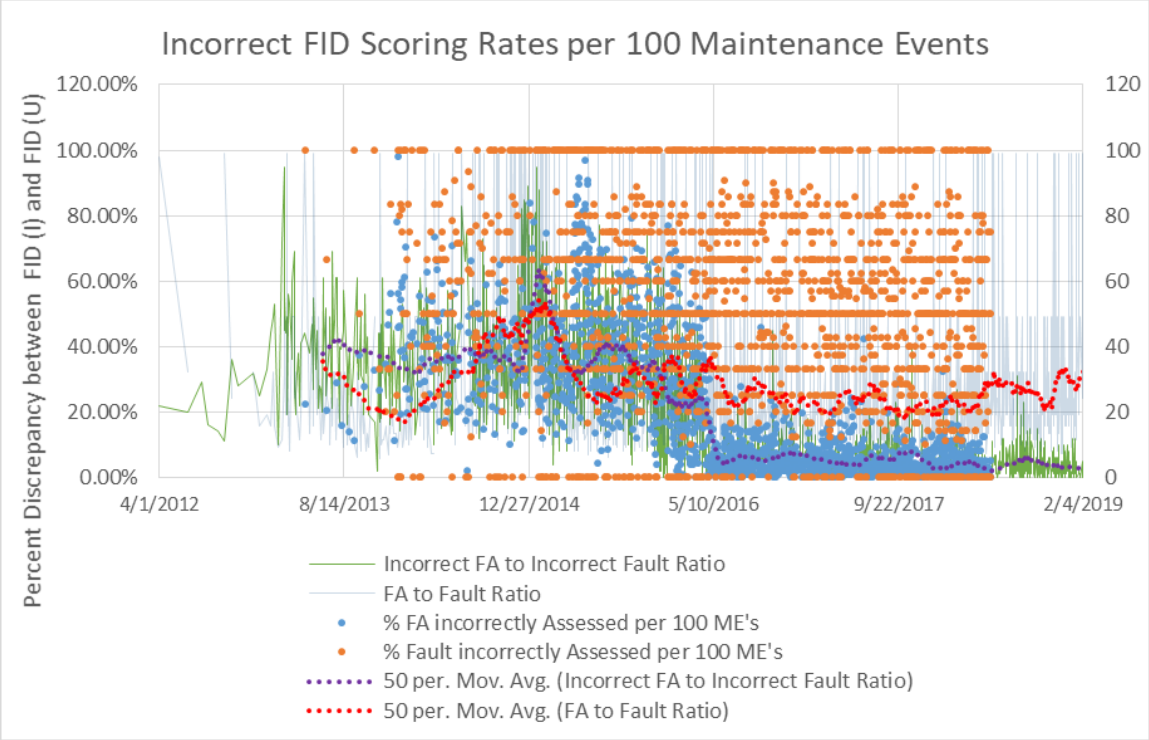


Figure 5-1. Trend Change in FID Code Scoring in 2016

As it was previously mentioned in the chapter 4, the incorrect assessment rates for faults were concentrated around 0%, 50% and 100%. In other words, maintainers either knew how to distinguish faults 100% of the times, randomly picked FID codes at the 50/50 chance or always picked the wrong or irrelevant FID Codes. In addition, 44% of the inconsistent FID codes for faults were in the 0% and 50% percent error rante and 56% of the results were within the 50% to 100% range. In other words, there were more maintenance events that were assigned with the wrong FID codes initially than the

number of correct assignments. Since more data points are present in the 50% to 100% range, it is reasonable to say that the maintainers did not know how to distinguish faults and false alarms.

Figure 4-12 shows that as the number of input data increased, the accuracy of identifying false alarm actually decreased. This behavior is caused by the introduction of more “fault” data. The first data point on the graph is the result after training the classification model with only 606 files. Considering only 5 % of the input data was faults, there was not enough “fault” data and the RUSBoostedTrees by design under-samples the imbalanced data. As more input data is introduced, the trained model will find the optimal balance between the accuracy of identifying FA (true negative rate) and the accuracy of identifying faults (true positive rate). While the 14% error rate is still better than the overall human performance of 19% error rate for the entire data points, the human performance based on the post-2016 data showed a lower error rate in identifying false alarms. This drastically lowered numbers could have resulted from the possible change in the FID scoring practice by the SMEs when updating FID (U) as it was explained in the previous section.

After conducting analyses for all five hypotheses, the RUSBoostedTrees algorithm was selected as the best classification algorithm for the enhanced filtering application. The rationale for choosing the best machine-learning algorithm will be explained in the chapter 5.

First, RUSBoostedTrees offers a great flexibility when dealing with imbalanced and uncleaned data. During the data preparation, there were lots of aircraft flight data that had missing information but the RUSBoostedTree algorithm made it possible to use this

data with missing gaps. Otherwise, the input data had to be a lot smaller in order to exclude any data with missing information.

Second, while it would be extremely helpful to fully understand all the failure modes and how each of the failure modes is linked to HRCs, but RUSBoostedTrees can allows user to use any input parameters and the algorithm does its own iterative process to determine which parameters are more important than the others. This makes it possible for the future research to investigate a variety of different predictors in order to increase the accuracy of the classification model.

5.2 Conclusions

This praxis has demonstrated the possibility and the benefits of using machine-learning classification models to distinguish false alarms and faults before HRCs are presented to the maintainers in order to avoid any unnecessary maintenance actions on false alarms. The supervised machine-learning classification models were trained with the PHM data from the 5th generation fighter aircrafts along with the labeled data from FRACAS. The results have shown that the RUSBoostedTrees algorithm yields the best results at identifying faults and false alarms with the accuracies of 85% and 55%, respectively, as compared to the human performance levels of 81% and 46%, respectively. Not only the machine learning algorithm distinguished false alarms and faults with higher accuracies, it could also eliminate the time it takes human to determine whether an HRC is a false alarm or not. Based on the FRACAS data, it took 11 minutes on average for the maintainers to spend the time on FA events and realize that it was a false alarm. Since the machine learning algorithms can be trained using the enterprise data and not just from the local units, the entire fleet's corporate knowledge can be

captured and be utilized by any sites that are newly standing with less experienced maintainers. It was also proven that the accuracy of the machine-learning algorithm can be improved with less number of data or maintenance events than what it takes for human to become better at distinguishing false alarms and faults.

As more aircrafts are delivered to the users, the need for this enhanced filtering capability will increase and the benefits of using the ML algorithms will be much more appreciated until all root causes of false alarms are completely identified and resolved at the source of the problems.

5.3 Contributions to Body of Knowledge

1. Using PHM data and FRACAS to generate the input parameters for supervised machine-learning classification models to mimic what an experienced maintainer would typically do to figure out whether an HRC is a false alarm or not is a new approach and it has been proven in this praxis that this method if feasible and can perform better than human in distinguishing faults and false alarms.
2. This praxis presented various input parameters that are valuable in determining whether an HRC is a false alarm or not.
3. The results have shown that RUSBoostedTrees is the optimal model for the machine-learning classification problem to distinguish false alarms and faults for the 5^{th} generation fighter aircraft application

5.4 Recommendations for Future Research

This praxis has introduced the idea and the possibility of using machine-learning classification models to distinguish aircraft faults and false alarms in order to create an

enhanced filtering in the off-board PHM system. The results have shown that the machine-learning algorithms can distinguish false alarms and faults at a higher accuracy and speed than human. The accuracies of the machine-learning model in identifying false alarms and faults were 85% and 55%, respectively, which are better than the human performance of 81% and 46%, respectively. However, it may take the accuracy to be in the upper 90% range before human maintainers might start trusting and using the machine-made decisions. Therefore, the future research can focus on improving the accuracies of the machine predictions.

One of the possible ways to improve the accuracy is by experimenting with different sets of input parameters. Since RUSBoostedTrees is flexible enough to entertain virtually any contributing factors whether they are important or not, many of the PHM data elements that have not been tested can be included. In addition, the understanding of each HRC and its source data at the BIT level would be extremely helpful to design supervised machine-learning algorithms to detect anomalies by implementing the similar approach as presented in Moreira's research (de Padua Moreira & Nascimento Jr 2012) in identifying false alarm in the health data of bleed valve or the ICU false alarm reduction studies done by Fahmi et al or Wang et al. (Fahmi et al., 2012 and Wang et al., 2015).

References

Bishop, Y.M.M., Fienberg, S.E. & Holland, P.W. (1975). Discrete Multivariate Analysis. Cambridge, MA: MIT Press. pp.18-37

Breiman, L. (1996). Bagging predictors. Machine Learning 24. pp. 123–140

De Mantaras & Armengol E. (1998). Machine learning from examples: Inductive and Lazy methods. Data & Knowledge Engineering 25. pp. 99- 123.

de Padua Moreira, R. & Nascimento Jr, C. (2012). Prognostics of aircraft bleed valves using a SVM classification algorithm. IEEE Aerospace Conference Proceedings. pp. 1-8. doi:10.1109/AERO.2012.6187377.

Dietterich, T.G. (2000). Ensemble Methods in Machine Learning. In: Multiple Classifier Systems. MCS 2000. Lecture Notes in Computer Science, Volume 1857. Springer, Berlin, Heidelberg. pp. 1-15

Doyle, P. (1973). The use of automatic interaction detector and similar search procedures. Oper. Res. Quart., 24, pp. 465–467.

Fahmi, B. R., Kaouther, N., Trabelsi, A. (2012). *Support Vector Machines Versus Multi-Layer Perceptrons for Reducing False Alarms in Intensive Care Units*, International Journal of Computer Applications, Volume 49– No.11. pp. 41-47

Gilmore, J. M. (2015). *Director, Operational Test and Evaluation FY 2015 Annual Report*, pp. 35-82. Retrieved from https://www.dote.osd.mil/pub/reports/FY2015/pdf/dod/2015f35jsf.pdf

Hodge, V., Austin, J. (2004), A Survey of Outlier Detection Methodologies, Artificial Intelligence Review, Volume 22, Issue 2, pp. 85-126.

Japkowicz N. and Stephen, S. (2002), The Class Imbalance Problem: A Systematic Study Intelligent Data Analysis, Volume 6, Number 5. pp. 429-449

Jayasree, S. and Gavya, A. (2014), *Addressing imbalance problem in the class – A survey,* International Journal of Application or Innovation in Engineering & Management (IJAIEM), Volume 03, Issue 09. pp. 239-243

Kotsiantis, S.B (2007) "Supervised Machine Learning: A Review of Classification Techniques", Informatica 31. pp. 249-268

Lawless, S. T., (1994). Crying wolf: false alarms in a pediatric intensive care unit. Critical care medicine, vol. 22, no. 6, pp. 981–985

Lockheed Martin (2018), *F-35 Lighting II Program Status and Fast Facts.* Retrieved from https://www.lockheedmartin.com/content/dam/lockheed-martin/aero/documents/F-35/F-35%20Fast%20Facts%20-%20August%202018.pdf

Richert, W. & Coelho, L. P. (2013) *Building Machine Learning Systems with Python,* Packt Publishing Ltd., ISBN 978-1-78216-140-0

U.S Environmental Protection Agency (2003), *On-Board Diagnostic (OBD) Regulations and Requirements: Questions and Answers* (EPA420-F-03-042). pp. 2. Retrieved from https://nepis.epa.gov/Exe/ZyPURL.cgi?Dockey=P100LW9G.TXT

Vasseur, P. (2018), *A Machine Learning Approach to Confirm and Reduce False Positive Alarms Generated by Data Leak Detection Processes.* Student-Faculty Research Day, CSIS, Pace University. pp. C4-1-9

Vennapusa, M., Bhyrapuneni, S., (2019) A Comprehensive Study of Machine Learning Mechanisms On Big data. International Journal of Recent Technology and Engineering (IJRTE). Volume 7. Issue 6S2. pp. 773-779

Wang, X., Gao, Y., Lin, J., Rangwala, H., Mittu. R., (2015). *A Machine Learning Approach to False Alarm Detection for Critical Arrhythmia Alarms*. IEEE 14th International Conference on Machine Learning and Applications. pp. 202-207

Zhang, G. (2000), Neural networks for classification: a survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C 30(4). pp. 451- 462.

Appendix A

Table A-1. Number of Discrepancies between FID (I) and FID (U)

Operational Sites	Fault to Fault	% Fault to Fault	Non-FA to FA	% Non-FA to FA	FA to FA	% FA to FA	Non-Fault to Fault	% Non-Fault to Fault	Total # of ME's	Incorrect Scoring
Site 1	705	1.96%	8579	23.84%	25190	70.00%	1511	4.20%	35985	28.04%
Site 2	743	2.48%	7213	24.05%	21674	72.28%	357	1.19%	29987	25.24%
Site 3	801	3.06%	4089	15.61%	20946	79.98%	354	1.35%	26190	16.96%
Site 4	147	0.61%	4329	17.88%	19463	80.38%	275	1.14%	24214	19.01%
Site 5	534	2.27%	4903	20.81%	17259	73.27%	860	3.65%	23556	24.47%
Site 6	298	1.74%	1605	9.36%	14851	86.61%	392	2.29%	17146	11.65%
Site 7	525	3.61%	770	5.29%	12775	87.78%	483	3.32%	14553	8.61%
Site 8	99	2.39%	1003	24.22%	2948	71.17%	92	2.22%	4142	26.44%
Site 9	44	1.33%	171	5.16%	3087	93.09%	14	0.42%	3316	5.58%
Site 10	62	2.54%	82	3.36%	2274	93.12%	24	0.98%	2442	4.34%
Site 11	8	1.16%	41	5.96%	625	90.84%	14	2.03%	688	7.99%
Site 12	8	1.99%	80	19.85%	312	77.42%	3	0.74%	403	20.60%
Site 13	12	3.35%	62	17.32%	282	78.77%	2	0.56%	358	17.88%
Site 14	7	3.29%	47	22.07%	151	70.89%	8	3.76%	213	25.82%
Site 15	3	1.49%	62	30.85%	135	67.16%	1	0.50%	201	31.34%
Site 16	0	0.00%	11	6.67%	10	6.06%	144	87.27%	165	93.94%
Site 17	9	5.39%	0	0.00%	158	94.61%	0	0.00%	167	0.00%
Site 18*	0	0.00%	4	14.29%	0	0.00%	24	85.71%	28	100.00%
Site 19	4	11.43%	3	8.57%	28	80.00%	0	0.00%	35	8.57%
Site 20	5	17.24%	0	0.00%	24	82.76%	0	0.00%	29	0.00%
Site 21*	0	0.00%	19	16.10%	0	0.00%	99	83.90%	118	100.00%
Site 22	2	10.00%	1	5.00%	14	70.00%	3	15.00%	20	20.00%
Site 23	0	0.00%	0	0.00%	12	100.00%	0	0.00%	12	0.00%
Site 24	1	8.33%	1	8.33%	9	75.00%	1	8.33%	12	16.67%
Site 25	0	0.00%	0	0.00%	2	100.00%	0	0.00%	2	0.00%
Site 26	1	50.00%	1	50.00%	0	0.00%	0	0.00%	2	50.00%
Avg**		5.65%		13.51%		75.05%		5.79%		19.3%
StDev**		10.31%		12.13%		24.25%		17.67%		20.03%

* Developmental Test Sites

** Excluding DT Sites

Table A-2. Various FA to Fault Data Sampling Ratio

Model	Data Sampling Ratio	True Negative Rate Human (FA-FA)	True Positive Rate Human (Fault-Fault)	Human Accuracy	TP (TN FA)	FN (I FN Fault	FN (F FP - FA)	TN (Fa TP - Fault)	True Positive Rates (FA-FA)	True Negative Rates (Fault-Fault)	FA Positive Predictive Values	Fault Negative Predictive Values	Accuracy	ML Better than Human	Rank (Based on Accuracy)
FineTree	0.2FA to 1Fault	81%	46%	79%	34	143	49	843	19%	95%	41%	85%	82.0%	No	N/A
	0.4FA to 1Fault	81%	46%	79%	126	228	64	828	36%	93%	66%	78%	76.6%	No	N/A
	0.6FA to 1Fault	81%	46%	79%	236	295	119	773	44%	87%	66%	72%	70.9%	No	N/A
	0.8FA to 1Fault	81%	46%	79%	332	376	143	749	47%	84%	70%	67%	67.6%	No	N/A
	1FA to 1Fault	81%	46%	79%	637	255	307	585	71%	66%	67%	70%	68.5%	No	N/A
	1.5FA to 1Fault	81%	46%	79%	1155	183	440	452	86%	51%	72%	71%	72.1%	Yes	21
	2FA to 1Fault	81%	46%	79%	1544	240	464	428	87%	48%	77%	64%	73.7%	Yes	16
	2.5FA to 1Fault	81%	46%	79%	1995	235	472	420	89%	47%	81%	64%	77.4%	Yes	8
	3FA to 1Fault	81%	46%	79%	2438	238	472	420	91%	47%	84%	64%	80.1%	Yes	3
	13FA to 1Fault	81%	46%	79%	10863	367	608	284	97%	32%	95%	44%	92.0%	No	N/A
MediumTree	0.2FA to 1Fault	81%	46%	79%	30	147	41	851	17%	95%	42%	85%	82.4%	No	N/A
	0.4FA to 1Fault	81%	46%	79%	124	230	81	811	35%	91%	60%	78%	75.0%	No	N/A
	0.6FA to 1Fault	81%	46%	79%	239	292	118	774	45%	87%	67%	73%	71.2%	No	N/A
	0.8FA to 1Fault	81%	46%	79%	350	358	153	739	49%	83%	70%	67%	68.1%	No	N/A
	1FA to 1Fault	81%	46%	79%	636	256	320	572	71%	64%	67%	69%	67.7%	No	N/A
	1.5FA to 1Fault	81%	46%	79%	1155	183	426	466	86%	52%	73%	72%	72.7%	Yes	18
	2FA to 1Fault	81%	46%	79%	1570	214	459	433	88%	49%	77%	67%	74.9%	Yes	14
	2.5FA to 1Fault	81%	46%	79%	1998	232	475	417	90%	47%	81%	64%	77.4%	Yes	8
	3FA to 1Fault	81%	46%	79%	2439	237	490	402	91%	45%	83%	63%	79.6%	No	N/A
	13FA to 1Fault	81%	46%	79%	10975	255	617	275	98%	31%	95%	52%	92.8%	No	N/A
CoarseTree	0.2FA to 1Fault	81%	46%	79%	25	152	43	849	14%	95%	37%	85%	81.8%	No	N/A
	0.4FA to 1Fault	81%	46%	79%	138	216	73	819	39%	92%	65%	79%	76.8%	No	N/A
	0.6FA to 1Fault	81%	46%	79%	229	302	105	787	43%	88%	69%	72%	71.4%	No	N/A
	0.8FA to 1Fault	81%	46%	79%	331	377	112	780	47%	87%	75%	67%	69.4%	No	N/A
	1FA to 1Fault	81%	46%	79%	636	256	296	596	71%	67%	68%	70%	69.1%	No	N/A
	1.5FA to 1Fault	81%	46%	79%	1149	189	426	466	86%	52%	73%	71%	72.4%	Yes	20
	2FA to 1Fault	81%	46%	79%	1518	266	436	456	85%	51%	78%	63%	73.8%	Yes	15
	2.5FA to 1Fault	81%	46%	79%	1959	271	441	451	88%	51%	82%	62%	77.2%	Yes	10
	3FA to 1Fault	81%	46%	79%	2380	296	442	450	89%	50%	84%	60%	79.3%	Yes	4
	13FA to 1Fault	81%	46%	79%	10861	369	574	318	97%	36%	95%	46%	92.2%	No	N/A
LogisticRegression	0.2FA to 1Fault	81%	46%	79%	89	88	232	660	50%	74%	28%	88%	70.1%	No	N/A
	0.4FA to 1Fault	81%	46%	79%	201	153	352	540	57%	61%	36%	78%	59.5%	No	N/A
	0.6FA to 1Fault	81%	46%	79%	268	263	270	622	50%	70%	50%	70%	62.5%	No	N/A
	0.8FA to 1Fault	81%	46%	79%	446	262	281	611	63%	68%	61%	70%	66.1%	No	N/A
	1FA to 1Fault	81%	46%	79%	574	318	286	606	64%	68%	67%	66%	66.1%	No	N/A
	1.5FA to 1Fault	81%	46%	79%	971	367	352	540	73%	61%	73%	60%	67.8%	No	N/A
	2FA to 1Fault	81%	46%	79%	1375	409	344	548	77%	61%	80%	57%	71.9%	No	N/A
	2.5FA to 1Fault	81%	46%	79%	1784	446	380	512	80%	57%	82%	53%	73.5%	No	N/A
	3FA to 1Fault	81%	46%	79%	2166	510	384	508	81%	57%	85%	50%	74.9%	No	N/A
	13FA to 1Fault	81%	46%	79%	10623	607	530	362	95%	41%	95%	37%	90.6%	No	N/A
LinearSVM	0.2FA to 1Fault	81%	46%	79%	8	169	0	892	5%	100%	100%	84%	84.2%	No	N/A
	0.4FA to 1Fault	81%	46%	79%	49	305	9	883	14%	99%	84%	74%	74.8%	No	N/A
	0.6FA to 1Fault	81%	46%	79%	188	343	91	801	35%	90%	67%	70%	69.5%	No	N/A
	0.8FA to 1Fault	81%	46%	79%	353	355	139	753	50%	84%	72%	68%	69.1%	No	N/A
	1FA to 1Fault	81%	46%	79%	776	116	501	391	87%	44%	61%	77%	65.4%	No	N/A
	1.5FA to 1Fault	81%	46%	79%	1236	102	547	345	92%	39%	69%	77%	70.9%	No	N/A
	2FA to 1Fault	81%	46%	79%	1684	100	591	301	94%	34%	74%	75%	74.2%	No	N/A
	2.5FA to 1Fault	81%	46%	79%	2142	88	633	259	96%	29%	77%	75%	76.9%	No	N/A
	3FA to 1Fault	81%	46%	79%	2606	70	658	234	97%	26%	80%	77%	79.6%	No	N/A
	13FA to 1Fault	81%	46%	79%	11220	10	802	90	100%	10%	93%	90%	93.3%	No	N/A
QuadraticSVM	0.2FA to 1Fault	81%	46%	79%	20	157	15	877	11%	98%	57%	85%	83.9%	No	N/A
	0.4FA to 1Fault	81%	46%	79%	115	239	43	849	32%	95%	73%	78%	77.4%	No	N/A
	0.6FA to 1Fault	81%	46%	79%	215	316	89	803	40%	90%	71%	72%	71.5%	No	N/A
	0.8FA to 1Fault	81%	46%	79%	340	368	125	767	48%	86%	73%	68%	69.2%	No	N/A
	1FA to 1Fault	81%	46%	79%	765	127	474	418	86%	47%	62%	77%	66.3%	Yes	23
	1.5FA to 1Fault	81%	46%	79%	1202	136	515	377	90%	42%	70%	73%	70.8%	No	N/A
	2FA to 1Fault	81%	46%	79%	1663	121	514	378	93%	42%	76%	76%	76.3%	No	N/A
	2.5FA to 1Fault	81%	46%	79%	2092	138	556	336	94%	38%	79%	71%	77.8%	No	N/A
	3FA to 1Fault	81%	46%	79%	2567	109	584	308	96%	35%	81%	74%	80.6%	No	N/A
	13FA to 1Fault	81%	46%	79%	11167	63	700	192	99%	22%	94%	75%	93.7%	No	N/A

Table A-2. Various FA to Fault Data Sampling Ratio (Cont'd)

CubicSVM	0.2FA to 1Fault	81%	46%	79%	25	152	26	866	14%	97%	49%	85%	83.3%	No	N/A
	0.4FA to 1Fault	81%	46%	79%	111	243	64	828	31%	93%	63%	77%	75.4%	No	N/A
	0.6FA to 1Fault	81%	46%	79%	231	300	104	788	44%	88%	69%	72%	71.6%	No	N/A
	0.8FA to 1Fault	81%	46%	79%	334	374	143	749	47%	84%	70%	67%	67.7%	No	N/A
	1FA to 1Fault	81%	46%	79%	738	154	466	426	83%	48%	61%	73%	65.2%	Yes	25
	1.5FA to 1Fault	81%	46%	79%	1176	162	500	392	88%	44%	70%	71%	70.3%	No	N/A
	2FA to 1Fault	81%	46%	79%	1608	176	515	377	90%	42%	76%	68%	74.2%	No	N/A
	2.5FA to 1Fault	81%	46%	79%	2048	182	546	346	92%	39%	79%	66%	76.7%	No	N/A
	3FA to 1Fault	81%	46%	79%	2487	189	541	351	93%	39%	82%	65%	79.5%	No	N/A
	13FA to 1Fault	81%	46%	79%	11028	202	647	245	98%	27%	94%	55%	93.0%	No	N/A
FineGaussianSVM	0.2FA to 1Fault	81%	46%	79%	0	177	0	892	0%	100%	0%	83%	83.4%	No	N/A
	0.4FA to 1Fault	81%	46%	79%	3	351	0	892	1%	100%	100%	72%	71.8%	No	N/A
	0.6FA to 1Fault	81%	46%	79%	9	522	0	892	2%	100%	100%	63%	63.3%	No	N/A
	0.8FA to 1Fault	81%	46%	79%	38	670	0	892	5%	100%	100%	57%	58.1%	No	N/A
	1FA to 1Fault	81%	46%	79%	616	276	417	475	69%	53%	60%	63%	61.2%	No	N/A
	1.5FA to 1Fault	81%	46%	79%	1328	10	853	39	99%	4%	61%	80%	61.3%	No	N/A
	2FA to 1Fault	81%	46%	79%	1773	11	867	25	99%	3%	67%	69%	67.2%	No	N/A
	2.5FA to 1Fault	81%	46%	79%	2219	11	863	29	100%	3%	72%	73%	72.0%	No	N/A
	3FA to 1Fault	81%	46%	79%	2665	11	874	18	100%	2%	75%	62%	75.2%	No	N/A
	13FA to 1Fault	81%	46%	79%	11198	32	867	25	100%	3%	93%	44%	92.6%	No	N/A
MediumGaussianSVM	0.2FA to 1Fault	81%	46%	79%	4	173	1	891	2%	100%	80%	84%	83.7%	No	N/A
	0.4FA to 1Fault	81%	46%	79%	27	327	7	885	8%	99%	79%	73%	73.2%	No	N/A
	0.6FA to 1Fault	81%	46%	79%	152	379	59	833	29%	93%	72%	69%	69.2%	No	N/A
	0.8FA to 1Fault	81%	46%	79%	316	392	140	752	45%	84%	69%	66%	66.8%	No	N/A
	1FA to 1Fault	81%	46%	79%	760	132	495	397	85%	45%	61%	75%	64.9%	No	N/A
	1.5FA to 1Fault	81%	46%	79%	1231	107	576	316	92%	35%	68%	75%	69.4%	No	N/A
	2FA to 1Fault	81%	46%	79%	1701	83	616	276	95%	31%	73%	77%	73.9%	No	N/A
	2.5FA to 1Fault	81%	46%	79%	2167	63	667	225	97%	25%	76%	78%	76.6%	No	N/A
	3FA to 1Fault	81%	46%	79%	2638	38	715	177	99%	20%	79%	82%	78.9%	No	N/A
	13FA to 1Fault	81%	46%	79%	11226	4	839	53	100%	6%	93%	93%	93.0%	No	N/A
CoarseGaussianSVM	0.2FA to 1Fault	81%	46%	79%	0	177	0	892	0%	100%	0%	83%	83.4%	No	N/A
	0.4FA to 1Fault	81%	46%	79%	0	354	0	892	0%	100%	0%	72%	71.6%	No	N/A
	0.6FA to 1Fault	81%	46%	79%	18	513	1	891	3%	100%	95%	63%	63.9%	No	N/A
	0.8FA to 1Fault	81%	46%	79%	125	583	31	861	18%	97%	80%	60%	61.6%	No	N/A
	1FA to 1Fault	81%	46%	79%	683	209	512	380	77%	43%	57%	65%	59.6%	No	N/A
	1.5FA to 1Fault	81%	46%	79%	1319	19	834	58	99%	7%	61%	75%	61.7%	No	N/A
	2FA to 1Fault	81%	46%	79%	1782	2	884	8	100%	1%	67%	80%	66.9%	No	N/A
	2.5FA to 1Fault	81%	46%	79%	2229	1	891	1	100%	0%	71%	50%	71.4%	No	N/A
	3FA to 1Fault	81%	46%	79%	2676	0	892	0	100%	0%	75%	0%	75.0%	No	N/A
	13FA to 1Fault	81%	46%	79%	11230	0	892	0	100%	0%	93%	0%	92.6%	No	N/A
BoostedTree	0.2FA to 1Fault	81%	46%	79%	153	24	724	168	86%	19%	17%	0%	30.0%	No	N/A
	0.4FA to 1Fault	81%	46%	79%	126	228	80	812	36%	91%	61%	78%	75.3%	No	N/A
	0.6FA to 1Fault	81%	46%	79%	242	289	107	785	46%	88%	69%	73%	72.2%	No	N/A
	0.8FA to 1Fault	81%	46%	79%	361	347	136	756	51%	85%	73%	69%	69.8%	No	N/A
	1FA to 1Fault	81%	46%	79%	649	243	302	590	73%	66%	68%	71%	69.5%	No	N/A
	1.5FA to 1Fault	81%	46%	79%	1166	172	439	453	87%	51%	73%	72%	72.6%	Yes	19
	2FA to 1Fault	81%	46%	79%	1579	205	454	438	89%	49%	78%	68%	75.4%	Yes	12
	2.5FA to 1Fault	81%	46%	79%	2025	205	502	390	91%	44%	80%	66%	77.4%	No	N/A
	3FA to 1Fault	81%	46%	79%	2453	223	486	406	92%	46%	83%	65%	80.1%	No	N/A
	13FA to 1Fault	81%	46%	79%	10979	251	623	269	98%	30%	95%	52%	92.8%	No	N/A
BaggedTree	0.2FA to 1Fault	81%	46%	79%	21	156	13	879	12%	99%	62%	85%	84.2%	No	N/A
	0.4FA to 1Fault	81%	46%	79%	134	220	54	838	38%	94%	71%	79%	78.0%	No	N/A
	0.6FA to 1Fault	81%	46%	79%	278	253	111	781	52%	88%	71%	76%	74.4%	No	N/A
	0.8FA to 1Fault	81%	46%	79%	474	234	179	713	67%	80%	73%	75%	74.2%	No	N/A
	1FA to 1Fault	81%	46%	79%	691	201	237	655	77%	73%	74%	77%	75.4%	No	N/A
	1.5FA to 1Fault	81%	46%	79%	1190	148	356	536	89%	60%	77%	78%	77.4%	Yes	7
	2FA to 1Fault	81%	46%	79%	1644	140	445	447	92%	50%	79%	76%	78.1%	Yes	5
	2.5FA to 1Fault	81%	46%	79%	2089	141	469	423	94%	47%	82%	75%	80.5%	Yes	2
	3FA to 1Fault	81%	46%	79%	2545	131	492	400	95%	45%	84%	75%	82.5%	No	N/A
	13FA to 1Fault	81%	46%	79%	11102	128	651	241	99%	27%	94%	65%	93.6%	No	N/A
RUSBoostedTrees	0.2FA to 1Fault	81%	46%	79%	143	34	387	505	81%	57%	27%	94%	60.6%	No	N/A
	0.4FA to 1Fault	81%	46%	79%	289	65	397	495	82%	55%	42%	88%	62.9%	Yes	27
	0.6FA to 1Fault	81%	46%	79%	439	92	428	464	83%	52%	51%	83%	63.5%	Yes	26
	0.8FA to 1Fault	81%	46%	79%	582	126	427	465	82%	52%	58%	79%	65.4%	Yes	24
	1FA to 1Fault	81%	46%	79%	752	140	433	459	84%	51%	63%	77%	67.9%	Yes	22
	1.5FA to 1Fault	81%	46%	79%	1164	174	430	462	87%	52%	73%	73%	72.9%	Yes	17
	2FA to 1Fault	81%	46%	79%	1544	240	419	473	87%	53%	79%	66%	75.4%	Yes	12
	2.5FA to 1Fault	81%	46%	79%	1915	315	411	481	86%	54%	82%	60%	76.7%	Yes	11
	3FA to 1Fault	81%	46%	79%	2294	382	424	468	86%	52%	84%	55%	77.4%	Yes	6
	13FA to 1Fault	81%	46%	79%	9572	1658	399	493	85%	55%	96%	23%	83.0%	Yes	1

Table A-3. Various Data Sample Sizes

Training Data Size (# of Files)	True Negative Rate Human (FA-FA)	True Positive Rate Human (Fault-Fault)	Percent Incorrect FA by Human	Percent Incorrect Fault by Human	Human Accuracy	TN (FA-FA)	FN (FA-Fault)	FP (Fault-FA)	TP (Fault-Fault)	True Negative Rates (FA-FA)	True Positive Rates (Fault-Fault)	FA Negative Predictive Values	Fault Positive Predictive Values	Accuracy	Incorrect FA per 100 ME's (ML)	Incorrect Fault per 100 ME's (ML)	Actual 2016 Percent Incorrect FA per 100 ME's (Human)	Actual 2016 Percent Incorrect Fault per 100 ME's (Human)	Percent Incorrect Fault per 100 ME's (Human) Based on trend line	Percent Incorrect FA per 100 ME's (Human) Based on trend line
606	81%	46%	19%	54%	79%	10174	494	734	112	95%	13%	93%	18%	89.3%	5%	87%	18.86%	88.89%	61%	6.98%
849	81%	46%	19%	54%	79%	9953	491	681	147	95%	18%	94%	23%	89.6%	5%	82%	4.31%	75.00%	61%	6.97%
1091	81%	46%	19%	54%	79%	9275	945	613	197	91%	24%	94%	17%	85.9%	9%	76%	5.76%	75.00%	61%	6.96%
1333	81%	46%	19%	54%	79%	9010	984	578	215	90%	27%	94%	18%	85.5%	10%	73%	1.74%	66.67%	61%	6.95%
1576	81%	46%	19%	54%	79%	9058	712	570	205	93%	26%	94%	22%	87.8%	7%	74%	4.50%	51.11%	61%	6.94%
1818	81%	46%	19%	54%	79%	8756	789	543	214	92%	28%	94%	21%	87.1%	8%	72%	4.26%	28.89%	61%	6.93%
2061	81%	46%	19%	54%	79%	8465	855	536	204	91%	28%	94%	19%	86.2%	9%	72%	5.52%	28.89%	61%	6.92%
2303	81%	46%	19%	54%	79%	8058	1039	475	246	89%	34%	94%	19%	84.6%	11%	66%	5.99%	50.00%	61%	6.91%
2546	81%	46%	19%	54%	79%	7745	1127	441	262	87%	37%	95%	19%	83.6%	13%	63%	5.53%	45.56%	61%	6.90%
2788	81%	46%	19%	54%	79%	7851	796	442	244	91%	36%	95%	23%	86.7%	9%	64%	6.55%	42.78%	60%	6.89%
3031	81%	46%	19%	54%	79%	7414	1008	401	267	88%	40%	95%	21%	84.5%	12%	60%	4.50%	15.00%	60%	6.88%
3273	81%	46%	19%	54%	79%	7345	853	375	275	90%	42%	95%	24%	86.1%	10%	58%	3.17%	41.67%	60%	6.87%
3515	81%	46%	19%	54%	79%	7015	958	390	242	88%	38%	95%	20%	84.3%	12%	62%	4.50%	33.33%	60%	6.86%
3758	81%	46%	19%	54%	79%	6944	805	375	239	90%	39%	95%	23%	85.9%	10%	61%	3.47%	60.00%	60%	6.85%
4000	81%	46%	19%	54%	79%	6735	789	334	263	90%	44%	95%	25%	86.2%	10%	56%	2.12%	26.67%	60%	6.84%
4243	81%	46%	19%	54%	79%	6389	910	329	250	88%	43%	95%	22%	84.3%	12%	57%	3.55%	60.00%	60%	6.83%
4485	81%	46%	19%	54%	79%	6172	903	306	255	87%	45%	95%	22%	84.2%	13%	55%	2.80%	33.33%	60%	6.82%
4728	81%	46%	19%	54%	79%	5844	1006	281	262	85%	48%	95%	21%	82.6%	15%	52%	2.88%	61.11%	60%	6.81%
4970	81%	46%	19%	54%	79%	5804	822	288	237	88%	45%	95%	22%	84.5%	12%	55%	6.29%	52.78%	60%	6.80%
5212	81%	46%	19%	54%	79%	5517	884	229	278	86%	55%	96%	24%	83.9%	14%	45%	7.28%	80.05%	60%	6.79%
5455	81%	46%	19%	54%	79%	5400	776	259	231	87%	47%	95%	23%	84.5%	13%	53%	6.27%	68.94%	60%	6.78%
5697	81%	46%	19%	54%	79%	5187	765	233	239	87%	51%	96%	24%	84.5%	13%	49%	8.06%	77.27%	60%	6.77%
5940	81%	46%	19%	54%	79%	5007	720	218	236	87%	52%	96%	25%	84.8%	13%	48%	10.07%	83.33%	60%	6.76%
6182	81%	46%	19%	54%	79%	4786	716	220	217	87%	50%	96%	23%	84.2%	13%	50%	6.88%	100.00%	60%	6.75%
6425	81%	46%	19%	54%	79%	4569	709	226	192	87%	46%	95%	21%	83.6%	13%	54%	3.51%	83.33%	60%	6.74%
6667	81%	46%	19%	54%	79%	4373	680	199	202	87%	50%	96%	23%	83.9%	13%	50%	5.21%	72.22%	60%	6.73%
6910	81%	46%	19%	54%	79%	4183	645	192	192	87%	50%	96%	23%	83.9%	13%	50%	3.16%	55.56%	60%	6.72%
7152	81%	46%	19%	54%	79%	3953	651	181	184	86%	50%	96%	22%	83.3%	14%	50%	5.72%	66.67%	60%	6.71%
7394	81%	46%	19%	54%	79%	3797	583	157	190	87%	55%	96%	25%	84.3%	13%	45%	6.08%	69.44%	60%	6.70%
7637	81%	46%	19%	54%	79%	3575	580	146	183	86%	56%	96%	24%	83.8%	14%	44%	9.61%	66.11%	60%	6.69%

Table A-4. Top 20 RUSBoostedTrees Results with Various Predictors

TVE	SDL	Total # of HRCs	HRC	LCN Group	HAMI	Type of Test	# of HRC within Z Min Window	Time Occurred	RST Count	Set Count	RES Count	Nz Max	Hp Max	Vcas Max	TN (FA-FA)	FN (FA-Fault)	FP (Fault-FA)	TP (Fault-Fault)	True Negative Rates (FA-FA)	True Positive Rates (Fault-Fault)	FA Negative Predictive Values	Fault Positive Predictive Values	Accuracy	ML Better than Human	Rank (Based on Accuracy)
1	0	1	1	0	1	1	0	0	0	0	1	1	0	1	1572	212	411	481	88%	54%	79%	69%	76.7%	Yes	1
1	0	0	1	1	1	1	1	1	0	0	0	1	0	1	1564	220	405	487	88%	55%	79%	69%	76.6%	Yes	2
1	1	0	1	0	0	0	1	1	0	0	0	1	0	1	1555	229	399	493	87%	55%	80%	68%	76.5%	Yes	3
1	0	1	1	0	0	0	1	0	1	0	0	0	1	1	1577	207	422	470	88%	53%	79%	69%	76.5%	Yes	4
1	0	1	1	0	1	0	1	1	0	1	0	1	1	1	1560	224	406	486	87%	54%	79%	68%	76.5%	Yes	5
1	0	1	1	1	1	0	1	0	1	0	1	0	1	1	1549	235	395	497	87%	56%	80%	68%	76.5%	Yes	5
1	1	1	1	0	1	0	0	1	1	0	0	1	1	1	1555	229	402	490	87%	55%	79%	68%	76.4%	Yes	7
1	1	0	1	0	1	1	1	0	1	0	1	0	1	1	1565	219	412	480	88%	54%	79%	69%	76.4%	Yes	7
1	0	0	1	1	1	0	0	1	0	1	0	0	1	1	1562	222	409	483	88%	54%	79%	69%	76.4%	Yes	7
1	0	1	1	1	1	1	0	1	1	0	0	0	1	1	1543	241	390	502	86%	56%	80%	68%	76.4%	Yes	7
1	0	0	1	1	1	1	0	1	0	1	1	1	0	1	1558	226	405	487	87%	55%	79%	68%	76.4%	Yes	7
1	1	1	1	0	1	1	1	1	0	0	1	1	1	1	1553	231	401	491	87%	55%	79%	68%	76.4%	Yes	12
1	0	0	1	1	0	0	1	0	1	1	1	1	0	1	1559	225	407	485	87%	54%	79%	68%	76.4%	Yes	12
1	0	0	1	0	0	1	1	0	1	0	1	0	1	1	1571	213	420	472	88%	53%	79%	69%	76.3%	Yes	15
1	1	0	1	0	0	1	1	1	0	0	1	0	1	1	1564	220	413	479	88%	54%	79%	69%	76.3%	Yes	15
1	0	1	1	1	1	1	1	0	0	0	0	0	1	1	1569	215	418	474	88%	53%	79%	69%	76.3%	Yes	15
0	1	1	1	0	1	1	0	0	0	1	1	1	1	1	1552	232	402	490	87%	55%	79%	68%	76.3%	Yes	18
1	1	1	1	1	0	1	1	1	0	1	0	0	1	1	1561	223	411	481	88%	54%	79%	68%	76.3%	Yes	18
1	0	0	1	0	0	1	1	0	1	1	1	1	0	1	1562	222	412	480	88%	54%	79%	68%	76.3%	Yes	18
1	1	0	1	0	0	0	0	0	0	1	1	1	0	1	1557	227	407	485	87%	54%	79%	68%	76.3%	Yes	18

Table A-5. Sample Machine Learning Input Data

TVE	SDL	Total # of HRCs	HRC	LCN Group	HAMI	Type of Test	# of HRCs set within +- 2 min	Time occurred	RST Count	SET Count	RES Count	NzMax	HPmax	VCASmax	FID Code (U)
XGY:AAMQ	X65YAAXQA-AAXK	56	EKMPAPAPA AMA7	EK	PA	Flight	5	1:21:02	0	1	0	-0.29672	24832.33266	240.84690	FA
XGY:AAAV	X65YAAXQA-AAXP	16	XKMXAXAPA MVVE	XK	PA	Engine Run	2	0:08:49	0	1	0	0.00000	0.00000	0.00000	FA
XGY:AAME	X65YAXMAA-AAAX	43	KXXAAPAPA A7PK	KX	PA	Flight	3	0:18:46	0	1	0	-0.30922	28286.28022	290.43135	FA
XGY:AAPX	X65YAAXQA-AAXK	58	XEAAAAAKN MAAA	XE	KN	Flight	1	1:58:28	0	3	2	-0.43603	1806.50458	259.47867	FA
XGY:AAPK	X65YAAXQA-AAXK	37	EEPAAAAPA VXMQ	EE	PA	Engine Run	8	0:32:07	0	1	0	0.00000	0.00000	0.00000	FA
XGY:AAKM	X65YAXMAA-AAAX	38	EEKAAAAPA XXQ7	EE	PA	Flight	1	1:59:13	0	1	0	-0.30939	11839.78680	282.50378	FA
XFY:AA7K	X65YAXMAA-AAAX	28	EEXAAAAPA PMAM	EE	PA	Flight	1	2:12:27	0	1	0	0.00000	0.00000	0.00000	FA
X9Y:AAXA	X65YAXMAA-AAAX	35	MKPAAPAPA A7EP	MK	PA	Engine Run	35	0:02:16	0	1	0	0.00000	0.00000	0.00000	FA
X9Y:AAPX	X65YAAXQA-AAXK	40	XKVAA7APA APAA	XK	PA	Flight	8	0:21:40	0	2	0	-0.37830	12775.56986	259.19393	FA
XFY:AAPV	X65YAAXQA-AAXP	38	MKXAAAAPA APKA	MK	PA	Flight	2	0:39:30	0	1	0	-0.28907	24767.26370	318.15298	FA
XFY:AAKK	X65YAAXQA-AAXK	32	EEPAAAAPA VNAM	EE	PA	Flight	15	1:40:22	0	3	2	0.03346	18916.98435	356.36789	FA
X9Y:AAXA	X65YAXMAA-AAAX	37	KNAAAAAPA AXPN	KN	PA	Engine Run	2	0:30:56	1	3	2	0.00000	0.00000	0.00000	FA
XGY:AAAE	X65YAAXQA-AAXK	52	XVKXAPAXE EAAX	XV	XE	Engine Run	14	0:09:38	0	1	0	0.00000	0.00000	0.00000	FA
X9Y:AAAV	X65YAAXQA-AAXK	34	X7AAAAAKN A7PX	X7	KN	Flight	1	1:30:07	0	1	0	-0.20896	28909.49613	278.54847	FA
XFY:AAMM	X65YAAXQA-AAXK	40	EEPAAAAPA VMMK	EE	PA	Flight	7	1:46:10	0	1	0	-0.11319	31987.68022	391.49351	FA
XGY:AAPK	X65YAAXQA-AAXK	12	XKMPAXAPA MQQN	XK	PA	Flight	12	0:02:37	0	1	0	0.00000	0.00000	0.00000	FA
X9Y:AAPM	X65YAAXQA-AAXK	36	XKAAAAMPA AXQK	XK	PA	Flight	7	0:02:02	0	1	0	0.00000	0.00000	0.00000	FA
X9Y:AAA7	X65YAAXQA-AAXK	55	K7AAAAPKN AAXK	K7	KN	Flight	16	0:11:12	0	1	0	-0.24956	18737.32513	322.12483	FA
XFY:AAPV	X65YAAXQA-AAXP	35	KNAAAAAPA PPAQ	KN	PA	Engine Run	19	0:16:42	0	1	0	0.00000	0.00000	0.00000	FA
XGY:AAKX	X65YAXMAA-AAAX	49	EEPAAAAPA VPNA	EE	PA	Flight	4	0:10:02	0	2	1	-0.25820	30549.47532	270.37717	FA
XGY:AAMN	X65YAXMAA-AAAX	46	EKVAAAAPA AMQA	EK	PA	Flight	3	1:36:48	0	1	0	0.00000	0.00000	0.00000	FA
X9Y:AAPM	X65YAAXQA-AAXK	25	XNXAAAPKN NVAX	XN	KN	Flight	1	0:40:02	0	1	0	0.03872	18041.36034	324.08308	FA
XGY:AAMV	X65YAAXQA-AAXK	27	EEKAAAAPA AK7K	EE	PA	Engine Run	3	0:07:47	0	1	0	0.00000	0.00000	0.00000	FA
XFY:AAVN	X65YAXMAA-AAA7	29	XNXAAAAPA XPPA	XN	PA	Flight	24	0:05:53	0	1	0	0.00000	0.00000	0.00000	FA
XGY:AAXM	X65YAAXQA-AAXK	39	EKXAAAAPE PAKM	EK	PE	Flight	15	0:08:44	0	1	0	-0.21318	18098.59849	272.61491	FA
XGY:AAXQ	X65YAAXQA-AAXK	60	XPAAAAAPA AA7M	XP	PA	Flight	22	1:21:11	0	1	0	-0.39206	26055.07619	313.54884	FA
XFY:AAQM	X65YAXMAA-AAAX	44	EEMAAPAPA MPK7	EE	PA	Flight	6	2:30:45	0	1	0	-0.14781	31389.02590	406.75802	FA
XFY:AAMM	X65YAAXQA-AAXK	40	EEPAAAAPA VMPK	EE	PA	Flight	7	1:46:10	0	1	0	-0.11319	31987.68022	391.49351	FA
XGY:AAXQ	X65YAAXQA-AAXK	62	EEPAAAAPA XPAN	EE	PA	Flight	2	1:06:35	0	3	2	-0.47674	15579.30784	276.42936	FA
XFY:AAKV	X65YAAXAA-AAP7	49	EEPAAAAPA 7X7X	EE	PA	Flight	2	0:09:15	0	1	0	0.00000	26632.45985	318.89704	FA
X9Y:AAPE	X65YAXMAA-AAAX	49	XKXAAPPPA AMMK	XK	PA	Engine Run	27	0:22:04	0	2	0	0.00000	0.00000	0.00000	FA
X9Y:AAA7	X65YAAXAA-AAP7	49	XKMXAQXPA AAEQ	XK	PA	Flight	2	2:13:46	0	2	0	0.00000	28500.37705	264.94594	FA
XFY:AAKM	X65YAAXAA-AAP7	55	KNAAAAAPA AXAK	KN	PA	Engine Run	42	0:07:30	0	1	0	0.00000	0.00000	0.00000	FA
XGY:AAA7	X65YAAXQA-AAXK	35	KNAAAAAPA PXKP	KN	PA	Flight	6	0:27:23	0	1	0	0.00000	0.00000	0.00000	FA
XGY:AAPP	X65YAAXQA-AAXK	72	XVPMAPXPX PAQX	XV	PX	Flight	15	1:36:27	0	1	0	-0.23852	26430.43352	348.57710	FA
XFY:AAXP	X65YAAXQA-AAXP	57	XVPMAPXPX PAAQ	XV	PX	Flight	9	0:34:29	0	1	0	-0.24770	32597.59755	301.64495	FA
X9Y:AAPM	X65YAAXQA-AAXK	8	XPAAAAAPA APA7	XP	PA	Flight	8	0:02:43	0	1	0	0.00000	0.00000	0.00000	FA
XGY:AAPK	X65YAAXQA-AAXK	29	XKVAA7APA APAP	XK	PA	Flight	2	0:01:53	0	1	0	-0.18204	21828.94906	265.29779	FA